Walter Senn · Marcello Sanguineti ·
Ausra Saudargiene · Igor V. Tetko ·
Alessandro E. P. Villa · Viktor Jirsa ·
Yoshua Bengio (Eds.)

LNCS 16071

Artificial Neural Networks and Machine Learning – ICANN 2025

34th International Conference on Artificial Neural Networks
Kaunas, Lithuania, September 9–12, 2025
Proceedings, Part IV





Springer

MOREMEDIA

Lecture Notes in Computer Science **16071**

Founding Editors

Gerhard Goos
Juris Hartmanis

Editorial Board Members

Elisa Bertino, *Purdue University, West Lafayette, IN, USA*
Wen Gao, *Peking University, Beijing, China*
Bernhard Steffen, *TU Dortmund University, Dortmund, Germany*
Moti Yung, *Columbia University, New York, NY, USA*

The series Lecture Notes in Computer Science (LNCS), including its subseries Lecture Notes in Artificial Intelligence (LNAI) and Lecture Notes in Bioinformatics (LNBI), has established itself as a medium for the publication of new developments in computer science and information technology research, teaching, and education.

LNCS enjoys close cooperation with the computer science R & D community, the series counts many renowned academics among its volume editors and paper authors, and collaborates with prestigious societies. Its mission is to serve this international community by providing an invaluable service, mainly focused on the publication of conference and workshop proceedings and postproceedings. LNCS commenced publication in 1973.

Walter Senn · Marcello Sanguineti ·
Ausra Saudargiene · Igor V. Tetko ·
Alessandro E. P. Villa · Viktor Jirsa ·
Yoshua Bengio
Editors

Artificial Neural Networks and Machine Learning – ICANN 2025

34th International Conference on Artificial Neural Networks
Kaunas, Lithuania, September 9–12, 2025
Proceedings, Part IV

Springer

Editors
Walter Senn
University of Bern
Bern, Switzerland

Ausra Saudargiene
Vytautas Magnus University
Kaunas, Lithuania

Alessandro E. P. Villa
University of Lausanne
Lausanne, Switzerland

Yoshua Bengio
Université de Montréal
Montréal, QC, Canada

Marcello Sanguineti
University of Genoa
Genoa, Italy

Igor V. Tetko
Helmholtz Zentrum München
Neuherberg, Germany

Viktor Jirsa
Aix-Marseille Université
Marseille, France

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-032-04554-6 ISBN 978-3-032-04555-3 (eBook)
https://doi.org/10.1007/978-3-032-04555-3

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2026

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

If disposing of this product, please recycle the paper.

Preface

The 34th International Conference on Artificial Neural Networks (ICANN 2025) was held in Kaunas, Lithuania, from September 9 to 12, 2025.

The International Conference on Artificial Neural Networks (ICANN) serves as a global platform for presenting the latest breakthroughs in artificial intelligence, neural networks, deep learning, and brain-inspired computing. Organized annually by the European Neural Network Society (ENNS), ICANN has, since its inception in 1992, evolved into one of Europe's most prestigious conferences at the intersection of Artificial Intelligence and Neuroscience. It offers researchers, practitioners, and industry professionals a unique opportunity to share innovations, discuss theoretical advances, and explore practical applications of neural computation. The conference strongly values the synergy between theoretical progress and impactful real-world applications, and actively encourages contributions that demonstrate how artificial neural networks are being used to address pressing societal and technological challenges.

We received 375 submissions for the main research track. The research track accepted 170 papers (45%). In addition, there were 42 papers submitted to workshops and special sessions, which were mainly invited contributions (of which 29 were accepted). Each paper was assigned three reviewers, although due to availability and completion rates, an average of 2.2 reviews per paper was obtained, totaling 807 reviews. The review process followed a double-blind peer-review model, and area chairs oversaw quality and fairness. The papers submitted to the main research track were presented over two days. Additionally, one day was devoted to tutorials and one day to workshops.

The General Co-chairs of ICANN 2025 were distinguished experts in the fields of Artificial Intelligence and Neuroscience:

- Yoshua Bengio (Université de Montréal, Canada) is one of the world's leaders in Artificial Intelligence and Deep Learning, the most cited computer scientist worldwide, the recipient of the 2018 A.M. Turing Award, considered to be the "Nobel prize of computing". He is Co-president and Scientific Director of LawZero, a nonprofit organization committed to advancing research and developing technical solutions for safe-by-design AI systems, the Founder and Scientific Advisor of Mila and a Canada CIFAR AI Chair.
- Viktor Jirsa (Aix-Marseille Université, France; EBRAINS Chief Science Officer) is renowned for his fundamental work in theoretical neuroscience, and for open-source simulation platform The Virtual Brain, used to build personalised brain models for epilepsy surgery and other brain disorders. He served as a lead investigator in the EU Flagship Human Brain Project and its successor infrastructure EBRAINS (European Brain REsearch INfrastructureS). Jirsa's contributions have been recognised with numerous honours, including the Human Brain Project Innovation Prize (2021).

The conference featured five keynote talks spanning a wide spectrum of topics, reflecting both emerging scientific challenges and the growing convergence of theoretical

insights, real-world applications, and ethical considerations in AI and neuroscience. These talks addressed the frontiers of generative models, causality, brain simulation, ethics, and efficient deep learning, showcasing the diverse landscape ICANN 2025 aimed to represent:

- Bernhard Schölkopf (Max Planck Institute for Intelligent Systems, Tübingen, Germany): *AI Extracting Causality*
- Viktor Jirsa (Aix-Marseille University, France): *Virtual Brain Twins for Health and Disease*
- Gintarė Karolina Džiugaitė (DeepMind, USA): *On Memorization and Unlearning*
- Emmanuel Bengio (McGill University; Recursion/Valence Labs, Canada): *AI and Generative Models*
- Christiane Woopen (University of Bonn, Germany): *Ethics in AI, Neuroscience, Digital Twins, and Consciousness Research Neuroscience and AI for a Flourishing Life*
- Alessio Micheli (University of Pisa, Italy): *A Journey through Efficient Deep Learning on Graphs*

The invited speakers covered topics at the intersection of neuroscience and artificial intelligence such as neuromorphic computing, biophysical modeling, engineering network-based brain therapies, AI oversight, and reasoning-capable robotics in the physical world:

- Mihai Petrovici (NeuroTMA Lab, University of Bern, Switzerland): *Teacher Propagation Through Space, Time and the Brain*
- Mehmet Fatih Yanik (ETH Zürich, Switzerland): *Engineering Brain Activity Patterns for Therapeutics*
- Michael Deistler (Machine Learning in Science, University of Tübingen, Germany): *Building Mechanistic Models of Neural Computations with Simulation-based Machine Learning*
- Ameya Pandurang Prabh (Tübingen AI Center, University of Tübingen, Germany): *Scalable Strategies for AI Oversight of AI - Progress, Pitfalls and Path Forward*
- Ignas Budvytis (Resilient Robotics, UK): *Embodied Reasoning in the Physical World*

Participants had the exclusive opportunity to attend the public Panel Discussion 'Where Does AI Lead To? Opportunities and Risks', where the leading experts Yoshua Bengio, Viktor Jirsa, Christiane Woopen, Gintarė Karolina Džiugaitė, and Bernhard Schölkopf addressed the interaction between AI and Neuroscience, ethical challenges, AI's impact on human identity, and questions about the safe and responsible development of AI.

The ICANN 2025 Organizing Committee committed itself to supporting Diversity and Inclusion within the Artificial Intelligence and Neuroscience communities. As part of this commitment, ICANN 2025 awarded two scholarships of €1000 and €500 to early-career researchers from developing countries and/or underrepresented communities in science and technology. These scholarships were intended to support travel and accommodation costs, helping recipients attend the conference, present their research, and become active members of the ICANN community.

By enabling broader participation, ICANN 2025 continued efforts towards fostering a more inclusive and representative research environment, encouraging the exchange of ideas across cultures, disciplines, and career stages.

We extend our sincere thanks to all the authors, workshop and tutorial organizers, and participants whose scientific contributions and engagement made ICANN 2025 such a dynamic and inspiring event. Crafting a rich and interdisciplinary program would not have been possible without the dedication and significant commitment of the area chairs, the program committee led by Walter Senn (University of Bern, Switzerland) and the organizing committee, especially Ausra Saudargiene (Lithuanian University of Health Sciences/Vytautas Magnus University, Lithuania) and Linas Petkevičius (Vilnius University, Artificial Intelligence Association of Lithuania). We are also deeply grateful to the many volunteers and session chairs, whose behind-the-scenes work ensured the smooth running of the conference, as well as to all reviewers.

We thank Springer for their continued collaboration in publishing the conference proceedings, and Microsoft for providing access to the CMT conference management system and offering technical assistance throughout the organization process.

We deeply appreciate the ongoing support and guidance of the European Neural Network Society (ENNS), whose leadership and vision continue to shape the ICANN community.

September 2025

Ausra Saudargiene
Walter Senn
Igor V. Tetko
Alessandro E. P. Villa
Marcello Sanguineti
Viktor Jirsa
Yoshua Bengio

Organization

General Chairs

Yoshua Bengio	Université de Montréal, Canada
Viktor Jirsa	Aix-Marseille Université, France

Program Committee

Walter Senn (Chair)	University of Bern, Switzerland
Marcello Sanguineti	University of Genova, Italy
Igor V. Tetko	Helmholtz Munich, Germany and BIGCHEM GmbH, Germany
Alessandro E. P. Villa	University of Lausanne, Switzerland
Ausra Saudargiene	Lithuanian University of Health Sciences/Vytautas Magnus University, Lithuania

Honorary Chairs

Vera Kurkova	Czech Academy of Sciences, Czech Republic
Alessandro E. P. Villa	University of Lausanne, Switzerland
Stefan Wermter	University of Hamburg, Germany

Communication Committee

Sebastian Otte	University of Lübeck, Germany
Kristína Malinovská	Comenius University in Bratislava, Slovakia

Local Organizing Committee

Ausra Saudargiene	Lithuanian University of Health Sciences/Vytautas Magnus University, Lithuania
Linas Petkevičius	Vilnius University, Lithuania

Workshop, Tutorials and Special Session Chairs

Povilas Daniusis	Vytautas Magnus University, Lithuania
Dalius Matuzevičius	VilniusTech, Lithuania

Diversity and Inclusion Chair

Kamilė Dementavičiūtė	Vinted, Lithuania

ICANN Steering Committee

Stefan Wermter	University of Hamburg, Germany
Angelo Cangelosi	University of Manchester, UK
Igor Farkaš	Comenius University in Bratislava, Slovakia
Christina Jayne	Teesside University, UK
Matthias Kerzel	University of Hamburg, Germany
Alessandra Lintas	University of Lausanne, Switzerland
Kristína Malinovská	Comenius University in Bratislava, Slovakia
Jaakko Peltonen	Tampere University, Finland
Brigitte Quenet	ESPCI Paris, France
Roseli Wedemann	Rio de Janeiro State University, Brazil
Sebastian Otte	University of Lübeck, Germany
Ausra Saudargiene	Lithuanian University of Health Sciences/Vytautas Magnus University, Lithuania

Keynote Talks Abstracts

Virtual Brain Twins for Health and Disease

Viktor Jirsa

Institut de Neurosciences des Systèmes (INS), Aix-Marseille Université (AMU), France

Abstract. In the past twenty years, we have made significant progress in creating digital models of an individual's brain, so called virtual brain twins. By combining brain imaging data with mathematical models, we can predict outcomes more accurately than using each method separately. Our approach has helped us understand normal brain states, their operation and conditions like healthy aging, dementia and schizophrenia. We illustrate the virtual brain workflow along the example of drug resistant epilepsy, the so-called Virtual Epileptic Patient (VEP): we reconstruct the connectome of an epileptic patient using DTI and co-register other potential imaging data from the same individual (anatomical MRI, computer tomography (CT)). Each brain region is represented by neural population models, which are derived using mean field techniques from statistical physics expressing ensemble activity via collective variables. Subsets of brain regions generating seizures in patients with refractory partial epilepsy are referred to as the epileptogenic zone (EZ). During a seizure, paroxysmal activity is not restricted to the EZ, but may recruit other healthy brain regions and propagate activity through large brain networks. The identification of the EZ is crucial for the success of neurosurgery and presents one of the historically difficult questions in clinical neuroscience. The application of Bayesian inference and model inversion, in particular Hamiltonian Monte Carlo, allows the estimation of the patient's EZ, including estimates of confidence and diagnostics of performance of the inference. In summary, the Virtual Brain Twin augments the value of empirical data by completing missing data, allowing clinical hypothesis testing and optimizing treatment strategies for the individual patient. Virtual Brain Twins are part of the European infrastructure called EBRAINS, which supports researchers worldwide in digital neuroscience.

Biography. Viktor Jirsa, PhD, is Director of the Inserm Institut de Neurosciences des Systèmes (INS U1106) at Aix-Marseille Université and Director of Research (DRCE) at CNRS in Marseille, France. Trained in theoretical physics and applied mathematics (PhD 1996), he pioneered large-scale brain-network models that combine biologically grounded neural dynamics with individual connectomes, establishing a mathematical framework now central to network science in medicine. He is the scientific architect of the open-source simulation platform The Virtual Brain and served as a lead investigator in the EU Flagship Human Brain Project and its successor infrastructure EBRAINS, driving clinical translation of personalised brain models for epilepsy surgery and other disorders. Jirsa's contributions have been recognised with numerous honours, including the Human Brain Project Innovation Prize (2021).

Causal Representations, World Models and Digital Twins

Bernhard Schölkopf[1,2]

[1] Max Planck Institute for Intelligent Systems & ELLIS Institute Tübingen, Germany
[2] Professor at ETH, Zürich, Switzerland

Abstract. Research on understanding and building artificially intelligent systems has moved from symbolic approaches to statistical learning, and is now beginning to study interventional models relying on concepts of causality. Some of the hard open problems of machine learning and AI are intrinsically related to causality, and progress may require advances in our understanding of how to model and infer causality from data, as well as conceptual progress on what constitutes a causal representation and a causal world model. I will present basic concepts and thoughts, as well some applications to astronomy.

Biography. Bernhard Schölkopf's studies machine learning and causal inference, with applications to fields ranging from astronomy to biomedicine, computational photography, music and robotics. Originally trained in physics and mathematics, he earned a Ph.D. in computer science in 1997 and became a Max Planck director in 2001. His awards include the ACM AAAI Allen Newell Award, the BBVA Foundation Frontiers of Knowledge Award, the Leibniz Award, and the Royal Society Milner Award. He is a Professor at ETH Zurich, a Fellow of the ACM and of the CIFAR Program "Learning in Machines and Brains", and a member of the German Academy of Sciences. He helped start the MLSS series of Machine Learning Summer Schools, the ELLIS society, and the Journal of Machine Learning Research, an early development in open access and today the field's flagship journal. In 2023, he founded the ELLIS Institute Tübingen, and acts as its scientific director.

On Memorization and Unlearning

Gintarė Karolina Džiugaitė

Google DeepMind, USA

Abstract. Deep learning models exhibit a complex interplay between memorization and generalization. This talk will begin by exploring the ubiquitous nature of memorization, drawing on prior work on "data diets", example difficulty, pruning, and other empirical evidence. But is memorization essential for generalization? Our recent theoretical work suggests that eliminating it entirely may not be feasible. Instead, I will discuss strategies to mitigate unwanted memorization by focusing on better data curation and efficient unlearning mechanisms. Additionally, I will examine the potential of pruning techniques to selectively remove memorized examples and explore their impact on factual recall versus in-context learning.

Biography. Gintarė is a senior research scientist at Google DeepMind, based in Toronto, an adjunct professor in the McGill University School of Computer Science, and an associate industry member of Mila, the Quebec AI Institute. Prior to joining Google, Gintarė led the Trustworthy AI program at Element AI/ServiceNow, and obtained her Ph.D. in machine learning from the University of Cambridge, under the supervision of Zoubin Ghahramani. Gintarė was recognized as a Rising Star in Machine Learning by the University of Maryland program in 2019. Her research combines theoretical and empirical approaches to understanding deep learning, with a focus on generalization, memorization, unlearning, and network compression.

AI and Generative Models

Emmanuel Bengio

McGill University; Recursion/Valence Labs, Montreal, Canada

Abstract. In this talk I will share some recent progress on training energy-based generative models using the GFlowNet framework and applying them to scientific problems, as well as speculate on the bigger picture; using GFNs to be good Bayesians, and create models that reason about their environment, concluding with an open discussion of next steps and open problems.

Biography. Emmanuel Bengio is a senior ML Scientist at Valence Labs @ Recursion, working on GFlowNet and drug discovery from Mila, Montreal, Canada. He is mainly interested in Machine Learning, especially Deep Learning and Reinforcement Learning and mixing both. Lately, he has been working at the intersection of ML and drug design using the new GFlowNet framework.

Neuroscience and AI for a Flourishing Life

Christiane Woopen

Heinrich Hertz Professor at the University of Bonn, Germany

Abstract. Neuroscience and artificial intelligence are rapidly advancing fields that combine remarkable scientific progress with ambitious visions for the future. At the same time, they face deep conceptual disagreements, ethical concerns, and dystopian anxieties. Their findings—and the methodological innovations they employ, such as digital twins—raise complex ethical questions concerning autonomy, justice, privacy, and sustainability. Moreover, they are grounded in contested assumptions about the brain and its significance for personal identity and agency. These debates inevitably lead to broader philosophical questions about the nature of human beings and their potential distinctiveness within the continuum of life and consciousness. This lecture examines the ethical implications of these interrelated developments and levels of thought, and it argues for the relevance of human flourishing as a guiding ethical principle in addressing them.

Biography. Christiane Woopen is Heinrich Hertz Chair for Life Ethics at the University of Bonn and founding director of the Center for Life Ethics since October 2021. In addition to leading national and international research projects on ethics in digital technologies, sciences and health she is involved in policy advice, including as Chair of the German Ethics Council (2012–2016), as President of the Global Summit of National Ethics Councils (2014–2016), as a member of the UNESCO International Bioethics Committee until 2017, as Co-speaker of the German Data Ethics Commission (2018–2019), and as Chair of the European Group on Ethics in Science and New Technologies (EGE, 2017–2021). Woopen is a member of several academies of sciences (NRW, BBAW, Academia Europaea, National Academy of Medicine in Mexico) and was awarded the Order of Merit of the State of North Rhine-Westphalia as well as the Federal Cross of Merit 1st Class.

A Journey Through Efficient Deep Learning on Graphs

Alessio Micheli

Head of the Computational Intelligence and Machine Learning Group, Department of Computer Science, University of Pisa, Italy

Abstract. Although the investigation on learning in structured domains started in the late 1990s, recently deep learning for graphs has attracted tremendous research interest and increasing attention for applications. Indeed, graphs are powerful and flexible tools for representing relationships among data at different levels of abstraction. Unsurprisingly, the range of applications includes many fields, including biology, chemistry, network science, computer vision, natural languages and many others. On the other hand, extending the data domain to graphs opens new challenges for the field of deep learning, particularly regarding efficiency, which is related to the important aspect of environmental sustainability. The talk will very briefly introduce the area of deep learning for graphs, with a focus on its origins. We will then move on to discuss advanced topics and current open issues by providing an overview of recent progress in my research group. We will pay particular attention to efficiency, the interplay between model depth and learning complex data representations, and explainability on graphs.

Biography. Alessio Micheli is Full Professor at the Department of Computer Science of the University of Pisa, where he is the head and scientific coordinator of the Computational Intelligence & Machine Learning Group (CIML), part of the CAIRNE.eu Research Network. His research interests include machine learning, neural networks, deep learning, learning in structured domains (sequence, tree, and graph data), recurrent and recursive neural networks, reservoir computing, and probabilistic and kernel-based learning for non-vectorial data, with a particular focus and pioneering works on efficient neural networks for learning from graphs. Micheli is the national coordinator of the "Italian Working group on Machine Learning and Data Mining" of the Italian Association for Artificial Intelligence and he has been co-founder/chair of the IEEE CIS Task Force on Reservoir Computing. He is an elected member of the Executive Committee of the European Neural Network Society – ENNS. He serves as an Associate Editor for Neural Networks and IEEE Transactions on Neural Networks and Learning Systems.

Teacher Propagation Through Space, Time and the Brain

Mihai Petrovici

NeuroTMA Lab, University of Bern, Switzerland

Abstract. Whether biological or artificial, intelligence ultimately boils down to the ability of physical substrates to perform (complex) computations efficiently. Which algorithms can "run" on the dynamics of a given substrate? And conversely, which substrates should we build to implement our algorithms of choice? In my talk, I will discuss variants of these questions that are inspired by recent insights from neurobiology and recent developments in neuromorphic engineering. In particular, I will address several challenges of credit assignment in physical neuronal networks and show how they can be met with rigorous math and creative physics.

Biography. Already since my earliest days as an aspiring physicist, I have worked in several fields that are marked by emergent phenomena arising from complex interactions, from high-multiplicity particle collisions to ultracold glasses and, ultimately, neuromorphic systems. It is this "science of complexity" that continues to intrigue and inspire me. Following my PhD with Karlheinz Meier at Heidelberg University, I moved to the University of Bern, where I am now leading the Neuro-inspired Theory, Modeling and Applications (NeuroTMA) Lab.

I believe there is much to learn from brains about cognition, but taking steps beyond biology may well be warranted when building physical substrates for artificial intelligence – there are good reasons for airplanes not to flap their wings. Therefore, in our group, we combine knowledge and methods from a variety of fields - neuroscience, mathematics, physics, machine learning and microelectronics - to understand biological intelligence and extract its key features for subsequent implementation in silico.

Engineering Brain Activity Patterns for Therapeutics

Mehmet Fatih Yanik

ETH Zurich, Chair of Neurotechnology Laboratories, NSC Study Director, Co-director of Institute of Neuroinformatics, ETH Artificial Intelligence Center

Abstract. Brain networks are disrupted in numerous disorders. I will first show how aberrant brain-wide activity patterns can be corrected by targeting distinct network motifs with multiple neuromodulators using a vertebrate model of human epilepsy and autism. This systematic approach rescues behavior unlike any other. Next, I will present two technologies to realize such network corrections in humans: (1) Biocompatible ultraflexible tentacle electrodes that allow single-neuron-resolution network recordings simultaneously from many brain areas, where we can track inter-areal neuronal ensembles year long, and we are currently starting the first human acute recordings. (2) AU-FUS technology that allows us to non-invasively deliver drugs/RNA to specific brain circuits with 1,300x enhanced focal concentration with millimeter precision across the BBB; we are performing the first pre-clinical large animal studies. I will show how targeting dorsal ACC reduces chronic anxiety without motor side effects.

Biography. Yanik received his BS and MS in Engineering and Physics at MIT in 2000, and PhD in Computational and Applied Physics at Stanford in 2006. He completed brief postdoctoral work in Stanford Bioengineering and Neurosurgery. He subsequently served as Assistant (2006) and later as Associate Professor (2009) till his tenure at MIT. He is currently a professor at ETH Zurich in the Institute for Neuroinformatics. His studies are recognized by a NIH Director's Pioneer Award, NIH Director's New Innovator Award, NIH Transformative Research Award, ERC Consolidator Award, SNSF (~ERC) Advanced, Bridge Discovery Award, Packard Award in Engineering and Science, Alfred Sloan Award in Neuroscience, NIH Eureka Award, NSF Career Award, Silicon Valley's Innovator's Challenge Award, Technology Review Magazine's "World's top 35 innovators under age 35", and Junior Chamber International's "Outstanding Young Person", among others. His work has been highlighted by The Economist, BBC, Popular Mechanics, Nature, Guardian, ABC, Boston Globe, Scientific American, MIT Technology Review, The Scientist, National Geographic, Blick, and others.

Building Mechanistic Models of Neural Computations with Simulation-Based Machine Learning

Michael Deistler

Machine Learning in Science, University of Tübingen, Germany

Abstract. A central challenge in neuroscience is that many properties of neural systems cannot be measured exactly. This limits our understanding of these systems and our ability to build simulations that match experimental recordings or predict neural responses to unseen stimuli. Inference allows scientists to identify parameters—the properties that cannot be measured exactly—such that biophysical simulations are consistent with experimental measurements of neural activity.

I will present machine learning-based inference methods for biophysical simulations in neuroscience. While the focus is on neuroscience, the methods are broadly applicable. First, I will discuss simulation-based Bayesian inference using neural networks. Second, I will introduce differentiable simulation as a powerful approach for parameter inference in large-scale biophysical models. This allows us to directly optimize parameters using gradient-based methods, even in morphologically detailed models of single cells or networks—overcoming the scalability barriers of previous approaches.

Together, these results demonstrate how machine learning opens new possibilities for constructing and fitting biophysical simulations in neuroscience, helping to bridge the gap between experimental data and mechanistic understanding.

Biography. My research lies at the intersection of machine learning and science. I build inference methods that enable the tuning of mechanistic simulators to match empirical measurements. I have a particular focus on biophysical simulations in neuroscience, where I develop machine learning methods to build large-scale and data-driven models of biological intelligence.

Scalable Strategies for AI Oversight of AI - Progress, Pitfalls and Path Forward

Ameya Pandurang Prabh

Tübingen AI Center, University of Tübingen, Germany

Abstract. As foundation models scale, the aspiration of "AI providing oversight for AI" is becoming practical reality. Recent advances, including large-language-model (LLM) judges that auto-grade outputs and weak-to-strong pipelines where smaller models critique larger ones, demonstrate that automated oversight can effectively augment limited human supervision. However, new evidence exposes a critical flaw: great models increasingly think alike. We show that model errors become more correlated as capabilities improve. The similarity that drives high accuracy also biases LLM judges towards models resembling themselves, and weak-to-strong training becomes less effective as supervisors and students share similar blind spots. This talk outlines a research agenda and practical engineering solutions to maintain trustworthy automated supervision despite model convergence.

Biography. Ameya Prabhu is a postdoctoral researcher at the Tübingen AI Center, hosted by Matthias Bethge. His work focuses on data-centric methodologies, scalable lifelong benchmarks, and oversight mechanisms for foundation models. He earned his PhD at the University of Oxford under the guidance of Adel Bibi and Philip Torr, studying foundations of continual learning, machine unlearning, and lifelong model benchmarking. His current research interests are centered around developing Scientist AI (also known as STEM AI), inspired by the vision of Yoshua Bengio and Evan Hubinger. Specifically, he is working towards creating large language models (LLMs) as non-agentic cognitive systems that explain observed data through generating theories, rather than imitating or pleasing humans. His goal is to build powerful, trustworthy AI systems designed explicitly for safe, sandboxed exploration of STEM problems, without external human modeling or intervention.

Embodied Reasoning in the Physical World

Ignas Budvytis

Resilient Robotics, Cambridge, UK

Abstract. Robotics has recently experienced several significant breakthroughs in various capabilities, including improvements in 3D shape estimation and re-rendering algorithms, open-world perception, video generation-based world models, and solving complex manipulation tasks via imitation learning. However, widespread adoption of generalist robotic systems in the physical world still requires further research breakthroughs. In particular, safely interacting with humans, adapting to changes in tasks or environments, and efficiently learning from human feedback remain critical challenges. All these capabilities rely on embodied reasoning—the ability to understand one's own form factor, action space, and impact on the operating environment.

This talk aims to examine AI reasoning from its origins to recent advancements, providing inspiration for novel embodied reasoning systems while highlighting past pitfalls. The presentation reviews early contributions by Turing, McCarthy, and Hayes, and discusses challenges such as the framing problem. Additionally, it covers key milestones in neurosymbolic AI, early visual reasoning, and current efforts in embodied reasoning, including multi-modal foundational models and agentic AI reasoning frameworks. The talk concludes with a brief overview of remaining challenges and future directions for systems that reason effectively in the physical world.

Biography. Ignas Budvytis is an independent researcher and former academic specialising in computer vision, machine learning, and robotics. He is dedicated to developing advanced technologies that enhance how machines perceive and interpret the world. Ignas holds a Bachelor's degree in Computer Science and a PhD in Computer Vision from the University of Cambridge. He recently served as an Assistant Professor of Computer Vision and Robotics in the Department of Engineering at the University of Cambridge.

Contents – Part IV

Epilepsy Prediction Based onIntra- and Inter-channel Feature Mixing 1
Guang Chen, Ziheng Sui, Meng Xu, Chenglou Zou, Jianhui Liu, Jian Cao, Xixin Cao, Yuan Wang, and Xing Zhang

Fine-Grained Recognition of Arteriovenous Fistula Stenosis Using Blood Flow Sounds: An Animal Model-Based Dataset and a Frequency-Aware Decoupling Network 12
Shanlin Xiao, Yangyi Zhou, Jie Zhu, Jincen Wang, Yan Zhao, Yuan Zong, and Jie Kong

SMART-RetroNet: A Framework for Chemical Retrosynthesis Prediction 26
Xiaobo Cheng, Yi Ding, Yi Cao, and Yuehui Chen

Few-Shot Learning for Syndrome Differentiation with Two Prompts 39
Xian Zhou, Sophia Yat Mei Lee, Xinhe Gang, Yichen Yang, Yu Liu, Shengfeng Ju, and Shoushan Li

Neural QSLIM for Mesh Autoencoders 51
Haoliang Zhang, Xintong Li, Jonghoon Kim, Samuel Cheng, and Christian El Amm

Evolving Spatially Embedded Recurrent Spiking Neural Networks for Control Tasks 66
Alexandru Vasilache, Jona Scholz, Yulia Sandamirskaya, and Jürgen Becker

Amortizing Personalization in Virtual Brain Twins 78
Nina Baldy, Marmaduke M. Woodman, and Viktor K. Jirsa

MPCCP: A Multi-chain Perception Crime Charge Prediction Method 89
Congshan Huang, Tianshuo Jiao, Qiao Hu, Yupeng Hu, and Bianxia Du

Learning a Kernel-Based Beran Estimator Using Nearest-Neighbours and Its Application to Reliability Analysis 102
Danilo Samuel Jodas, Christian Laurence Almeida Barry, Guilherme Brandão Martins, Marcos Cleison Santana, Andre Luis Severino Abrego, Danilo Colombo, and João Paulo Papa

Conformalized Causal Learning for Uncertainty-Aware Mineral Prospectivity Mapping 115
Evelyn Jessica Jaya, Qinying Gu, Xinbing Wang, and Nanyang Ye

PhysMamba: Synergistic State Space Duality Model for Remote Physiological Measurement 128
Zhixin Yan, Yan Zhong, Hongbin Xu, Wenjun Zhang, Shangru Yi, Wenxiong Kang, and Lin Shu

Proactive Depot Discovery: A Generative DRL Framework for Adaptive Location-Routing 141
Site Qu and Guoqiang Hu

Learning Joint General and Specific Representation with Masked Auto-Encoder for Radiology Report Generation 154
Tongfei Shen, Zixuan Teng, and Dong Zhang

Process Adaptive Learning for Visual-Language Navigation 166
Chaoqi Gao, Boyuan Zhang, and Yahong Han

Audio-Driven Talking Head Generation with Emotion Based on FLAME Geometry Model 179
Yahui Li, Liejun Wang, Yinfeng Yu, and Shengjie Shen

Studying the Generalization Behavior of Surrogate Models for Punch-Bending by Generating Plausible Counterfactuals 192
Andreas Mazur, Henning Peters, André Artelt, Lukas Koller, Christoph Hartmann, Ansgar Trächtler, and Barbara Hammer

NGAT: A Node-Level Graph Attention Network for Long-Term Stock Prediction 204
Yingjie Niu, Mingchuan Zhao, Valerio Poti, and Ruihai Dong

CSM: Corn Instance Segmentation Model Fusing Dilated Residual Networks and Low-Rank Adaptation 216
Deli Zhu, Xinjie Li, and Liang He

Sensor-Enhanced PINNs for Contaminant Dispersion Modeling 228
Zhijing Feng, Elisa Y. M. Ang, Tay Bee Kiat, Koh Wai Heng, and Peng Cheng Wang

Uniform Representation of Parametric CAD Models for Generative Application 239
Shengling Duan, Jiali Feng, and Yue Qi

Surrogate-Assisted Multi-objective Design of Complex Multibody Systems 251
Augustina C. Amakor, Manuel B. Berkemeier, Meike Wohlleben, Walter Sextro, and Sebastian Peitz

KANLoc: WiFi Localization with A Lightweight KAN 263
Yunlong Gu, Meng Xu, Mengshan Li, Jiawei Li, Jiguang Li, Lixin Guan, and Zhao Huang

DualGF: Example-Driven Path Planning via Dual Gradient Fields 276
Mingdong Wu, Fangwei Zhong, Yulong Xia, Yizhou Wang, and Hao Dong

Improving Physics-Informed Neural Network Extrapolation via Transfer Learning and Adaptive Activation Functions 289
Athanasios Papastathopoulos-Katsaros, Alexandra Stavrianidi, and Zhandong Liu

A Spiking Central Pattern Generator Capable of Adaptive Gait Control in Quadruped Locomotion 302
Narumitsu Ikeda

ViSMoE: Visual-Aware Sparse Mixture-of-Experts for Embodied Referring Expression Grounding 314
Shuo Feng and Piji Li

FDFRL: Credit Card Fraud Detection Based on Federated Reinforcement Learning 326
Nana Zhang, Qin Li, Kun Zhu, and Dandan Zhu

MENGLAN: Multiscale Enhanced Nonparametric Gas Analyzer with Lightweight Architecture and Networks 339
Zhenke Duan, Jiani Tu, and Jiqun Pan

Targeted Trust-Based Merging of Customers' Opinions 351
Jurij Ružejnikov and Tatiana Valentine Guy

DA-NeRF: High-Fidelity Talking Face Generation From Speech With Neural Radiance Fields 366
Yali Cai, Peng Qiao, and Dongsheng Li

Beyond Reconstruction: A Physics Based Neural Deferred Shader for Photo-Realistic Rendering 378
Zhuo He, Paul Henderson, and Nicolas Pugeault

Accurate SDF Reconstruction with Geometric-Differential Regularization and Categorized Sampling Strategy 390
Kaiheng Li, Jiahui Chen, Chuanfeng Yang, Ziheng Zhang, Xuan Wei, Mingyu Shao, and Qingqi Hong

Optimized Supervised Control of Stochastic Timed Discrete Event Systems Using Supervisory Control Theory and Reinforcement Learning 402
Yingjun Liu, Fuchun Liu, and Hongzhen Zhu

A Classification Algorithm for Bronchiolitis Obliterans in Pediatric CT Images with Extreme Class Imbalance 415
Lu Liu, Xibin Feng, Ye Yuan, Wei Xu, and Wenjun Tan

DISEncoder: A Dual-Branch Query Encoder Using Graph Models for Distributed Databases 428
Jianwen Yang, Qiuhong Zhang, Jin Yan, Shuo Zhang, Zhiming Ding, Meiling Zhu, and Xinrun Xu

A Subject-Independent Stress Detection Model Based on Temporal Feature Disentanglement 441
Qingwei Zeng, Peibo Duan, Yongsheng Huang, and ChangSheng Zhang
Long Abstracts

Long Abstracts

Modulation of Brain Activity via Deep Brain Stimulation in Parkinson's Disease Patients: Individualized EEG Based Assessment 455
Gustavas Davidavicius, Vytautas Kucinskas, Pierpaolo Sorrentino, Andrius Radziunas, Augustinas Fedaravicius, Ieva Puleikyte, Karolina Reinyt, and Ausra Saudargiene

Author Index 457

Epilepsy Prediction Based on Intra- and Inter-channel Feature Mixing

Guang Chen[1(✉)], Ziheng Sui[2], Meng Xu[3(✉)], Chenglou Zou[4], Jianhui Liu[1], Jian Cao[1], Xixin Cao[1], Yuan Wang[1], and Xing Zhang[1]

[1] Peking University, Beijing, China
chguang@pku.edu.cn
[2] Shanxi University of Electronic Science and Technology, Linfen, Shanxi, China
[3] University of International Business and Economics, Beijing, China
xumeng@uibe.edu.cn
[4] Xinqiao Hospital, Army Medical University, Chongqing, China

Abstract. Frequent seizures of epilepsy affect millions of people worldwide. The ability to predict seizures in advance could significantly enhance the quality of life for patients by timely interventions. Electroencephalography (EEG) is an important auxiliary examination method for epilepsy. The intra-channel of EEG is temporal information, and the inter-channel of EEG is spatial information, which are different. However, few existing prediction methods based on EEG pay attention to this difference. To address this issue, an EEG Intra- and Inter-Channel Feature Mixing (I^2CM) is proposed to predict epilepsy seizure in this paper. The overall architecture includes three components: pre-process, I^2CM and prediction head. The raw EEG is segmented and filtered in pre-process. I^2CM mixes temporal and spatial information using intra-channel feature extractor and inter-channel feature extractor, respectively. Prediction head outputs whether epilepsy onset is about to occur. Moreover, Multi-Layer Perceptron (MLP) and Kolmogorov-Arnold Network (KAN) are utilized in intra-channel feature extractor, inter-channel feature extractor and prediction head for feature extraction. Our method achieves an average accuracy of 96.34%, a specificity rate of 96.21%, and a sensitivity rate of 96.47% on CHB-MIT dataset, an average accuracy of 98.21%, a specificity rate of 98.64%, and a sensitivity rate of 98.58% on Kaggle dataset , offering promising prospects for application in automatic seizure prediction.

Keywords: Epilepsy seizure prediction · EEG · Intra- and inter-channel · Feature Mixing

1 Introduction

Epilepsy is a neurological disorder that causes sudden and abnormal discharges of neurons in the brain, leading to abnormalities in consciousness, feeling, emotion, and behavior [1]. Epilepsy seizure are sudden, unpredictable, causing great

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 1–11, 2026.
https://doi.org/10.1007/978-3-032-04555-3_1

distress to patients and families. The ability to predict seizures in advance, an important issue in biomedical field for a new era [2], could significantly enhance the quality of life for patients by timely interventions.

Electroencephalography (EEG) records brain information and neural activity in the form of electrical signals [3]. By analyzing and processing patient EEG data, a prediction model for epileptic seizures can be constructed to help doctors formulate more effective treatment plans. Predicting epileptic seizures before they occur and take appropriate intervention and protective measures in a timely manner may reduce the damage caused by epilepsy [4].

A typical EEG of epilepsy patients can be divided into the following states: interictal, preictal, seizure prediction horizon (SPH), ictal and postictal state. As shown in Fig. 1, the interictal refers to the brain is in a normal state; the preictal refers to the state before the seizure of epilepsy; the seizure prediction horizon (SPH) is a short time interval between the preictal and ictal state, and the prediction alarm is expected to issue for doctors to intervene; the ictal refers to the actual epileptic seizure event; the postictal refers to the period after an epileptic seizure occurs when the brain recovers to a normal state. The epilepsy seizure prediction (ESP) mainly finds a seizure from preictal state by distinguishing between interictal and preictal states.

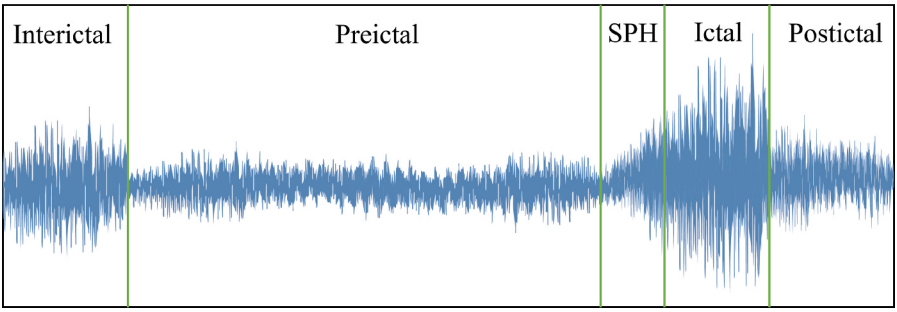


Fig. 1. The main states of epilepsy in EEG.

In recent years, with the development of deep learning technology, studies on epilepsy prediction have progressed rapidly and many important results have been achieved [5–15]. These deep learning models can identify early signs of seizure activity and predict the seizures. Although there are various networks among these models, their architecture can generally be divided into three main parts: preprocessor, feature extractor, and prediction head. The preprocessor performs operations such as filtering and time-frequency transformation, mainly to reduce the interference of eye and muscle movements on the EEG signals. The feature extractor extracts EEG signal features through backbone networks, such as CNN [5], LSTM [9], Vision Transformer [11], etc. The prediction head is usually the Fully Connected (FC) layer, which classifies the extracted features to determine whether the features are interictal or preictal.

EEG device usually has multiple electrodes, which are reflected as multiple channels in the EEG. The time series of potentials collected by each electrode is recorded in the corresponding channel. The intra-channel of EEG is temporal information, and the inter-channel is spatial information, while many methods pay little attention to the difference between intra- and inter-channel. In this paper, an EEG Intra- and Inter-Channel Feature Mixing (I^2CM) is proposed to predict epilepsy seizures. The major contributions of this paper are as follows:

- An epilepsy seizure prediction architecture based on EEG I^2CM is proposed to fuse temporal and spatial information, which contains pre-process, I^2CM and prediction head.
- MLP and KAN, instead of CNN, are used in I^2CM to extract the feature.
- Our seizure prediction method on CHB-MIT dataset is evaluated, and the accuracy, specificity and sensitivity are better than CNN, which validates the effectiveness of our algorithm.

2 Methodology

2.1 Overall Architecture

The raw EEG is used in our method to predict epilepsy seizures and intra- and inter-channel feature mixing is designed for EEG channel feature fusion. The overall architecture is shown in Fig. 2.

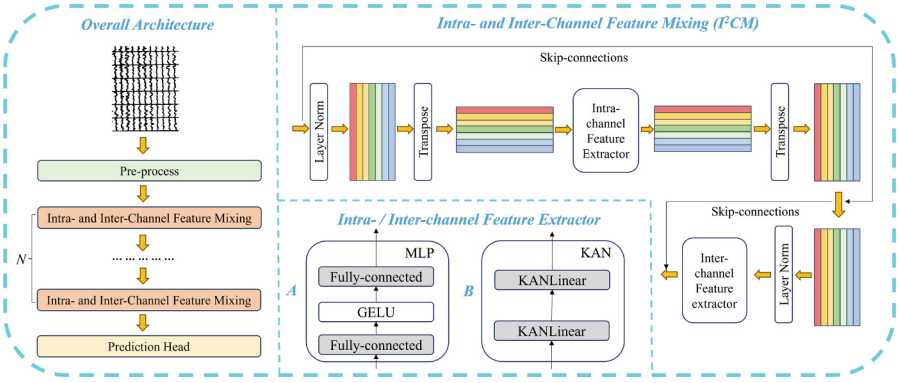


Fig. 2. The Proposed Method for Seizure Prediction based on EEG I^2CM.

The raw EEG is pre-processed firstly, and it is divided into interictal, preictal, SPH, ictal and postictal state. It is also filtered to reduce power-line interference. Then, the processed EEG is sent to N I^2CM blocks for feature extraction. In this block, the input sample is transposed for intra-channel feature extractor and transposed for inter-channel feature extractor. MLP [16] and KAN [17] are used for intra/inter-channel feature extractor. Finally, the feature is fed for prediction head to classify whether it is interictal or preictal state.

2.2 Dataset and Pre-process

We choose CHB-MIT Scalp EEG Database [18] and American Epilepsy Society Seizure Prediction Challenge (Kaggle) Database [19] to evaluate our work. CHB-MIT Scalp EEG Database consists of EEG recordings from 23 pediatric subjects with intractable seizures. These EEG signals are collected at a sampling rate 256 Hz. Kaggle Database comprises 48 seizures and over 700 h of interictal intracranial EEG (iEEG) recordings from five dogs and two patients. The iEEG recordings were sampled 400 Hz for dogs 1–4, using 16 implantable electrodes, while dog 5 utilized 15 electrodes. For the two patients, the sampling rate was 5 kHz, with 15 electrodes for patient 1 and 24 electrodes for patient 2.

We select an intervention time (SPH) of 5 min between the preictal and the ictal. The preictal state is defined 30 min before SPH [20], and the interictal period is defined as 4 h before the ictal and 4 h after the ictal. Based on this setting, for CHB-MIT, chb12 doesn't have a suitable interictal recording. Chb24 lacks the start and end time for the files and cannot determine the interictal state. Therefore, we use the remaining cases (chb1 chb11, chb13 chb23) for our study. In those cases, Channel 15 and Channel 23 are the same scalp electrode, so we use 22 channels in our study. For Kaggle Database, dog2, dog3, dog4 and dog5 is used for our study.

For pre-process, the EEG data is filtered by 6th-order Butterworth filter, filtering out the frequency components of 57∼63 Hz and 117∼123 Hz because of 60 Hz power-line interference. Furthermore, the EEG is segmented into windows of 5 s. The number of windows between interictal and preictal is unbalanced due to our state duration settings. In order to address this imbalance, we increase the number of windows for the fewer one by using an adjustable stride to generate overlapping windows.

2.3 I^2CM

The segmented EEG window is sent to N I^2CM blocks for feature extraction. For each I^2CM block, the input EEG window is normalized by layer normalization, and then transposed for intra-channel feature extractor. Intra-channel feature extractor fuses the EEG signal sequence feature inside each channel, and mixer temporal information. After intra-channel feature, the feature is transposed and normalized by layer normalization for inter-channel feature extractor. Inter-channel feature extractor fuses the EEG channel feature, and mixes spatial information. In the feature mixing, skip-connections are used to keep the gradient.

2.4 Prediction Head

The mixing feature extracted by I^2CM finally is fed to prediction head. Prediction head is actually a classifier, which classifies interictal or preictal state. A single-layer MLP and KAN are also used in prediction head.

2.5 Feature Mixing Based on MLP

Multi-Layer Perceptron (MLP) consists of an input layer, one or more hidden layers and an output layer (Fig. 3). Each layer is composed of multiple neurons, and there are fully-connected between neurons in adjacent layers. There are usually activation functions, such as RELU, tanh, sigmoid, GELU [21], etc., applied to the neurons of each hidden layer to increase the expressive ability of the network. The GELU activation function is used in our intra/inter-channel feature extractor. The output of 1-layer fully connection is:

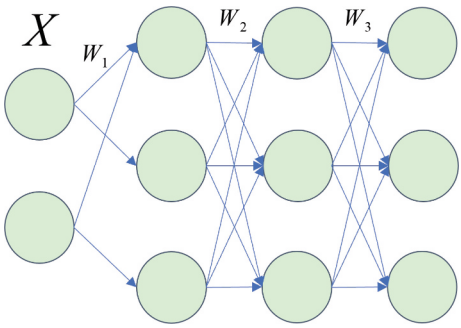


Fig. 3. Multi-Layer Perceptron (MLP).

$$Y = \text{GELU}(WX + b) \tag{1}$$

where W is weight matrix and b is bias.

The output for MLP in intra/inter-channel feature extractor can be expressed as:

$$Y = W_2\text{GELU}(W_1X + b_1) + b_2 \tag{2}$$

The intra-channel feature mixing acts on channel (column) i of X (it is applied to a transposed input), and the MLP is shared across all channels. It mixes temporal information for each channel, and can be written as follows:

$$X^{intra}_{*,i} = W_2\text{GELU}(W_1\text{LN}(X)_{*,i} + b_1) + b_2 \tag{3}$$

$$U_{*,i} = X_{*,i} + X^{intra}_{*,i} \tag{4}$$

where W_1, W_2 are weight matrix on MLP, b_1, b_2 are bias on MLP, $\text{LN}(\bullet)$ is the layer normalization and $U_{*,i}$ is the intra-channel mixing feature for channel i.

The inter-channel feature mixing acts on timestep (row) j of U, and the MLP is shared across all timesteps. It mixes spatial information, and can be written as follows:

$$U^{inter}_{j,*} = W_4\text{GELU}(W_3\text{LN}(U)_{j,*} + b_3) + b_4 \tag{5}$$

$$Y_{j,*} = X_{j,*} + U_{j,*}^{inter} \tag{6}$$

where W_3, W_4 are weight matrix on MLP, b_3, b_4 are bias on MLP, LN(•) is the layer normalization and $Y_{j,*}$ is the inter-channel mixing feature.

2.6 Feature Mixing Based on KAN

Kolmogorov-Arnold Network (KAN), inspired by the Kolmogorov-Arnold representation theorem [22], is a promising alternatives for MLP.

Kolmogorov-Arnold representation theorem states that for any multivariate continuous function f on a bounded domain, there exists a finite composition of univariate continuous functions G_i, $\phi_{i,j}$ such that

$$f(x_1, \cdots, x_n) = \sum_{i=1}^{2n+1} G_i(\sum_{j=1}^{n} \phi_{i,j}(x_j)) \tag{7}$$

where indicates that if we can find proper univariate functions G_i and $\phi_{i,j}$, f can be approximated.

The prototype of KAN, parametrizing each 1D function G_i and $\phi_{i,j}$ as a trainable b-spline curve [23], is a two-layer network with activation functions placed on edges and simple summation on nodes (Fig. 4). A generalized KAN is proposed to approximate function more well with wider and deeper layers. A KANLinear can be defined as a matrix of 1D functions $\Phi = \{\phi_{i,j}\}$, $i = 1, 2, \cdots, N_{in}$, $j = 1, 2, \cdots, N_{out}$, where N_{in} and N_{out} are input dimensions and output dimensions, respectively. A general KAN can be expressed by the composition L layers:

$$Y = \Phi_L(\Phi_{L-l}(\cdots(\Phi_2(\Phi_l(X)))\cdots)) \tag{8}$$

where Φ_L is the function matrix of the l-th KANLinear.

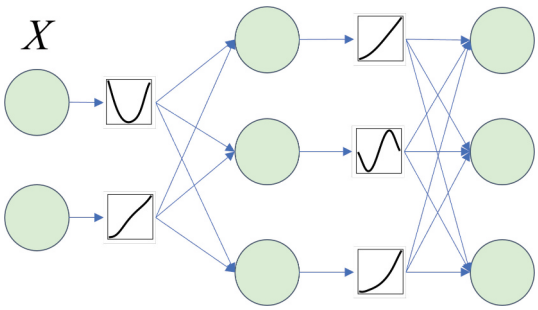


Fig. 4. Kolmogorov-Arnold Network (KAN).

The output for KAN in our intra/inter-channel feature extractor can be expressed as:

$$Y = \Phi_2(\Phi_1(X)) \tag{9}$$

The intra-channel feature mixing, acting on channel i of X, can be written as follows:

$$U_{*,i} = X_{*,i} + \Phi_2(\Phi_l(\mathrm{LN}(X)_{*,i})) \tag{10}$$

where $\mathrm{LN}(\bullet)$ is the layer normalization and $U_{*,i}$ is the intra-channel mixing feature for channel i. KAN is shared across all channels.

The inter-channel feature mixing, acting on timestep j of U, can be written as follows:

$$Y_{j,*} = U_{j,*} + (\Phi_4(\Phi_3(\mathrm{LN}(U)_{j,*})) \tag{11}$$

where $\mathrm{LN}(\bullet)$ is the layer normalization and $Y_{j,*}$ is the inter-channel mixing feature for timestep j. KAN is shared across all timesteps.

3 Experimental Results and Discussion

3.1 Experimental Setting

For each case in CHB-MIT Database, we use segmented EEG window with size of [1280, 22] as input for our model. The topology of both MLP and KAN in intra-channel feature extractor, inter-channel feature extractor, prediction head is 1280-32-1280, 22-32-22, 22-2, respectively. The number of I^2CM block N is set to 4. Cross-entropy loss function is used for model training. A 5-fold cross-validation technique is performed to evaluate our model. The experiment is implemented on NVIDIA A100 with Python 3.9 and Pytorch 1.12.1.

Accuracy (ACC), specificity (SPEC) and sensitivity (SENS) are utilized as evaluation metrics. They are defined as follows:

$$ACC = (TP + TN)/(TP + FN + FP + TN) \tag{12}$$

$$SPEC = TN/(TN + FP) \tag{13}$$

$$SENS = TP/(TP + FN) \tag{14}$$

where TP is the number of correctly classified preictal state, TN is the number of correctly classified interictal state, FP is the number of interictal state misclassified into preictal, and FN is the number of preictal state misclassified into interictal. The value of metrics of CHB-MIT dataset can be obtained by averaging those of 22 cases.

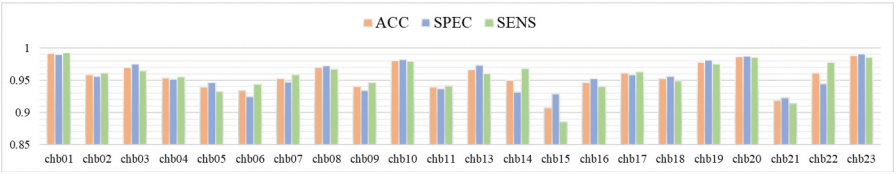


Fig. 5. The Result on CHB-MIT dataset based on MLP.

3.2 Result and Discussion

We test our model according to experimental setting, the results of each case on CHB-MIT dataset based on MLP and KAN are shown in Fig. 5 and Fig. 6, respectively. We can see that KAN is better than MLP in most cases, while MLP is better than KAN in 6 cases (chb01, chb03, chb07, chb10, chb17 and chb23).

The effect of the number of I^2CM is also studied. We keep the same setting except for the number of I^2CM and the result for different number of I^2CM is shown in Table 1. More I^2CM blocks yield better results. When there are 4 blocks, our approach can reach an average accuracy of 96.34%, a specificity rate of 96.21% and a sensitivity rate of 96.47%.

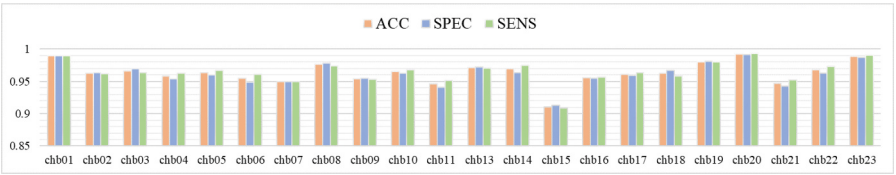


Fig. 6. The Result on CHB-MIT dataset based on KAN.

Table 1. Result Comparison for Different Number of I^2CM.

Method	N	ACC (%)	SPEC (%)	SENS (%)
MLP	1 I^2CM	94.78	94.67	94.89
KAN		95.26	95.03	95.50
MLP	2 I^2CM	95.53	95.22	95.85
KAN		96.06	95.97	96.16
MLP	4 I^2CM	95.69	95.68	95.69
KAN		**96.34**	**96.21**	**96.47**

RNN, BiLSTM, CNN are utilized on the same data to compare the effect of KAN with that of MLP (Table 2, Table 3). The topology of RNN and BiLSTM

are all 22-28-28, following a MLP or KAN with a topology of 28–22; the topology of CNN is TC (Conv16@(1,5)-MaxPool(1,2)-Conv32@(1,5)-MaxPool(1,4)-Conv64@(1,3)-MaxPool(1,5)-Conv64@(3,3)-MaxPool(4,4)), following a MLP or KAN with a topology of 2560-100-32-2. We can see KAN can achieve better results than MLP. Also, our methods based on MLP and KAN are both better than CNN with topology of TC. That is to say, I^2CM based on MLP and KAN perform better than TC in extracting feature.

Table 2. Result Comparison on Kaggle Database for MLP and KAN.

Method	Topology	ACC (%)	SPEC (%)	SENS (%)
RNN+MLP	22-28-28-22	85.26	84.35	86.76
RNN+KAN		87.46	86.28	88.56
BiLSTM+MLP	22-28-28-22	87.59	86.27	89.78
BiLSTM+KAN		89.36	87.67	90.25
CNN+MLP	TC-2560-100-32-2	96.58	94.82	96.98
CNN+KAN		97.17	95.65	98.04
Ours(MLP)	4 I^2CM	97.93	96.85	97.95
Ours(KAN)		**98.21**	**98.64**	**98.58**

Table 3. Result Comparison on CHB-MIT Database for MLP and KAN.

Method	Topology	ACC (%)	SPEC (%)	SENS (%)
RNN+MLP	22-28-28-22	73.21	71.15	75.33
RNN+KAN		75.24	73.56	76.94
BiLSTM+MLP	22-28-28-22	76.77	74.41	79.15
BiLSTM+KAN		77.59	75.22	79.94
CNN+MLP	TC-2560-100-32-2	94.02	93.22	94.83
CNN+KAN		94.67	94.13	95.21
Ours(MLP)	4 I^2CM	95.69	95.68	95.69
Ours(KAN)		**96.34**	**96.21**	**96.47**

Furthermore, we compare our method with other work (Table 4). Our model can obtain the highest accuracy and specificity among these models, and a sensitivity comparable to that of Spiking Conformer [12].

Table 4. Result Comparison with Other Works.

Model	ACC (%)	SPEC (%)	SENS (%)
CNN+SVM [7]	–	90.8	92.7
Two-Layer LSTM [9]	85.1	–	86.8
Sel-JPM-ViT [11]	93.65	92.78	94.70
Spiking Conformer [12]	93.1	89.5	**96.8**
MCNN+AT+BiLSTM [13]	91.39	–	91.30
Ours	**96.34**	**96.21**	96.47

4 Conclusion

This paper focuses on EEG intra- and inter-channel feature mixing and presents an epilepsy seizure prediction architecture to fuse temporal and spatial information. The architecture consists of pre-process, I^2CM and prediction head. Preprocess module performs segmentation and filtering of raw EEG. I^2CM uses intra-channel feature and inter-channel feature extractors built by MLP or KAN to mix feature. Prediction head classifies interictal or preictal state. Additionally, our method is evaluated on CHB-MIT dataset. The results show that I^2CM performs better than CNN with topology of TC, and KAN performs better than MLP, which demonstrates the tremendous potential of our implementation for automatic epilepsy seizure prediction.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Kandar, H.K.M.C.C., Das, S.K., Ghosh, L., Gupta, B.K.: Epilepsy and its management: a review. J. PharmaSciTech **1**(2), 20–26 (2012)
2. Kuhlmann, L., Lehnertz, K., Richardson, M.P., Schelter, B., Zaveri, H.P.: Seizure prediction—ready for a new era. Nat. Rev. Neurol. **14**(10), 618–630 (2018)
3. Mahmoodian, N., Haddadnia, J., Illanes, A., Boese, A., Friebe, M.: Seizure prediction with cross-higher-order spectral analysis of EEG signals. SIViP **14**, 821–828 (2020)
4. Iasemidis, L.D.: Epileptic seizure prediction and control. IEEE Trans. Biomed. Eng. **50**(5), 549–558 (2003)
5. Khan, H., Marcuse, L., Fields, M., Swann, K., Yener, B.: Focal onset seizure prediction using convolutional networks. IEEE Trans. Biomed. Eng. **65**(9), 2109–2118 (2017)
6. Daoud, H., Bayoumi, M.A.: Efficient epileptic seizure prediction based on deep learning. IEEE Trans. Biomed. Circuits Syst. **13**(5), 804–813 (2019)
7. Usman, S.M., Khalid, S., Aslam, M.H.: Epileptic seizures prediction using deep learning techniques. IEEE Access **8**, 39998–40007 (2020)

8. Xu, Y., Yang, J., Zhao, S., Wu, H., Sawan, M.: An end-to-end deep learning approach for epileptic seizure prediction. In: 2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), pp. 266–270. IEEE (2020)
9. Varnosfaderani, S.M., et al.: A two-layer LSTM deep learning model for epileptic seizure prediction. In: 2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS), pp. 1–4. IEEE (2021)
10. Zhao, S., Yang, J., Sawan, M.: Energy-efficient neural network for epileptic seizure prediction. IEEE Trans. Biomed. Eng. **69**(1), 401–411 (2021)
11. Qi, N., Piao, Y., Zhang, H., Wang, Q., Wang, Y.: Seizure prediction based on improved vision transformer model for EEG channel optimization. In: Computer Methods in Biomechanics and Biomedical Engineering, pp. 1–12 (2024)
12. Chen, Q., Sun, C., Gao, C., Liu, S.-C.: Epilepsy seizure detection and prediction using an approximate spiking convolutional transformer. In: 2024 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–5. IEEE (2024)
13. Ahmad, I., et al.: Multi-feature fusion-based convolutional neural networks for EEG epileptic seizure prediction in consumer internet of things. IEEE Trans. Consum. Electron. **70**(3), 5631–5643 (2024)
14. Mao, T., Li, C., Song, R., Xu, G., Chen, X.: Online seizure prediction via fine-tuning and test-time adaptation. IEEE Internet Things J. (2024)
15. Georgis-Yap, Z., Popovic, M.R., Khan, S.S.: Supervised and unsupervised deep learning approaches for EEG seizure prediction. J. Healthc. Inform. Res. **8**(2), 286–312 (2024)
16. Tolstikhin, I.O., et al.: MLP-mixer: an all-MLP architecture for vision. In: Advances in Neural Information Processing Systems, vol. 34, pp. 24261–24272 (2021)
17. Liu, Z., et al.: Kan: Kolmogorov-Arnold networks. arXiv preprint arXiv:2404.19756 (2024)
18. Shoeb, A.H.: Application of machine learning to epileptic seizure onset detection and treatment. Ph.D. thesis, Massachusetts Institute of Technology (2009)
19. Brinkmann, B.H., et al.: Crowdsourcing reproducible seizure forecasting in human and canine epilepsy. Brain **139**(6), 1713–1722 (2016)
20. Gao, Y., et al.: Pediatric seizure prediction in scalp EEG using a multi-scale neural network with dilated convolutions. IEEE J. Transl. Eng. Health Med. **10**, 1–9 (2022)
21. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016)
22. Schmidt-Hieber, J.: The Kolmogorov-Arnold representation theorem revisited. Neural Netw. **137**, 119–126 (2021)
23. De Boor, C.: On calculating with b-splines. J. Approx. Theory **6**(1), 50–62 (1972)



Fine-Grained Recognition of Arteriovenous Fistula Stenosis Using Blood Flow Sounds: An Animal Model-Based Dataset and a Frequency-Aware Decoupling Network

Shanlin Xiao[1,2], Yangyi Zhou[3], Jie Zhu[2], Jincen Wang[1,2], Yan Zhao[4], Yuan Zong[1,2(✉)], and Jie Kong[3(✉)]

[1] School of Biological Science and Medical Engineering, Southeast University, Nanjing 211189, China
{xiaosl,xhzongyuan}@seu.edu.cn

[2] Key Laboratory of Child Development and Learning Science of Ministry of Education, Southeast University, Nanjing 211189, China

[3] Department of Intervention, Nanjing First Hospital, Nanjing 210006, China
kilogram@163.com

[4] College of Computer Science and Software Engineering, Hohai University, Nanjing 211100, China

Abstract. In this paper, we investigate the feasibility of automatically recognizing the fine-grained stenosis rate of arteriovenous fistula (AVF) by analyzing blood flow sounds. Due to the challenges of recruiting patients with varying stenosis rates of AVF for data collection, we initially developed an AVF animal model using Beagle dogs. We manipulated the stenosis rates in their AVFs and recorded the corresponding blood flow sound data, resulting in the creation of the BeagleAVF dataset. This dataset consists of 3,351 blood flow sound segments from five Beagle dogs, each labeled with six distinct stenosis rates using synchronized digital subtraction angiography. We also introduce a novel deep learning method, the Frequency-Aware Decoupling Network (FAD-Net), designed to address the challenge of recognizing fine-grained stenosis rates of AVF through blood flow sounds. The core concept of FAD-Net is to seek stenosis-related frequency bands within the spectrums of blood flow sounds, thereby facilitating the learning of stenosis rate-discriminative features. To evaluate the FAD-Net, we conducted extensive experiments using the BeagleAVF dataset. The results demonstrate the effectiveness and superiority of FAD-Net compared to several baseline deep neural networks. Furthermore, our work provides experimental evidence supporting the feasibility of recognizing fine-grained stenosis rates of AVF through the analysis of blood flow sounds.

Keywords: Arteriovenous fistula (AVF) · Blood flow sound · Stenosis rate · Deep learning

S. Xiao and Y. Zhou—Equal contributions.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 12–25, 2026.
https://doi.org/10.1007/978-3-032-04555-3_2

1 Introduction

End-stage renal disease (ESRD) is the final stage of various chronic kidney diseases, affecting approximately 759 individuals per million people worldwide, making it one of the major global health challenges [1]. For patients with ESRD, hemodialysis is a vital kidney replacement therapy that sustains life, and the arteriovenous fistula (AVF) is considered the ideal vascular access for the dialysis process, often referred to as the "lifeline" of hemodialysis [2,3]. However, over time, AVF may develop stenosis due to calcification or abnormal physical pressure, which not only reduces dialysis efficiency but also increases the risk of multiple complications [4,5]. According to research from the National Kidney Foundation (NKF), the primary patency rate of AVF in the first year after creation ranges from 40% to 70% [6]. Therefore, accurate diagnosis of AVF stenosis and timely intervention are critical to ensuring the health and survival of patients with ESRD.

Clinically, AVF stenosis can be diagnosed using digital subtraction angiography (DSA), Doppler ultrasound (DU) and auscultation. DSA is considered the "gold standard" for diagnosing AVF stenosis. However, it is invasive and requires the use of contrast agents, which can cause side effects [7]. DU is a non-invasive diagnostic method, but relies on specialized equipment and expert judgment, limiting its convenience [8]. In contrast, auscultation evaluates AVF stenosis by detecting changes in the bruit or thrill of the blood flow sounds [9]. This method is simple, quick, and non-invasive, reducing patient discomfort. However, auscultation is influenced by the experience and expertise of the physician, making it difficult to ensure its objectivity and accuracy.

In recent years, to improve the accuracy and objectivity of auscultation-based diagnosis, numerous studies have constructed private blood flow sound datasets and proposed various methods to optimize stenosis detection, achieving significant advancements. Wang et al. [10] used a wireless blood flow sound recorder to collect AVF blood sounds and combined them with Doppler ultrasound for stenosis labeling. They extracted time-frequency features using the S-transform and classified stenosis with an RBF neural network. Higashi et al. [11] collected blood flow sounds using an electronic stethoscope and used ultrasound for stenosis labeling. They developed an SVM classification model using features such as frequency power ratio and mel-frequency cepstral coefficients. With the development of deep learning, Park et al. [7] used a wireless electronic stethoscope to collect blood flow sounds 1–2 cm distal to the anastomosis site, with stenosis labeled based on DSA. They used a convolutional neural network (CNN) and employed Mel spectrograms as input, demonstrating for the first time the feasibility of deep learning in binary classification of stenosis detection. Zhou et al. [12] collected blood flow sounds from six different sites, combined them with Doppler ultrasound for stenosis labeling, and compared the performance of CNN, ResNet-50, and Vision Transformer (ViT) models. Their results indicated that, on their private dataset, the ViT model outperformed the other models in stenosis recognition.

However, existing studies have primarily focused on binary classification of stenosis and have not addressed fine-grained AVF stenosis grading. Fine-grained AVF stenosis grading refers to classifying the fistula stenosis into multiple grades from patency to occlusion, which not only aids patients in understanding disease progression but also enables more precise timing for clinical intervention. Nevertheless, due to difficulties in recruiting patients with different stenosis grades, existing studies have not labeled fine-grained stenosis grades, limiting the exploration of fine-grained recognition of AVF stenosis.

Early blood dialysis vascular access experimental models often use canine species, particularly Beagle dogs, due to their hemodynamic characteristics being similar to those of humans, making them highly favored in vascular access-related research [13,14]. Inspired by this, we developed a Beagle dog AVF model, surgically controlled the stenosis grade, and collected blood flow sound data to establish the BeagleAVF dataset. This dataset covers six stenosis grades and includes 3,351 five-second blood flow sound segments, labeled by physicians based on DSA, providing the first high-quality dataset for fine-grained recognition of AVF.

Based on this dataset, we propose a frequency-aware decoupling network (FAD-Net). This network utilizes a "band decomposition - feature decoupling - feature aggregation" three-stage cascade mechanism, aiming to autonomously learn and focus on frequency bands related to stenosis and extract hidden stenosis-related features from the blood flow sounds. Extensive experiments conducted on the BeagleAVF dataset demonstrated the effectiveness of this method and further confirmed the feasibility of fine-grained recognition of AVF stenosis using blood flow sounds, providing new insights and inspiration for future research in this field.

2 BeagleAVF Dataset

This section provides a detailed description of the construction process of the BeagleAVF dataset, as illustrated in Fig. 1. The establishment of the animal model and data collection were conducted by experienced physicians. Note that the animal experiments in this research have been approved by the Nanjing First Hospital Ethics Committee with the Approval No. KY20230912-08.

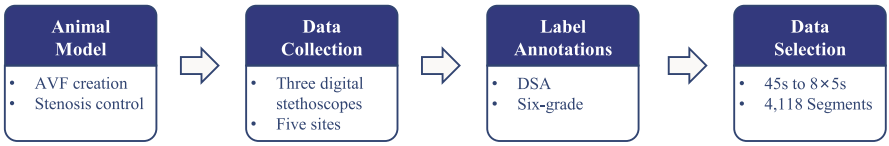
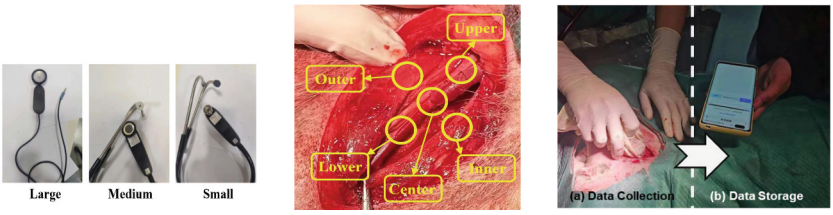


Fig. 1. Flowchart for establishing the BeagleAVF dataset: a comprehensive four-step process involving Animal Model, Data Collection, Label Annotation, and Data Selection.

2.1 Animal Model

In this study, five female Beagle dogs were selected, each weighing approximately 10 kg, with a body length ranging from 30 to 40 cm and an age of 1 to 1.5 years. The establishment of the animal model consisted of two phases:

1. **Fistula Creation**: Under isoflurane anesthesia (induction concentration: 5%, maintenance concentration: 1.5%–2%), the femoral artery and vein were exposed through an inguinal incision. A standard AVF model was created by performing an end-to-side microsurgical anastomosis between the femoral vein and femoral artery using a 6-0 polypropylene suture.
2. **Stenosis Control**: After a four-week recovery for fistula maturation, a secondary surgery was performed. A progressive stenosis model was created by applying an adjustable Fisherman's knot 3 cm distal to the anastomosis, gradually constricting the lumen diameter to simulate varying stenosis grades.



(a) Stethoscopes **(b)** Colletion sites **(c)** Data collection

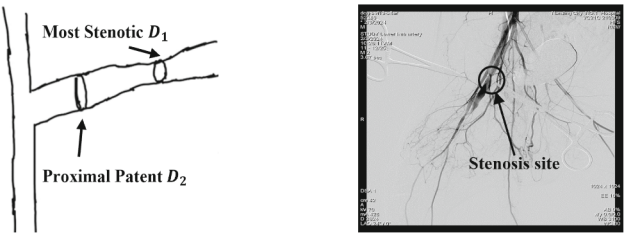
Fig. 2. Illustration of data collection process: (a) Three types of stethoscopes used for data collection. (b) Five collection sites around the AVF anastomosis. (c) The data collection procedure performed by experienced physicians.

2.2 Data Collection

The data collection process is illustrated in Fig. 2. To enhance data diversity and evaluate the impact of different stethoscopes, this study utilized three digital stethoscopes with diaphragm diameters of 47 mm, 36.7 mm, and 23 mm, labeled as "large," "medium," and "small," respectively. The models included the Exagiga ETZ-2 and HMYL POPULAR-3, with the latter available in both 36.7 mm and 23 mm versions.

Audio recordings were obtained at five designated sites, including the AVF anastomosis and four adjacent locations (upper, lower, inner, and outer) approximately 0.5–1 cm away, as recommended by physicians. For each stenosis grade, recordings were sequentially collected using all three stethoscopes at these five sites, with each recording lasting 45 s. All data were collected by professional physicians and stored in cloud storage.

2.3 Label Annotations



(a) Stenosis diagram **(b)** DSA image

Fig. 3. Label annotation process. (a) Stenosis diagram illustrating the AVF stenosis, D_1 is the most stenotic segment and D_2 is the proximal patent segment. (b) DSA image used by physicians to calculate the stenosis rate.

Based on the revised Proudilit coronary artery stenosis grading method [15] and the recommendations of physicians, this study classifies AVF stenosis into six grades. As the grade increases, the severity of stenosis worsens: Grade 1 indicates a patent AVF, while Grades 2 to 6 correspond to stenosis rates of 0–30%, 30–50%, 50–90%, 90–100%, and occlusion, respectively. The stenosis rate is calculated using the following equation:

$$R = \left(1 - \frac{D_1}{D_2}\right) \times 100\%, \tag{1}$$

where D_1 and D_2 are the diameters of the most stenotic and proximal patent segments, respectively, and R is the stenosis rate. Two physicians independently measured the vascular diameter based on the DSA, as shown in the Fig. 3 and the final result was determined by a senior physician.

Table 1. Number of audio segments for different Beagle IDs and stethoscopes, where G1-G6 represent stenosis grade 1–6.

Beagle ID	Large#						Medium#						Small#					
	G1	G2	G3	G4	G5	G6	G1	G2	G3	G4	G5	G6	G1	G2	G3	G4	G5	G6
01	35	40	40	39	40	40	40	37	38	40	38	40	39	40	40	32	37	40
02	40	33	8	22	16	40	40	10	36	26	22	40	40	13	30	40	37	40
03	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
04	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
05	39	40	38	40	32	40	38	32	40	40	40	40	32	32	40	40	40	40

2.4 Data Selection

Due to noise generated during the placement and activation of the stethoscope, the first 5 s of each 45-s recording were removed. The remaining 40 s were split into eight non-overlapping 5-s segments, totaling 3,600 segments. After excluding 249 segments with poor quality, 3,351 high-quality segments were retained. All segments were stored in WAV format at a 16,000 Hz sampling rate, annotated with stenosis grade, collection site, stethoscope, and Beagle ID. The dataset was divided into three sub-datasets by stethoscope, as shown in Table 1.

3 Methods

The overall architecture of the frequency-aware decoupling network (FAD-Net) that we propose is shown in the Fig. 4.

Our design is inspired by the spectral variations of blood flow sounds induced by fistula stenosis. According to the Venturi effect, fistula stenosis leads to a reduction in cross-sectional area, resulting in an increase in blood flow velocity, which in turn causes changes in the frequency components within specific frequency bands [16]. Therefore, identifying stenosis-related frequency bands and extracting their features is crucial.

However, the upper cutoff frequency of the stenosis-related bands remains difficult to determine. To address this challenge, we adopt a frequency band decomposition strategy, where the original blood flow sound signal is decomposed into multiple frequency bands with different cutoff frequencies, and features are extracted from each band. Subsequently, we introduce a frequency contribution attention mechanism, enabling the model to automatically focus on the most related frequency bands for stenosis classification while effectively suppressing irrelevant frequency components.

Furthermore, blood flow sounds inherently contain individual-specific and stethoscope characteristics, potentially interfering with the model's ability to learn discriminative features. To address this issue, we propose disentangling stenosis-irrelevant information from the blood flow sounds to reduce their interference. The resulting features, with irrelevant information removed, are then utilized for stenosis classification and are referred to as stenosis-related features.

3.1 Frequency Band Decomposition

We propose a squeezing frequency band decomposition strategy, where a Butterworth low-pass filter bank with progressively shrinking cutoff frequencies decomposes the original signal.

For a blood flow sound segment $X \in \mathbb{R}^T$, considering that blood flow sound signals are primarily distributed 1500 Hz, we apply a Butterworth low-pass filter bank $\{H_k\}_{k=1}^{6}$ to decompose the signal into 6 frequency bands with cutoff frequencies $\{250, 500, 750, 1000, 1250, 1500\}$ Hz. The filtered signal for each subband k is:

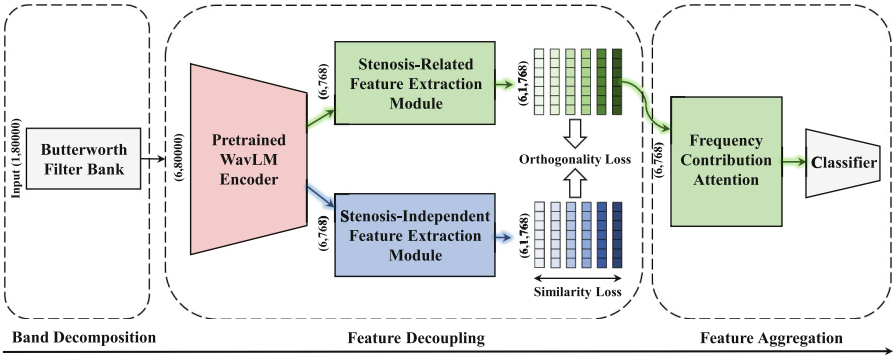


Fig. 4. Overall architecture of the proposed frequency-aware decoupling network, composed of three modules: band decomposition, feature decoupling, and feature aggregation. The features extracted from the upper branch are utilized for classification.

$$X_k = H_k(X), \tag{2}$$

where X_k denotes the output of the k-th filter applied to blood flow sound segment X.

3.2 Feature Decoupling

The feature decoupling module consists of feature extraction module and dual-path feature decoupling module. Given the excellent time-frequency modeling capabilities demonstrated by self-supervised speech models such as WavLM [17] and Whisper [18] in tasks like speech recognition, and recent studies that have proven the effectiveness of using pre-trained models for audio classification tasks [19], we employ the pre-trained WavLM-Base model to extract features from the blood flow sound signals. This process can be expressed as follows:

$$\mathbf{x}_k = \text{WavLM}(X_k), \quad \mathbf{x}_k \in \mathbb{R}^d, \tag{3}$$

where WavLM$(\cdot)$ denotes the feature extraction process by the WavLM-Base model, and $\mathbf{x}_k$ is the feature vector extracted from the k-th frequency band X_k, with d denoting the feature dimension.

The dual-path feature decoupling module disentangles the extraction of stenosis-related and stenosis-independent features, enhancing the discriminative capacity of stenosis-related features through inter- and intra-branch constraints.

In the stenosis-related feature extraction module, a fully connected layer with shared weights is utilized to extract stenosis-related representations. This process is formulated as:

$$\mathbf{x}_k^r = f^r(\mathbf{x}_k), \quad \mathbf{x}_k^r \in \mathbb{R}^d, \tag{4}$$

where $f^r(\cdot)$ denotes the stenosis-related feature extraction module, $\mathbf{x}_k^r$ is stenosis-related feature of the k-th frequency band.

In the stenosis-independent feature extraction module, to comprehensively capture noise information across different frequency bands, we utilize six independent fully connected layers to extract stenosis-independent representations from each frequency band. This process is formulated as follows:

$$\mathbf{x}_k^i = f_k^i(\mathbf{x}_k), \quad \mathbf{x}_k^i \in \mathbb{R}^d, \tag{5}$$

where $f_k^i(\cdot)$ denotes the stenosis-independent feature extraction module for the k-th frequency band, $\mathbf{x}_k^i$ is stenosis-independent feature of the k-th frequency band.

Meanwhile, leveraging the invariance of stenosis-independent factors, such as physiological characteristics and stethoscope type, within the same signal segment, we employ a consistency loss to constrain stenosis-independent features across different frequency bands:

$$\mathcal{L}_{\text{con}} = \frac{1}{K(K-1)} \sum_{k=1}^{K} \sum_{j \neq k} \left(1 - \cos\left(\mathbf{x}_k^i, \mathbf{x}_j^i\right)\right), \tag{6}$$

where $\cos(\cdot)$ denotes cosine similarity, and $\mathbf{x}_k^i$ and $\mathbf{x}_j^i$ are the feature representations of the k-th and j-th stenosis-independent features, respectively.

Finally, to enforce the independence of stenosis-related and stenosis-independent features while imposing constraints between the two modules, we incorporate an orthogonal loss. This loss mitigates redundant information interference, enhancing feature discriminability and purity:

$$\mathcal{L}_{\text{ortho}} = \sum_{k=1}^{K} \left(\frac{\mathbf{x}_k^r \cdot \mathbf{x}_k^i}{\|\mathbf{x}_k^r\|_2 \|\mathbf{x}_k^i\|_2} \right)^2, \tag{7}$$

where $\|\cdot\|_2$ denotes the L_2-norm, and $\mathbf{x}_k^r \cdot \mathbf{x}_k^i$ represents the dot product of the two feature vectors.

3.3 Feature Aggregation

Instead of directly flattening stenosis-related features for each frequency band, we incorporate a frequency contribution attention to assess the importance of the features from each frequency band:

$$\alpha_k = \frac{\sigma(g(\mathbf{x}_k^r))}{\sum_{j=1}^{K} \sigma(g(\mathbf{x}_j^r))}, \tag{8}$$

where $g(\cdot)$ denotes the operation performed by the FC layer, and $\sigma(\cdot)$ represents the *Sigmoid* function. Subsequently, the weighted features are concatenated and then fed into the classifier.

3.4 Total Loss

The total loss function of the proposed FAD-Net is obtained by combining the loss functions presented in Eqs. (6) and (7), along with the cross-entropy loss. The resulting expression is given as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ce}} + \lambda_1 \mathcal{L}_{\text{con}} + \lambda_2 \mathcal{L}_{\text{ortho}}, \tag{9}$$

where $\mathcal{L}_{\text{total}}$ and $\mathcal{L}_{\text{ce}}$ represent the total loss and the cross-entropy loss, λ_1 and λ_2 are hyperparameters that control the importance of the consistency loss and orthogonal loss, respectively.

4 Experiments

4.1 Experimental Setup

In this section, we conducted extensive experiments on the BeagleAVF dataset and its three sub-datasets. We applied Leave-One-Subject-Out (LOSO) cross-validation protocol for evaluation. For each fold of the experiment, we use the blood flow sounds of one Beagle dog as the testing set, while the remaining Beagle dogs blood flow sounds are used as the training set. The performance metric we adopted is *Accuracy*, and its calculation formula is as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%, \tag{10}$$

where TP, FP, FN, and TN represent the True Positive, False Positive, False Negative, True Negative respectively, over all experiment folds.

We compared the FAD-Net with several deep learning methods, including CNN models such as VGG-11 [20], ResNet-18 [21], and MobileNetV2 [22], as well as pre-trained speech models like WavLM, Whisper and HuBERT [23] for feature extraction, followed by classification using a simple multilayer perceptron (MLP).

Additionally, we performed coarse-grained stenosis recognition experiments, categorizing stenosis into 2-grade and 4-grade. The 2-grade classification consists of $< 50\%$ stenosis and $> 50\%$ stenosis, while the 4-grade classification includes patency, mild stenosis (0–50%), severe stenosis (50–100%), and occlusion.

Furthermore, to explore the impact of different Butterworth filter banks on the results, we also designed two different frequency band divisions for comparison, with cutoff frequencies of $\{250, 500, 750\}$ Hz and $\{500, 1000, 1500\}$ Hz, referred to as FAD-Net V1 and FAD-Net V2, respectively.

4.2 Implementation Details

For feature extraction, we use the pre-trained WavLM-Base model, followed by mean pooling, producing a 768-dimensional feature vector. The stenosis-related and stenosis-independent feature extraction modules consist of fully connected layers with dimensions 768, 1024, and 768. The frequency contribution attention

module uses fully connected layers with dimensions 768, 256, and 1 to learn the frequency band weights. During training, we use an NVIDIA RTX 4090 GPU with a batch size of 128. The optimizer used is Adam, with a learning rate of 0.0001. To ensure that the auxiliary losses are effective without interfering with the training of the main network, the hyperparameters for consistency loss and orthogonality loss are both set to 0.05.

4.3 Results and Discussion

Table 2. Comparison of accuracy for fine-grained stenosis recognition under the LOSO protocol. The best result is highlighted in bold.

Method	Large#	Medium#	Small#	BeagleAVF#
VGG-11	40.29	44.23	41.34	40.08
ResNet-18	38.66	39.12	37.99	38.50
MobileNet-V2	38.48	41.63	37.81	33.36
Hubert	39.66	40.11	36.31	36.44
Whisper	40.56	37.87	40.28	38.68
WavLM	42.92	41.90	37.90	38.38
FAD-Net V1	40.93	44.76	41.87	38.62
FAD-Net V2	42.20	43.87	41.25	38.44
FAD-Net	**43.56**	**47.18**	**42.49**	**40.14**

Table 3. Experimental results of coarse-grained stenosis recognition using FAD-Net under the LOSO protocol.

Classification	Large#	Medium#	Small#
2-Grade	76.14	71.36	77.03
4-Grade	49.73	49.42	48.23
6-Grade	43.56	47.18	42.49

The experimental results of all methods are shown in Table 2. Most methods achieve classification accuracies exceeding 40% across the three sub-datasets. Among the baseline models, VGG-11 overall exhibits the best performance, with accuracy rates of 40.29%, 44.23%, 41.34%, and 40.08%, respectively. These results validate the feasibility of using blood flow sound signals for fine-grained recognition of AVF stenosis and provide a solid experimental foundation for subsequent research. Notably, our proposed FAD-Net outperforms all comparative

methods, achieving accuracy of 43.56%, 47.18%, 42.49%, and 40.14%, demonstrating significant improvements over other methods. This further confirms the effectiveness and superior performance of FAD-Net in fine-grained stenosis recognition tasks. Moreover, the training module of FAD-Net is composed of several structurally simple and lightweight MLPs, with a total number of parameters of approximately 19.6M. In experiments, the model's batch inference speed on an RTX GEFORCE 4090 GPU reaches 0.0197 s, demonstrating promising potential for future deployment on mobile devices.

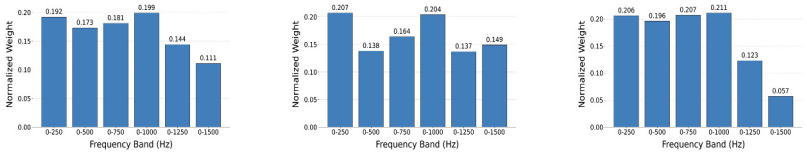
As expected, the choice of stethoscope has a significant impact on recognition performance. Among the three stethoscopes, the "medium" stethoscope performs the best, achieving an accuracy 3% higher than the other two in certain methods. Additionally, the overall classification performance for the BeagleAVF dataset is lower than that of the three sub-datasets. This may be due to the BeagleAVF dataset containing data from different stethoscopes, with samples that are more diverse or complex, which presents greater challenges for the model during the training process. These experimental results highlight an important factor that has been overlooked in previous research, namely the significant impact of different data collection devices on the recognition of fine-grained stenosis. This suggests that in both scientific research and clinical applications, it is crucial to select the appropriate stethoscope to improve diagnostic accuracy.

The experimental results using different Butterworth low-pass filter banks show that, although the frequency band division affects the model's recognition performance, it still outperforms the baseline model. This improvement is primarily attributed to the model's adaptive focusing capability on stenosis-related frequency bands. Furthermore, compared to FAD-Net, the performance of FAD-Net V1 and FAD-Net V2 is relatively poor. FAD-Net V1 uses the frequency band division of $\{250, 500, 750\}$ Hz, which may lead to the loss of important information in the 750–1500 Hz band, thus affecting recognition performance. On the other hand, FAD-Net V2 uses a coarser frequency band division of $\{500, 1000, 1500\}$ Hz, which, compared to FAD-Net's finer frequency band partitioning, may struggle to accurately locate key frequency bands, resulting in interference from irrelevant information and potentially leading to model confusion in some samples.

For coarse-grained tasks, as shown in Table 3, the accuracy for 2-grade and 4-grade stenosis recognition exceeds 77% and 49%, respectively. The simplicity of these tasks compared to the 6-grade fine-grained task contributes to the higher accuracy. The contrast between coarse-grained and fine-grained results underscores the challenges of fine-grained classification, suggesting that further research should focus on enhancing model accuracy and stability.

As shown in Fig. 5, we visualized the normalized band weights of samples with three stenosis grades (30–50%, 50–90%, and 90–100%) using Beagle 01 as the test set in the Small sub-dataset. The 0–250 Hz and 0–1000 Hz bands contribute most significantly to stenosis classification, indicating that stenosis may be associated with features in low-frequency and some higher-frequency bands. However, features in the 0–1250 Hz and 0–1500 Hz bands may introduce

high-frequency noise and redundant information, reducing the model's discriminative ability. These findings highlight the importance of band attention for model performance. Notably, FAD-Net does not rely on any prior knowledge but can automatically focus on the most informative frequency bands.



(a) 30–50% stenosis **(b)** 50–90% stenosis **(c)** 90–100% stenosis

Fig. 5. Normalized frequency band weights for 30–50%, 50–90%, and 90–100% stenosis grades in the Small sub-dataset, using Beagle 01 as the test set.

4.4 Data Collection

Moreover, benefiting from the feature decoupling mechanism and the frequency-band attention module, FAD-Net can effectively focus on critical frequency bands and disentangle stenosis-related features, even in the presence of variations in spectral distributions between human and Beagle blood flow sounds. Therefore, FAD-Net demonstrates strong potential for generalization to human datasets, and we plan to validate and extend this approach using clinical patient data in future work

5 Conclusion

In this study, we introduced the task of fine-grained recognition of AVF stenosis for the first time. To address the challenge of limited data, we constructed BeagleAVF, the first animal model-based fine-grained AVF stenosis blood flow sound dataset. We further proposed FAD-Net, which automatically identifies stenosis-related frequency bands and learns discriminative features for stenosis classification. Extensive experiments on BeagleAVF validated the effectiveness of our approach and demonstrated the feasibility of fine-grained recognition of AVF stenosis using blood flow sounds. In the future, we will explore more efficient deep learning algorithms to further improve the accuracy of fine-grained recognition of AVF stenosis. We will also collect blood flow sound data from human patients with fine-grained stenosis grading and apply prior knowledge from animal models, including deep learning algorithms, to the diagnosis of human patients. These studies will contribute to the development of more accurate, convenient, and cost-effective diagnostic tools for patients undergoing hemodialysis.

Acknowledgement. This work was supported in part by the YESS Program by CAST under the Grant 2023QNRC001, in part by the Zhishan Young Scholar Program of Southeast University, and in part by the Jiangsu Medical Association Special Fund Project under the Grant SYH-3201140-0094 (2023041).

References

1. Paletti, S.K., et al.: Kidney: a review on end stage renal disease, dialysis and transplant. Int. Clin. Med. Case Rep. J. **8** (2023)
2. Park, H.S., et al.: Comparison of outcomes with arteriovenous fistula and arteriovenous graft for vascular access in hemodialysis: a prospective cohort study. Am. J. Nephrol. **43**(2), 120–128 (2016)
3. Viecelli, A.K., et al.: Report of the Standardized Outcomes in Nephrology-Hemodialysis (SONG-HD) consensus workshop on establishing a core outcome measure for hemodialysis vascular access. Am. J. Kidney Dis. **71**(5), 690–700 (2018). https://doi.org/10.1053/j.ajkd.2017.12.003
4. Kirkeeide, R.L., et al.: Wall vibrations induced by flow through simulated stenoses in models and arteries. J. Biomech. **10**(7), 431–441 (1977)
5. Fredberg, J.J.: Origin and character of vascular murmurs: model studies. J. Acoust. Soc. Am. **61**(4), 1077–1085 (1977)
6. Al-Jaishi, A.A., et al.: Patency rates of the arteriovenous fistula for hemodialysis: a systematic review and meta-analysis. Am. J. Kidney Dis. **63**(3), 464–478 (2014)
7. Park, J.H., et al.: Feasibility of deep learning-based analysis of auscultation for screening significant stenosis of native arteriovenous fistula for hemodialysis requiring angioplasty. Korean J. Radiol. **23**(10), 949 (2022)
8. Vesquez, P.O., et al.: Arteriovenous fistula stenosis detection using wavelets and support vector machines. In: 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 1298–1301 (2009)
9. Lin, C.H., et al.: Arteriovenous shunt stenosis assessment based on empirical mode decomposition and 1D-convolutional neural network: clinical trial stage. Biomed. Sig. Process. Control **66**, 102461 (2021)
10. Wang, H.Y., et al.: Novel noninvasive approach for detecting arteriovenous fistula stenosis. IEEE Trans. Biomed. Eng. **61**(6), 1851–1857 (2014)
11. Higashi, D., et al.: Classification of shunt murmurs for diagnosis of arteriovenous fistula stenosis. In: Proceedings of Asia-Pacific Signal Information Processing Association Annual Summit Conference (APSIPA ASC), pp. 665–669 (2018)
12. Zhou, G., et al.: Deep learning analysis of blood flow sounds to detect arteriovenous fistula stenosis. NPJ Digit. Med. **6**(1), 163 (2023)
13. Kokozidou, M., et al.: Vascular access animal models used in research. Ann. Anat. **225**, 65–75 (2019)
14. Hatzibaloglou, A., et al.: Experimental acute and chronic vascular access through the femoral vessels: hemodynamic study in dogs. Artif. Organs **13**(5), 438–441 (1989)
15. Proudfit, W.L.: Prognostic value of coronary arteriography. Cardiovasc. Clin. **12**(2), 1–8 (1981)
16. Yeih, D.F., et al.: Physiology-based diagnosis algorithm for arteriovenous fistula stenosis detection. In: Proceedings of 36th Annual International Conference IEEE Engineering in Medicine and Biology Society, pp. 4619–4622 (2014)

17. Chen, S., et al.: WavLM: large-scale self-supervised pre-training for full stack speech processing. IEEE J. Sel. Top. Sig. Process. **16**(6), 1505–1518 (2022)
18. Radford, A., et al.: Robust speech recognition via large-scale weak supervision. In: International Conference on Machine Learning, pp. 28492–28518 (2023)
19. Gong, Y., et al.: Whisper-at: noise-robust automatic speech recognizers are also strong general audio event taggers. arXiv preprint arXiv:2307.03183 (2023)
20. Simonyan, K., et al.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
21. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
22. Sandler, M., et al.: Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018)
23. Hsu, W.-N., et al.: Hubert: self-supervised speech representation learning by masked prediction of hidden units. IEEE/ACM Trans. Audio Speech Lang. Process. **29**, 3451–3460 (2021)

SMART-RetroNet: A Framework for Chemical Retrosynthesis Prediction

Xiaobo Cheng[1], Yi Ding[1], Yi Cao[2], and Yuehui Chen[1(✉)]

[1] School of Information Science and Engineering, University of Jinan, Jinan, China
{xbcheng,dingyi}@stu.ujn.edu.cn

[2] Shandong Key Laboratory of Ubiquitous Intelligent Computing (School of Information Science and Engineering), University of Jinan, Jinan, China
{ise_caoy,yhchen}@ujn.edu.cn

Abstract. Retrosynthesis prediction of chemical reactions is a fundamental yet challenging problem in the field of chemistry. Existing methods, especially those based on deep learning such as Transformer models, tend to learn the map between reactants and products directly from chemical reaction datasets. However, these models often neglect prior knowledge from the chemistry domain. Without the prior knowledge, the performance of models heavily relies on the quality of reaction dataset, instead of understanding molecule. To address these issues, we propose the SMART-RetroNet framework. Our approach involves a pre-training task of mapping different SMILES representations to their canonical forms, during which the model could successfully learn fundamental prior knowledge in chemistry field. Experiment results demonstrate that SMART-RetroNet can achieve significant performance improvements in chemical reaction one-step retrosynthesis task. This work provides an effective approach for chemical reaction retrosynthesis prediction and a way to transfer prior knowledge in related chemistry tasks.

Keywords: Chemical reaction retrosynthesis prediction · Deep learning · Pre-training · Representational learning

1 Introduction

Retrosynthesis of chemical reactions is a crucial problem in the field of chemistry. It aims to determine the reactants and reaction pathways that can lead to a desired chemical product [1]. In the process of drug development, for example, chemists often need to design synthetic routes for target compounds [2]. Accurate retrosynthesis prediction can greatly accselerate the research and development process and reduce costs [3].

Deep learning techniques have brought new research perspectives for chemical reaction retrosynthesis. Deep learning models [4,9] have shown potential in handling complex chemical reaction data. Existing deep learning-based methods for chemical reaction retrosynthesis are mainly end-to-end trained models that take the chemical reaction product as input and output predicted reactants. During the training phase, a critical challenge is the innate variety in Simplified Molecular Input Line Entry System (SMILES) when denoting a single molecule,

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 26–38, 2026.
https://doi.org/10.1007/978-3-032-04555-3_3

which may complicate the learning process [7]. To address this issue, researchers often turn to use data augmentation, which helps in increasing the diversity of the training data by generating different but equivalent SMILES representations [7]. Data alignment [8], on the other hand, ensures that the molecular representations are in a consistent format. In this way, end-to-end models could efficiently capture the latent distributions within the dataset, facilitating efficient model training and accurate retrosynthesis predictions

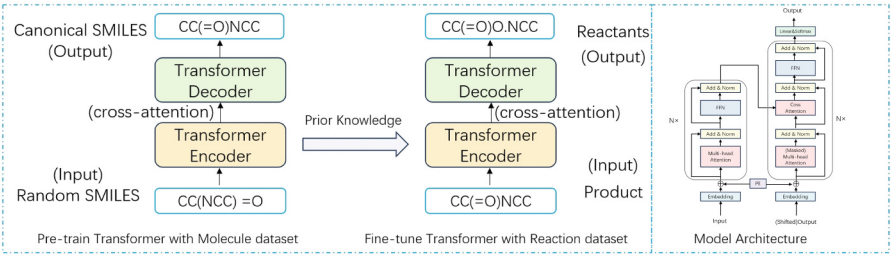


Fig. 1. SMART-RetroNet Framework and Architecture.

Usually, the task of predicting chemical reaction retrosynthesis involves a wide range of chemical prior knowledge, like atom types, hybridization states, types of chemical bonds, chirality centers, and stereoisomers. Without the fundamental knowledge, the model performance could be heavily influenced by data quality. In order to solve this problem, we proposed the SMART-RetroNet framework. During the pre-training phase, we introduced a simple but general task, which is beneficial for a variety of downstream tasks related to chemistry. Given a random SMILES representation of a molecule, the model is required to produce its canonical form (see Fig. 1). In this way, the model may learn different levels of chemical prior knowledge, including molecular structure, functional group precedence, molecular symmetry, and so on. Moreover, we employed data augmentation techniques such as atom deletion and bond deletion to reinforce the model's perception of each atom's environment within the molecule, thereby enhancing its understanding of molecular configurations and interactions. Then, the model is fine-tuned on reaction dataset to learn the relationship between reactants and products in chemical reactions. Through this learning method, our model has achieved performance improvement in the chemical reaction retrosynthesis task. For the first time, we introduce the similarity score to evaluate the similarity of the SMILES strings between products and reactants in a reaction of the dataset, which would enable us to better understand and analyze how prior knowledge plays a crucial role in the task of chemical reaction retrosynthesis.

2 Related Works

Transformer Based Methods. Transformer-based models for chemical reaction prediction were initially introduced by Schwaller et al. [9]. Compared with

previous sequence models that relied on Recurrent Neural Networks (RNNs), Transformers have brought about remarkable improvements in the field of chemistry. The self-attention in Transformers can effectively capture global dependencies within the sequence, enabling more accurate prediction of chemical reactions [9,11]. Following the initial proposal, researchers have put forward a variety of Transformer-based models [11–13]. For example, RetroPrime [11] employs a two-stage retrosynthetic strategy, which enhances its interpretability like human being specilists. RetroFormer [13] utilizes both global and local attention and combines them with the prediction of reaction sites, and makes its application more flexible. These models have demonstrated promising results in different chemical prediction tasks.

Pre-training. Pre-training has achieved remarkable success in computer vision and natural language processing. Pre-trained vision models such as ResNet and CLIP have been widely adopted as feature extractors for downstream tasks, significantly improving performance for downstream tasks [14,15]. Pre-trained language models such as BERT and GPT have conqued numerous language-related tasks [16,17].

In the realm of chemical retrosynthesis prediction, pre-training is also an effective technique. ChemFormer [18] is a model pre-trained on large molecule database of PubChem, which includes over 1 million small molecules represented in SMILES, to learn the universal feature distribution of molecules from an extensive dataset. However, our proposed framework focuses more intently on how to adequately learn the prior knowledge relevant to downstream chemical retrosynthesis task during the pre-training phase, and then use the prior knowledge effectively during the fine-tuning stage to enhance the final performance.

3 Methods

3.1 Overview

Consider one-step retrosynthesis, where the input $\mathbf{X}$ is the representation of product in a chemical reaction and the output $\mathbf{Y}$ is the representation of reactants. We aim to train a deep learning model $\mathbf{F}$ to generate the prediction $\mathbf{Y}{=}\mathbf{F}(\mathbf{X})$. Model $\mathbf{F}$ is trained on a set of paired data $\mathbf{D_{train}} = \{(\mathbf{X}_i, \mathbf{Y}_i)|i = 1, \ldots, N\}$.

In this paper, we introduce a new framework for the training of model $\mathbf{F}$ as the following process:

(1) In the pre-training phase, the dataset $\mathbf{D_{pre\text{-}train}} = \{\mathbf{X}_i|i = 1, \ldots, N\} \cup \{\mathbf{Y}_i|i = 1, \ldots, N\}$ consists the products and reactants from the original reaction dataset of $\mathbf{D_{train}}$. For each molecule M_1 in $\mathbf{D_{pre\text{-}train}}$, we randomly apply a data augmentation technique to generate a new molecule M_2. Then the molecule M_2 will determine the input and output of the pre-trained model. We denote $\mathbf{X^r}$ as the random format SMILES representation of M_2 and $\mathbf{X^c}$ as its canonical format SMILES representation. The objective of this phase is to train a model $\mathbf{F_{pre}}$ to learn the mapping from the random SMILES to the canonical SMILES for each augmented molecule, which can be expressed as $\mathbf{X^c} = \mathbf{F_{pre}}(X^r)$.

(2) In the fine-tuning phase, the model $\mathbf{F_{pre}}$ will be fine-tuned to capture the relationship between the products and reactants on $\mathbf{D_{train}}$, obtaining the model $\mathbf{F}$ of $\mathbf{Y^r} = F(X_r)$.

3.2 Pre-training

In this study, we developed a specialized task that takes a molecule represented by a random SMILES string as input to predict its canonical SMILES representation. Our model contains a 6-layer encoder and a 6-layer decoder. Each transformer layer incorporates 8 attention heads. The input embedding dimension is set to 512. The encoder and decoder are connected via a cross-attention.

Molecule Dataset. We extract molecules from the USPTO-50K dataset [5, 20] for pre-training. For each reaction, the distinct molecule of reactants and products together form the data corpus for our pre-training model (see Fig. 2). To avoid any confusion, we refer to the new dataset as the molecule dataset throughout this paper.

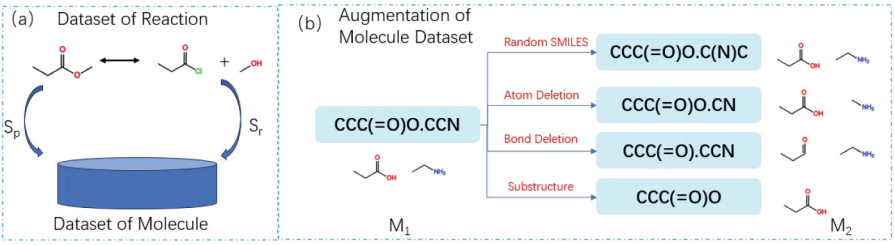


Fig. 2. (a) Molecule Dataset building: SMILES s_p and SMILES s_r are derived from the original reaction dataset. (b) Molecule SMILES augmentation: each molecule has four potential ways of augmentation: atom deletion, bond deletion, random SMILES and substructure. During pretraining, each M_1 randomly selects one augmentation.

Data Augmentation. To further enhance the performance of our pre-training model, we have implemented several data augmentation techniques. These measures include random SMILES, bond deletion, atom deletion, and substructure selection (see Fig. 2). All these augmentation methods are deployed dynamically with the help of the open-source chemical software tool of RDKit. By altering the traversal sequence of atomic nodes in the molecular graph, we enumerate random SMILES representations for each molecule. In the method of atom or bond deletion, we randomly remove one atomic node or bond from the molecular graph obtained from the original SMILES to generate new SMILES, which differs from the original but maintains similar structure. For substructure, as shown in Fig. 2, this method is only effective for a portion of the molecules in the molecule dataset. We randomly select one of the multiple reactant molecules SMILES and discard the others. All generated molecules are validated by RDkit, ensuring its chemical rationality. Data augmentation drives the model to learn molecule structure, functional groups, and chirality in a molecule. Consequently, during pre-training, the model's ability to learn the prior knowledge is enhanced.

Model Training. The loss function [19] for a batch of N_b samples of the pre-training phase is:

$$L(\hat{\mathbf{X}^{(\mathbf{c})}}, \mathbf{X}^{(\mathbf{c})}) = -\frac{1}{N_b} \sum_{b=1}^{N_b} \sum_{t=1}^{T} X_{bt}^{(c)} \log(\hat{X}_{bt}^{(c)}) \tag{1}$$

where L denotes the cross-entropy loss, and $X_{bt}^{(C)}$ is the real labels, and $\hat{X}_{bt}^{(C)}$ is the output of the pre-training model, and T is the number of classes for characters in SMILES. The top-k precision metric is used to demonstrate the pre-training model's performance.

3.3 Fine-Tuning

Dataset Preprocessing. We utilized the well-established USPTO-50K dataset as our training data. We employed four methods to derive variations from the original dataset: canonical data, augmented data, aligned data, and mixed data. The details of each approach are as follows:

Canonical Data: This involved preprocessing the raw USPTO-50K dataset to remove any inconsistencies and to ensure a uniform format for all entries.

Augmented Data: Each molecule has different SMILES and we follow Igor' augmentation strategy to get Aug × N dataset [7]. The method for generating random smiles here is the same as that in Sect. 3.2, but the difference lies in the fact that they are generated in pairs to increase the data variety in the fine-tune process.

Aligned Data: We follow Zhong's strategy to get Align × N SMILES [8]. In the data alignment process, the product molecule uses an enumeration method to randomly generate product SMILES. The new representation of the reactant is obtained by using the atomic mapping between the reactant and product pairs, instead of enumerating randomly.

Mix Data: We swap the positions of the product and reactant in the Align × N dataset and add a prefix '.' in the reactant; and then merge the new Align × N dataset with the original Align × N dataset to form the Mix × 2N dataset. [7]

Model Training. In this study, during the fine-tuning phase, the model retains the exact same architecture as that of the pre-training phase. Given the product molecules as input, the model predicts all corresponding reactants. The loss function [19] for a batch of N_b samples is:

$$L(\hat{\mathbf{Y}}, \mathbf{Y}) = -\frac{1}{N_b} \sum_{b=1}^{N_b} \sum_{t=1}^{T} Y_{bt} \log(\hat{Y}_{bt}) \tag{2}$$

where L denotes the cross-entropy loss, and Y_{bt} is the label, and $\hat{Y}_{bt}$ is the output of the model, and T is the number of classes for characters in SMILES. The evaluation method used for our model is the top-k accuracy metric.

4 Experiments and Results

4.1 Performance Evaluation

To prove our hypothesis that incorporation of domain-specific prior knowledge from chemistry can enhance the model's performance on the task of retrosynthesis, we deployed SMART-RetroNet on the USPTO-50K dataset. For the pre-training process, we used the combination methods of atom deletion, bond deletion, random SMILES to augment the molecule dataset. The pre-training process took a total of 150000 iterations with batch size of 64, requiring 15 h to finish on a single NVIDIA GeForce RTX 3090 GPU. For the fine-tuning process, we froze the embedding layer and three encoder blocks close to the input of the pre-trained transformer. The fine-tuning process run 60000 steps on Mix-40 dataset with a batch size of 128, taking 6 h on the same device. As shown in Table 1 our model achieved a Top-1 accuracy of 53.31%, which is comparable to ChemFormer, while our Top-10 accuracy of 90.05% surpasses that of ChemFormer. Due to the prior knowledge acquired during the pre-training process, which is successfully integrated into downstream tasks, our model has achieved commendable performance. This integration of learned prior knowledge has enhanced generalization and adaptation to challenges presented by the retrosynthesis tasks.

Table 1. Performance comparison of models for one-step retrosynthesis.

Models	Top-1	Top-3	Top-5	Top-10
RetroPrime	51.40%	70.80%	74.00%	76.10%
RetroFormer	53.20%	71.10%	76.60%	82.10%
ChemFormer	**53.60%**	58.30%	61.10%	61.70%
ours	53.31%	**77.09%**	**84.58%**	**90.05%**

4.2 Pre-training Model Stability Analysis

Model Performance During Pre-training. To evaluate the training result of the pre-trained model on the task of SMILES transformation, we tested the pre-trained model using the molecule testset after the pre-training with fully hybrid augmentation. On test set, the model's prediction accuracy for random SMILES is 98.06%, 95.56%, 99.08% and 99.24% for top-1, top-3, top-5 and top-10, respectively. This result indicates that the model has already acquired fundamental prior knowledge corresponding to SMILES transformations during the pre-training phase, including atom types, functional group priorities, chirality, and stereochemical rules, all of which play a pivotal role in various tasks within the chemical domain.

To prove that our pre-training task can enhance performance on downstream retrosynthesis task, we fine-tuned the pre-trained model under fine-tuning condition in Sect. 4.1 (without freezing layers) on different datasets of Canonical, Aug × 5, Aug × 20, Align × 5, Align × 20 and Mix × 40. We chose the same transformer model without pre-train as the baseline model. The results of the experiment are shown in Table 2. Compared to the baseline model, our model shows performance improvements in four datasets.

Table 2. Evaluation results of fine-tune phase for one-step retrosynthesis.

Dataset	Models	Top-1/%	Top-2/%	Top-3/%	Top-5/%	Top-10/%
Canonical	baseline	43.68 ± 0.52	56.78 ± 0.42	62.80 ± 0.53	69.86 ± 0.62	76.33 ± 0.72
Canonical	ours	43.72 ± 0.14	59.16 ± 0.26	65.43 ± 0.30	63.88 ± 0.35	81.35 ± 0.45
Aug × 5	baseline	20.79 ± 0.67	26.68 ± 0.86	30.44 ± 0.85	34.41 ± 0.95	38.59 ± 1.01
Aug × 5	ours	45.84 ± 0.24	58.98 ± 0.33	65.69 ± 0.32	72.70 ± 0.42	78.73 ± 0.50
Aug × 20	baseline	6.35 ± 0.65	8.69 ± 0.66	10.17 ± 0.80	11.69 ± 0.97	14.32 ± 1.15
Aug × 20	ours	45.98 ± 0.20	59.34 ± 0.25	65.87 ± 0.32	72.30 ± 0.37	78.41 ± 0.40
Align × 5	baseline	47.93 ± 0.29	63.99 ± 0.33	70.89 ± 0.48	78.49 ± 0.49	85.64 ± 0.61
Align × 5	ours	51.61 ± 0.27	65.93 ± 0.31	72.74 ± 0.40	78.87 ± 0.43	84.32 ± 0.49
Align × 20	baseline	50.69 ± 0.14	65.79 ± 0.26	73.34 ± 0.32	80.95 ± 0.37	88.10 ± 0.40
Align × 20	ours	52.17 ± 0.14	67.15 ± 0.27	74.28 ± 0.33	80.91 ± 0.55	86.24 ± 0.42
Mix × 40	baseline	48.57 ± 0.15	64.69 ± 0.28	73.34 ± 0.34	81.71 ± 0.46	88.58 ± 0.48
Mix × 40	ours	52.63 ± 0.26	68.74 ± 0.30	75.71 ± 0.35	83.24 ± 0.45	89.00 ± 0.53

Contribution of Data Augmentation. To quantify the impact of our data augmentation methods used in our pre-training task on the downstream task, we compared the performance of models trained with and without augmented data (see Fig. 3). We conducted experiments on Mix × 40 dataset under the same experimental conditions as described in Sect. 4.1 (without freezing the parameters). Without data augmentation, the pre-trained model achieves a top-1 accuracy of 46% for chemical reaction task prediction. However, by adopting the strategies of selecting substructures, randomly deleting atoms, randomly deleting bonds, randomizing SMILES, and the fully hybrid, the yield top-1 accuracies are 49.01%, 52.37%, 52.17%, 53.35%, respectively. This demonstrates that the data augmentation used in our pre-training task has enhanced the model's performance on downstream task. It can be attributed to the following reasons. First, at the level of data distribution, by deleting local structures to form new molecules, the diversity of the data is increased, allowing the model to learn a wider range of features to avoid overfitting. Second, since randomly deleted local structures could be key features, such as functional groups, our data augmentation methods enable the model to identify and focus on these key features to

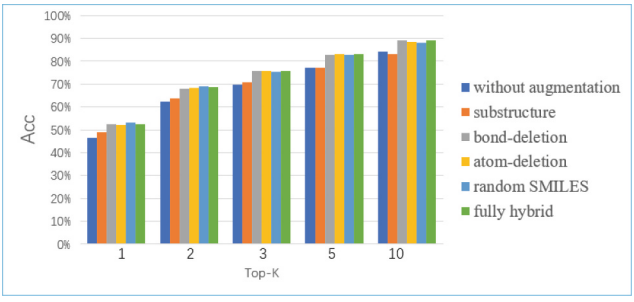


Fig. 3. The contribution of augmentation methods during pre-training phase.

get better understanding of molecule structure. These precisely correspond to the learning process of prior chemistry knowledge. However, too much deletion of original structure may decrease the effect of augmentation. So our suggested augmentation methods are bond-deletion, atom-deletion and random SMILES, or their combinations.

4.3 Performance Analysis by Similarity Score

Comparing Mix × 40 with Align × 20 from Table 2, we observed that the performance of our model increased from 52.17% to 52.63%, while the baseline model decreased from 50.69% to 48.57%. Additionally, we noted that for a chemical reaction represented in SMILES in the fine-tuning task dataset USPTO-50K, there are some similar fragments present in both the reactants and products, leading us to speculate that these similar fragments in reactant-product pairs may have a certain impact on the retrosynthesis prediction. To prove this, we adopted sequence alignment inspired by bioinformatics to evaluate the similarity between reactants and product SMILES strings [21].

To apply sequence alignment to SMILES, we treat each SMILES string as a sequence of characters, which denote atoms, double bonds, chirality center signs, and so on. Each token can represent an atom, or a bond. The alignment is performed at character level to exhibit the minimal difference between two strings. According to the alignment result, we can get two new strings of R' and P' (see Fig. 4). Let M denote a match between two characters, and D denote a mismatch.

The similarity score is a measure that represents the proportion of matching states in alignment of product and reactant strings in a reaction (see Fig. 4), averaged across an entire dataset. By this similarity score, we can quantitatively assess the difficulty of molecule generation.

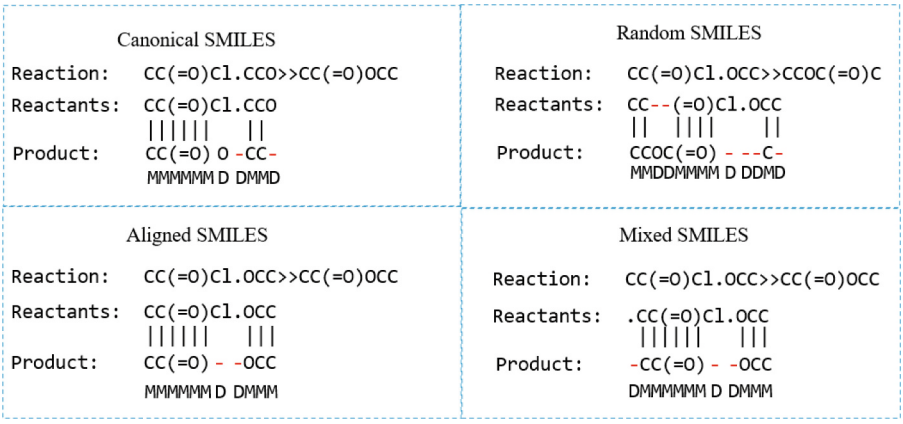


Fig. 4. Examples of different dataset and matching states between aligned product and reactants.

To gain a better understanding of the disparities among the datasets constructed before, we compute the average similarity score between product and reactants. The results of these calculations are presented in Table 3.

Table 3. The similarity score of reaction datasets

Dataset	Canonical	Aug × 5	Aug × 20	Align × 5	Align × 20	Mix × 40
Score	0.6280	0.4526	0.4196	0.7248	0.7193	0.7106

As can be seen in Table 3, the similarity between the product SMILES and reactant SMILES in the canonical USPTP-50K dataset is 0.6280. After augmentation using the random SMILES strategy, the similarity score decreased to the range of 0.42–0.45. After augmentation through data alignment, the similarity score increased to approximately 0.72. After mixing the datasets, the similarity score was approximately 0.71. Therefore, the random SMILES augmentation and mixing strategy reduces the similarity between reactant and product molecules, while data alignment enhances the similarity between them.

The baseline model's Top-1 accuracy dropped from 45% to 6%–20% when random SMILES data augmentation reduced data similarity. Due to the increase in similarity, the Align × 5 dataset has mitigated the problem of reduced prediction accuracy. Our framework, however, solves this problem by learning a more general representation.

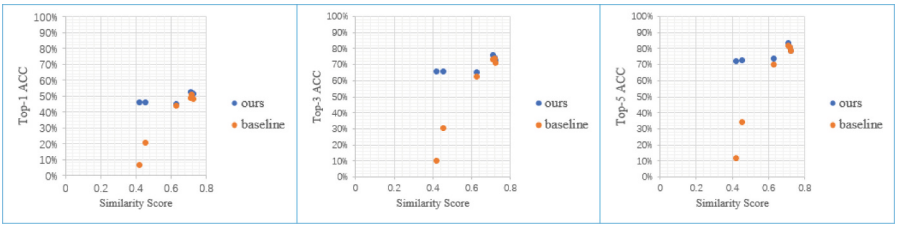


Fig. 5. Performance of model with respect to the similarity score of reaction dataset.

Figure 5 shows the relationship between the model's top-k performance and the dataset similarity scores. The result reveals that the performance of the baseline model is roughly positively correlated to the datasets' similarity score. Without prior knowledge, baseline models trained on datasets with high similarity may treat the structure of the product sequence as a template to be applied during the generation of the reactant sequence (see Fig. 4), which could lead the model to focus excessively on sequence matching within the dataset and neglect a deeper understanding of the intrinsic molecular structure. However, our model, having learned prior knowledge during the pre-training phase, is less affected by the similarities within the dataset during fine-tuning. Consequently, our model demonstrates an improved performance on the Mix × 40 dataset compared to the Align × 20 dataset because of better understanding of molecule.

4.4 Layer Freezing in Fine-Tuning

To further analyze how the prior knowledge contributes to its performance during fine-tuning, we employed a layer-freezing fine-tuning strategy. Specifically, we observed that initially freezing the lower encoder layers and progressively unfreezing them during fine-tuning results in a slight performance improvement. (see Fig. 6) This is likely because the lower layers capture general molecular representations that are beneficial for the task. However, fully unfreezing all layers eventually leads to performance decline.

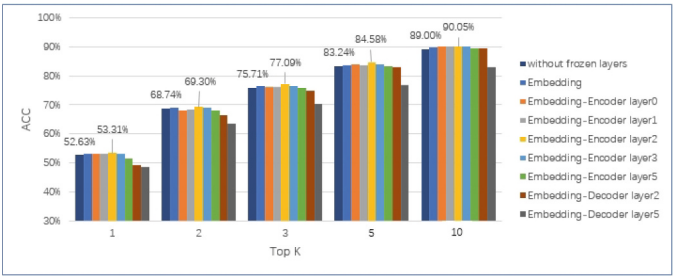


Fig. 6. Performance with frozen layers.

5 Discussion

Importance of Prior Knowledge. Our pre-training task inherently helps to encode valuable chemical knowledge into the model. The SMILES representation, with its various encoding possibilities for the same molecule, provides a rich training signal. Through this process, the model develops a strong Functional Group Awareness. The heatmap in Fig. 7 illustrates that, for different SMILES representations of the same molecule, the attention scores in the first block of the model's encoder are higher for functional group compared to non-functional group parts, which is the ester group. This knowledge aids in identifying potential reaction sites and predicting likely reactants during retrosynthesis. Furthermore, it also helps to learn Long-Range Dependencies in SMILES strings, allowing it to understand the relationships between distant atoms or substructures within a molecule.

Comparison Between Graph-Based Models and Transformer-Based Models. Both Transformer-based models and graph-based models aim to capture the relationships between atoms in a molecule. GNNs explicitly treat a molecule as a graph and directly learn the atomic and bond relationships to understand its structure. In contrast, Transformers learn these relationships implicitly from SMILES strings via global self-attention. Although the learning is indirect, it is more convenient for model design and data augmentation.

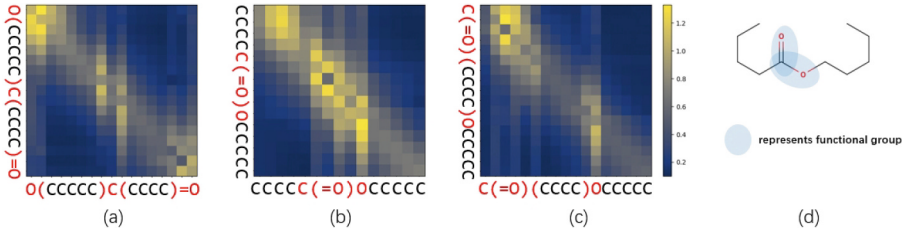


Fig. 7. Visualization of attention scores in the first block of encoder. (a) (b) and (c) are three different SMILES representations of pentyl pentanoate, whose chemical structure is shown in (d).

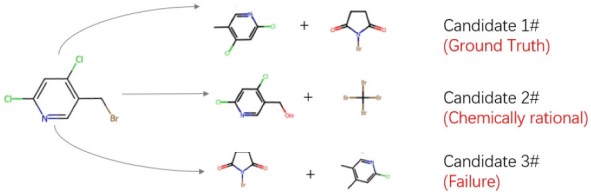


Fig. 8. Retrosynthesis prediction of a molecule in real-world scenario.

Generalization Evaluation. Figure 8 demonstrates a real-world pharmaceutical intermediate synthesis involving a multi-component cascade reaction. Our model successfully predicted the correct reaction product (candidate 1#) as the ground truth. Additionally, it generated a chemically rational candidate solution (candidate 2#). This result suggests that it can effectively capture reaction patterns in molecule structure.

Failure Cases. A failure case is also provided in Fig. 8. Candidate 3# contains a correct reactant and an erroneous one with a Chlorine atom missing. Due to such a detailed flaw, this reaction cannot produce the desired outcome. The reason is that it is easier for a model to learn general structures in molecules with the assistance of prior knowledge. However, the model is not guaranteed to capture the rigorous chemical rules dominating the molecule conformation and reactions. To resolve this issue, special design is required to drive the model to obey such unbreakable rules.

6 Conclusion

We have proposed a novel framework of SMART-RetroNet in this study. Specifically, we introduced a pre-training task of mapping different SMILES representations, during which the model could successfully learn fundamental prior knowledge in chemistry field. Our experimental findings demonstrate that the knowledge acquired during pre-training effectively transfers to single-step chemical reaction retrosynthesis tasks during fine-tuning, enhancing model performance. This study not only delivers an effective approach for chemical reaction

retrosynthesis prediction but also find a way on the transferability of prior knowledge in related chemistry tasks.

Acknowledgments. This work was supported in part by the University of Jinan Disciplinary Cross-Convergence Construction Project 2023 (XKJC-202308), and in part by Shandong provincial Natural Science Foundation, China (ZR2024MF124, ZR2021MF036).

References

1. Long, L., Li, R., Zhang, J.: Artificial intelligence in retrosynthesis prediction and its applications in medicinal chemistry. J. Med. Chem. (2025)
2. Gu, S., Zhe, W., Ho, T.: The combination of molecular dynamics simulations and machine learning for applications in the field of drug design. Mod. Appl. Pharm. China (2022)
3. DiMasi, J., Grabowski, G., Hansen, R.: Innovation in the pharmaceutical industry: new estimates of R&D costs. J. Health Econ. **47**, 20–33 (2016)
4. Zhu, Y., Du, Y., Wang, Y.: A survey on deep graph generation: methods and applications. In: Learning on Graphs Conference, p. 47-1. PMLR (2022)
5. Schneider, N., Stiefl, N., Landrum, G.: What's what: the (nearly) definitive guide to reaction role assignment. J. Chem. Inf. Model. **56**(12), 2336–2346 (2026)
6. Jin, W., Coley, C., Barzilay, R.: Predicting organic reaction outcomes with Weisfeiler-Lehman network. In: Advances in Neural Information Processing Systems (2017)
7. Tetko, I. V., Karpov, P., Van Deursen, R.: State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis. Nat. Commun. **11**(1), 5575(2020)
8. Zhong, Z., Song, J., Feng, Z.: Root-aligned SMILES: a tight representation for chemical reaction prediction. Chem. Sci. **13**(31), 9023–9034 (2022)
9. Schwaller, P., Laino, T., Gaudin, T.: Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction. ACS Cent. Sci. **5**(9), 1572–1583 (2019)
10. Schwaller, P., Gaudin, T., Lanyi, D.:"Found in translation": predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models. Chem. Sci. **9**(28), 6091–6098 (2018)
11. Wang, X., Li, Y., Qiu, J.: RetroPrime: a diverse, plausible and Transformer-based method for single-step retrosynthesis predictions. Chem. Eng. J. **420**, 129845 (2021)
12. Kim, E., Lee, D., Kwon, Y.: Valid, plausible, and diverse retrosynthesis using tied two-way transformers with latent variables. J. Chem. Inf. Model. **61**(1), 123–133 (2021)
13. Wan, Y., Hsieh, C. Y., Liao, B.: Retroformer: pushing the limits of end-to-end retrosynthesis transformer. In: International Conference on Machine Learning, pp. 22475–22490. PMLR (2022)
14. Han, K., Wang, Y., Chen, H.: A survey on vision transformer. IEEE Trans. Pattern Anal. Mach. Intell. **45**(1), 87–110 (2022)
15. Fang, A., Ilharco, G., Wortsman, M.: Data determines distributional robustness in contrastive language image pre-training (CLIP). In: International Conference on Machine Learning, pp. 6216–6234. PMLR (2022)

16. Kenton, J., Toutanova, L.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT, pp. 1–2 (2019)
17. Radford, A., Narasimhan, K., Salimans, T.: Improving language understanding by generative pre-training (2018)
18. Irwin, R., Dimitriadis, S., He, J.: Chemformer: a pre-trained transformer for computational chemistry. Mach. Learn. Sci. Technol. **3**(1), 015022 (2022)
19. De Boer, P., Kroese, D., Mannor, S.: A tutorial on the cross-entropy method. Ann. Oper. Res. **134**(1), 19–67 (2005)
20. Lowe, D. M. Extraction of chemical structures and reactions from the literature (2012)
21. Jararweh, Y., Al-Ayyoub, M., Fakirah, M.: Improving the performance of the Needleman-Wunsch algorithm using parallelization and vectorization techniques. Multimedia Tools Appl. **78**, 3961–3977 (2019)

Few-Shot Learning for Syndrome Differentiation with Two Prompts

Xian Zhou[1], Sophia Yat Mei Lee[2], Xinhe Gang[1], Yichen Yang[1], Yu Liu[3], Shengfeng Ju[3], and Shoushan Li[1(✉)]

[1] Natural Language Processing Lab, Soochow University, Suzhou 215006, China
20234227070@stu.suda.edu.cn, lishoushan@suda.edu.cn
[2] The Hong Kong Polytechnic University, Hong Kong, China
[3] Changshu Rural Commercial Bank, Suzhou 215500, China

Abstract. Currently, the majority of approaches to syndrome differentiation in Traditional Chinese Medicine (TCM) conceptualize this endeavor as a text classification challenge. The state-of-the-art technique in text classification is the few-shot fine-tuning approach grounded in large language models. This approach minimizes the need for extensive manual annotation while yielding remarkable classification outcomes with a limited number of labeled samples. In this paper, we propose a few-shot fine-tuning approach that integrates two distinct types of prompts tailored for the syndrome differentiation task. One type of prompt directly inquires about the syndrome type, while the other indirectly seeks to infer the syndrome type. Specifically, our approach initially adopts the prefix-tuning approach, which utilizes two prefix vectors to combine these two types of prompts. Subsequently, we fine-tune the prefix-related parameters and combine the results from the two prefix fine-tunings to generate the final output. Experimental results demonstrate the superiority of our proposed approach.

Keywords: Few-shot learning · Syndrome differentiation · Traditional Chinese Medicine

1 Introduction

Syndrome differentiation in Traditional Chinese Medicine (TCM) involves a comprehensive analysis of a patient's clinical details, including symptoms, physical manifestations, medical background, and other relevant factors, to determine the individual's specific pathological state. This process is pivotal in TCM practice, as it guides the selection of appropriate treatment strategies. Natural Language Processing (NLP)-based syndrome differentiation leverages NLP techniques to analyze and interpret a patient's clinical data, thereby identifying the underlying pathological condition. For instance, as shown in Fig. 1, the input of the NLP-based syndrome differentiation approach is a patient's clinical symptoms, and the output is the corresponding syndrome differentiation type, such as "*Phlegm-dampness amassing in lung* (痰湿蕴肺证)". By employing NLP techniques in

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 39–50, 2026.
https://doi.org/10.1007/978-3-032-04555-3_4

syndrome differentiation, TCM practitioners can enhance both the precision and speed of their diagnostic procedures, ultimately contributing to better patient care and outcomes.

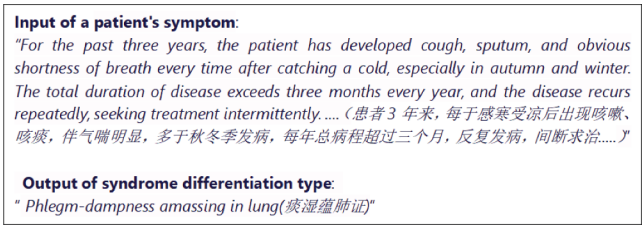


Fig. 1. An example of syndrome differentiation based on patient's clinical symptoms.

Syndrome differentiation can be regarded as a text classification task. In this paper, we address the challenges of TCM syndrome differentiation, which include high data annotation costs and complex dialectical logic. In small-sample scenarios, traditional machine learning approaches struggle to capture the deep associations between symptoms and syndromes. The current state-of-the-art approach for text classification is few-shot fine-tuning based on large language models (LLMs). For example, fine-tuning techniques such as LoRA [1] or prefix-tuning [2] enable effective training of LLMs with a limited number of labeled samples. Compared with previous fully supervised learning approaches like BERT, this approach achieves comparable performance while requiring significantly fewer manually annotated samples. As a result, it drastically reduces dependence on extensive manual annotation, lowers costs, and enables rapid adaptation across different domains.

However, the effectiveness of fine-tuning approaches based on large language models often hinges on human-designed prompts. In the context of syndrome differentiation, specifically for distinguishing between *Deficiency* (虚) and *Excess* (实) syndromes based on a patient's symptoms, we can design a direct prompt: "*Which type of syndrome in Traditional Chinese Medicine (TCM) does this symptom belong to? A. Deficiency Syndrome B. Excess Syndrome.*" Alternatively, an indirect approach can be employed by focusing on the treatment strategy. For instance, in TCM theory, tonification is used to treat *Deficiency* syndromes, while purgation is used for *Excess* syndromes. Therefore, we can devise an indirect prompt that indicates the syndrome type, and an indirect prompt could be: "*Which treatment approach in TCM is needed for this syndrome? A. Tonification B. Purgation.*" This raises the question: which type of prompt is more effective for distinguishing between *Deficiency* and *Excess* syndromes in TCM?

In this paper, we propose a few-shot fine-tuning approach for the syndrome differentiation task, leveraging the fine-tuning capabilities of large language models. Our approach integrates two distinct prompts - one direct and one indirect

- using prefix-tuning technology. First, two prefix vectors are constructed for the two prompts, respectively. Second, the weight parameters of the transformer model are trained separately for each prompt and its respective prefix vector. To reduce the cost of fine-tuning, we fix the weight parameters of the original transformer model and only adjust the weight parameters corresponding to the two prefix vectors. Finally, the results from the fine-tuning of the two prefixes are fused to produce the final classification outcome. Experimental results demonstrate that our approach achieves better classification performance in few-shot learning tasks for syndrome differentiation compared to single-prompt approaches and other existing few-shot learning approaches.

2 Related Work

2.1 Syndrome Differentiation in TCM

In the study of syndrome differentiation in TCM, text classification approaches are commonly employed and can be categorized into two main types: using non-pretrained models and pretrained models.

As for non-pretrained models, Pang et al. (2020) [3] propose an attention mechanism-enhanced multilayer perceptron model (ATT-MLP) for syndrome differentiation TCM. Chen et al. (2021) [4] propose the S-TextBLCNN model, which employs deep learning and natural language processing to classify the efficacy of TCM formulas. Huang et al. (2022) [5] propose a novel intelligent dialectical model that utilizes a convolution neural network to address the high-dimensional sparse vector classification task of syndrome differentiation.

As for pre-trained models, Peng et al. (2019) [6] propose the Biomedical Language Understanding Evaluation (BLUE) benchmark, which includes five tasks and ten datasets aimed at advancing pre-training language representations in the biomedicine domain. Song et al. (2019) [7] propose a multi-layer semantic-enhanced framework that combines character-level BERT with an optimized Text-CNN for accurate classification of Traditional Chinese Medicine medical cases, where instance- and attribute-level semantic expansion enriches case information, the character-level BERT encodes texts, and the refined Text-CNN delivers superior performance over alternative representation-classifier combinations. Ren et al. (2022) [8] propose the TCM-SD benchmark, the first large-scale public dataset for syndrome differentiation in TCM, and introduce a domain-specific pre-trained language model, ZY-BERT, to enhance the performance of deep neural networks. Chen et al. (2024) [9] propose the TCM-BERT-CNN model, a deep learning-based holistic syndrome differentiation framework that utilizes BERT [10] and CNN to classify multiple TCM syndromes.

2.2 Few-Shot Learning in Text Categorization

For few-shot classification, Lei et al. (2023) [11] propose the Task-Adaptive Reference Transformation (TART) network to enhance few-shot text classification by transforming class prototypes into task-adaptive fixed reference points in

metric spaces. Pauli et al. (2023) [12] propose AncSetFit, an efficient few-shot classification approach that employs sentence embeddings and contrastive learning with triplet loss in low-data scenarios. Varia et al. (2023) [13] propose a unified framework for Aspect-based Sentiment Analysis (ABSA) that fine-tunes a T5 model with instructional prompts in a multi-task few-shot learning approach. Nguyen et al. (2024) [14] propose a benchmarking approach to evaluate three verbalizer baselines for topic classification in few-shot learning. Wang et al. (2024) [15] propose a retrieval framework for few-shot aspect-based sentiment analysis (ABSA) that enhances in-context learning (ICL) effectiveness by utilizing contrastive learning to simultaneously consider overall semantics. Lee et al. (2024) [16] propose SuperST, a self-training framework for few-shot text classification that employs superficial learning to selectively acquire necessary context from noisy pseudo-labels while fine-tuning on less noisy data.

Specifically, regarding the fine-tuning techniques for few-shot learning with large models, Li et al. (2021) [2] propose prefix tuning, a lightweight alternative to fine-tuning for natural language generation tasks. Hu et al. (2021) [1] propose Low-Rank Adaptation (LoRA), an approach that freezes pre-trained model weights and incorporates trainable rank decomposition matrices into transformer layers.

3 Our Approach

The input for the syndrome differentiation task consists of patient information, such as symptoms and signs, represented as x. The output of the task is the syndrome differentiation type, such as *Deficiency* (虚) and *Excess* (实), represented as y. In this study, we assume that x is textual information and y is syndrome category information. Therefore, this task can be defined as a classic text classification problem. Our approach ensembles two prompts using prefix-tuning fine-tuning with the QWen model [17], which is a large language model. The overall framework of this approach is illustrated in Fig. 2.

3.1 Input Layer of the Model

Our approach has two types of inputs. The first type consists of the input text x and two manually designed templates, denoted as $Prompt_1$ and $Prompt_2$, i.e.,

$$X_1, X_2 = [x, Prompt_1], [x, Prompt_2]$$

Then, the inputs X_1 and X_2 are encoded through two word embedding layers to obtain word vectors E_1, E_2.

$$E_1, E_2 = \text{Embedding}_{\text{QWen}}(X_1), \text{Embedding}_{\text{QWen}}(X_2)$$

Where $\text{Embedding}_{\text{QWen}}$ refers to the embedding layer of the QWen model.

The second type of input consists of a set of trainable prefix vectors, which allows the pre-trained language model to adapt to specific tasks without modifying its original parameters.

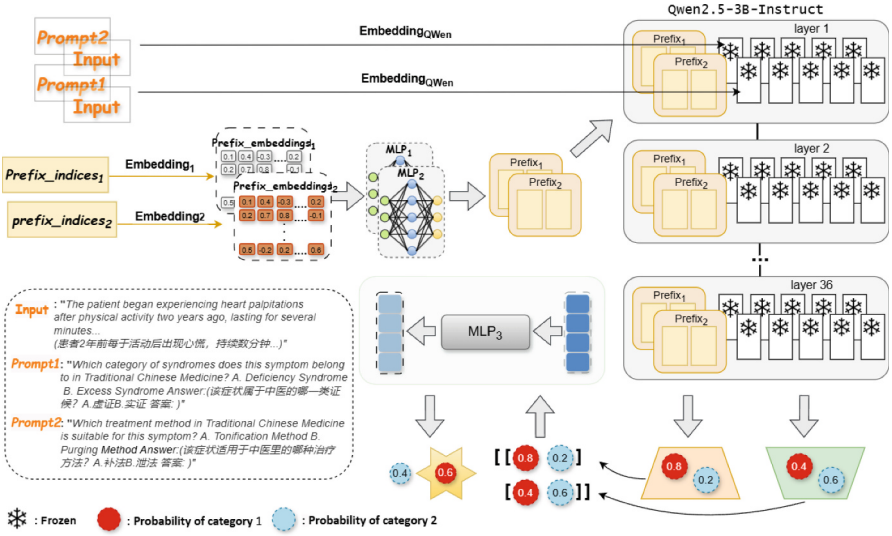


Fig. 2. The overall framework of our approach.

First, we construct two prefix sequences $Prefix_indices_1$, $Prefix_indices_2$, and obtain the embeddings through the word embedding layer, i.e.,

$$Prefix_embeddings_1 = \text{Embedding}_1(Prefix_indices_1)$$

$$Prefix_embeddings_2 = \text{Embedding}_2(Prefix_indices_2)$$

Where Embedding_1 and Embedding_2 are different embedding layers.

The prefix embeddings $Prefix_embeddings_1$ and $Prefix_embeddings_2$ are input into their respective fully connected layers (MLP layers) to obtain new prefix representations $Prefix_1, Prefix_2$.

$$Prefix_1 = \text{MLP}_1(Prefix_embeddings_1)$$

$$Prefix_2 = \text{MLP}_2(Prefix_embeddings_2)$$

3.2 Intermediate Layers of the Model

The intermediate layers of the model utilize a transformer architecture, which consists of two parts. The first part is the original transformer layers of the QWen model. Specifically, according to the calculation approach of attention [18], the input representations E_1 and E_2 are transformed through a linear transformation to generate the K and V matrices.

The second part consists of the transformer layers for the newly added prefix section. Specifically, according to the calculation approach of attention [18],

the input representation vectors $Prefix_1$ or $Prefixs_2$ are decomposed into two parts, K_p and V_p.

The matrices from these two parts are then concatenated, i.e.,

$$K' = \begin{bmatrix} K_p \\ K \end{bmatrix}, V' = \begin{bmatrix} V_p \\ V \end{bmatrix}$$

The QWen transformer employs a multi-head attention mechanism, which splits Q, K', V' into multiple heads for parallel computation, and then concatenates the results.

$$MultiHead(Q, K', V') = Concat(head_1, ..., head_h)W^O$$

Where Q, K', V' represent the query, key, and value vectors, respectively. h denotes the number of heads; $head_i$ represents the output of the i-th head. W_O is the output transformation matrix.

The output $head_i$ of each head can be expressed as follows,

$$head_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Where QW_i^Q, KW_i^K, VW_i^V represent the query, key, and value transformation matrices for the i-th head, respectively. The attention function is denoted as Attention [18].

During the fine-tuning phase, we freeze the parameters of the original transformer in the QWen model, such as K and V, while adjusting only the smaller parameter quantities from the second step, such as K_P and V_P.

3.3 Model Output

The newly trained model generates a probability distribution for predicting the next word after training:

$$P(Next_token = t_{next}|X_1) = \text{QWen}(Prefix_1, E_1)$$

$$P(Next_token = t_{next}|X_2) = \text{QWen}(Prefix_2, E_2)$$

To obtain the final classification label, we define a mapping function M that maps the task categories c to the corresponding words v in the model's vocabulary.

$$M : c \rightarrow v$$

For example, in a binary classification task between *Excess* syndrome and *Deficiency* syndrome, since our prompt involves selecting A or B to determine whether it is an *Excess* syndrome or a *Deficiency* syndrome, we can map the *Deficiency* syndrome category to "A" in the vocabulary and the *Excess* syndrome category to "B" in the vocabulary. Finally, the probability distribution of X_1 or X_2 belonging to category c is determined by the following formula, i.e.,

$$P_1 = P(Next_token = M(c)|X_1)$$

$$P_2 = P(Next_token = M(c)|X_2)$$

To combine the two results mentioned above, we concatenate the category probability distributions P_1 and P_2 obtained from the two models, i.e.,

$$P_{Concat} = \text{Concat}(P_1, P_2)$$

The concatenated vector is passed through a fully connected classification layer to obtain the probability of x belonging to category c, i.e.,

$$P(c|x) = \text{MLP}_3(P_{Concat})$$

4 Experiment

In this section, we systematically validate the effectiveness of the proposed approach.

4.1 Experimental Setting

We conduct our empirical study using the dataset used in [8], which represents the first public large-scale syndrome differentiation benchmark. However, we observe some intersections and overlaps among the syndrome categories in this dataset. For example, two categories in the classification system provided by the paper "*Liver-Stomach Disharmony*" and "*Liver Qi Invading Stomach*" are essentially identical dialectical types. Therefore, to address this issue and reduce category overlap, we reorganize the dialectical types based on the criteria of *Deficiency* (虚) and *Excess* (实) from [19], categorizing all dialectical types into two groups: *Deficiency* (虚) and *Excess* (实). For instance, the category "*Liver and kidney yin Deficiency*" is classified as *Deficiency* (虚), while "*Rheumatic depression*" is classified as *Excess* (实). The motivation for adopting the *Deficiency* (虚) and *Excess* (实) differentiation lies in its significance in TCM, as it is widely applicable in both acupuncture treatments and herbal medicine.

Experimental Setup: For the training set, we randomly select training sets of sizes 16, 32, and 64 for each category. The number of instances in the development sets is equal to that of the training set. The test set consists of a fixed number of 400 instances (200 for *Deficiency* (虚) syndrome and 200 for *Excess* (实) syndrome). The evaluation metrics used are standard accuracy, precision, recall, and F1 score.

For comparison, we implement the following baseline approaches:

(1) GPT-3.5 [20]: This is a zero-shot learning approach that does not use any training samples. We utilize GPT-3.5 for text classification. The prompt used for classification is: "*Does this symptom belong to the Deficiency syndrome or the Excess syndrome in traditional Chinese medicine? The answer must be Deficiency syndrome or Excess syndrome*".
(2) BERT [10]: This is a supervised learning approach that employs an autoencoding pretrained approach. We use the 6-layer BERT-base model.

(3) QWen [17]: This is a prompt-based few-shot learning approach that incorporates an autoregressive pretrained approach. The size of the learning model is 1.5b.
(4) ZY-BERT [8]: This is a supervised learning approach that involves retraining a BERT model specifically on a large-scale TCM text corpus.
(5) SETFIT [21]: This is an efficient approach designed for fine-tuning sentence transformers (ST) in few-shot learning tasks. The approach first fine-tunes a pretrained sentence transformer model on a small number of text pairs using a contrastive learning approach in a Siamese network setup. Then, the fine-tuned model generates text embeddings, which are used to train a classification head.
(6) TRD-FSL [22]: This approach uses the pretrained ELECTRA, and it reformulates tasks as token-replacement detection problems for few-shot learning.
(7) DLM-CSC [23]: This approach leverages the autoencoding pretrained ELECTRA model to perform few-shot text classification based on semantic consistency and outperforms existing approaches on multiple tasks.
(8) Our Approach: This is our approach. During the fine-tuning process, we set the number of training epochs to 20, the learning rate to 5e-5, and the batch size to 1. The fine-tuning is conducted on a machine with a single 4090 GPU. In our approach, two different prompts are used, i.e., $Prompt_1$ and $Prompt_2$. $Prompt_1$ is "*Which category of syndromes does this symptom belong to in Traditional Chinese Medicine? A. Deficiency Syndrome B. Excess Syndrome Answer:*". $Prompt_2$ is "*Which treatment approach in Traditional Chinese Medicine is suitable for this symptom? A. Tonification approach B. Purging approach Answer:*".

4.2 Experimental Results

Table 1. The experimental results of baseline approaches and our approach with 16 labeled samples in each category

Approach	*Deficiency* (虚)			*Excess* (实)			Total	
	Precision	*Recall*	*F1*	*Precision*	*Recall*	*F1*	*Accuracy*	*Macro-F1*
GPT-3.5	0.396	0.440	0.417	0.370	0.330	0.349	0.385	0.383
BERT	0.754	0.460	0.571	0.612	0.850	0.711	0.655	0.641
QWen($Prompt_1$)	0.712	0.545	0.618	0.632	0.780	0.698	0.663	0.658
QWen($Prompt_2$)	0.682	0.525	0.593	0.614	0.755	0.677	0.640	0.635
ZY-BERT	**0.802**	0.445	0.572	0.615	**0.890**	**0.728**	0.668	0.650
SETFIT	0.611	0.575	0.593	0.599	0.635	0.617	0.605	0.605
TRD-FSL	0.687	0.505	0.582	0.609	0.770	0.680	0.638	0.631
DLM-SCS	0.605	0.627	0.616	0.640	0.618	0.629	0.623	0.622
Our approach	0.678	**0.685**	**0.682**	**0.682**	0.675	0.678	**0.680**	**0.680**

Table 2. The experimental results of baseline approaches and our approach with 32 labeled samples in each category

Approach	*Deficiency* (虚)			*Excess* (实)			Total	
	Precision	*Recall*	*F1*	*Precision*	*Recall*	*F1*	*Accuracy*	*Macro-F1*
GPT-3.5	0.396	0.440	0.417	0.370	0.330	0.349	0.385	0.383
BERT	0.649	0.740	0.692	0.697	0.600	0.645	0.670	0.668
QWen($Prompt_1$)	0.833	0.525	0.644	0.653	**0.895**	**0.755**	0.710	0.700
QWen($Prompt_2$)	0.687	0.615	0.649	0.652	0.720	0.684	0.668	0.667
ZY-BERT	0.727	0.745	0.736	0.739	0.720	0.729	0.733	0.733
SETFIT	0.684	0.670	0.677	0.677	0.690	0.683	0.680	0.680
TRD-FSL	0.711	0.750	0.730	0.735	0.695	0.715	0.723	0.722
DLM-SCS	**0.865**	0.627	0.727	0.485	0.782	0.598	0.675	0.663
Our approach	0.731	**0.800**	**0.764**	**0.779**	0.705	0.740	**0.753**	**0.752**

Table 3. The experimental results of baseline approaches and our approach with 64 labeled samples in each category

Approach	*Deficiency* (虚)			*Excess* (实)			Total	
	Precision	*Recall*	*F1*	*Precision*	*Recall*	*F1*	*Accuracy*	*Macro-F1*
GPT-3.5	0.396	0.440	0.417	0.370	0.330	0.349	0.385	0.383
BERT	0.713	0.795	0.752	0.768	0.680	0.722	0.738	0.737
QWen($Prompt_1$)	0.761	0.795	**0.778**	0.785	0.750	0.767	0.773	0.772
QWen($Prompt_2$)	0.751	0.740	0.746	0.744	0.755	0.750	0.748	0.748
ZY-BERT	0.706	0.815	0.756	0.781	0.660	0.715	0.738	0.736
SETFIT	0.706	0.770	0.737	0.747	0.680	0.712	0.725	0.724
TRD-FSL	0.724	**0.825**	0.771	**0.797**	0.685	0.737	0.755	0.754
DLM-SCS	0.740	0.736	0.738	0.735	0.739	0.737	0.738	0.738
Our approach	**0.818**	0.720	0.766	0.750	**0.840**	**0.793**	**0.780**	**0.779**

Tables 1, 2 and 3 present the classification performance of various approaches across different sizes of training sets. From the tables, we observe that: (1) The GPT-3.5 model does not achieve satisfactory classification results on the syndrome differentiation task, with its performance being nearly equivalent to random guessing. This suggests that for very large models, such as GPT-3.5, do not perform well on the syndrome differentiation task without fine-tuning on training samples. (2) Across all three training set sizes, QWen ($Prompt_1$) outperforms BERT. Nevertheless, when using QWen ($Prompt_2$), BERT attains slightly better classification results for training set sizes of 16 and 32. These findings indicate that QWen's performance is sensitive to the prompts used. (3) The direct approach ($Prompt_1$) is more efficient with abundant data but less robust, while the indirect approach ($Prompt_2$) is less efficient with small data but more stable

via domain knowledge, making it ideal for small-sample scenarios. (4) Among all approaches, our approach consistently achieves the best performance, surpassing both BERT and QWen with a single prompt, as well as other baseline classification approaches, such as ZY-bert, SETFIT, TRD-FSL, and DLM-SCS.

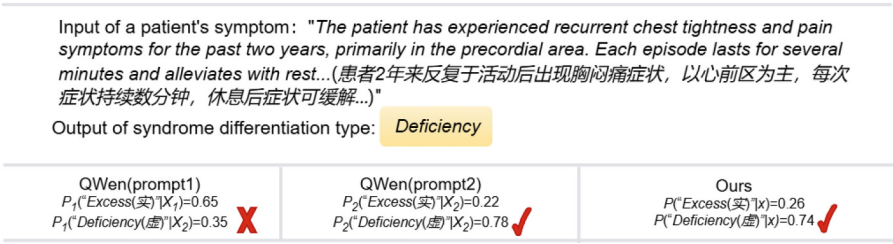


Fig. 3. An example for case study.

Figure 3 provides an example for case study. In this example, the doctor's diagnosis is "*Qi Deficiency and blood stasis*", which is classified as a type of *Deficiency* syndrome. The annotation in this case, stating "*alleviates with rest*", suggests a higher likelihood of the patient having a Deficiency syndrome. For this case, the approach using $Prompt_1$ misclassified the diagnosis by focusing only on chest tightness and pain and wrongly associating pain with *Excess* syndromes, whereas the approach using $Prompt_2$ classified it correctly by seizing the pathological essence of *Deficiency* syndromes. Our approach ultimately arrives at the correct classification. This example demonstrates the ability of our approach to handle complex cases in classification tasks, even with a limited number of training samples, resulting in improved classification performance.

5 Conclusion

In this paper, we propose a prefix-tuning-based few-shot learning approach that integrates two distinct prompts for application in the syndrome differentiation task in TCM. These two prompts are specifically designed: one directly inquires about the TCM syndrome differentiation results, while the other indirectly derives the results by asking about treatment approaches. Experimental results demonstrate that our approach outperforms the prefix-tuning approach using a single prompt, across various numbers of limited labeled samples. Furthermore, our approach exhibits superior performance compared to other existing few-shot learning approaches.

For our future work, we plan to extend our approach in two directions. First, we aim to expand from integrating two prompts to multiple prompts, exploring how this impacts classification performance. Second, we intend to broaden the scope of our experiments to include other TCM syndrome differentiation classifications such as Yin-Yang dialectics, Cold-Heat dialectics, and Interior-Exterior dialectics, to evaluate the effectiveness of our approach in these contexts.

Acknowledgment. This work is supported by Hong Kong PolyU grants #P00525-92 and the project "Co-establishing an Innovation and Entrepreneurship Education Practice Base with Changshu Bank" from Soochow University and Changshu Rural Commercial Bank.

References

1. Hu, E.J., Shen, Y., Wallis, P., et al.: Lora: low-rank adaptation of large language models. *CoRR*, abs/2106.09685 (2021)
2. Li, X.L., Liang, P.: Prefix-tuning: optimizing continuous prompts for generation. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 4582–4597. Association for Computational Linguistics (2021)
3. Pang, H., Wei, S., Zhao, Y., et al.: Effective attention-based network for syndrome differentiation of AIDS. BMC Med. Info. Decis. Mak. **20**(1), 264 (2020)
4. Cheng, N., Chen, Y., Gao, W., et al.: An improved deep learning model: S-TextBLCNN for traditional Chinese medicine formula classification. Front. Genet. **12**, 12 (2021)
5. Huang, Z., Miao, J., Chen, J., et al.: A traditional Chinese medicine syndrome classification model based on cross-feature generation by convolution neural network: model development and validation. JMIR Med. Inform. **10**(4), e29290 (2022)
6. Peng, Y., Yan, S., Lu, Z.: Transfer learning in biomedical natural language processing: an evaluation of BERT and elmo on ten benchmarking datasets. In: *Proceedings of the 18th BioNLP Workshop and Shared Task, BioNLP@ACL 2019, Florence, Italy, August 1, 2019*, pages 58–65. Association for Computational Linguistics (2019)
7. Song, Z., Xie, Y., Huang, W., et al. Classification of traditional Chinese medicine cases based on character-level BERT and deep learning. In: *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, pp. 1383–1387 (2019)
8. Ren, M., Huang, H., Zhou, Y., et al.: TCM-SD: a benchmark for probing syndrome differentiation via natural language processing. In *Chinese Computational Linguistics - 21st China National Conference, CCL 2022, Nanchang, China, October 14-16, 2022, Proceedings*, vol. 13603 of *LNCS*, pp. 247–263. Springer (2022)
9. Chen, Z., Zhang, D., Liu, C., et al.: Traditional Chinese medicine diagnostic prediction model for holistic syndrome differentiation based on deep learning. Int. Med. Res. **13**(1), 101019 (2024)
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics (2019)
11. Lei, S., Zhang, X., He, J., et al.: TART: improved few-shot text classification using task-adaptive reference transformation. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, (vol. 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 11014–11026. Association for Computational Linguistics (2023)

12. Pauli, A., Derczynski, L., Assent, I.: Anchoring fine-tuning of sentence transformer with semantic label information for efficient truly few-shot classification. In: Bouamor, H., Pino, J., Bali, K., (eds.) *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 11254–11264, Singapore (2023). Association for Computational Linguistics
13. Varia, S., Wang, S., Halder, K., et al.: Instruction tuning for few-shot aspect-based sentiment analysis. In: *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, WASSA@ACL 2023, Toronto, Canada, July 14, 2023*, pp. 19–27. Association for Computational Linguistics (2023)
14. Nguyen, Q.A., Tomeh, N., Lebbah, M., et al.: Enhancing few-shot topic classification with verbalizers. a study on automatic verbalizer and ensemble methods. In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pp. 5956–5965. ELRA and ICCL (2024)
15. Wang, Q., Xu, H., Ding, K., et al.: In-context example retrieval from multi-perspectives for few-shot aspect-based sentiment analysis. In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pp. 8975–8985. ELRA and ICCL (2024)
16. Lee, J.H., Hahn, J., Seo, H.T., Park, J., Han, Y.S.: SuperSt: superficial self-training for few-shot text classification. In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pp. 15436–15447. ELRA and ICCL (2024)
17. Bai, J., Bai, S., Chu, Y., et al.: Qwen technical report. *CoRR*, abs/2309.16609 (2023)
18. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. *CoRR*, abs/1706.03762 (2017)
19. Wang, Y.: *Traditional Chinese Medical Diagnosis*. Higher Education Press (2012)
20. Brown, T.B., Mann, B., Ryder, N., et al.: Language models are few-shot learners. In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual* (2020)
21. Tunstall, L., Reimers, N., Jo, U.E.S., et al.: Efficient few-shot learning without prompts. *CoRR*, abs/2209.11055 (2022)
22. Li, Y., Li, Z., Chen, Y., Li, S.: Word-level prefix/suffix sense detection: a case study on negation sense with few-shot learning. In: *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 7651–7658. Association for Computational Linguistics (2023)
23. Xie, Z., Li, Y.: Discriminative language model as semantic consistency scorer for prompt-based few-shot text classification. In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pp. 4968–4977. ELRA and ICCL (2024)



Neural QSLIM for Mesh Autoencoders

Haoliang Zhang[1(✉)], Xintong Li[2], Jonghoon Kim[3], Samuel Cheng[1], and Christian El Amm[1]

[1] University of Oklahoma, Norman, OK, USA
mars_zhang@ou.edu
[2] Beijing University of Posts and Telecommunications, Beijing, China
[3] Chungnam National University, Daejeon, Republic of Korea

Abstract. Mesh autoencoders rely heavily on fixed quadric sampling schemes, which preserve only topological information, often leading to poor-quality shape up-sampling and overfitting. This paper introduces Neural QSLIM, a novel framework for geometry-aware mesh up-sampling. By randomly generating bijective coarse-to-fine mesh counterparts, our method trains a neural network to learn complex mesh shapes. During decoding, our approach takes a coarse triangle mesh from the bottleneck and reconstructs finer geometry using QSLIM-guided topological updates, while predicting vertex positions with the trained neural network. This design improves the accuracy of vertex positions while maintaining the desired topological structure, rather than relying on simple point projection as in classical mesh autoencoders. We demonstrate that our method enables more effective non-linear mesh up-sampling, resulting in efficient and flexible representations across various mesh autoencoder models.

Keywords: Mesh autoencoder · Geometry · QSLIM

1 Introduction

Mesh autoencoders have become popular for learning latent representations of registered meshes, relying heavily on a multiscale hierarchical representation. This architecture requires convolution-like operations to capture local geometric features on graphs. However, classical pooling methods cannot be applied directly to 3D meshes [1–4] due to irregular sampling and complex connectivity. Ranjan et al. [5] introduced the CoMA model, which first applied mesh sampling operators for down-sampling and up-sampling in neural networks. In the mesh sampling process, the contracted vertices are stored at their barycentric locations in the transformation matrices during downsampling and later used in the decoder stage. This approach enables the model to capture both global and local context and has been adopted by subsequent works [6–11].

H. Zhang and X. Li—Contributed equally.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 51–65, 2026.
https://doi.org/10.1007/978-3-032-04555-3_5

Classical mesh autoencoders rely on pointwise loss, which requires the input and output shapes to have the same number of vertices. While MeshCNN [12] and Neural Subdivision [13] are specifically designed for mesh down-sampling and up-sampling, respectively, without user guidance, they alter the mesh topology, leading to features with different dimensionalities. The Fully convolutional mesh autoencoder proposed in [14] achieves high reconstruction accuracy but relies on a large number of parameters, making it computationally expensive compared to more lightweight architectures.

Furthermore, current methods apply mesh sampling once on a template to construct down-sampling and up-sampling transformation matrices, which are then shared across all meshes. This approach leads to template overfitting. An alternative method is to apply mesh sampling individually to all training meshes and average the barycentric coordinates. However, Quadric Simplification (QSLIM) [15,16] a classical method that simplifies meshes via iterative edge collapses while minimizing geometric error, requires closest-point-on-mesh calculations (similar to the concept of Chamfer distance [17,18]), which are computationally expensive for high-resolution meshes, and averaging operations cause all projected coordinates to converge toward the centroid, leading to loss of geometric detail. To address these problems, we propose *Neural QSLIM* (as shown in Fig. 1), which updates mesh vertices using a transformation matrix while predicting vertex positions with a trained neural network. The transformation matrix preserves mesh topology and adjusts vertex positions based on a neural network conditioned on the mesh geometry. Compared to existing methods, our

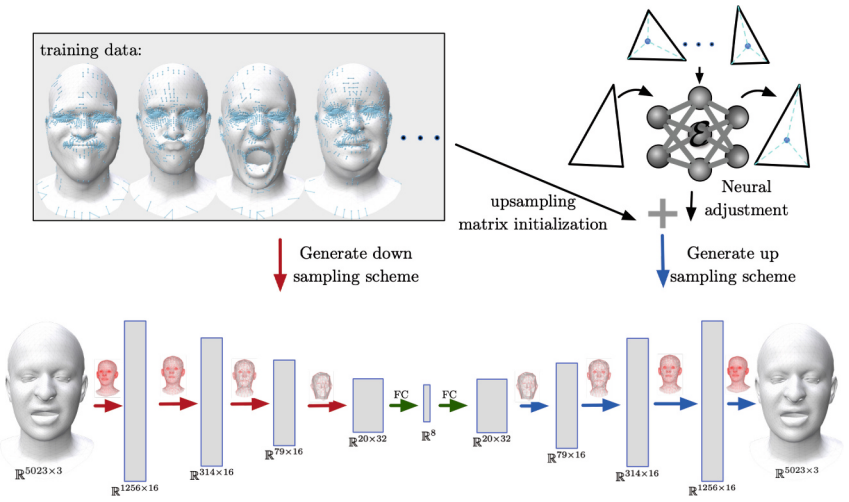


Fig. 1. Unlike standard mesh autoencoders that build down/up matrices from a single mesh, our method employs a novel coarsening scheme and neural unpooling layer on a fixed topology mesh dataset. The resulting transformation matrices or hierarchies can be seamlessly integrated into mesh generative models without altering the user's network architecture.

mesh unpooling eliminates template overfitting. By training across a set of morphable meshes, our approach learns a generalizable rule based on local patches from multiple training meshes rather than relying on a single template.

In summary, our work introduces a mechanism for up-sampling meshes in mesh autoencoders. Our method enhances mesh autoencoder representations by providing more accurate vertex initialization in upsampling. Our main contributions are:

1. We propose a geometry-aware mesh up-sampling method that provides more generalized mesh representations than traditional approaches.
2. We show that our method effectively eliminates overfitting caused by sharing transformation matrices across meshes.
3. We demonstrate that a shallow 3-layer multi-layer perceptron (MLP) is sufficient for predicting vertex positions.

2 Related Works

Our work is designed for geometric deep learning, and its decoder stage resembles subdivision methods. In this section, we introduce previous mesh representation models and compare mesh autoencoders with mesh subdivision approaches.

2.1 Geometric Deep Learning

CoMA [5] introduces the first sampling operations for mesh autoencoders using precomputed down-sampling and up-sampling matrices. Its distinctive pyramid-shaped structure optimizes memory efficiency and enables the model to capture both global and local context.

Following the CoMA architecture, later studies [6,7,9,19–23] improved mesh convolution networks while inheriting the down-sampling and up-sampling operations. Neural3DMM [6] and SpiralNet++ [7] replace spectral convolution layers with operators that convolve along a fixed spiral serialization [24] of neighboring vertices, outperforming previous approaches. Alternative implementations, such as [14], define a shared weight basis kernel at every vertex location but come at the cost of increased parameters. MeshCNN [12] instead selects edges as input features and applies filters directly to each edge, making it well-suited for classification and segmentation tasks. In contrast, our work focuses on latent representation and mesh generation.

Previous research improves convolution networks but retains the same down-sampling and up-sampling operations, which rely solely on topological information without considering geometry. Furthermore, constructing the transformation matrix from a single mesh leads to severe overfitting. In contrast, our method proposes an efficient geometry-aware mesh up-sampling operation to enhance the mesh autoencoder.

2.2 Comparing with Subdivision Surfaces

The primary objective of a mesh autoencoder is to learn non-linear representations of meshes. This framework assumes that the input mesh is an exact description of the shape of interest. The output of the decoder should precisely preserve the given shape, even if the input mesh only approximates the true geometry (e.g., 3D scans). If defects such as topological errors or noise are present, these artifacts are also retained by the model. Mesh autoencoders are designed to maintain the topological structure of the input while approximating geometric details as accurately as possible. However, while they effectively learn latent representations of meshes, they do not improve the quality of the underlying geometry.

In contrast, the idea of subdivision is to "define a smooth surface as the limit of a sequence of successive refinements" [25]. However, they are incapable of recuperating lost details or semantic features. Traditional methods like subdivision methods [26–28] process the mesh by combinatorial update (splitting faces, adding vertices, or/and fipping edges [29]) and vertex smoothing. On the other hand, contemporary non-linear neural subdivision methods [30] and subdivision-based networks (SubdivNet) [31] use fixed and uniform topological updates while leveraging neural networks to capture intricate details. Although SubdivNet introduces pooling and unpooling layers, it cannot be directly applied to arbitrary meshes because it requires the input to maintain subdivision connectivity.

Geometry generation models are generally evaluated based on reconstruction losses, assessing the output mesh's fidelity in approximating a known target. For a mesh autoencoder, both input and output meshes share the same topological structure, and the model aims to minimize vertex-to-vertex loss. In contrast, subdivision-based models typically employ point-to-surface distances, establishing correspondences through closest-point queries. In essence, while subdivision alters the original topological structure, it fails to generate meaningful semantic details from highly coarse meshes. A detailed exploration of this issue is beyond the scope of this paper.

3 Method

In this section, we provide an overview of the main components of Neural QSLIM: the construction of the transformation matrix, data generation, and the network architecture.

3.1 Build Transformation Matrix

Although the dataset shares the same topology, the down-sampling results obtained by QSLIM [15] vary significantly. To achieve optimal performance, we contract vertices based on frequency. Specifically, by applying QSLIM to a randomly selected 1% of meshes from the training dataset, we sort the collapsed

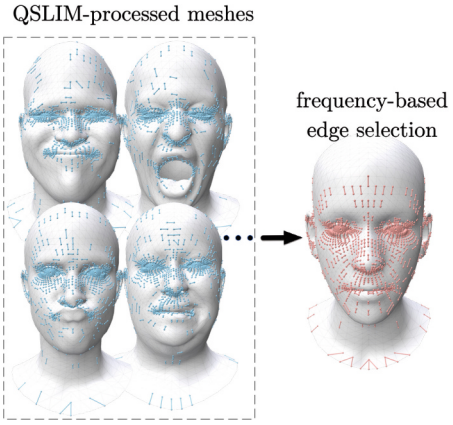


Fig. 2. QSLIM is applied to the meshes, and edges are selected based on their frequency. Red edges show those contracted during down-sampling. (Color figure online)

edges by their occurrence frequency (as shown in Fig. 2). The most common edges are then used to construct the down-sampling transformation matrix.

Vertices discarded during down-sampling are mapped onto the surface of the down-sampled mesh using barycentric coordinates. Following [5], we denote the down-sampling and up-sampling transformation matrices as $Q_d \in \{0,1\}^{n \times m}$ and $Q_u \in \{0,1\}^{m \times n}$, respectively. For the up-sampled mesh $\mathcal{V}_u \in \mathbb{R}^{m \times 3}$ and the corresponding down-sampled mesh $\mathcal{V}_d \in \mathbb{R}^{n \times 3}$, the relationships are given by $\mathcal{V}_u = Q_u \mathcal{V}_d$ and $\mathcal{V}_d = Q_d \mathcal{V}_u$. By applying QSLIM to different mesh expressions and selecting the most common contracted edges, our method generates a more generalized transformation matrix.

Following CoMA [5], Q_d and Q_u are implemented as sparse selector matrices with binary entries indicating retained or restored vertices. In our method, Q_u provides an initial estimate for upsampled geometry, which is further refined by a shallow MLP to capture local shape variations.

3.2 Data Generation and Training

Our approach is inspired by neural subdivision, but our upsampling is guided by a precomputed transformation matrix instead of using classic Loop subdivision [27]. This allows our method to generate meshes with any desired number of vertices and work with mesh autoencoders.

We first generate pairs of meshes with different discretizations to train the neural network. Specifically, we apply random edge collapses to create coarse meshes and use bijective mappings to locate corresponding positions on the ground-truth mesh.

Mullen et al. [32] introduced free-boundary conformal parameterizations for a triangular mesh. In the context of an edge collapse, this conformal parameterization [32] can be employed to map the vertices of the 1-ring of the edge into its

Algorithm 2: Generate Down/Initial-Up Matrix

Input : M, n ▷ selected meshes, target edge count
Output: Q_d,Q_u ▷ down & up sampling matrices

▷ initialization
$E \leftarrow$ EmptyPriorityQueue ( ) ▷ edge frequency
$O \leftarrow$ EmptyVertexBarycentric ( ) ▷ barycentric coordinate
foreach *mesh* $m \in M$ **do**
  $e \leftarrow$ QSLIM (m) ▷ extract collapsed edges
  UpdateBarycentric (O, m, e)
  Enqueue (E, e)
▷ build
while *VertexPopCount*$(Q) < n$ ***and!****Empty* (Q) **do**
  $e_{ij} \leftarrow$ Pop (E) ▷ extract maximum-frequency edge
  UpdateDownMatrix (Q_d, e_{ij})
  UpdateUpMatrix (Q_u, O, e_{ij}) ▷ initial vertex & barycentric coordinate
return (Q_d, Q_u);

local UV space, as illustrated in Fig. 3. This UV parameterization serves as a verification tool. A critical aspect to note is the shared boundary vertices between the contracted point post-edge collapse and the collapsed edge. Given the UV boundary constraints, the post-collapse vertex retains identical UV boundaries.

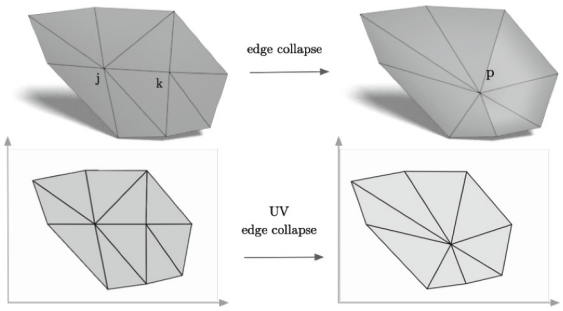


Fig. 3. Utilizing conformal parameterization to map the 1-ring of a collapsed edge to the UV space. Notably, the vertices post and pre-collapse maintain identical boundary vertices.

We adopt sequential downsampling UV mapping as an alternative to the traditional approach, ensuring accurate triangle projection across each mesh and preventing the inside-out issue associated with naive bundle closest projection. Figure 4 illustrates an external vertex projecting onto an internal face along a saddle-shaped surface due to internal heat source interference, disrupting expected heat diffusion along the surface. We employ a highly sparse matrix $Q_d \in \{0,1\}^{m \times n}$, where $m > n$, in the pooling layer to derive coarse meshes.

Here, $Q_d(p, q) = 1$ indicates retention of the q-th vertex is kept. Analogous to the template-only method, we compute the transformation matrix Q_d only once. This practice sustains the model's efficiency while enhancing its generalization performance.

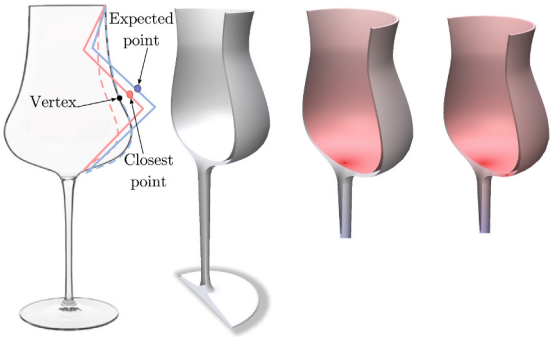


Fig. 4. Heat diffusion illustrated on a wine glass. Through local mapping, inside-out projection is avoided, resulting in an outward heat spread on the surface (center right), as opposed to the incorrect heat flow caused by closest point projection (right).

3.3 Up-Sampling

Down-sampling, while essential for mesh simplification, represents only one of the two core components integral to the functionality of the mesh autoencoder model. Given that lossless pooling/unpooling is unattainable, previous methods constructed an un-sampling matrix during the down sampling phase. In these earlier approaches, discarded vertices of the fine mesh were projected onto the nearest triangle in the down-sampled mesh, a process articulated through barycentric coordinates. However, as depicted in Fig. 5, templates prove inadequate in capturing varying geometric features. This limitation persists even when identical edge contractions are applied, leading to barycentric coordinates that are not aligned with the template.

Our training data is constructed by applying the transformation matrix decided in down-sample step to selected meshes. During the down-sampling, we collect each triangle and corresponding barycentric coordinates pair. Suppose $\mathcal{E}(v)$ is the barycentric coordinates output by the network $\mathcal{E}$ (Fig. 6), and $f(v)$ denote the actual barycentric coordinates. We measure the per-vertex loss with the $\mathcal{L} = ||f(v) - \varepsilon(v)||_2$.

Network Architecture: Our network is inspired by classical mesh autoencoder which has two operators: (1) mesh sampling from template, and (2) the convolution operator. The two pivotal steps in our process are depicted in Fig. 1. The down-sample matrix and the initial up-sample matrix are determined from selected meshes. Simultaneously, we gather each triangle and the corresponding

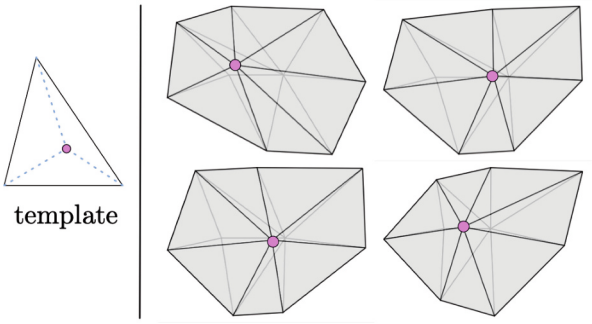


Fig. 5. Barycentric coordinates resulting from mapping discarded vertices to the closest triangle are depicted. Despite contracting the same edge to the same end, the projection point differs depending on the given geometry. Relying solely on the template representation leads to an inaccurate up-sampling matrix.

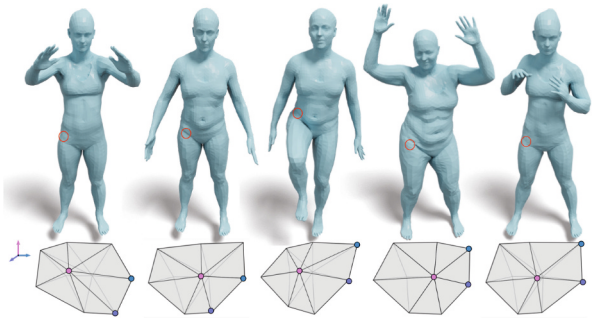


Fig. 6. Registered data allows us to order the three vertexes of the triangle canonically. We feed vertex features into multi-layer perceptrons (MLP).

pairs of barycentric coordinates for training the network $\mathcal{E}$. One distinguishing feature of our approach is that we apply training meshes, instead of relying solely on templates. This strategy enables us to utilize neighborhood information derived from a broader range of mesh expressions.

We detail the process through which our model acquires rules from the local patches across all training meshes. During each instance of edge collapse, rather than employing fixed weights from a template mesh, we integrate a training module to the mesh. This allows the network to learn the rules associated with the bijective mapping of the local patch dynamically. Our methodology is characterized by the advantage of constant-cost mapping ("constant-cost mapping" refers to using precomputed barycentric coordinates from the template mesh, which are shared across all training shapes.). We formulate pairs of local coarse and fine meshes and utilize shallow multilayer perceptrons to learn the nuances of the upsampling mappings.

The final step in our architecture design is the representation of the input and output. Consider a naive approach that use global xyz-coordinates of triangle and projection point pairs. This approach has a major caveat. Cartesian coordinates of a triangle must be described with repect to some global coordinate system. This lead to incorrect and badly training set up because the rotation and transplation of mesh. To mitigate this issue, we incorporate the use of local coordinates of the mesh, ensuring invariance to rigid transformations. Within the context of this study, the half-edge data structure is employed for mesh storage, offering a unique canonical orientation for the vertices of triangular faces. The local coordinate frame is defined by the three vertices of the mesh, enabling the construction of input features in a canonical manner. This approach underscores each mesh's reliance on its intrinsic coordinate system, a factor that not only expedites the training process but also augments the model's generalization capabilities.

Our refinement network is a shallow MLP with three layers (64, 64, 3) and ReLU activations. We experimented with deeper networks (up to 5 layers) and wider hidden dimensions (128), but observed no significant improvement in accuracy or convergence speed. For training loss, we compared ℓ_2 and Huber loss, and found ℓ_2 slightly more stable across different meshes.

4 Applications

In this section, we present a series of experimental results to evaluate our operator. We begin by demonstrating its invariance to rigid motions and its ability to improve reconstruction quality. We commence by illustrating its capability to remain invariant to rigid motions, a feature that notably enhances the quality of the outcomes. Our presentation extends to a visual demonstration, underscoring the superiority of our method in delivering enhanced unpooling results, especially when juxtaposed against the closest point projection in tasks associated with mesh reconstruction. It is imperative to note that, to engender a robust and equitable comparison, we ensured uniformity in the model architectures and maintained constancy in the kernel size across the diverse convolutions under consideration.

We implemented our network using the PyTorch [33]. We use Relu activation [34], and the ADAM optimizer [35] with a learning rate of 0.002. We implemented our technique using the Spectra [36], and tested it on a workstation with an Intel Core i7 2.6 GHz CPU, and 16 GB of RAM. For the mesh reconstruction task, each method was trained and evaluated using a single NVIDIA RTX 3080 Ti. Our method incurs negligible training overhead: the base architecture matches prior work, and the added MLP takes under 10 min to train.

In practical applications, it is commonplace for modelers to adjust the mesh into various poses prior to the application of alternative operators. Figure 7 exemplifies the paramount importance of invariant representation in optimizing the performance of a mesh autoencoder. Our findings reveal a notable vulnerability in global-based models; even with training on an identical shape, a minor rigid

motion significantly undermines the applicability of the learned weights. In contrast, the incorporation of local coordinates as input features not only enhances the model's resilience to such perturbations but also augments the pace of convergence by approximately 2%-3.5%. While these enhancements may appear marginal, they contribute incremental improvements in the training phase.

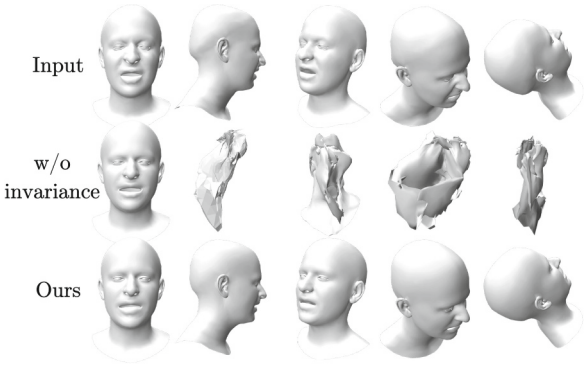


Fig. 7. Employing local coordinates embedded within each mesh as both inputs and outputs, our methodology achieves invariance to rigid motions. This adaptation yields a marked enhancement in quality relative to preceding methods that lack such invariance.

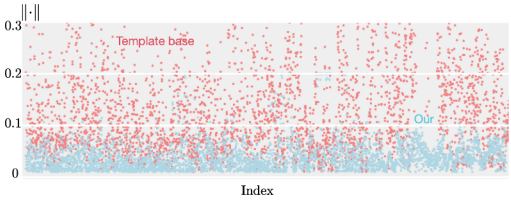


Fig. 8. By looking at the error in the function of the Euclidean domain of upsampling mesh. We can see that the shared weight cannot handle the diverse mesh. We randomly picked mesh from the CoMA dataset, the reduction from 5024 to 315 vertices (6%) then upsampling by a different strategy. Each point illustrates the error between estimated and original coordinates.

We further evaluated successive local mapping on mesh generative tasks. For registered data such as CoMA [5] or FAUST [37], modelers often manipulate meshes with different expressions and embed the GNN in a pooling/unpooling layer. Earlier models that employed in-network sampling and shared the transformation matrices of the template across all meshes encountered a pervasive issue of template overfitting. As depicted in Fig. 8, we diverge from this approach by upsampling the coarse mesh without relying on the shared weights of

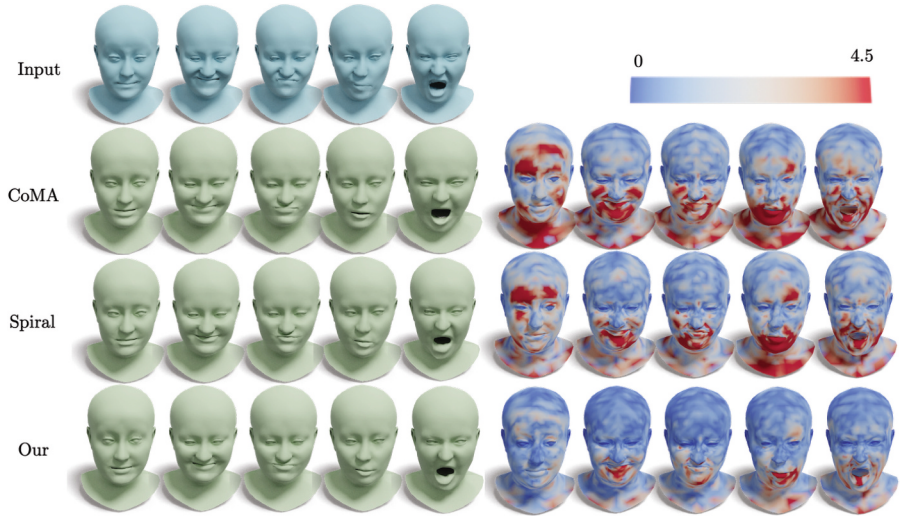


Fig. 9. Qualitative assessment of reconstruction outcomes. Each vertex's Euclidean error is color-coded, with the error values capped at 4.5 mm. Warmer hues denote larger errors.

the template, demonstrating that our model is adept at offering accurate initial estimations.

Our method for pooling and unpooling is highly compatible with numerous standard mesh processing pipelines. It is engineered to facilitate local bijective mapping while meticulously preserving the intrinsic properties of the original mesh. We exemplify this attribute within the context of a mesh autoencoder application.

For a quantitative examination of our network's generalization capabilities to accommodate unseen shapes, we employed the CoMA dataset. This dataset encompasses a diverse collection of over 20K+ 3D meshes, featuring 12 distinct subjects each exhibiting 12 extreme facial expressions, with each registration consisting of 5023 vertices. We split the dataset in training and test samples with a ratio of 9 : 1.

We employed a conventional autoencoder architecture, composed of an encoder and a decoder. The encoder is structured as: $3 \times \{Conv(16) \rightarrow Pool(4)\} \rightarrow \{Conv(32) \rightarrow Pool(4)\} \rightarrow FC(8)$, accompanied by an ELU activation function post each Conv layer. The decoder mirrors the encoder's structure in reverse order, substituting pooling layers with unpooling layers. The integration between layers is facilitated through a spiral convolution, the efficacy of which is illustrated through a visual comparison of the reconstruction outcomes depicted in Fig. 9 and Fig. 10. A comprehensive list of mean errors, accompanied by standard deviations, is presented in Table 1 for an exhaustive comparative analysis.

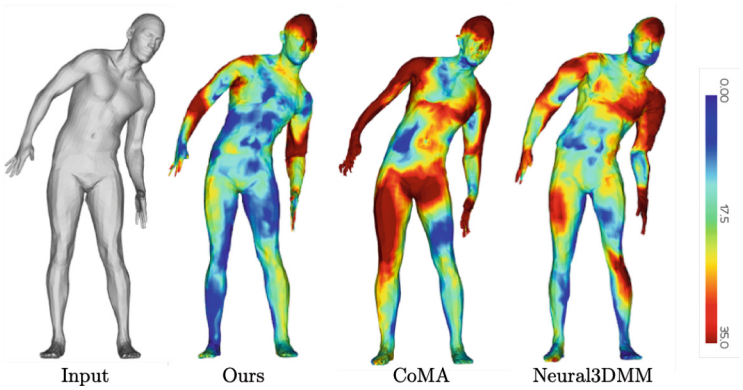


Fig. 10. Colour coding of the per vertex euclidean error of the reconstructions. The error values are saturated at 35 (millimeters).

Table 1. Comparison with CoMA and Neural3DMM. Errors are in millimeters. Simplified Neural3DMM with filter size [16,16,16,32].

	Mean Error	#Parameters
Ours	$\mathbf{0.541 \pm 0.588}$	37,563
Neural3DMM [6]	0.804 ± 0.877	48,067
CoMA [5]	0.845 ± 0.994	33,856
PCA	1.639 ± 1.638	120,552

5 Conclusion and Limitations

We proposed a neural-driven quadric mesh unpooling framework that improves vertex positioning and captures complex non-linear transformations across diverse meshes. Our method learns local vertex features to guide downsampling and upsampling, effectively preserving fine geometric details and mitigating template dependence. Experimental results show that our model outperforms template-based mesh autoencoders, achieving lower point-to-point reconstruction error and better generalization across different mesh expressions. Our current framework assumes topologically consistent, registered meshes during training. As such, it is not directly applicable to partial or noisy scans with missing geometry or varying connectivity. Extending the method to handle topological noise would require rethinking the construction of transformation matrices and refinement targets, which we leave for future work.

References

1. Bagautdinov, T., Wu, C., Saragih, J., Fua, P., Sheikh, Y.: Modeling facial geometry using compositional vaes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3877–3886 (2018)
2. Sinha, A., Bai, J., Ramani, K.: Deep learning 3d shape surfaces using geometry images. In: European Conference on Computer Vision, pp. 223–240. Springer (2016)
3. Sinha, A., Unmesh, A., Huang, Q., Ramani, K.: Surfnet: generating 3d shape surfaces using deep residual networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6040–6049 (2017)
4. Maron, H., et al.: Convolutional neural networks on surfaces via seamless toric covers. ACM Trans. Graph. **36**(4), 71–1 (2017)
5. Ranjan, A., Bolkart, T., Sanyal, S., Black, M.J.: Generating 3D faces using convolutional mesh autoencoders. In: European Conference on Computer Vision (ECCV), pp. 725–741 (2018)
6. Bouritsas, G., Bokhnyak, S., Ploumpis, S., Bronstein, M., Zafeiriou, S.: Neural 3d morphable models: Spiral convolutional networks for 3d shape representation learning and generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 7213–7222 (2019)
7. Gong, S., Chen, L., Bronstein, M., Zafeiriou, S.: Spiralnet++: a fast and highly efficient mesh convolution operator. In: Proceedings of the IEEE International Conference on Computer Vision Workshops (2019)
8. Tretschk, E., Tewari, A., Zollhöfer, M., Golyanik, V., Theobalt, C.: DEMEA: deep mesh autoencoders for non-rigidly deforming objects. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12349, pp. 601–617. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58548-8_35
9. Chen, Z., Kim, T.-K.: Learning feature aggregation for deep 3d morphable models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13164–13173 (2021)
10. Pichi, F., Moya, B., Hesthaven, J.S.: A graph convolutional autoencoder approach to model order reduction for parametrized pdes. J. Comput. Phys. **501**, 112762 (2024). ISSN 0021-9991. https://doi.org/10.1016/j.jcp.2024.112762, https://www.sciencedirect.com/science/article/pii/S0021999124000111
11. Almalki, A., Latecki, L.J.: Self-supervised learning with masked autoencoders for teeth segmentation from intra-oral 3d scans. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 7820–7830 (2024)
12. Hanocka, R., Hertz, A., Fish, N., Giryes, R., Fleishman, S., Cohen-Or, D.: Meshcnn: a network with an edge. ACM Trans. Graph. (TOG) **38**(4), 90:1–90:12 (2019)
13. Liu, H.T.D., Kim, V.G., Chaudhuri, S., Aigerman, N., Jacobson, A.: Neural subdivision. ACM Trans. Graph. **39**(4), (2020). ISSN 0730-0301. https://doi.org/10.1145/3386569.3392418
14. Zhou, Y., et al.: Fully convolutional mesh autoencoder using efficient spatially varying kernels. In: Advances in Neural Information Processing Systems (2020)
15. Garland, M., Heckbert, P.S.: Surface simplification using quadric error metrics. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, pp. 209–216 (1997)
16. Cohen, J., Olano, M., Manocha, D.: Appearance-preserving simplification. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1998, pp. 115–122. ACM. New York, NY, USA (1998). ISBN 0897919998. https://doi.org/10.1145/280814.280832

17. Barrow, H.G., Tenenbaum, J.M., Bolles, R.C., Wolf, H.C.: Parametric correspondence and chamfer matching: two new techniques for image matching. In: Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI 1977, pp. 659–663, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc (1977)
18. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 605–613 (2017)
19. Gao, Z., Yan, J., Zhai, G., Zhang, J., Yang, Y., Yang, X.: Learning local neighboring structure for robust 3d shape representation. In: AAAI Conference on Artificial Intelligence (2020). https://api.semanticscholar.org/CorpusID:219402126
20. Bouzid, H., Ballihi, L.: Spatr: mocap 3d human action recognition based on spiral auto-encoder and transformer network. Comput. Vis. Image Underst. **241**, 103974 (2024)
21. Zhang, H., Cheng, S., El Amm, C., Kim, J.: Efficient pooling operator for 3d morphable models. IEEE Trans. Visual Comput. Graphics **30**(7), 4225–4233 (2024). https://doi.org/10.1109/TVCG.2023.3255820
22. Zhang, H., Na, W., Kim, J.: State-of-charge estimation of the lithium-ion battery using neural network based on an improved thevenin circuit model. In: 2018 IEEE Transportation Electrification Conference and Expo (ITEC), pp. 342–346 (2018). https://doi.org/10.1109/ITEC.2018.8450162
23. Zhang, H., Tang, W., Na, W., Lee, P.Y., Kim, J.: Implementation of generative adversarial network-cls combined with bidirectional long short-term memory for lithium-ion battery state prediction. J. Energy Storage **31**, 101489 (2020). https://doi.org/10.1016/j.est.2020.101489, https://www.sciencedirect.com/science/article/pii/S2352152X19306930
24. Lim, I., Dielen, A., Campen, M., Kobbelt, L.: A simple approach to intrinsic correspondence learning on unstructured 3d meshes. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops, (2018)
25. Zorin, D., Schroder, P., DeRose, A., Kobbelt, L., Levin, A., Sweldens, W.: Subdivision for Modeling and Animation. ACM Press/Addison-Wesley Publishing Co. (2000)
26. Doo, D., Sabin, M.: Behaviour of recursive division surfaces near extraordinary points. Comput. Aided Des. **10**(6), 356–360 (1978)
27. Loop, C.: Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, Department of Mathematics (1987)
28. Catmull, E., Clark, J.: Recursively generated b-spline surfaces on arbitrary topological meshes. In: Seminal Graphics: Pioneering Efforts that Shaped the Field, pp. 183–188. ACM Press/Addison-Wesley Publishing Co. (1998)
29. Kobbelt, L.: 3-subdivision. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, pp. 103–112, USA (2000). ACM Press/Addison-Wesley Publishing Co. ISBN 1581132085
30. Liu, H.T.D., Kim, V.G., Chaudhuri, S., Aigerman, N., Jacobson, A.: Neural subdivision. ACM Trans. Graph. **39**(4) (2020). ISSN 0730-0301
31. Hu, S.-M., et al.: Subdivision-based mesh convolution networks. ACM Trans. Graph. (TOG), **41**(3), 1–16 (2022)
32. Mullen, P., Tong, Y., Alliez, P., Desbrun, M.: Spectral conformal parameterization. In: Computer Graphics Forum, vol. 27, pp. 1487–1494. Wiley Online Library (2008)
33. Paszke, A., et al.: Pytorch: an imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems, vol. 32 (2019)

34. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Icml (2010)
35. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization (2017). https://arxiv.org/abs/1412.6980
36. Qiu, Y.: Spectra: a header-only c++ library for large scale eigenvalue problems. https://github.com/yixuan/spectra (2015)
37. Bogo, F., Romero, J., Pons-Moll, G., Black, M.J.: Dynamic FAUST: registering human bodies in motion. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017



Evolving Spatially Embedded Recurrent Spiking Neural Networks for Control Tasks

Alexandru Vasilache[1,2(✉)], Jona Scholz[2], Yulia Sandamirskaya[3], and Jürgen Becker[2]

[1] FZI Research Center for Information Technology, Karlsruhe, Germany
vasilache@fzi.de
[2] Karlsruhe Institute of Technology, Karlsruhe, Germany
[3] ZHAW Zurich University of Applied Sciences, Wädenswil, Switzerland

Abstract. Spiking neural networks (SNNs), inspired by biological brains, use discrete spikes for communication, offering potential advantages in energy efficiency and temporal processing. These properties make them attractive for low-power, real-time control, but optimizing their structure and parameters is challenging. This work investigates the impact of spatial embedding on recurrent SNN performance and efficiency in continuous control tasks. We evolve SNNs with neurons positioned in a 3D Euclidean space, where connection probabilities and strengths decrease with distance. A genetic algorithm optimizes neuron parameters, connection weights, and network topology. Evaluating various spatial embeddings (none, 1D, 2D, and 3D) across multiple reinforcement learning environments, we find that spatially embedded networks outperform non-embedded counterparts within our framework. We also find that the 2D embedding generally achieves the best performance. Spatial embedding also leads to highly sparse networks, with over 95% of weights being zero. Compared to state-of-the-art deep reinforcement learning models, our evolved SNNs achieve similar performance on simpler tasks using as little as 1.46% of the non-zero weights. A performance gap on more complex tasks suggests the need for future research on larger, spatially embedded networks. These findings show the potential of spatial embedding for designing efficient SNNs for neuromorphic control.

Keywords: Spiking Neural Networks (SNNs) · Topology Optimization · Spatial Embedding · Genetic Algorithms · Control Tasks

1 Introduction

Spiking Neural Networks (SNNs). SNNs are a type of artificial neural network inspired by the biological nervous system. Unlike conventional Artificial Neuronal Networks (ANNs), which rely on continuous activation values, SNNs

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 66–77, 2026.
https://doi.org/10.1007/978-3-032-04555-3_6

utilize discrete events called spikes to transmit information between neurons [13]. This event-driven nature, coupled with the statefulness and temporal dynamics of spiking neurons, enables sparse activations, making SNNs promising candidates for energy-efficient, low-latency computation. Recent advancements in training algorithms and neuromorphic hardware [16] further bolster the potential of SNNs for various machine learning tasks, including robot control.

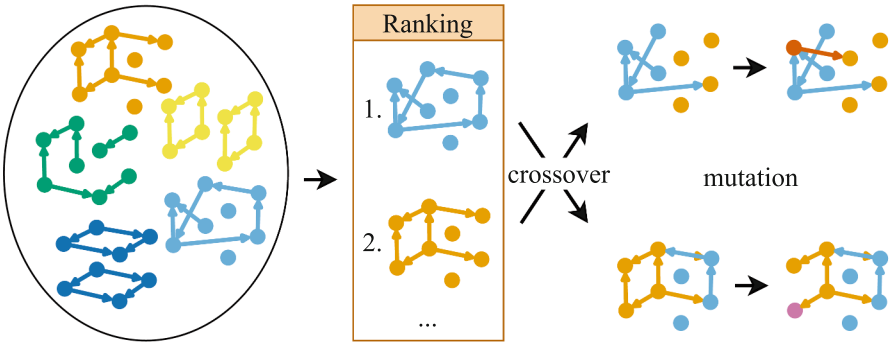


Fig. 1. Conceptual visualization of the proposed framework. Neurons are arranged in an n-dimensional grid. The method evaluates a population of networks and ranks them. High-ranking genes produce modified duplicates through crossover and mutation.

SNNs for Control Tasks. Research on Neural Networks (NNs) for control falls broadly into two categories: Evolutionary Algorithms (EAs) and Reinforcement Learning (RL) strategies. While both approaches show promise, SNNs present a unique challenge: effectively tuning numerous neuron-specific parameters (e.g., firing thresholds, time constants), potentially leading to smaller, more efficient networks that achieve better performance. This is where gradient-free EAs offer a distinct advantage over RL methods, which are typically limited to adjusting only synaptic weights through gradient-based methods.

EAs draw inspiration from natural selection to directly optimize all SNN parameters. For instance, Qiu et al. [22] used the NEAT algorithm [26] to evolve a recurrent spiking controller on a pole balancing task. Other works have explored EAs for optimization of various aspects of SNNs for control [12,23,30], highlighting their potential for such tasks.

RL methods, conversely, train SNNs by interacting with the environment and learning from rewards, inspired by natural learning. Recent advances focus on increasing training efficiency and leveraging temporal information processing capabilities. For example, Zanatta et al. [33] developed a framework for faster SNN training with Proximal Policy Optimization (PPO) [24]. Other studies have explored incorporating learning mechanisms like R-STDP [11] and temporal coding [31], and addressing challenges like domain adaptation [3].

Topology Optimization of SNNs. Neural architecture search (NAS) [7], while successful in optimizing ANN architectures, is still nascent for SNNs. Existing approaches often adapt NAS techniques from ANNs, neglecting the unique temporal dynamics of SNNs and primarily focusing on feedforward architectures [9,14,15] [20], which may limit the discovery of SNN-optimized architectures. Recurrent SNNs, commonly referred to as Liquid State Machines (LSMs) [34], offer a biologically plausible alternative to NAS for topology optimization [18] and have been proven effective for robot control tasks [17]. Beyond optimizing the overall architecture, connection optimization adjusts SNNs during training, pruning less important connections and growing new ones [21,22,25,30]. Notably, Wang et al. [30] demonstrate the effectiveness of evolving connection probabilities in SNNs with sparse 1-bit connections on robotic locomotion tasks.

Spatial Embedding of Neural Networks (NNs). In a study by Deb et al. [5], weights of different network types were reshaped into 2D cortical sheets. Through the use of a topographic loss the networks learned a shape that exhibited higher weight sparsity and better parameter efficiency. Elbrecht and Schuman [6] used an evolutionary algorithm to evolve Compositional Pattern Producing Networks (CPPs) that define the connectivity, weights, delays, and neuron thresholds of SNNs whose neurons are arranged in a 2D euclidean space. Achterberg et al. [2] show that embedding Recurrent Neural Networks (RNNs) within a 3D Euclidean space and applying biophysical constraints during training leads to the emergence of brain-like characteristics, such as modularity and energy-efficient coding. Furthermore, the structure of spatially embedded networks may allow for more efficient implementations on neuromorphic chips, by aligning the physical hardware substrate more closely with the algorithm. This can be seen in work by Bogdan et al. [4], who implemented a 2D SNN on the SpiNNaker neuromorphic platform, where neurons were placed on a grid and connection rules were distance-dependent. Using Spike-Timing-Dependent Plasticity (STDP) the connection weights were evolved over time and pruned based on their magnitude. To our knowledge, our work is the first to compare the impact of different spatial embeddings on the performance SNNs.

2 Methods

For training and evaluation, we used the Gymnasium library [29], which provides standard RL environments like MuJoCo [28], ideal for testing control algorithms.

Each environment has a unique observation space, with parameters ranging from joint positions and angles to velocities, spanning a range of $(-\infty, +\infty)$, but typically within a limited range (e.g., $[0, 10]$). To provide meaningful network inputs, we determine effective ranges through multiple random runs, normalizing observations between 0 and 1. These normalized values are passed to the hidden neuron layer after being multiplied by the input connections.

The network's output is decoded by aggregating the dendritic currents (sum of synaptic inputs) of output neurons over multiple timesteps (n_{ts}). Outputs are

normalized using a `sigmoid` function and scaled to fit action bounds, mapping neuronal activity to control actions.

The hidden population in the 3D SNN is implemented with a single layer of LIF neurons with sparse recurrent connections. The connectivity is based on the spatial arrangement of neurons, as detailed in Sect. 2.1 Network Representation. The network runs for a set number of timesteps (n_{ts}) per environment step, allowing for input integration. Neuron states reset only at the end of each environment episode, preserving information across steps.

2.1 Network Representation

Euclidean Space Encoding. Neurons in the recurrent layer are assigned fixed positions in an n-dimensional grid. These positions are used in the evolution process, to determine the connection probability between two neurons in newly generated networks. It is based on the Euclidean distance d_{ij} between them:

$$P(w_{ij} \neq 0) = \begin{cases} e^{-1}, & \text{if } d_{ij} = 0 \\ e^{-d_{ij}^2}, & \text{otherwise} \end{cases} \tag{1}$$

This spatial dependence fosters locally connected networks. A distance-based scaling factor f_{ij}, ranging from 0 to 1, attenuates long-range connection strength:

$$f_{ij} = \begin{cases} 1.0, & \text{if } d_{ij} = 0 \\ (e^{-d_{ij}+1})^2, & \text{otherwise} \end{cases} \tag{2}$$

Input and output neurons are conceptually placed outside of the hidden recurrent network. The distance from input/output neurons (n_i) to any recurrent layer neuron (n_j) is artificially set to 1, ensuring a connection probability of e^{-1}, and no weight scaling.

Neurons and Synapses. The basic unit in our network is the Leaky Integrate-and-Fire (LIF) neuron, implemented using the Norse framework [19]. New networks have a ratio of 80% excitatory neurons (polarity of neuron i: $p_i = 1$) and 20% inhibitory ($p_i = -1$). The membrane potential $v_i(t)$ of neuron i follows:

$$\dot{v_i}(t) = \tau_{mem_i}(v_{leak_i} - v_i(t) + I_i(t)) \tag{3}$$

where τ_{mem_i} is the membrane time constant, v_{leak_i} is the leak potential, and $I_i(t)$ is the synaptic input current. The synaptic input current is:

$$\dot{I}_i(t) = -I_i(t)\tau_{syn_i} + \sum_j z_j(t) p_j w_{ij} f_{ij} \tag{4}$$

where τ_{syn_i} is the synaptic time constant. Each connection from presynaptic neuron j to postsynaptic neuron i has a polarity p_j, strength w_{ij}, and scaling factor f_{ij}. The outgoing value of neuron j ($z_j(t)$) is either a binary spike (hidden

neurons) or a numeric value (input neurons). When $v_i(t)$ reaches the threshold v_{th_i}, the neuron spikes:

$$z_i(t) = \Theta(v_i(t) - v_{th_i}) \tag{5}$$

where Θ is the Heaviside step function. After firing, $v_i(t)$ resets to v_{res_i}. In our implementation, continuous dynamics are approximated with discrete updates:

$$\begin{aligned} i_{\text{jump}} &= I_i(t) + \sum_j z_j(t) p_j w_{ij} f_{ij} \\ dv &= \Delta t \cdot \tau_{mem_i} \left(v_{leak_i} - v_i(t) + i_{\text{jump}}\right) \\ v_{\text{decayed}} &= v_i(t) + dv \\ di &= -\Delta t \cdot \tau_{syn_i} \cdot i_{\text{jump}} \\ I_i(t+1) &= i_{\text{jump}} + di \\ z_i(t+1) &= \Theta(v_{\text{decayed}} - v_{th_i}) \\ v_i(t+1) &= (1 - z_i(t+1)) \cdot v_{\text{decayed}} + z_i(t+1) \cdot v_{res_i} \end{aligned}$$

All the neuron and synapse-specific parameters are optimized through evolution, with their ranges and other hyperparameters summarized in Table 1.

Table 1. Parameter Values

Variable	Value Range
v_{res}	0, 0
v_{leak}	0, 127
v_{th}	1, 1000
w	0.1, 127
τ_{mem}	10, 1000
τ_{syn}	10, 1000

Mutation	Probability
add synapse	0.01
delete synapse	0.05
change weight	0.1
change polarity	0.05
change v_{th}	0.1
change v_{res}	0.1
change v_{leak}	0.1
change τ_{mem}	0.1
change τ_{syn}	0.1

Hyperparameter	Value
$\#seeds_{train}$	3
$\#seeds_{test}$	100
n_p	100
n_g	1000
n_{ts}	64
Δt	0.001
excitatory ratio	0.8

2.2 Evolutionary Algorithm

The EA iteratively refines a population of SNNs across $n_g = 1000$ generations through selection, crossover, and mutation (Fig. 1). Individuals are ranked using a weighted combination of performance and behavioral descriptors. A curriculum learning strategy accelerates evolution and improves sample efficiency. The maximum episode length starts at 10 steps, is multiplied by 1.1 after each generation, and is clipped at 1000 steps. This allows rapid evaluation and pruning of poor initial networks, while gradually increasing episode length to enable discovery of complex, long-term behaviors as the population improves.

Initialization. A population of $n_p = 100$ SNNs is initialized with parameters randomly sampled within the bounds defined in Table 1.

Evaluation. Each SNN is evaluated on the control task for $\#seeds_{train} = 3$ fixed environment seeds during training and for $\#seeds_{test} = 100$ for testing. The fitness is the mean reward across all training seeds. The following behavioral descriptors (scaled between [0,1]) are also measured:

- f_1: Number of active output neurons (quantifying joint usage).
- f_2: Standard deviation of active output neuron values over time (measuring joint utilization, promoting either erratic or smooth movements).
- f_3: Connection sparsity between synapses (measuring solution efficiency).

Selection and Variation. A ranking identifies $m = \lfloor \sqrt{n_p} \rfloor = 10$ elite networks and is determined by a weighted sum of the fitness score and behavioral descriptors. The weights for reward, f_3, f_1, and f_2 are 100, 10, 1, and 0.1, respectively. 90% of the next generation comprises the elites (10), a mutated copy of each elite (10) and offspring (70). The 10% rest of the pool consists of (10) randomly initialized networks. Offspring are produced by applying mutation to the result of crossover between two elites. Elite pairs for crossover are selected sequentially from a list ordered by rank: [0,1], ..., [0,m], then [1,2], ..., [1,m], and so on. The variation operators are defined as follows, based of values in Table 1:
Crossover: Combines two parent networks and creates two offspring:

- Half of the neurons from each parent are randomly selected.
- Selected neurons, along with their neuron-specific parameters (including polarities), are exchanged between parents.
- Outgoing synaptic connections (including recurrent, input-to-hidden, and hidden-to-output) of the selected neurons are swapped.

Mutation: Introduces random modifications:

- Synaptic connections: Added with probability $add\ synapse = 0.01$, removed with probability $delete\ synapse = 0.05$, or have their weight modified with probability $change\ weight = 0.1$ (within $w \in (0.1, 127)$).
- Neuron properties: Polarity is altered with probability $change\ polarity = 0.05$; threshold with probability $change\ v_{th} = 0.1$; reset potential with probability $change\ v_{res} = 0.1$; leak potential with probability $change\ v_{leak} = 0.1$; membrane time constant with probability $change\ \tau_{mem} = 0.1$; and synaptic time constant with probability $change\ \tau_{syn} = 0.1$.

Network Cleanup. After crossover and mutation, neurons without input connections (excluding self-connections), neurons without output connections (excluding self-connections), and isolated sub-networks (chains of neurons not connected to input or output neurons) are removed. This ensures all neurons are functionally connected.

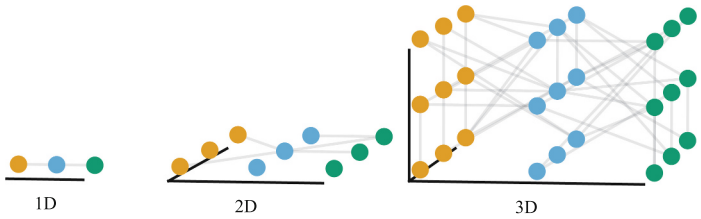


Fig. 2. Conceptual visualization of the different spatial embeddings investigated.

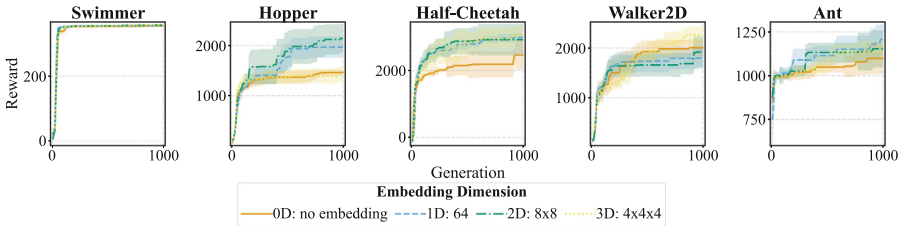


Fig. 3. Performance comparison of spatial embeddings across five environments. Plots show reward during training, evaluated on 100 unseen test seeds. Results are averaged over three independent runs; shaded regions represent standard error.

3 Experiments and Results

3.1 Spatial Embedding

To assess the impact of spatial embedding, we investigate four 64-neuron configurations, visualized in Fig. 2: 0D (no spatial embedding, connections with probability 0.1), 1D (64 neuron line), 2D (8×8 square), and 3D ($4 \times 4 \times 4$ cube). We select these embeddings as a way to investigate the effect of biological constraints on network performance and sparsity. Furthermore, higher dimensional embeddings did not present a meaningful choice, given the relatively small numbers of neurons in our network.

These embeddings are evaluated across five Gymnasium [29] environments: Swimmer-v5, Hopper-v5, Half-Cheetah-v5, Walker2D-v5, and Ant-v5. Figure 3 shows the test fitness of the best individual of each generation for each spatial embedding and environment.

3.2 Sparsity

Spatially embedded networks show a high number of near-zero weights (Fig. 4), a consequence of the distance-based weight scaling factor f_{ij} (Eq. 2).

To exploit this, we prune network weights ($w \in (0, 127)$) using absolute thresholds of 0, 0.01, 0.1, 3, 10, and 30. We then evaluate their sparsity and performance. Figure 5 shows the results, with sparsity defined as the percentage of zero weights. The total number of weights per environment are: Swimmer: 4736, Hopper: 4992, Half-Cheetah: 5568, Walker2D: 5568, and Ant: 11328.

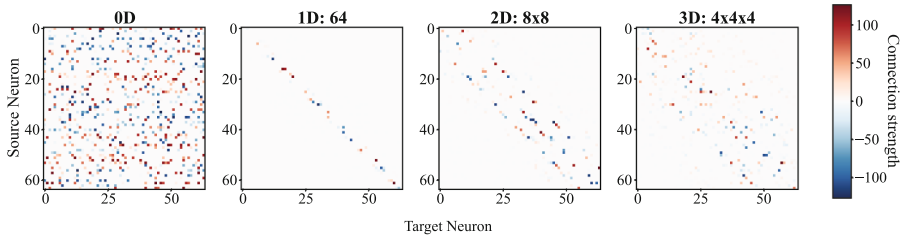


Fig. 4. Connectivity strength between hidden neurons for different dimensional embeddings. In 0D, connection probability is uniform. In nD, distance determines connection likelihood and strength. Higher dimensions allow more connections per neuron.

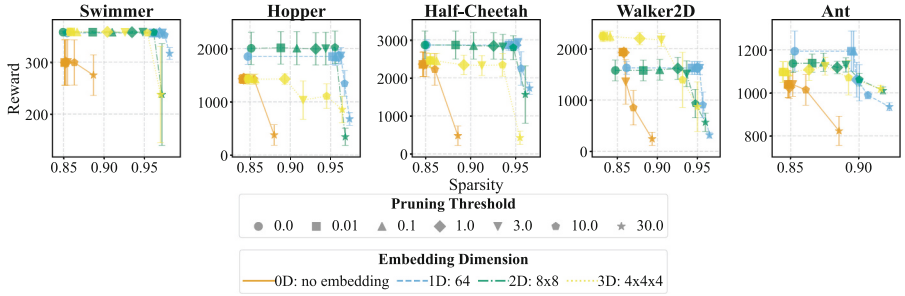


Fig. 5. Performance-sparsity plots of pruned SNNs across five environments. Markers represent different weight pruning thresholds. Sparsity is the percentage of zero weights. Error bars represent standard error over three training runs. Total weight counts are: Swimmer: 4736, Hopper: 4992, Half-Cheetah: 5568, Walker2D: 5568, Ant: 11328.

3.3 Comparison with State-of-the-Art Baselines

Table 2 compares our spatially embedded models to the best performing models ("expert") from the Farama-Minari repository [32], trained with PPO [24], SAC [8], and TQC [10], by analyzing their architectures [1] and weight counts.

Baseline models have between 9,408 and 475,392 weights, while ours have significantly fewer: 137 (Swimmer), 222 (Hopper), 703 (HalfCheetah), 694 (Walker2d), and 1208 (Ant). This corresponds to parameter reductions of 98.54% to 99.94%. Relative performance is 99.48% (Swimmer), 66.03% (Hopper), 20.81% (HalfCheetah), 35.82% (Walker2d), and 23.04% (Ant).

4 Discussion

Comparing the performance of spatial embeddings, we observe that they consistently outperformed non-embedded networks within our framework (Fig. 3). Furthermore, the 2D spatial embedding generally achieved the best performance especially in less complex environments, which may represent evidence for the similar 2D sheet-like structures observed in the brain, as cortical sheets [27].

Table 2. Comparison of Spatially Embedded SNNs vs. SOTA Baselines with standard deviation shown. Our models are embedded in 2D (Swimmer, Hopper, HalfCheetah), 3D (Walker2d), and 1D (Ant). Pruning thresholds are: 30.0 (Swimmer), 10.0 (Hopper), 0.01 (HalfCheetah), 1.0 (Walker2d), and 0.1 (Ant). "NZW" is the number of non-zero weights. "RWC" (relative weight count) and "RP" (relative performance) are expressed as a percentage of the baseline values.

Environment	SOTA Baseline		Spatially Embedded			
	NZW	Reward	NZW	RWC	Reward	RP
Swimmer	9,408	363.70 ± 0.06	137	1.46%	361.81 ± 0.17	99.48%
Hopper	347,392	4098.17 ± 7.83	222	0.06%	2706.10 ± 7.37	66.03%
HalfCheetah	384,256	17641.80 ± 1.96	703	0.18%	3670.43 ± 119.83	20.81%
Walker2d	359,680	6956.61 ± 0.50	694	0.19%	2491.96 ± 24.77	35.82%
Ant	475,392	5846.28 ± 43.80	1,208	0.25%	1346.85 ± 3.51	23.04%

The 1D and 3D embeddings proved especially efficient for specific environments. These preliminary findings should be confirmed by future studies through evaluation on an increased number of control environments, as the current number may be insufficient to draw definitive conclusions and to fully capture the diversity of continuous control.

Furthermore, spatially embedded networks achieved high sparsity levels (over 95%) with minimal performance impact (Fig. 5). The 1D embedding was particularly robust, allowing for pruning of weights below 3 without any performance degradation in most environments.

Table 2 reveals a substantial non-zero weight (NZW) reduction (98.54%-99.94%) compared to state-of-the-art models, highlighting the efficiency achievable through spatial embedding and pruning. However, a performance trade-off exists. While our best performing model achieved near-identical performance (99.48%) to the baseline on the simpler Swimmer environment using only 1.46% of the NZWs, more complex environments exhibited a larger performance gap (23.04% for Ant, 20.81% for HalfCheetah), which generally increased with task complexity. This suggests that while the current network size of 64 neurons is sufficient for simple tasks, larger networks are likely necessary for more complex problems. Despite this, our models still achieved meaningful performance on these complex tasks (outperforming the "simlpe" models in the Farama-Minari repository), while using less than 0.3% of the NZWs of the baselines.

A key limitation of the current study is the computational cost of the framework. Simulating larger networks, which preliminary experiments suggest would improve performance, is currently infeasible due to the implementation's inefficiency. Evaluating a single environment for five embeddings across 1000 generations and three training seeds required approximately 150 h on a server equipped with 2x AMD EPYC 7713 64-Core Processors, 3x NVIDIA RTX A6000 48 GB, 4x NVIDIA RTX 6000 48 GB, and 512 GB DDR4-3200 RAM.

5 Conclusion

We introduced a novel evolutionary framework for optimizing recurrent SNNs. Instead of treating SNNs as abstract networks, we positioned neurons within a 3D Euclidean space, mirroring the physical organization of biological brains. This spatial embedding, coupled with distance-dependent connections, naturally fostered the evolution of locally connected, sparse networks. A genetic algorithm then refined not just the network's connections, but also the parameters of each neuron (such as firing thresholds and time constants), a capability often lacking in gradient-based reinforcement learning approaches.

Spatially embedded SNNs consistently surpassed the performance of non-embedded networks within our framework. Moreover, the evolved networks were remarkably sparse, with over 95% of all possible connections eliminated while maintaining the same performance. Compared to state-of-the-art deep reinforcement learning models, our SNNs achieved similar performance on simpler tasks while using as little as 1.46% of the non-zero weights (NZWs). Despite the extreme NZW reduction (some models using less than 0.1% of the baseline NZWs), with increasing task complexity, a larger performance gap emerged. This is due to the limited size of our networks, using only 64 neurons, as this work is intended as a proof-of-concept regarding the potential of evolving spatially embedded SNNs.

These findings point to spatial embedding as a candidate for the practical application of SNNs in low-power control. Future work will focus on scaling this approach to larger networks and improving the computational efficiency of the evolutionary optimization, to tackle more complex control tasks. The ultimate goal is to develop SNN-based controllers that are not only powerful but also highly efficient, ideal for deployment in real-world, resource-limited scenarios.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

Code Availability. The source code for our experiments is available on GitHub: https://github.com/Alex-Vasilache/Spatial-SNNs

References

1. farama-minari (Minari), Februay 2025, https://huggingface.co/farama-minari, Accessed 05 May 2025
2. Achterberg, J., Akarca, D., Strouse, D.J., Duncan, J., Astle, D.E.: Spatially embedded recurrent neural networks reveal widespread links between structural and functional neuroscience findings. Nat. Mach. Intell. **5**(12), 1369–1381 (2023). https://doi.org/10.1038/s42256-023-00748-9
3. Akl, M., Sandamirskaya, Y., Ergene, D., Walter, F., Knoll, A.: Fine-tuning deep reinforcement learning policies with r-STDP for domain adaptation. In: Proceedings of the International Conference on Neuromorphic Systems 2022, pp. 1–8. ACM, Knoxville TN USA, July 2022. https://doi.org/10.1145/3546790.3546804

4. Bogdan, P.A., Rowley, A.G., Rhodes, O., Furber, S.B.: Structural plasticity on the spinnaker many-core neuromorphic system. Front. Neurosci. **12**, 434 (2018)
5. Deb, M., Deb, M., Murty, N.: Toponets: high performing vision and language models with brain-like topography. arXiv preprint arXiv:2501.16396 (2025)
6. Elbrecht, D., Schuman, C.: Neuroevolution of spiking neural networks using compositional pattern producing networks. In: International Conference on Neuromorphic Systems 2020, pp. 1–5. ACM, Oak Ridge TN USA, July 2020. https://doi.org/10.1145/3407197.3407198
7. Elsken, T., Metzen, J.H., Hutter, F.: Neural architecture search: a survey (2019)
8. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor (2018)
9. Kim, Y., Li, Y., Park, H., Venkatesha, Y., Panda, P.: Neural architecture search for spiking neural networks, July 2022, arXiv:2201.10355 [cs, eess]
10. Kuznetsov, A., Shvechikov, P., Grishin, A., Vetrov, D.: Controlling overestimation bias with truncated mixture of continuous distributional quantile critics (2020)
11. Liu, Y., Pan, W.: Spiking neural-networks-based data-driven control. Electronics **12**(2), 310 (2023). https://doi.org/10.3390/electronics12020310
12. Lu, J., Hagenaars, J.J., de Croon, G.C.H.E.: Evolving-to-learn reinforcement learning tasks with spiking neural networks, February 2022, arXiv:2202.12322 [cs]
13. Maass, W.: Networks of spiking neurons: the third generation of neural network models. Neural Netw. **10**(9), 1659–1671 (1997)
14. Man, Y., Xie, L., Qiao, S., Zhou, Y., Shang, D.: Differentiable architecture search with multi-dimensional attention for spiking neural networks. Neurocomputing **601**, 128181 (2024). https://doi.org/10.1016/j.neucom.2024.128181
15. Na, B., Mok, J., Park, S., Lee, D., Choe, H., Yoon, S.: AutoSNN: towards energy-efficient spiking neural networks. In: Proceedings of the 39th International Conference on Machine Learning, pp. 16253–16269. PMLR, June 2022, iSSN: 2640–3498
16. Neuromorphic, O.: Neuromorphic Hardware Guide, Accessed 05 June 2025
17. Pan, W., Zhao, F., Shen, G., Zeng, Y.: Multi-scale evolutionary neural architecture search for deep spiking neural networks, November 2023, arXiv:2304.10749
18. Pan, W., Zhao, F., Zhao, Z., Zeng, Y.: Brain-inspired evolutionary architectures for spiking neural networks, September 2023, arXiv:2309.05263 [cs]
19. Pehle, C.G., Egholm Pedersen, J.: Norse-a deep learning library for spiking neural networks. Zenodo (2021)
20. Putra, R.V.W., Shafique, M.: SpikeNAS: a fast memory-aware neural architecture search framework for spiking neural network-based autonomous agents, April 2024, arXiv:2402.11322 [cs]
21. Qi, Y., et al.: Jointly learning network connections and link weights in spiking neural networks, p. 1603, July 2018. https://doi.org/10.24963/ijcai.2018/221
22. Qiu, H., Garratt, M., Howard, D., Anavatti, S.: Evolving spiking neural networks for nonlinear control problems. In: 2018 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1367–1373, November 2018. https://doi.org/10.1109/SSCI.2018.8628848, arXiv:1903.01180 [cs]
23. Rafe, A.W., Garcia, J.A., Raffe, W.L.: Exploration of encoding and decoding methods for spiking neural networks on the cart pole and lunar lander problems using evolutionary training. In: 2021 IEEE Congress on Evolutionary Computation (CEC), pp. 498–505. IEEE, Kraków, Poland, June 2021. https://doi.org/10.1109/CEC45853.2021.9504921
24. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. CoRR **abs/1707.06347** (2017)

25. Shen, J., Xu, Q., Liu, J.K., Wang, Y., Pan, G., Tang, H.: ESL-SNNs: an evolutionary structure learning strategy for spiking neural networks. Proc. AAAI Conf. Artif. Intell. **37**(1), 86–93 (2023). https://doi.org/10.1609/aaai.v37i1.25079
26. Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. Evol. Comput. **10**(2), 99–127 (2002). https://doi.org/10.1162/106365602320169811
27. Tax, C.M., et al.: Quantifying the brain's sheet structure with normalized convolution. Med. Image Anal. **39**, 162–177 (2017)
28. Todorov, E., Erez, T., Tassa, Y.: Mujoco: a physics engine for model-based control. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5026–5033. IEEE (2012)
29. Towers, M., et al.: Gymnasium, March 2023. https://doi.org/10.5281/zenodo.8127026
30. Wang, G., Sun, Y., Cheng, S., Song, S.: Evolving connectivity for recurrent spiking neural networks. In: Advances in Neural Information Processing Systems, vol. 36, pp. 2991–3007. Curran Associates, Inc. (2023)
31. Wu, G., Liang, D., Luan, S., Wang, J.: Training spiking neural networks for reinforcement learning tasks with temporal coding method. Front. Neurosci. **16**, 877701 (2022). https://doi.org/10.3389/fnins.2022.877701
32. Younis, O.G., Perez-Vicente, R., Balis, J.U., Dudley, W., Davey, A., Terry, J.K.: Minari, September 2024. https://doi.org/10.5281/zenodo.13767625
33. Zanatta, L., Barchi, F., Manoni, S., Tolu, S., Bartolini, A., Acquaviva, A.: Exploring spiking neural networks for deep reinforcement learning in robotic tasks: a comparative study, June 2024. https://doi.org/10.21203/rs.3.rs-4470688/v1
34. Zhou, Y., Jin, Y., Ding, J.: Surrogate-assisted evolutionary search of spiking neural architectures in liquid state machines. Neurocomputing **406**, 12–23 (2020). https://doi.org/10.1016/j.neucom.2020.04.079



Amortizing Personalization in Virtual Brain Twins

Nina Baldy(✉), Marmaduke M. Woodman, and Viktor K. Jirsa

Aix Marseille Univ, INSERM, INS, Inst Neurosci Syst, Marseille, France
{nina.baldy,marmaduke.woodman}@univ-amu.fr

Abstract. Virtual brain twins are personalized digital models of individual human subject or patient's brains, allowing for mechanistic interpretation of neuroimaging data features. Training & inference with these models however presents a pair of challenges: (a) large shared infrastructure do not allow for use of personal data and (b) inference in clinical applications should not require significant resources. We introduce "anonymized personalization" to address both by expanding model priors to include personalization which under amortized inference allows training to be performed anonymously, while inference is both personalized and lightweight. We illustrate the basic approach, demonstrate reliability in an example, and discuss the impact on both experimental and computational neuroscience. Code is available at https://github.com/ins-amu/apvbt

Keywords: Virtual brain twins · Amortized inference · Personalized medicine

1 Introduction

Virtual brain twins (VBTs) represent a synthetic approach in computational neuroscience, offering personalized digital models of individual brains that enable mechanistic interpretations of neuroimaging data features [6,7,13,20] and the underlying causal organization [5,10]. These models hold significant potential for advancing our understanding of brain function, neurological disorders, and personalized medicine. However, their broad application present significant challenges that must be addressed to fully realize their clinical and research utility.

One major challenge lies in the training and inference processes required for these models. Traditional approaches often rely on large shared infrastructure [17], which poses a critical barrier due to the sensitivity of brain imaging data. The use of such infrastructure requires stringent privacy protections, complicating the integration of sensitive patient information into model development. Additionally, use in the clinic requires efficient and resource-light inference methods to ensure practicality in real-world settings.

To overcome these challenges, we propose expanding model priors to incorporate personalization within an amortized inference framework. This approach

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 78–88, 2026.
https://doi.org/10.1007/978-3-032-04555-3_7

allows for anonymous training while enabling personalized and lightweight inference, addressing both the privacy concerns associated with sensitive data and the computational demands of clinical applications. By integrating prior knowledge about brain structure and function into the model architecture, we can train models without direct access to individual patient data, thereby preserving privacy while maintaining the ability to generate accurate and personalized VBTs.

This paper is organized as follows: Sect. 2 outlines our methodology, detailing how we expand model priors for personalization and implement amortized inference. Section 3 presents our results, including an assessment on a test case. Finally, Sect. 4 discusses the broader implications of this work for computational neuroscience and clinical applications.

2 Methods

2.1 Virtual Brain Twins

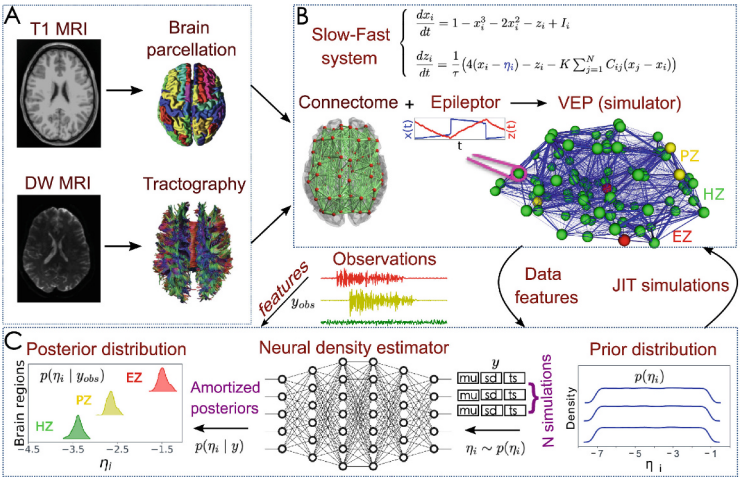


Fig. 1. Virtual brain twins provide mechanistic interpretation of neuroimaging data. An example of amortized inference workflow as developed by Hashemi et al. [7] for clinical epilepsy. MRI images (T1 & diffusion-weighted) are used to reconstruct brain geometry, parcellation & connectome. Nonlinear dynamics are assigned to each node, with connectome providing the interactions between nodes. The model is simulated for many samples in parameter space, training a deep Bayesian estimator to map data features to posterior distributions over model parameters. Once trained, the estimator can be rapidly applied to unseen data to provide new inferences. This figure from Hashemi et al. [7] was reproduced with permission.

VBT models describe whole brain neural activity, and, through forward models, neuroimaging data such as EEG or fMRI. The models are composed of a

set of nodes representing regions of the brain, each with dynamics described by differential equations, and a connectome, which describes the weighted connections between the regions, usually derived from diffusion weighted imaging, as described below in Sect. 2.2. For the models used here, we employ a low dimensional description of neural dynamics [14], in terms of N nodes' mean firing rates $\boldsymbol{r}(t)$ and mean membrane potentials $\boldsymbol{v}(t)$,

$$\dot{\boldsymbol{r}} = \Delta/\pi + 2\boldsymbol{r}\boldsymbol{v} + k\boldsymbol{C}\boldsymbol{r} \tag{1}$$

$$\dot{\boldsymbol{v}} = \boldsymbol{v}^2 + \eta + J\boldsymbol{r} + I - \pi^2\boldsymbol{r}^2 \tag{2}$$

where $\boldsymbol{C}$ is the matrix of connectome weights, and default parameter values $\tau = 1.0$, $I = 0.0$, $\Delta = 1.0$, $J = 15.0$, $\eta = -5.0$ generate bistable dynamics. Simulations of the VBT model involves sampling parameters of interest, integrating the differential equations (potentially stochastic and/or delayed), computing relevant data features. For our tests, we consider as parameters only coupling scaling k and noise scaling D as parameters of interest, with uninformative uniform priors, i.e. $k \sim U(0.1, 0.3)$ and $D \sim U(0.2, 0.4)$. The data feature used for inference is the mean value of $\boldsymbol{r}_i(t)$ at steady state.

A typical workflow for inference with a VBT is illustrated in Fig. 1. The canonical computational implementation for simulating such models is provided by The Virtual Brain software package [16], freely available from https://thevirtualbrain.org.

2.2 Personal Data: Connectomes

In typical use of VBTs, the personal data of interest is the set of connection weights describing how regions in the brain communicate, enabling the model to provide personalized predictions of brain dynamics. In this report, we make use of MRI scans 461 subjects across a German cohort in the 1000BRAINS project (1KB) over a range of ages collected by [3] and an American cohort of subjects part of the Human Connectome Project (HCP) described by [19], with the connectomes reconstructed under 20 parcellations by [12]. Because the connectomes represent raw counts of tracks between regions with large variance, they are preprocessed by (a) clipping values to 1 to 99th percentiles, (a) normalizing between 0 and 1, and (c) taking the square root per connectome, then removing the mean over all connectomes, such that subsequent steps work with residual deviations around preprocessed mean connectomes.

2.3 Priors over Personal Data

While inference with VBTs has generally taken the connectome as a fixed piece of data on which the model and posterior are conditioned, we will instead consider the connectome as a parameter to be inferred, alongside the other parameters of interest. Human connectomes derived from diffusion-weighted imaging are symmetric, leading to additional $(N^2 - N)/2$ parameters (ignoring self connections). We regularize this high-dimensional set of parameters by introducing a

low-rank representation, in an approach similar to an autoencoder [8] that we term *cross-coder* since it collects information across parcellations: each subject in the dataset has their connectome represented in 20 parcellations. We train a two layer architecture to encode the connectome of a subject shown in Fig. 2, from all parcellations into the same latent space, and decode from that latent space to all parcellations, with a l_2 loss:

$$L_2 = \sum_{i,j,k} \| \boldsymbol{c}_{j,k} - \left((\boldsymbol{c}_{i,k} \boldsymbol{W}_i^e + \boldsymbol{b}_i^e)^T \boldsymbol{W}_j^d + \boldsymbol{b}_j^d \right)^T \| \tag{3}$$

where $\boldsymbol{c}_{i,k}$ is a vector containing the upper triangle of the connectome for the kth subject in the ith parcellation, $\boldsymbol{W}_i^e$ and $\boldsymbol{b}_i^e$ are the weights matrices and bias vectors for the encoder from the ith parcellation, $\boldsymbol{W}_i^d$ and $\boldsymbol{b}_i^d$ similarly for the decoder. Stochastic gradient descent is performed with the Adam optimizer [11] in JAX [9], with a learning rate of 3e-4, until convergence, typically 1000 iterations, with batch size 64, requiring 2 min on an RTX 4090 or 20 min on Apple M3. The choice of a simple linear representation ensures (a) fast convergence despite small sample size (b) robust generalization and (c) inability to learn training data. A set of encoders & decoders were trained per cohort, and for each cohort one half of the connectomes were used for training (train-test split of 0.5).

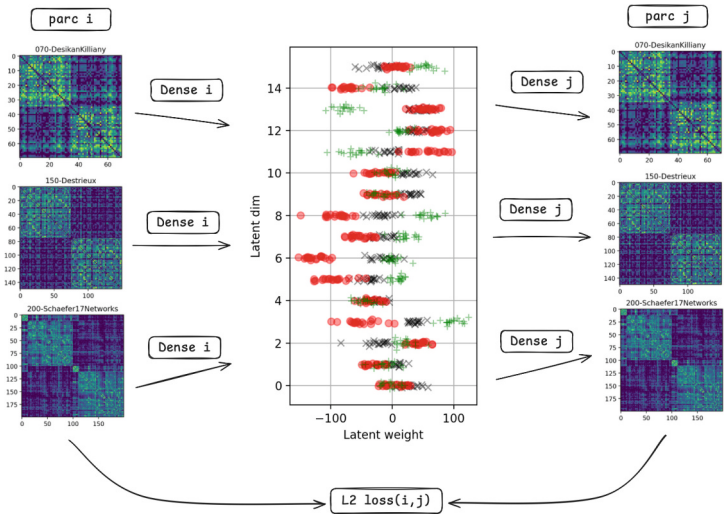


Fig. 2. Cross-coder architecture encodes connectomes. The cross-coder architecture, inspired by autoencoder, takes the subject connectome data in the ith parcellation, encodes with a single dense layer to a latent space and decodes to the jth parcellation, with a l_2 loss on the reconstruction error. Each encoder is informed not only by its own parcellation but also errors due to decoding to all other parcellations, and similarly for each decoder.

While the loss function drives the optimization, our application is to personalization, and we therefore assess the final result in terms of confusion rate, i.e. the probability of the cross-coder producing a connectome from subject k which is more similar to another subject l than k, with similarity assessed via Euclidean distance.

Finally, to obtain a Bayesian prior, we compute the mean and covariance over the latent project of all (training) subjects' connectomes from all parcellations. The resulting mean and covariance parameterize a multivariate normal distribution usable for sampling. We remark that while this distribution is derived from personal data, because it averages over many subjects, it is no longer personal data subjected to privacy concerns. Such prior distribution of connectivity could be referred to as a cohort-based prior.

2.4 Simulation Based Inference

Simulation-based inference (SBI) is a family of techniques that formulates Bayesian inference as a Bayesian regression problem, in which the data features parameterize (via a neural network) a distribution over the parameters [4]. The process starts with a training step in which the prior is sampled, and a simulator computes corresponding data features for samples from the prior, which then allows training the neural network. The posterior distribution is thus shaped through a sequence of invertible transformations, and once trained, can be quickly sampled from by providing data features. Here we use amortized neural posterior estimation [15] as implemented by [18] in the PyTorch-based Python package `sbi`, https://sbi-dev.github.io/sbi/latest/. We evaluate the results with model-based calibration metrics [1], specifically posterior shrinkage and posterior z-score

$$s_n = 1 - \frac{\sigma_n^2(y)}{\tau_n^2(y)} \tag{4}$$

$$z_n = \left| \frac{\mu_n(y) - \theta_n}{\sigma_n(y)} \right| \tag{5}$$

where τ_n and σ_n are standard deviations of the prior and posterior, μ_n is the posterior mean and θ_n is the true parameter value; and posterior predictive 90% confidence intervals. An acceptable inference process should show high shrinkage, low z-score, and the true value should lie in the 90% confidence region. As the main question studied here is replacing per-subject SBI (where the connectome is taken as fixed data and not estimated) by cohort-level SBI (i.e. absorbing the connectome as a parameter with the prior), we focus on evaluating the degree to which cohort-level SBI matches subject-level SBI performance.

3 Results

3.1 Cross-Coding Sufficient for Personalized Connectomes

Training the cross-coder architecture yields a low rank representation of connectomes, via gradient descent on a training set of 230 connectomes and 230

connectomes across the 1KB and HCP datasets, for latent dimension of 16. Once validation loss converges, the confusion rate is 2.3% on connectomes in the validation set, confirming that low rank representation successfully preserves the personalization of the connectomes.

Because the cross-coder uses the same latent space for all parcellations, we expected the different parcellations to project the same subject to similar latent coordinates. The middle panel in Fig. 2 shows three example subjects projected to the latent space of a trained cross-coder, showing much lower intra-subject variability compared to inter-subject variability. This verifies a crucial aspect of maintaining personalization of the connectomes in the low rank latent space.

3.2 Cohort SBI Provides Reliable Subject-Level Inference

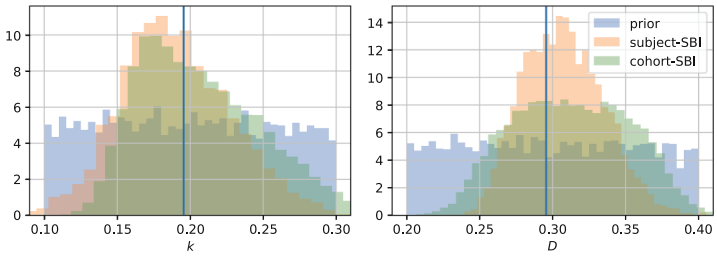


Fig. 3. Example of subject vs cohort-level inference. The prior (blue, uniform) contracts to the posterior in both subject-level (orange) and cohort-level SBI (green). The ground truth value is indicated by the blue vertical line. Both inferences provide similar posteriors. (Color figure online)

We evaluated cohort-level SBI, i.e. SBI over the parameters including the latent connectome parameters, vs subject-level SBI, where the connectome is fixed. The results shown are for the 79-node Shen 2013 parcellation only. Figure 3 shows the prior distribution and posteriors for the two parameters of interest: coupling scaling k and noise scaling D. While the posterior obtained with cohort-level SBI does not contract as significantly as for the subject-level SBI, there is a clear contraction towards the true value.

We next sought to evaluate the performance across all subjects in the test set (N=230), by performing subject-level SBI for each, and comparing with posterior obtained with the cohort-level SBI. In Fig. 4, the metrics reveal a very similar (correlation 0.8) distribution, and the rates at which the true value lies in the 90% CI of the posterior are 0.89 and 0.95 for cohort and subject-level SBI respectively. These results suggests that the cohort-level SBI provides a useful approximation of subject-level SBI, leveraging an amortized posterior that can be evaluated for any subject - while having been trained only once, with only

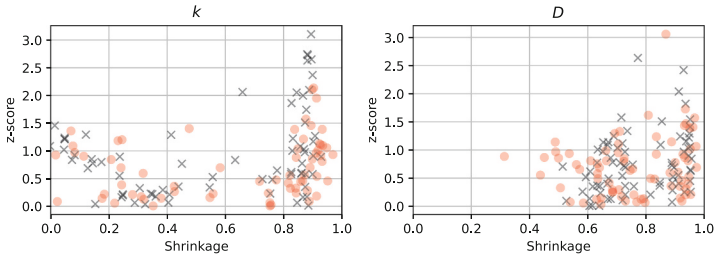


Fig. 4. Cohort-level SBI approximates subject-level SBI. Distributions of model-based calibration scores for all subjects in the test set (N = 230). Black cross represent scores of the cohort-level SBI (applied to each subject), while red dots indicate scores of each subject-level SBI. (Color figure online)

as many simulations as its subject-level counterpart. The computational complexity of the simulation process, for N subjects and M simulations, is O(NM) for per subject SBI, and O(M) for cohort SBI.

3.3 Cohort Inference Identifies the Connectome

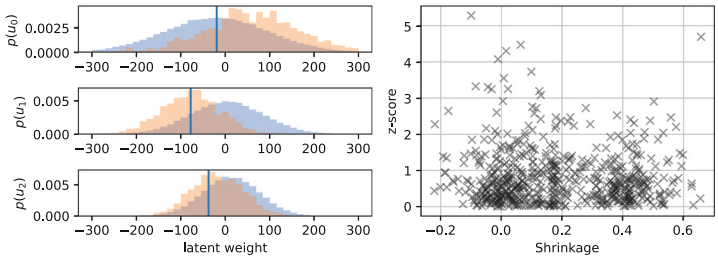


Fig. 5. Cohort-level SBI partially identifies latent connectome parameters. *Left*: The prior (blue) contracts to the posterior cohort-level SBI (orange) in most cases, here shown three latent connectome parameters. *Right*: Posterior shrinkage & z-scores for all connectomes in the test set, as identified via cohort-level SBI, showing a tendency toward identification if not as a strong as for parameters of interest. (Color figure online)

Lastly, for cohort-level SBI, we sought to evaluate how well the connectome is identified as a parameter. As shown in Fig. 5, there is a tendency towards identifying the latent connectome parameters, but not as strong as for parameters of interest. This may suggest that the model in the regime parameterized for each subject may not always be sensitive to the contribution of the individual connectome. In such case, the average connectome derived from a population could provide a reasonable fit to a given individual. In the more general case, the use of a cohort-based Bayesian prior allows this simplification to take place

(through the averaging over the latent project) without excluding outliers that do not match this average. However, given the metabolic cost of maintaining the connectome [2], shrinkage of the posterior over connectomes may signal a physiologically-relevant regime of the parameters of interest, helping to guide model construction and evaluation.

4 Discussion

The development of VBTs with expanded model priors represents a significant advancement in computational neuroscience. We have demonstrated that incorporating cohort-based priors over personal data enables amortized inferences on par with per-subject personal inference, allowing new applications where personal data is unavailable (because it was not recorded or because of privacy protections). While this technique does not replace state of the art, computationally intensive workflows involving e.g. MCMC, but rather to a strong compromise between performance, flexibility and inference efficiency.

4.1 Limitations

The proposed cohort-level inference serves as a useful approximation of subject-specific models but does not fully capture all the individual details of a person's brain connectivity. This is evidenced by the finding that the posterior distributions from cohort-level SBI do not contract as significantly as those from subject-level SBI. Furthermore, while the model parameters of interest like coupling and noise are well-identified, the framework is less successful at strongly identifying the individual latent connectome parameters themselves. Rather than a methodological deficit, we suggest this as degree of model degeneracy where, in certain parameter regimes, the model's dynamics may not be highly sensitive to the fine-grained details of an individual's connectome.

4.2 Implications for Translational Research

This framework significantly lowers the barrier for translating VBTs into clinical and large-scale research settings. The "anonymized personalization" approach allows the computationally expensive training of a model to be performed on large, shared high-performance computing infrastructure without requiring access to sensitive personal data. The resulting trained model can then be used by clinicians for lightweight and personalized inference on new patients without needing complex retraining or patient-specific tuning, enabling practical real-time analysis and decision support. This approach also makes it feasible to scale personalized inferences across thousands of subjects, overcoming the prohibitive computational burden typically associated with large cohorts (computational complexity for N subjects and M simulations is reduced from O(NM) for per-subject SBI to O(M) for cohort SBI).

Our approach also addresses the common research problem of incomplete datasets, enabling new analyses where they were previously impossible: per-subject diffusion imaging needed to create a structural connectome is not always available. This method makes it possible to build and infer parameters for a VBT model in such cases by substituting the missing data with a cohort-based prior derived from a similar population.

4.3 Future Directions

A crucial next step is to validate the cohort-based inference framework in specific, well-established domains of whole-brain modeling, namely resting-state dynamics and clinical epilepsy. While this work demonstrates the method's general reliability, applying it to resting-state models would allow for rigorous validation against empirical functional connectivity, a standard benchmark in the field. In epilepsy, where personalized models are already used to delineate epileptogenic networks, this approach could significantly accelerate clinical translation by enabling models to be trained on large, anonymized cohorts and then rapidly applied to new patients, testing the framework's ability to capture both normative and pathological brain states.

A second extension exploiting amortization lies in high-resolution brain models. The brain models employed in this study are based on "low resolution" connectomes with tens to hundreds of regions. Recent work [20] clearly motivated the need for spatially detailed models for resolving causal effects, however these models require even more resources to run. To ensure inference is tractable for a wide range of users, our amortized approach should be evaluated.

Finally, the computational efficiency of the amortized inference process makes this approach a prime candidate for novel real-time clinical applications, such as model-based neurofeedback. Standard inference methods are too slow for real-world clinical use, but once a cohort model is trained, the amortized posterior can be evaluated almost instantly for a new subject. This speed is essential for neurofeedback, where a subject's brain state must be inferred and fed back to them with minimal delay. A VBT operating in real-time could offer mechanistic insights to help guide a patient in modulating their own brain activity, representing a powerful therapeutic avenue that is infeasible with current computationally intensive techniques.

4.4 Conclusions

Amortizing personalization in VBTs represents a significant step toward making sophisticated, mechanistic brain models practical tools for both science and medicine. By expanding model priors to include cohort-level information, our method successfully enables anonymous training and lightweight, efficient inference while still retaining individual personalization. This approach provides a strong compromise between performance and efficiency, overcoming critical barriers related to data privacy and computational cost that have hindered the widespread adoption of VBTs. Ultimately, this work helps bridge the gap

between complex research models and actionable insights in real-world clinical environments.

Acknowledgments. This project/research has received funding from the European Union's Horizon Europe Programme under the Specific Grant Agreements No. 101147319 (EBRAINS 2.0 Project) and No. 101137289 (Virtual Brain Twin Project).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Betancourt, M.: Calibrating model-based inferences and decisions. arXiv preprint arXiv:1803.08393 (2018)
2. Bullmore, E., Sporns, O.: The economy of brain network organization. Nat. Rev. Neurosci. **13**, 336–349 (2012). https://doi.org/10.1038/nrn3214
3. Caspers, S., et al.: Studying variability in human brain aging in a population-based German cohort–rationale and design of 1000brains. Front. Aging Neurosci. **6**, 149 (2014)
4. Cranmer, K., Brehmer, J., Louppe, G.: The frontier of simulation-based inference. Proc. Natl. Acad. Sci. **117**(48), 30055–30062 (2020)
5. Fousek, J., Rabuffo, G., Gudibanda, K., Sheheitli, H., Petkoski, S., Jirsa, V.: Symmetry breaking organizes the brain's resting state manifold. Sci. Rep. **14**(1), 31970 (2024)
6. Hashemi, M., et al.: Principles and operation of virtual brain twins. IEEE Rev. Biomed. Eng. 1–25 (2025). https://doi.org/10.1109/RBME.2025.3562951
7. Hashemi, M., et al.: Amortized Bayesian inference on generative dynamical network models of epilepsy using deep neural density estimators. Neural Netw. **163**, 178–194 (2023)
8. Hinton, G.E.: Connectionist learning procedures. Artif. Intell. **40**(1), 185–234 (1989). https://doi.org/10.1016/0004-3702(89)90049-0
9. JAX-authors: Jax: Composable transformations of Python+NumPy programs (2025). https://github.com/google/jax. Accessed 06 Feb 2025
10. Jirsa, V., Sheheitli, H.: Entropy, free energy, symmetry and dynamics in the brain. J. Phys. Complex. **3**(1), 015007 (2022)
11. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
12. Kyesam, J., Eickhoff, S.B., Popovych, O.V.: Parcellation-based structural and resting-state functional whole-brain connectomes of 1000brains cohort (v1.1) (2022)
13. Lavanga, M., et al.: The virtual aging brain: causal inference supports interhemispheric dedifferentiation in healthy aging. Neuroimage **283**, 120403 (2023)
14. Montbrió, E., Pazó, D., Roxin, A.: Macroscopic description for networks of spiking neurons. Phys. Rev. X **5**(2), 021028 (2015)
15. Papamakarios, G., Murray, I.: Fast ε-free inference of simulation models with Bayesian conditional density estimation. Adv. Neural Inf. Process. Syst. **29** (2016)
16. Sanz Leon, P., et al.: The virtual brain: a simulator of primate brain network dynamics. Front. Neuroinform. **7**, 10 (2013)

17. Schirner, M., et al.: Brain simulation as a cloud service: the virtual brain on ebrains. Neuroimage **251**, 118973 (2022)
18. Tejero-Cantero, A., et al.: sbi: a toolkit for simulation-based inference. J. Open Source Softw. **5**(52), 2505 (2020). https://doi.org/10.21105/joss.02505
19. Van Essen, D.C., et al.: The WU-Minn human connectome project: an overview. Neuroimage **80**, 62–79 (2013)
20. Wang, H.E., et al.: Delineating epileptogenic networks using brain imaging data and personalized modeling in drug-resistant epilepsy. Sci. Transl. Med. **15**(680), eabp8982 (2023)

MPCCP: A Multi-chain Perception Crime Charge Prediction Method

Congshan Huang[1], Tianshuo Jiao[1], Qiao Hu[1], Yupeng Hu[1], and Bianxia Du[2(✉)]

[1] Hunan University, Changsha, China
{huangcongshan,s2310w0963,huqiao,yphu}@hnu.edu.cn
[2] Changsha Vocational and Technical College, Changsha, China
13366969235@163.com

Abstract. In the context of intelligent justice constructing, using artificial intelligence technology to improve judicial efficiency and fairness has become an important research direction. Crime charge prediction, as an important task of intelligent justice, aims to accurately determine the corresponding crime based on the case description. However, most existing methods only focus on the semantic level and ignore that syntactic understanding can reduce misjudgments caused by semantic biases. Therefore, this paper proposes a multi-chain perception crime charge prediction method, MPCCP. This method aims to achieve a deeper semantic and syntactic understanding of case texts through graph neural networks. It designs a multi-chain perception mechanism to explore the potential connections between information. The mechanism captures the complex relationships between criminal behaviors through long-distance propagation and multi-step inference on semantic and syntactic information and introduces a rule engine to effectively predict the principal and subordinate crime charges that overlap in cases. Experimental results show that this method has achieved significant performance improvements in the crime charge prediction task.

Keywords: Crime Charge Prediction · Graph Neural Network · Multi-chain Perception · Rule Engine

1 Introduction

In the digital age, the legal field is accelerating its transformation towards informatization and intelligence. Artificial intelligence plays a significant role in enhancing judicial efficiency and fairness. Legal Judgment Prediction (LJP) predicts judgment outcomes by analyzing case facts and legal elements, while Crime Charge Prediction, as a key subtask [3], aims to accurately determine crimes based on case descriptions. Early research on crime charge prediction primarily relied on rule-based methods and machine learning methods, but these methods struggled to handle complex legal texts. With the advancement of deep neural networks, methods such as CNN [7], RNN [16], and BERT [4] have improved

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 89–101, 2026.
https://doi.org/10.1007/978-3-032-04555-3_8

performance by capturing deeper textual meanings. Although these methods excel at capturing sequential word relationships, graph neural networks (GNN), such as GCN [5], GAT [10], and GRN [6], are more effective in comprehensively modeling semantic relationships in legal texts.

However, most current GNN only on the semantic level, neglecting the fact that syntactic understanding can help reduce misjudgments caused by semantic biases. As shown in Fig. 1(a), merely based on semantic analysis, it could easily be mistaken for fraud. This is because the use of the word "deceived" implies the use of deceptive means, and "taking away 50,000 yuan" may be understood as the result of fraud. However, syntactic analysis shows that the "deceived" act led to Party B handing over the bank card and password, rather than directly transferring the property. "Without Party B's knowledge" serves as a conditional adverbial, indicating that the "taking away" act was carried out secretly, which supports the determination of theft. Therefore, when faced with complex legal logics, these methods struggle to accurately capture semantic and syntactic relationships and reason complex cases.

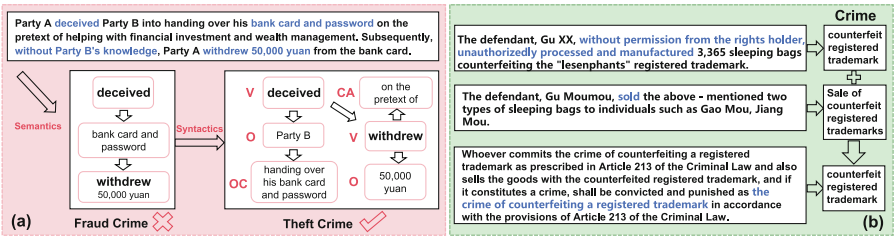


Fig. 1. Case studies: (a) Syntactic analysis reduces semantic bias in fraud/theft cases; (b) Principal-subordinate charge prediction in trademark infringement cases.

Inspired by the multi-defendant crime charge prediction in the MUD framework proposed by Wei et al. [14], we have identified significant shortcomings in existing methods when dealing with principal and subordinate crime charges. In such scenarios, one criminal act may be absorbed by another, resulting in only the absorbed offense being convicted and sentenced. However, determining which behavior is the absorbing behavior and accurately predicting the final principal and subordinate crime charges pose a significant challenge to existing methods, such as large language models(LLM) like GPT-3 [1]. As shown in Fig. 1(b), the crime of selling registered trademarks is absorbed by the crime of counterfeiting registered trademarks, with conviction and punishment based on the latter, in accordance with Article 213 of the Criminal Law.

Based on this, our crime charge prediction method needs to solve three key problems: (1) How to comprehensively capture the relationships within case texts from both semantic and syntactic perspectives; (2) How to design a mechanism that can accurately understand and capture key information in the graph

structure; (3) In the context of principal and subordinate crime charges, how to accurately determine the final convicted crime charge.

Therefore, this paper proposes a multi-chain perception crime charge prediction method, MPCCP. This method constructs syntactic and semantic graphs of criminal behavior descriptions through dependency analysis, enabling the model to further analyze the syntactic structure among words based on semantics. Then, by constructing a multi-chain mechanism, it enables information to perform multi-step reasoning and handle long-distance dependencies in the graph structure. The multi-chain transfer algorithm based on the attention mechanism dynamically adjusts the weights of information propagation according to the correlations and importance among nodes, thus capturing key information more effectively. Finally, by introducing a rule engine, the model can automatically detect and execute judicial rules before predicting criminal charges, ensuring that the prediction results are consistent with judicial practice.

The main contributions of this paper are as follows: (1) This method leverages the syntactic perspective and combines a multi-chain perception mechanism to understand the deep logical relationships between criminal behaviors; (2) This method introduces a rule engine, which effectively solves the prediction problem of principal and subordinate crime charges and significantly improves the accuracy of crime charge prediction; (3) Comprehensive experiments were conducted on two publicly available judicial datasets, demonstrating that the model outperforms existing competitive baselines in performance.

2 Related Work

2.1 Crime Charge Prediction

Crime charge prediction is generally regarded as a text classification task, in which crime charges serve as class labels. Current research mainly focuses on neural network-based methods. Deep learning possesses powerful feature learning capabilities, enabling it to more accurately understand complex case texts and extract key information. For example, Sukanya et al. [11] proposed a legal judgment prediction framework, MHAN, based on improved RNN and BiLSTM. These deep learning methods have shown significant performance improvements but still face many limitations when dealing with complex crime scenarios.

Recently, LLM [1,4,12] have demonstrated excellent performance in various conventional tasks. Chalkidis et al. [2] proposed Legal-BERT, a BERT model pre-trained on legal corpora, which outperforms general BERT in several legal natural language processing tasks. Li et al. [8] demonstrated GPT-3's fine-tuning capabilities in simple legal reasoning and legal rule classification tasks. However, the application of such LLM in complex legal judgment prediction tasks remains largely unexplored.

2.2 Graph Neural Networks

Early the GNN introduced graph convolution operations, enabling effective learning of graph structures. Subsequently, the proposal of the GAT further

promoted the development of GNN. The GAT enables the model to adaptively learn the interaction relationships between nodes by introducing an attention mechanism.

Currently, many crime charge prediction methods based on graph neural networks have demonstrated excellent performance. For example, inspired by graph convolution operators, Xu et al. [17] proposed the end-to-end model LADAN, which uses graph neural networks to learn the differences between easily confused crime charges. Zhao et al. [18] proposed the LA-MGFM model, which accurately integrates legal provision information through a multi-graph fusion mechanism and effectively handles easily confused crime charges. Pan et al. [10] proposed a context-based legal article graph (CLAG) and the CLAGN model, enabling the representations of facts and circumstances to enhance each other. Wang et al. [13] proposed the GraSCL model based on supervised contrastive learning to mine potential dependency structures and promote relationship learning under various graph topologies. However, these methods primarily capture semantic correlations through graph neural networks. When facing more complex criminal behaviors, they struggle to comprehensively capture associative relationships within case texts from both semantic and syntactic perspectives simultaneously, and they also perform poorly in handling long-distance dependencies.

3 Method

In this section, we will discuss each module of MPCCP (Fig. 2.) in detail.

3.1 Graph Module

Syntactic Graph. We use the HanLP toolkit to perform dependency syntactic analysis on the text describing criminal acts. For every word w_i in the text, we identify its syntactic dependency relationships with other words. For instance, in the sentence "Gu XX manufactured and produced sleeping bags with counterfeit 'lesenphants' registered trademarks without permission from the rights holder," the words "manufactured and produced" have a verb-object relationship with "sleeping bags." We represent this relationship as an edge in the syntactic graph. The syntactic graph $G_{syn} = (V_{syn}, E_{syn})$ is thus constructed, where $V_{syn} = \{w_1, w_2, \cdots, w_n\}$ denotes the set of words in the text, and $E_{syn} = \{(w_i, w_j)|r_{ij}\}$ represents the syntactic relationships between words, with r_{ij} indicating the specific type of syntactic relationship. As shown in Fig. 3(a), in the syntactic graph, the predicate "manufactured and produced" typically serves as the root node. The immediate child nodes connected to the root node form the first chain, which usually includes the subject "Gu XX," the object "sleeping bags," and the adverbial phrase "without permission." Grandchild nodes constitute the second chain, and so on.

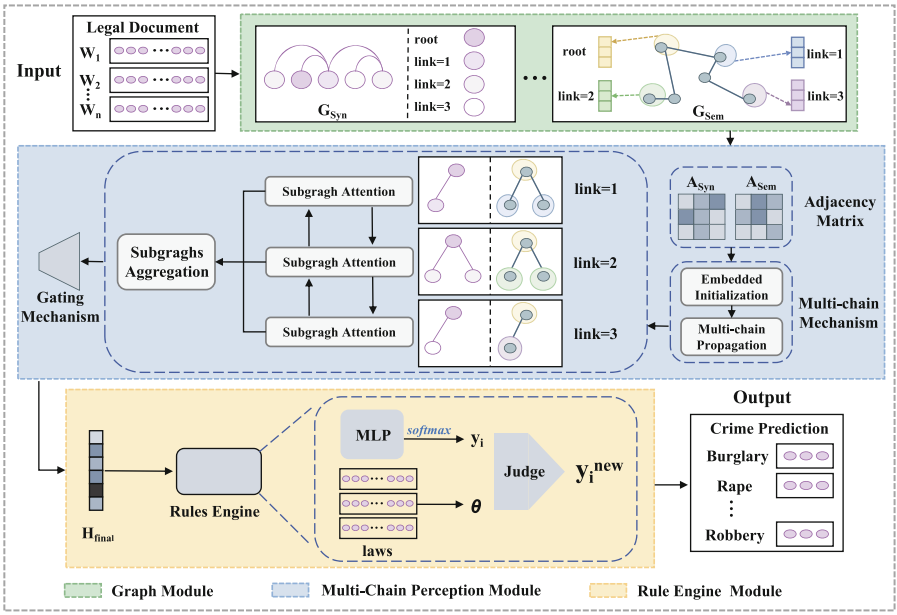


Fig. 2. Framework Diagram of MPCCP.

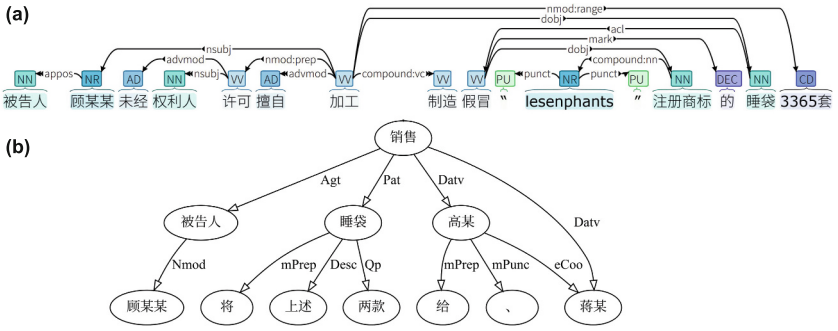


Fig. 3. (a) Syntactic Dependency Graph. (b) Semantic Dependency Graph.

Semantic Dependency Graph. Similar to the syntactic graph, we conduct semantic dependency parsing using HanLP. As shown in Fig. 4, the semantic analysis captures the agent-patient relationship between "sold" and "sleeping bags", as well as the action-recipient relationship between "sold" and "Gao, Jiang". These relationships are used to construct the semantic graph $G_{sem} = (V_{sem}, E_{sem})$, in which verbs usually serve as root nodes in the semantic graph.

3.2 Multi-chain Perception Module

Adjacency Matrix Generation. For the syntactic graph G_{syn}, we generate an adjacency matrix A_{syn} of size $n \times n$. If there is a syntactic relationship between w_i and w_j, the corresponding element $A_{syn}[i][j]$ is set to a value representing the strength of the relationship. This value can be determined based on the type of syntactic relationship or a pre-defined weight mapping. Typically, we determine the initial weight based on the number of chains and the components involved. For example, the first chain has a higher weight than the second chain, subjects and objects have higher weights than adverbials, and unconnected elements are set to 0. Similarly, for the semantic graph G_{sem}, we generate the adjacency matrix A_{sem} in the same way.

Multi-chain Mechanism. The multi-chain mechanism captures the long-range dependencies between words through multi-step propagation on different chains. Let h_t denote the hidden state of a certain node on the t-th propagation chain. The initial hidden state h_0 is initialized based on the word embeddings of the nodes in the syntactic graph and the semantic graph. The information propagation formula is as follows:

$$h_{t+1} = \sum_{i=1}^{n} \sum_{j=1}^{n} \tilde{A}_{syn}[i][j]\sigma(W_{syn}h_t[i]) + \tilde{A}_{sem}[i][j]\sigma(W_{sem}h_t[i]) \tag{1}$$

where W_{syn} and W_{sem} are the learnable weight matrices for the syntactic and semantic graphs, respectively, and $\tilde{A}$ is the normalized adjacency matrix.

Subgraph Attention Mechanism. We propose a subgraph attention mechanism to focus on key information in graphs and enhance the understanding of graph structures.This mechanism allows the model to dynamically allocate attention within each subgraph, highlighting nodes and edges that are more critical for the current task. For each node i in subgraphs with different chains, we compute attention scores between nodes. Let the nodes in the subgraph be represented as $v_1, v_2, \ldots, v_m$, and their corresponding feature vectors be $h_1, h_2, \ldots, h_m$. First, we map the node features to a new space using a shared linear transformation:

$$e_{ij} = LeakyReLU\left(W_1 \cdot [h_i; h_j]\right) \tag{2}$$

where W_1 is a learnable weight matrix, and e_{ij} represents the attention score of node i with respect to node j.

Next, we normalize the attention scores using the $softmax$ function to obtain the attention coefficients α_{ij}.

Finally, we perform a weighted sum of the node features using the attention coefficients to obtain the updated node features:

$$h'_i = \sum_{j=1}^{m} \alpha_{ij} \cdot h_j \tag{3}$$

Gating Mechanism. To control the flow of information and prevent overfitting, we introduce a gating mechanism. The gating signal g_i for each node i is computed as follows:

$$g_i = \sigma\left(W_g[h_i \parallel h_{t+1}[i]]\right) \tag{4}$$

where W_g is a learnable weight matrix.

The final node representation $H_{final}[i]$ after gating is given by:

$$H_{final}[i] = g_i \odot h_{t+1}[i] + (1 - g_i) \odot h_i \tag{5}$$

3.3 Rule Engine Module

After processing through the multi-chain perception module, the model obtains accurate feature representations. The role of the rule engine module is to adjust the model's predictions based on predefined rules to improve prediction accuracy in specific scenarios. In this module, a Multi-Layer Perceptron (MLP) is first used to map the high-dimensional features H_{final} to the probability distribution space of crime categories:

$$z = W_{fc}H_{\text{final}} + b_{fc} \tag{6}$$

We apply the $softmax$ function to normalize the output vector, thus obtaining the probability distribution y_i of each crime. Next, the rule engine is introduced to define rules for specific principal-subordinate relationships. We predefine a threshold θ. If the predicted scores for two crimes both exceed this threshold θ (which can be based on judicial experience), a principal-subordinate relationship may exist between them. In such cases, we assign the sum of the predicted scores to the principal crime and set the predicted score of the subordinate crime to 0 (absorption process). This rule-based adjustment aligns more closely with actual legal judgment logic. Let y_p and y_a denote the predicted scores for the principal and subordinate crime charges, respectively. The rule can be expressed as:

$$\begin{cases} y_i^{new} = \begin{cases} y_p + y_a & \text{if } y_p > \theta \text{ and } y_a > \theta \\ y_p & \text{otherwise} \end{cases} \\ y_i^{new} = \begin{cases} 0 & \text{if } y_p > \theta \text{ and } y_a > \theta \\ y_a & \text{otherwise} \end{cases} \end{cases} \tag{7}$$

Finally, the crime with the highest probability is selected as the final prediction result.

4 Experiments

4.1 Datasets

We evaluated our model on two publicly available legal datasets: CAIL2018 and LAIC2021. These datasets are widely used in intelligent justice research, providing diverse descriptions of criminal cases and their corresponding crime charges.

CAIL2018: The CAIL2018 dataset [15] was released by the China AI and Law Challenge (CAIL) and is categorized into CAIL-Large and CAIL-Small versions based on scale, with data sourced from China Judgments Online.

LAIC2021: The LAIC2021 dataset originates from the 2021 Legal AI Challenge (LAIC), containing factual descriptions, criminal charges, sentencing information, and other legal elements. Liu et al. [9].

4.2 Benchmarks

We compared the MPCCP with several baseline models that have shown excellent performance in the crime charge prediction task:

- MHAN [11]: A multi-head attention network that combines RNN and BiLSTM, used for legal judgment prediction.
- Legal-BERT [2]: A BERT model pre-trained on legal corpora, suitable for legal NLP tasks.
- LADAN [17]: A graph-based model that uses graph convolutional networks to learn the differences between easily confused crime charges.
- LA-MGFM [18]: A multi-graph fusion model that integrates legal provisions and case facts for crime charge prediction.
- CLAGN [10]: A context-aware legal article graph network that enhances the representation of facts and legal articles.
- GraSCL [13]: A graph-based supervised contrastive learning model that mines potential dependency structures for crime charge prediction.

4.3 Experimental Setup

To ensure the fairness of experimental comparisons, we strictly adhere to the official partitioning standards of both datasets. All models are implemented in PyTorch and evaluated on dual GeForce RTX 4090 GPUs, utilizing the Adam optimizer with an initial learning rate of 1e-5 that undergoes gradual decay at a rate of 1e-6 during training. Considering the differing average lengths of the two datasets, we set the maximum sequence lengths to 500 for CAIL and 300 for LAIC respectively, and the batch size is 32. Both semantic and syntactic dependency graphs are configured with 3 chains, and the rule threshold θ is fixed at 0.3. We use an early stop strategy for the label accuracy of the validation set, with a patience of 10 epochs. Performance is assessed using four metrics: Macro-Precision (MP), Macro-Recall (MR), Macro-F1 (MF1), and Accuracy (ACC).

4.4 Experimental Results

As shown in Table 1, the experimental results are based on the average value of 5 test runs. The best-performing results are highlighted in bold, and the second-best results are underlined. The results show that MPCCP outperforms

the SOTA models in four metrics. On the CAIL-Small dataset, compared with GraSCL, which performs the second best, MPCCP has improved by 1.28%, 0.48%, and 0.67% in terms of ACC, MP, and MR respectively. However, the MF1 has decreased by 1.38%, which is because certain infrequent crime categories in the CAIL dataset have a significant impact on the MF1 metric.

On the LAIC dataset, MPCCP is 0.41%, 0.35%, 0.44%, and 0.23% higher than GraSCL and LA-MGFM, which rank second, in terms of ACC, MP, MR, and MF1 respectively. Although LA-MGFM integrates information using a multi-graph fusion mechanism, it is less effective than the multi-chain perception of MPCCP in handling long-distance dependency relationships. On the other hand, GraSCL places too much emphasis on using graph reasoning and supervised contrastive learning to mine label relationships, ignoring the importance of syntactic information, and thus fails to accurately understand cases involving complex syntactic structures. The results show that MPCCP can effectively improve the performance of crime charge prediction.

Table 1. Results of Crime charge prediction.

Method	**CAIL-Small**				**LAIC**			
	ACC	MP	MR	MF1	ACC	MP	MR	MF1
MHAN	86.20	84.30	84.90	83.50	87.21	84.85	85.00	84.24
Legal-BERT	86.80	84.90	82.50	83.10	88.10	85.24	83.10	84.15
LADAN	84.83	83.33	82.80	82.85	84.53	83.01	82.21	83.04
LA-MGFM	88.65	87.74	87.90	87.96	89.25	<u>88.44</u>	<u>88.42</u>	<u>88.95</u>
CLAGN	85.63	83.48	83.89	83.03	85.32	84.18	84.74	84.41
GraSCL	<u>89.03</u>	<u>88.33</u>	<u>88.23</u>	**89.21**	<u>89.55</u>	88.33	88.12	88.76
MPCCP	**90.17**	**88.75**	**88.82**	<u>88.00</u>	**89.92**	**88.75**	**88.81**	**89.16**

4.5 Ablation Experiments

To verify the effectiveness of different modules of the MPCCP method, we compared it with the following variants:

- MPCCP¬SemG : Removing the semantic graph module, which makes the model lose the ability to perceive the semantic dependency structure of case documents.
- MPCCP¬SynG : Removing the syntax graph module, which makes the model lose the ability to perceive the syntactic dependency structure of case documents.
- MPCCP-MCGAT : Replace the multi-chain perception module with the multi-chain GAT to verify its effectiveness in dealing with long-range dependencies and complex legal reasoning.

– MPCCP¬RE : Removing the rule engine module, which makes the model lose the ability to handle the prediction scenarios of principal and subordinate crime charges.

As shown in Table 2, MPCCP¬SemG exhibits the most significant performance drop compared to MPCCP, while the other three variants also demonstrate decreased performance across both datasets. These results collectively indicate that: (1) the semantic graph module plays a crucial role in capturing semantic dependencies among case texts; (2) the syntactic graph module effectively facilitates the analysis of legal documents' syntactic structures; (3) the multi-chain perception module outperforms conventional GAT in information aggregation; and (4) the rule engine ensures better alignment between prediction outcomes and legal practice.

Table 2. Results of ablation study

Method	**CAIL-Small**		**LAIC**	
	ACC	MF1	ACC	MF1
MPCCP¬SemG	75.12	71.12	72.74	70.04
MPCCP¬SynG	86.50	82.21	86.28	85.51
MPCCP-MCGAT	84.14	83.02	83.41	83.57
MPCCP¬RE	87.54	86.20	87.74	86.84
MPCCP	**90.17**	**88.00**	**89.92**	**89.16**

4.6 Qualitative Analysis

To explore the impact of the perception range and the rule engine on crime charge prediction performance, we designed a series of experiments on the optimal number of chains k for semantic and syntactic graphs and the optimal threshold θ for principal-subordinate crime rules.

As shown in Fig. 4, when k is less than 3, the performance of the model improves as the number of chains increases. When $k = 3$, the model achieves the best performance, and indicators such as ACC reach their peak values. This is because when the number of chains is too small, the information propagation step length and the capture range of the multi-chain perception module are limited, making it difficult to understand the complex logical relationships between criminal behaviors. However, when k continues to increase, the performance of the model decreases instead. This indicates that an excessive number of chains may introduce an excessive amount of noise information, interfering with the model's judgment of key information and leading to a decrease in performance.

As shown in Fig. 4, when θ is small, specifically between 0.1 and 0.2, the rule engine may overactivate the judgment rules for principal and subordinate

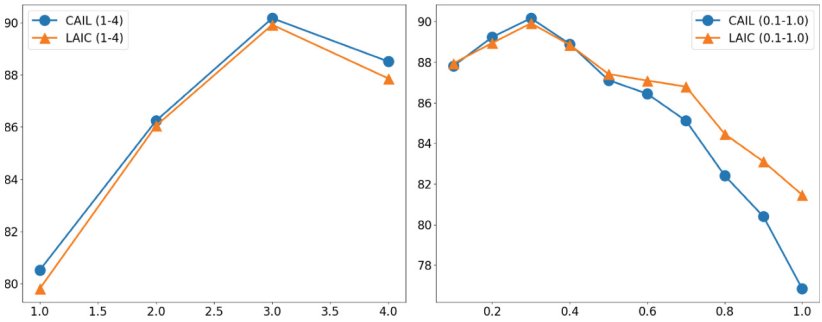


Fig. 4. Qualitative Analysis Results.

crime charges, misclassifying some crimes that should not be regarded as having a principal-subordinate relationship, thus resulting in poor performance. As θ gradually increases, the model can more reasonably identify the principal-subordinate relationships, and its performance improves step by step, reaching the best state when $\theta = 0.3$. When θ continues to increase, the rule engine module becomes overly conservative in judging the principal-subordinate relationships. Some crimes that actually have a principal-subordinate relationship are not correctly identified because their prediction scores do not meet the higher threshold, which in turn reduces the performance of the model. This indicates that when $\theta = 0.3$, the rule engine can adjust the prediction scores based on judicial experience, making the scores of principal crimes more accurately reflect their dominant role in the case, thereby enhancing the overall performance of the model.

5 Conclusions

This paper proposes a crime charge prediction method with multi-chain perception, MPCCP. This method can better comprehensively understand the information in case texts and effectively solve the problem of predicting principal and subordinate crime charges, ensuring that the prediction results are in line with judicial practice. Experiments have verified that MPCCP has achieved the best performance among current methods. For future work, it is planned to apply MPCCP to more scenarios such as legal article recommendation and sentence prediction, to verify its effectiveness and generalization ability in different judicial tasks.

Acknowledgments. This work is supported by the China National Key Research and Development Program (No. 2022YFC3301500) and the Changsha Municipal Natural Science Foundation (grant no. kq2502348).

References

1. Brown, T., et al.: Language models are few-shot learners. Adv. Neural. Inf. Process. Syst. **33**, 1877–1901 (2020)
2. Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., Androutsopoulos, I.: LEGAL-BERT: The Muppets straight out of law school. Findings of the Association for Computational Linguistics: EMNLP 2020 (2020)
3. Chao, Q., Shao, S., Hu, H.: Tree-CP: an interpretable crime prediction model based on tree structure and keyword extraction. In: Third International Conference on Advanced Algorithms and Neural Networks (AANN 2023). vol. 12791, pp. 265–273. SPIE (2023)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), pp. 4171–4186 (2019)
5. Gan, J., et al.: Multigraph fusion for dynamic graph convolutional network. IEEE Trans. Neural Netw. Learn. Syst. **35**(1), 196–207 (2022)
6. Ji, Z., Chen, K., He, Y., Pang, Y., Li, X.: Heterogeneous memory enhanced graph reasoning network for cross-modal retrieval. SCIENCE CHINA Inf. Sci. **65**(7), 172104 (2022)
7. LeCun, Y., et al.: Backpropagation applied to handwritten zip code recognition. Neural Comput. **1**(4), 541–551 (1989)
8. Liga, D., Robaldo, L.: Fine-tuning GPT-3 for legal rule classification. Comput. Law Secur. Rev. **51**, 105864 (2023)
9. Liu, Y., et al.: ML-LJP: multi-law aware legal judgment prediction. In: Proceedings of the 46th international ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1023–1034 (2023)
10. Pan, W., Chen, Y., Liu, Z., Li, X., Xu, Z.: Circumstance-aware graph neural network for legal judgment prediction. In: 2023 International Conference on Asian Language Processing (IALP), pp. 332–337. IEEE (2023)
11. Sukanya, G., Priyadarshini, J.: A meta analysis of attention models on legal judgment prediction system. Int. J. Adv. Comput. Sci. Appl. **12**(2) (2021)
12. Touvron, H., et al.: LLaMA: Open and efficient foundation language models (2023)
13. Wang, J., Le, Y., Cao, D., Lu, S., Quan, Z., Wang, M.: Graph reasoning with supervised contrastive learning for legal judgment prediction. IEEE Trans. Neural Netw. Learn. Syst. (2024)
14. Wei, X., Xu, Q., Yu, H., Liu, Q., Cambria, E.: Through the mud: a multi-defendant charge prediction benchmark with linked crime elements. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2864–2878 (2024)
15. Xiao, C., et al.: CAIL2018: A large-scale legal dataset for judgment prediction (2018)
16. Xiao, J., Zhou, Z.: Research progress of RNN language model. In: 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), pp. 1285–1288. IEEE (2020)

17. Xu, N., Wang, P., Chen, L., Pan, L., Wang, X., Zhao, J.: Distinguish confusing law articles for legal judgment prediction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics (2020)
18. Zhao, Q., Gao, T., Guo, N.: LA-MGFM: a legal judgment prediction method via Sememe-enhanced graph neural networks and multi-graph fusion mechanism. Inf. Process. Manag. **60**(5), 103455 (2023)

Learning a Kernel-Based Beran Estimator Using Nearest-Neighbours and Its Application to Reliability Analysis

Danilo Samuel Jodas[1(✉)], Christian Laurence Almeida Barry[1], Guilherme Brandão Martins[1], Marcos Cleison Santana[1], Andre Luis Severino Abrego[2], Danilo Colombo[2], and João Paulo Papa[1]

[1] São Paulo State University (UNESP), School of Sciences, Bauru, Brazil
{danilo.jodas,christian.barry,guilherme.brandao,joao.papa}@unesp.br
[2] CENPES/PETROBRAS, Rio de Janeiro, Rio de Janeiro, Brazil
{andre.abrego,colombo.danilo}@petrobras.com.br

Abstract. Reliability analysis plays a crucial role in various domains for modeling time-to-event data; however, traditional approaches often struggle with complex and nonlinear data distributions. Among the non-parametric methods, the Beran estimator stands out as a robust method employed for estimating the reliability of equipment in industrial environments. Built upon it, the Beran estimator with Neural Kernels (BENK) aims to address some limitations its counterpart faces, albeit at the cost of careful hyperparameter fine-tuning, particularly the number of neural subnetworks that implement the neural kernels, which can be addressed by using simpler methods, like the k-nearest neighbors algorithm. Although widely used, finding suitable values for the neighborhood's size is not straightforward. This work introduces a parameterless k-Nearest Neighbors algorithm to the context of kernel-based estimators, particularly the Beran estimator, and evaluates it in reliability analysis for downhole safety valves, an essential device concerning security issues in oil wells. We demonstrate that our parameterless approach maintains or surpasses the predictive capability of survival functions compared to traditional hyperparameter-dependent methods, providing a robust and adaptable tool for reliability analysis that effectively handles complex, nonlinear data distributions without requiring dataset-specific calibration.

Keywords: Beran estimator · Survival analysis · Reliability analysis · Nearest neighbors · Machine learning

1 Introduction

Reliability analysis is a robust statistical framework designed to investigate the time until the occurrence of critical events, like equipment failure [15] and disease relapse [13]. Unlike traditional regression models, reliability analysis accommodates censored data, i.e., instances where the event of interest has not occurred

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 102–114, 2026.
https://doi.org/10.1007/978-3-032-04555-3_9

within the observation period. The core of reliability analysis figures on the estimation of key functions like the *survival function*, which represents the probability of an individual or system surviving beyond a specific time, and the *hazard function*, which quantifies the instantaneous risk of event occurrence. Traditional methods such as the Kaplan-Meier estimator [11] and the Cox Proportional Hazards Model [4] represent the foundation for time-to-event analysis, making them robust statistical inferences under specific assumptions. However, machine learning techniques such as Random Survival Forests [8] have emerged to capture nonlinear relationships and interactions without strict parametric premises, thus improving prediction capabilities in a wide range of scenarios [7].

Among the nonparametric methods, the Beran estimator stands out as a robust method employed in conditional survival analysis [1]. One of its key strengths lies in its ability to effectively accommodate the effects of covariates without assuming a specific parametric form for the survival distribution. Traditionally, the Beran estimator uses Gaussian, Epanechnikov, or uniform kernels to measure the similarity between feature vectors when estimating reliability functions. However, according to Kirpichenko et al. [12], it is challenging to model complex structures in the data, especially when the feature vectors show intricate structures, such as in a spiral. To address this limitation, the authors introduced a novel method called Beran Estimator with Neural Kernels (BENK), built on the traditional Beran estimator with kernel functions based on neural networks - termed neural kernels.

Although BENK demonstrates strong performance in capturing the data complexities, it requires careful hyperparameter tuning, particularly regarding the number of neural subnetworks that implement the neural kernels and the subset sizes used during training. These hyperparameters can impact the model's performance, generalization ability, and computational efficiency, requiring a prior step for hyperparameter optimization. The k-Nearest Neighbors (k-NN) algorithm, yet simple and robust, is susceptible to changes when setting the number of neighbors, i.e., the k parameter. Jodas et al. [9,10] addresses such an issue by presenting the Parameterless k-Nearest Neighbors (PL-kNN) model. This k-NN variant bypasses the selection of the k parameter by determining the optimal number of neighbors based on the data distribution.

This work addresses the task of estimating the reliability function to predict the survival probability of Downhole Safety Valves (DHSVs), which are critical components in the oil and gas industry. Calculating the reliability function for DHSVs plays an important role in predictive maintenance and risk assessment in offshore activities, as it enables the determination of the likelihood that equipment will operate beyond a specified duration. We investigate a modification of the Beran estimator by integrating a parameterless k-NN approach to capture relationships among samples' covariates and eliminate the need for manual tuning of hyperparameters. This approach eliminates the computational burden of extensive hyperparameter optimization typically required in kernel-based methods while maintaining or improving predictive performance.

The key contributions of this research include: (i) the application of a parameterless kernel-based estimator for reliability analysis by integrating PL-kNN with the Beran estimator; (ii) the elimination for manual hyperparameter tuning in reliability analysis applications; (iii) a reduced computational complexity while maintaining competitive or superior predictive performance to similar kernel-based approaches; and (iv) the application of the proposed estimator to a real-world dataset of Downhole Safety Valves utilized in the oil an gas industry.

2 Theoretical Background

The Beran estimator is a nonparametric method used in reliability and survival analysis to estimate the *conditional reliability function*. It extends the Kaplan-Meier estimator by incorporating covariate information through kernel smoothing, thus allowing reliability probabilities to be calculated while accounting for individual characteristics. This flexibility makes this estimator a robustness tool for analyzing datasets where hazard rates *vary* with covariates, unlike traditional reliability models that assume *fixed* covariate effects.

In the context of reliability analysis, conditional reliability function estimation plays an important role in predicting the likelihood that a component, such as a DHSV, will continue to function beyond a specified time. The reliability function is a monotonic curve representing the operating probabilities of such systems or components. In other words, it estimates the remaining observations that are still active at a specific time, where at time zero, all components are assumed to be working, resulting in a 100% survival probability. As time passes, this probability tends to decrease due to various factors, such as malfunctioning, degradadation or reaching the lifespan limit. Even though the reliability curve is monotonically decreasing, its shape depends on the failure mode, physics of failures, composition of the components, and operational and environmental condition operation of the system/component.

The Beran estimator for a conditional reliability function $S(t|\boldsymbol{x})$ at time t given the covariate vector $\boldsymbol{x}$ is defined as follows:

$$S(t \mid \boldsymbol{x}) = \prod_{T_i \leq t} \left(1 - \frac{K(\boldsymbol{x}_i - \boldsymbol{x})\delta_i}{\sum_{j:T_j \geq T_i} K(\boldsymbol{x}_j - \boldsymbol{x})} \right), \tag{1}$$

where t is the maximum observed time, T_i represents the i-th observed time, δ_i is the event indicator of the i-th observation (1 if the event occurred or 0 if censored), $\boldsymbol{x}_i$ denotes the covariate vector for the i-th individual, and $K(\cdot)$ is a kernel function. The condition $T_j \geq T_i$ represents all remaining times after T_i. The kernel function is critical to the performance of the Beran estimator. Common kernels include Gaussian and Epanechnikov kernels [14], with bandwidth selection methods such as cross-validation to optimize the balance between bias and variance. A smaller bandwidth captures finer details but increases variance, while a larger bandwidth smooths estimates at the cost of reduced sensitivity to local variations.

An advantage of the Beran estimator concerns its ability to model complex covariate relationships without assuming a specific reliability distribution. This aspect can benefit biomedical research, reliability engineering, and other domains where reliability or survival analysis is affected by multiple, potentially nonlinear factors. The estimator also handles right-censored data, which ensures robust estimation in incomplete datasets. Nonetheless, *kernel smoothing* may introduce *boundary bias* near the extremes of covariate distributions. Additionally, the computational complexity may *increase* while estimating the bandwidth in the cross-validation approach [14]. Despite these challenges, advances in computational methods continue to enhance the estimator's applicability in large-scale survival analysis.

3 Proposed Method

The Beran estimator can be represented through a kernel function $K(\boldsymbol{x}^*, \mathcal{V}, \mathcal{Y})$, where $\boldsymbol{x}^*$ is the target covariate vector, $\mathcal{V}$ is the set of training samples, and $\mathcal{Y}$ represents the corresponding labels indicating failure or censoring. In this framework, the estimator learns the relationship between the covariates and the reliability probabilities via kernel-based smoothing. However, this process can be interpreted as a classification problem within the PL-kNN paradigm. The kernel function $K(\boldsymbol{x}^\star, \mathcal{V}, \mathcal{Y})$ computes a weight for each training observation based on its similarity to the target covariate $\boldsymbol{x}^*$. This weight is influenced by the proximity of the covariate vectors and the labels $\mathcal{Y}$, which provide crucial information about the survival or censoring status of the observation.

Algorithm 1 implements the proposed Beran estimator with PL-kNN as the kernel function[1]. The algorithm estimates the reliability function using the proposed kernel based on PL-kNN. The procedure operates in two main loops: (i) the first computes kernel weights for each time point, and (ii) the second estimates the reliability probability at each time step. The algorithm begins by initializing necessary variables at Lines 1–3. The total number of observations is stored in n (Line 1). A zero-initialized array, S, stores the reliability probabilities (Line 2). An empty list, M, is also prepared to accumulate the computed kernel weights for later use (Line 3).

In the first loop (Line 4), for each event time $T[i]$, the algorithm identifies the set R_i of observations still at risk, defined as a set containing the indices of observations for which the event times $T[j]$ meet the condition $T[j] \geq T[i]$ (Line 5). In other words, the set R_i will include indices $j = i, i+1, i+2, \ldots, n-1$ of observations $\boldsymbol{x_j}$ with times greater than or equal to the current time $T[i]$ being analyzed. Notice that the times are sorted in ascending order. The corresponding covariates set $\mathcal{V}$ and the at-risk individuals' censoring indicators $\mathcal{Y}$ are then extracted in Lines 6 and 7, respectively. A kernel function K is applied to the target covariates set $\boldsymbol{x}^\star$ and the at-risk training set $\mathcal{V}$ and $\mathcal{Y}$ to compute a weight representing the local influence of each observation. The pair ($\mathcal{V}$, $\mathcal{Y}$) is used to

[1] The code is available at http://github.com/danilojodas/beran-estimator.

Algorithm 1. Kernel-Based Reliability Function Estimation.

Require: The sorted list of times T, the training set $\mathcal{X}$, the set of fault or censoring indications Δ, a test sample $\boldsymbol{x}^*$, and a kernel function K
Require: auxiliary variables weight list M, reliability probabilities list S
1: $n \leftarrow |T|$
2: $S \leftarrow \text{zeros}(n)$
3: $M \leftarrow [\,]$
4: **for** $i = 0$ to $n - 1$ **do**
5: $\quad \mathcal{R}_i \leftarrow \{j \mid T[j] \geq T[i]\}$
6: $\quad \mathcal{V} \leftarrow \{\mathcal{X}[j] \mid j \in \mathcal{R}_i\}$
7: $\quad \mathcal{Y} \leftarrow \{\Delta[j] \mid j \in \mathcal{R}_i\}$
8: $\quad w_i \leftarrow K(\boldsymbol{x}^*, \mathcal{V}, \mathcal{Y})$
9: $\quad M \leftarrow M \parallel w_i$
10: **end for**
11: **for** $i = 0$ to $n - 1$ **do**
12: $\quad \mathcal{R}_i \leftarrow \{j \mid T[j] \geq T[i]\}$
13: $\quad \mathcal{P} \leftarrow \{M[j] \mid j \in \mathcal{R}_i\}$
14: $\quad \psi_i \leftarrow \frac{\mathcal{P}[i] \cdot \Delta[i]}{\sum_{k=i}^{n-1} \mathcal{P}[k] + \epsilon}$
15: $\quad$ **if** $i = 0$ **then**
16: $\quad\quad S_i \leftarrow 1 - \psi_i$
17: $\quad$ **else**
18: $\quad\quad S_i \leftarrow S_{i-1} \cdot (1 - \psi_i)$
19: $\quad$ **end if**
20: **end for**
21: **return** S

train PL-kNN to compute the weights for the test set $\boldsymbol{x}^\star$. This process is defined as follows:

$$K(\boldsymbol{x}^*, \mathcal{V}, \mathcal{Y}) = \frac{\sum_{\forall \boldsymbol{z} \in \mathcal{T} \subseteq \mathcal{V} \wedge \lambda(\boldsymbol{z}) = \omega_i} D_M(\boldsymbol{z}_j, \boldsymbol{x}^\star)^{-1} * W(\boldsymbol{z}_j, \boldsymbol{c}_i)}{\mid \mathcal{T} \mid}, \forall i \in \mathcal{Y}, \tag{2}$$

where $\mathcal{T} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m\}$ is the set of the nearest neighbors of $\boldsymbol{x}^\star$ derived from the training samples $\mathcal{V}$, $\lambda(\boldsymbol{z})$ represents the true label of the training sample $\boldsymbol{z}$, $D_M(\boldsymbol{z}_j, \boldsymbol{s})$ is the Manhattan distance between the test sample $\boldsymbol{x}^\star$ and its j-th nearest neighbor $\boldsymbol{z}_j$, $W(\boldsymbol{z}_j, \boldsymbol{c}_i)$ is a weight represented by the inverse distance between the j-th nearest neighbor sample $\boldsymbol{z}_j$ and its centroid $\boldsymbol{c}_i$ of the class i, and $\mathcal{Y} = \{\omega_1, \omega_2, \omega_3, \ldots, \omega_n\}$ is the set of classes from the dataset. Therefore, Eq. 2 computes the average of the weights from the nearest neighbors $\mathcal{T}$ of $\boldsymbol{x}^\star$. This weight is then appended to the M list (Line 9).

The second loop iterates through each event time $T[i]$ once more to estimate the reliability function (Line 11). The set of at-risk observations is identified (Line 12) along with the corresponding weights (Line 13). A weighted sum of the at-risk observations is computed using the kernel weights (Line 14). The hazard function for each time point i is then calculated as follows:

$$\psi_i = \frac{\mathcal{P}[i] \cdot \Delta_i}{\sum_{k=i}^{n-1} \mathcal{P}[k] + \epsilon}, \tag{3}$$

ensuring numerical stability with a small constant $\epsilon = 10^{-10}$ in the denominator. This equation pertains to the normalized kernel function $K(\cdot)$ as described in

Eq. 1. The reliability probability at the first time step is set as $S_0 = 1 - \psi_0$. For subsequent time steps, the survival probability is computed recursively as follows:

$$S_i = S_{i-1}\dot{(}1 - \psi_i). \tag{4}$$

This recursive approach follows the same principle of Kaplan-Meier estimation but incorporates kernel-based weighting to adapt to covariate information. Finally, the computed reliability function values represented by S are returned as the algorithm's output. It estimates the survival probability over time while accounting for covariate effects through kernel smoothing.

4 Methodology

Dataset: A dataset related to the evaluation of DHSVs has been employed during the study. It comprises 576 samples collected from oil wells and was split into 394 samples from training and 182 samples for testing to ensure the learning of the survival function. Each data sample is defined by a set of covariates, which are described in Table 1.

Table 1. Description of the covariates that represent the data samples.

Covariate	Description
Water Column (WC)	The water column depth at which the DHSV is installed in the well.
Manufacturer	The company responsible for manufacturing the DHSV.
Censoring indication	A binary event variable indicating whether the sample is censored (event = 0) or failed (event = 1).
Time to event	The recorded time until either a failure occurred or the observation was censored.

Experimental Setup. The operating depth and manufacturer were used as covariates in the predictive model. To ensure proper preprocessing, we standardized the operating depth using z-score normalization to prevent scale-related biases, especially for distance-based algorithms like k-NN. As a categorical variable, the manufacturer was encoded using one-hot encoding to convert it into a binary vector. Finally, the preprocessed features were concatenated as inputs for the model.

To evaluate the performance of our approach, we compared it against three baseline models: the Cox Proportional Hazards Model (Cox PHM), the Random Survival Forest (RSF), and the Beran Estimator with Neural Kernels (BENK). While the former two models serve as widely employed benchmarks in survival analysis, the primary focus of our comparison is the BENK estimator due to its extensive hyperparameter setup in modeling complex survival patterns. The BENK model primarily comprises a set of neural networks that receive a pair of samples' covariates. BENK allows two hyperparameters for sample subset

selection, namely a portion of the whole dataset (n) to compose the input for the neural networks and random selection repetition of this subset selection (N). Given the extensive hyperparameter tuning required for BENK, we tested 10 randomly selected values for N and n, thoroughly evaluating their behavior on the survival function generation. We adapted the original BENK code available at GitHub[2].

Evaluation Measures. To assess the models' performance, three evaluation metrics have been employed: Integrated Brier Score (IBS) [2], focused on capturing the model's discriminant capabilities in terms of accuracy, Concordance Index (C-Index) [5] and its variant approach with Inverse Probability of Censoring Weights (C-Index IPCW) [16], both employed for measuring the model's ranking ability. Those metrics are well-known and widely used in the literature for evaluating reliability analysis-based models [3,6].

5 Experiments and Discussion

This section presents the quantitative results estimated by IBS, C-Index and C-Index IPCW. In addition, we present the reliability curves to complement the qualitative analysis. The curve is expected to behave monotonically decreasing since the probability of the equipment continuing to function tends to reduce as time progresses, which is sometimes referred to as the probability of survival. In other words, it reflects the estimated probability that a typical equipment from the population will survive beyond a certain time. At the maximum time, the survival probability reaches its lowest value and may approach zero, depending on the operating conditions.

Following the method proposed by Kirpichenko et al. [12], we setup the values for the hyperparameters N and n of the BENK model. Such values are derived from the entire set of c training observations as follows:

$$\mathcal{C} = \{(\mathbf{x}_i, \delta_i, t_i),\ i = 1, \ldots, c\}\,.$$

Each i in the range $\{1, \ldots, c\}$ is used to generate N subsets $\mathcal{C}_i^{(r)}$, with $r = 1, \ldots, N$, containing n examples chosen randomly from $\mathcal{C} \setminus (\mathbf{x}_i, \delta_i, t_i)$, which are indexed by the index set $\mathcal{J}^{(r)}$. Specifically, the subsets $\mathcal{C}_i^{(r)}$ are expressed as follows:

$$\mathcal{C}_i^{(r)} = \left\{ \left(\mathbf{x}_k^{(r)}, \delta_k^{(r)}, t_k^{(r)} \right),\ k \in \mathcal{J}^{(r)} \right\}, \quad r = 1, \ldots, N.$$

These subsets are used for training the neural networks of the BENK method. Further details can be found in the study of Kirpichenko et al. [12]. Notice that the value for N is expected to start at 1 since we need at least one subset of random samples for $\mathcal{C}_i^{(r)}$.

5.1 Quantitative Analysis

Table 2 presents the results obtained from each baseline used for comparison and the same evaluation measures provided by the proposed method (ours).

[2] https://github.com/Stasychbr/BENK.

Table 2. Evaluation measures computed from the test set samples.

Estimator	IBS	C-index	C-index IPCW
Cox PHM	0.345	**0.650**	0.406
RSF	0.370	0.493	0.418
BENK N=1/n=28	**0.241**	0.465	0.512
BENK N=9/n=69	0.317	0.404	0.523
BENK N=10/n=49	0.316	0.514	0.466
BENK N=13/n=55	0.316	0.521	0.502
BENK N=14/n=85	0.322	0.511	0.500
BENK N=20/n=9	0.352	0.325	0.502
BENK N=25/n=65	0.319	0.476	0.460
BENK N=26/n=30	0.322	0.467	0.493
BENK N=35/n=54	0.323	0.447	0.435
BENK N=39/n=74	0.352	0.573	0.412
Ours	0.323	0.566	**0.525**

The best results—lowest IBS and highest C-index—are highlighted in bold. The reported metrics illustrate the performance of both baseline and proposed models on the DHSV dataset. Surprisingly, Random Survival Forest (RSF) showed weaker performance based on the C-index, while the Cox Proportional Hazards Model (Cox PHM) performed better, followed closely by our approach. The high censoring rate in the DHSV dataset notably affects models like RSF. However, since the C-Index PCW accounts for censored data, our method outperforms the baselines considering this measure. IBS also supports this, where our model achieved 0.323, compared to 0.370 for RSF and 0.345 for Cox PHM.

The proposed method attained comparable or even superior results to various BENK setups. Notably, the BENK model with N=1 and n=28 attained the lowest IBS among all baselines. However, a lower IBS does not necessarily indicate a higher C-index, as most BENK setups showed lower IBS and C-index values than Beran PL-kNN. In terms of C-index, the proposed method outperformed all BENK setups except for N=39 and n=74, though the difference is minimal. For C-index IPCW, the proposed model achieved the highest value among all BENK configurations and baselines.

5.2 Qualitative Analysis

We qualitatively assessed the performance of our proposed model by comparing its estimated survival curves to those generated by the Kaplan-Meier (KM) estimator and other baseline models. To incorporate the effect of the Water Column (WC) covariate, we stratified the data into groups based on low, medium, and high WC values. We estimated the reliability curve using KM for each group and compared it to the curves generated by our model and the baselines. Figure 1

shows the reliability functions for Beran PL-kNN, Cox PHM, and RSF. The functions exhibit the expected monotonic decreasing behavior, regardless of the model under analysis. Under different conditions, as determined by the WC, the curve is expected to behave differently, exhibiting the steepest descent and lower survival probabilities at an early stage when the water column value is high, corroborating to the fact that higher WC induces stronger degradation on DHSV, reducing its reliability. This scenario occurs for Beran PL-kNN in the plots of manufacturer C and WC with medium and high levels. However, such behavior varies based on the combination of the two features.

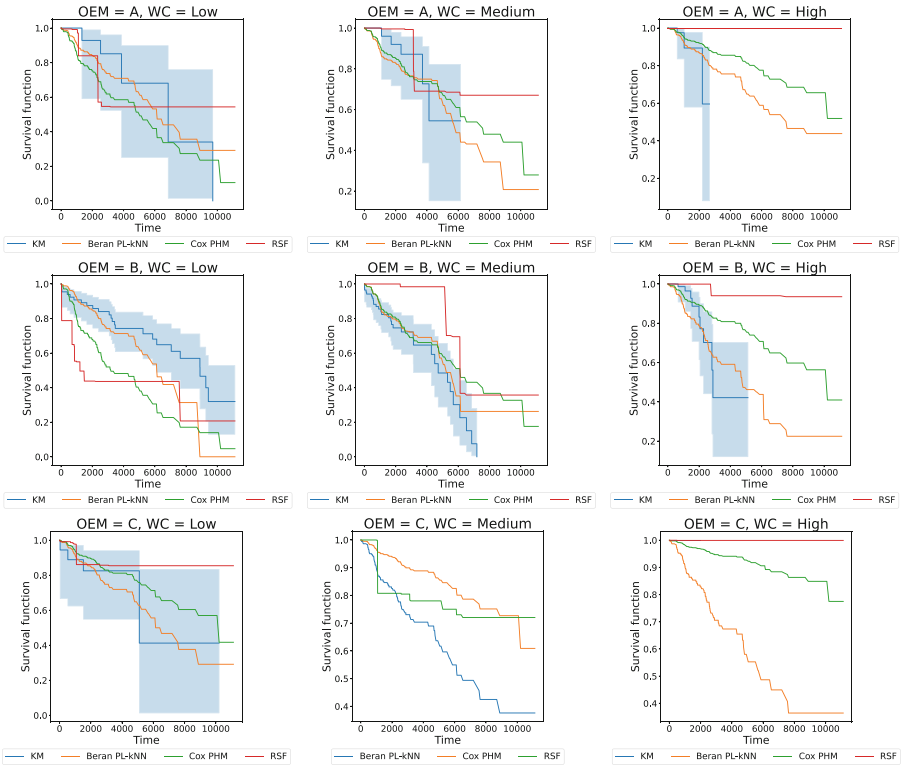


Fig. 1. Reliability functions for each combination of manufacturer and operational depth.

KM estimator curves can display plateaus of constant probabilities due to censored data, a phenomenon also seen in RSF, especially with small sample sizes and high censoring. This phenomenon is evident in groups with fewer failures, such as manufacturers A and C with high WC. In contrast, Cox PHM and Beran PL-kNN are less prone to prolonged constant survival periods. Beran PL-kNN outperforms KM, showing fewer such intervals; Beran PL-kNN closely

follows KM for low WC values, while Cox PHM underestimates survival for manufacturers A and B and overestimates it for C. In the medium WC range, Beran PL-kNN remains aligned with KM, whereas Cox PHM diverges, particularly for manufacturer C, at later time points. After 4,000 d, Beran PL-kNN increasingly differs from KM, reflecting a more realistic survival trend than Cox PHM. Remarkably, Beran PL-kNN remains stable even in the high WC group despite the reduced number of data points for certain manufacturers. Cox PHM appears to overestimate the survival curve relative to KM in this group, while the proposed approach continues to track the KM curves reasonably well. The case of manufacturer C with high WC is particularly noteworthy. The limited data availability distorts the survival curves, even for the KM estimator. However, Beran PL-kNN demonstrates significantly less distortion than other models, indicating that it is more robust and stable in the face of small sample sizes, at least from a qualitative perspective.

Figure 2 depicts the same reliability function curves estimated by the proposed approach but compares them to the BENK model under different setups. The BENK estimate results indicate its failure behavior is highly sensitive to the choice of initialization parameters. Qualitatively, it directly impacts the survival probabilities over time. On the other hand, Beran PL-kNN achieved similar results to those of many of the displayed BENK models. Notably, there is a lack of data observations when the water column is high, regardless of the selected manufacturer. Looking closely at the results concerning manufacturer A combined with WC High, such a scenario shows an increasing discrepancy between Beran PL-kNN and BENK estimates concerning survival probability over operational days. Approximately from 3,000 d, Beran PL-kNN shows higher survival estimates than BENK for all subsequent operational days. Beran PL-kNN's survival curve presents a gradual decline followed by a plateau, suggesting the model captures a combination of a relatively constant failure rate until 8,000 d of operation and rare failure events happened after that time for that specific covariates configuration.

Most of the survival curves estimated by Beran PL-KNN are similar to those of BENK and obtained under lower initialization requirements. From a qualitative standpoint, such results indicate the proposed approach presents less complexity while performing the survival prediction task with similar discriminative power to the BENK model. Finally, the results indicate that Beran PL-kNN tends to make more optimistic predictions than BENK, i.e., the former outputs higher probabilities of survival over the available lifetime than the latter.

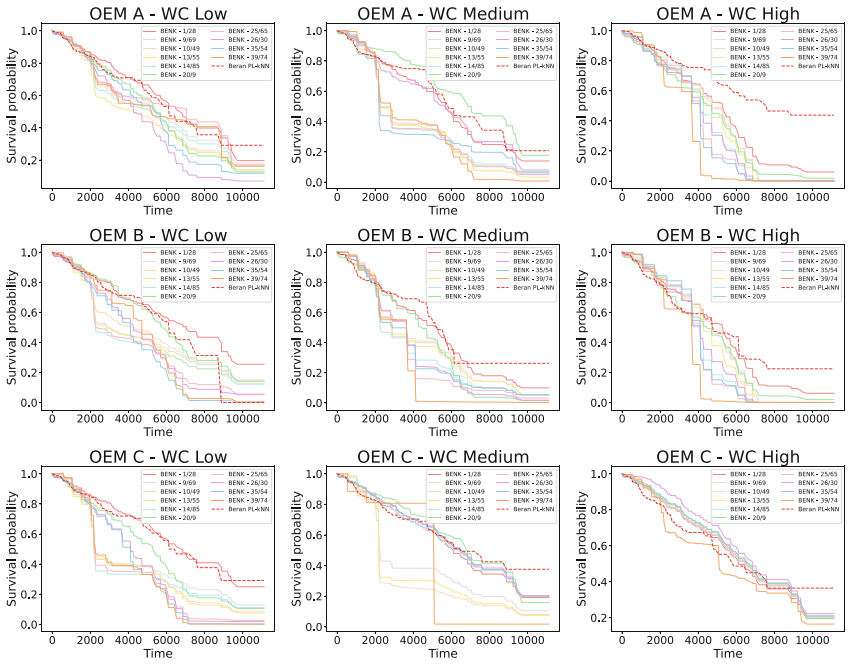


Fig. 2. Survival plots for each combination of manufacturer and operational depth.

6 Conclusions

This paper presented a reliability analysis approach that combines the Beran estimator with PL-kNN as the kernel function to promote nonparametric survival estimation and machine learning parameterless approaches. The proposed approach demonstrates the potential to improve reliability function estimation by adapting the problem as a classification task within PL-kNN. This perspective allows for more accurate weighting of training instances based on their relevance to the test sample, thus leading to a more robust survival analysis method. Future work will explore other kernel functions and a multi-kernel learning approach to improve robustness from multiple kernel functions. Moreover, we intend to explore additional datasets in future work to further validate the methodâĂŹs generalization capability beyond the current use case.

Acknowledgments. This study was partially financed by the São Paulo Research Foundation (FAPESP), Brazil, grant numbers 2013/07375-0, 2019/07665-4, 2023/14427-8, 2023/10823-6, 2023/03726-4, 2023/01374-3, and 2023/14354-0. The authors are also grateful to the Brazilian National Council for Scientific and Technological Development (CNPq) grant 308529/2021-9, to Unesp-IEPe-RC-#06/2023 PROPe grant, and to the Petrobrás Brazil grant #2017/00285-6 for their financial support.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Beran, R.: Nonparametric regression with randomly censored survival data. University of California, Berkeley, Tech. rep. (1981)
2. Graf, E., Schmoor, C., Sauerbrei, W., Schumacher, M.: Assessment and comparison of prognostic classification schemes for survival data. Stat. Med. **18**(17–18), 2529–2545 (1999)
3. Graf, E., Schmoor, C., Sauerbrei, W., Schumacher, M.: Assessment and comparison of prognostic classification schemes for survival data. Stat. Med. **18**(17–18), 2529–2545 (1999)
4. Harrell, Jr, F.E., Harrell, F.E.: Cox proportional Hazards Regression Model. Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis, pp. 475–519 (2015). https://doi.org/10.1007/978-1-4757-3462-1_19
5. Harrell, F.E., Jr., Lee, K.L., Mark, D.B.: Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. Stat. Med. **15**(4), 361–387 (1996). https://doi.org/10.1002/(SICI)1097-0258(19960229)15:4<361::AID-SIM168>3.0.CO;2-4
6. Harrell, F.E., Jr., Lee, K.L., Mark, D.B.: Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. Stat. Med. **15**(4), 361–387 (1996)
7. Huang, Y., Li, J., Li, M., Aparasu, R.R.: Application of machine learning in predicting survival outcomes involving real-world data: a scoping review. BMC Med. Res. Methodol. **23**(1), 268 (2023). https://doi.org/10.1186/s12874-023-02078-1
8. Ishwaran, H., Kogalur, U.B., Blackstone, E.H., Lauer, M.S.: Random survival forests. Ann. Appl. Stat. **2**(3), 841–860 (2008). https://doi.org/10.1214/08-AOAS169
9. Jodas, D.S., Passos, L.A., Adeel, A., Papa, J.P.: PL-kNN: a Python-based implementation of a parameterless k-Nearest Neighbors classifier. Software Impacts **15**, 100459 (2023). https://doi.org/10.1016/j.simpa.2022.100459
10. Jodas, D.S., Passos, L.A., Adeel, A., Papa, J.P.: PL-k NN: a parameterless nearest neighbors classifier. In: 2022 29th International Conference on Systems, Signals and Image Processing (IWSSIP), vol. CFP2255E-ART, pp. 1–4 (2022). https://doi.org/10.1109/IWSSIP55020.2022.9854445
11. Kaplan, E.L., Meier, P.: Nonparametric estimation from incomplete observations. J. Am. Stat. Assoc. **53**(282), 457–481 (1958). https://doi.org/10.2307/2281868
12. Kirpichenko, S., Utkin, L., Konstantinov, A., Muliukha, V.: BENK: the Beran estimator with neural kernels for estimating the heterogeneous treatment effect. Algorithms **17**(1), 40 (2024). https://doi.org/10.3390/a17010040
13. Limongi, J.E., Santos, K.A.R., Perissato, I.L., Pinto, R.d.M.C., Mendonça, T.M.d.S., Rinaldi, A.E.M.: Survival analysis of chagas disease patients, beneficiaries of social security and social assistance in Brazil, 1942–2016. Revista Brasileira de Epidemiologia **27**, e240020 (2024). https://doi.org/10.1590/1980-549720240020
14. López-Cheda, A., Cao, R., Jácome, M.A., Van Keilegom, I.: Nonparametric incidence estimation and bootstrap bandwidth selection in mixture cure models. Comput. Stat. Data Anal. **105**, 144–165 (2017). https://doi.org/10.1016/j.csda.2016.08.002

15. Papathanasiou, D., Demertzis, K., Tziritas, N.: Machine failure prediction using survival analysis. Fut. Int. **15**(5), 153 (2023). https://doi.org/10.3390/fi15050153
16. Uno, H., Cai, T., Pencina, M.J., D'Agostino, R.B., Wei, L.J.: On the C-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. Stat. Med. **30**(10), 1105–1117 (2011). https://doi.org/10.1002/sim.4154

Conformalized Causal Learning for Uncertainty-Aware Mineral Prospectivity Mapping

Evelyn Jessica Jaya[1], Qinying Gu[2(✉)], Xinbing Wang[1], and Nanyang Ye[1(✉)]

[1] Shanghai Jiao Tong University, Shanghai, China
ynylincoln@sjtu.edu.cn
[2] Shanghai Artificial Intelligence Laboratory, Shanghai, China
guqinying@pjlab.org.cn

Abstract. Uncertainty quantification and causal reasoning remain key challenges in deep learning for geospatial optimization. Traditional models for Mineral Prospectivity Mapping (MPM) often lack interpretability, struggle with out-of-distribution (OOD) generalization, and yield unreliable uncertainty estimates, leading to inefficient resource allocation and false positives. To address these limitations, this study introduces Conformalized Causal Learning (CCL), a novel deep learning framework that integrates causal feature selection, conformal prediction, and attention-based neural architectures to enhance robustness and trustworthiness. Unlike conventional models that prioritize predictive accuracy alone, CCL incorporates calibrated uncertainty estimation to support confidence-aware decision-making. Experimental evaluations across four diverse geological datasets show that CCL reduces predictive uncertainty by up to 90% while maintaining competitive AUROC scores. The framework consistently outperforms baseline methods in both robustness and interpretability. CCL offers a generalizable solution for high-stakes geospatial applications, including mineral exploration, environmental monitoring, climate modeling, and remote sensing—advancing the development of trustworthy, uncertainty-aware AI.

Keywords: Geospatial AI · Trustworthy AI · Uncertainty-Aware Deep Learning · Causal Learning

1 Introduction

Mineral Prospectivity Mapping (MPM) is a critical geospatial AI problem that enables the discovery of mineral resources essential for modern industries. The goal of MPM is to identify high-prospectivity regions while minimizing economic, environmental, and social costs. This task relies on integrating diverse geological, geophysical, and geochemical datasets into predictive models. With increasing demand for critical minerals in renewable energy and high-tech applications, the development of scalable, data-driven, and interpretable deep learning models for mineral exploration has become imperative.

Despite advancements, MPM approaches face challenges with high-dimensional, non-stationary, and spatially correlated datasets, leading to

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 115–127, 2026.
https://doi.org/10.1007/978-3-032-04555-3_10

overfitting and poor generalization in unexplored regions [11]. Additionally, the lack of uncertainty quantification raises false positive rates and inefficient resource allocation.

MPM methodologies have evolved from expert-driven to data-driven approaches. Early knowledge-driven methods relied on expert interpretation of geological maps, but their subjectivity and lack of scalability hindered performance in data-scarce regions [7]. Statistical techniques such as weights-of-evidence (WofE) and logistic regression (LR) introduced probabilistic modeling [6], but these approaches assumed independent and identically distributed (IID) data, which is rarely valid in real-world geoscience applications [10].

Machine learning (ML) and deep learning (DL) have improved MPM by enabling non-linear pattern recognition. Methods like Support Vector Machines (SVM) [21], Random Forests (RF) [19], and Artificial Neural Networks (ANNs) demonstrated better predictive capabilities. Deep learning architectures such as CNNs [9] and RNNs capture spatial and temporal dependencies. However, these models are often black boxes, limiting interpretability, and are sensitive to data biases, reducing their reliability in out-of-distribution (OOD) scenarios [1]. Uncertainty quantification has emerged as key for improving model reliability, but current probabilistic GIS models and transfer learning frameworks lack systematic approaches for calibrated uncertainty [14]. Optimization techniques like ensemble learning and genetic algorithms (GA) improve feature selection but fail to capture causal geological relationships.

To address these issues, this study introduces Conformalized Causal Learning (CCL), a novel hybrid deep learning framework integrating causal feature selection with conformal prediction. CCL enhances predictive confidence with well-calibrated uncertainty intervals, improving generalization and interpretability by focusing on geologically meaningful features. Unlike conventional deep learning models, CCL integrates uncertainty-aware learning and causal reasoning, ensuring confidence-aware decision-making. This approach advances research in trustworthy AI, geospatial deep learning, and uncertainty-aware scientific discovery.

2 Proposed Method

MPM identifies regions with high mineralization potential by integrating geological, geophysical, and geochemical datasets. The goal is to learn a predictive function $f : \mathbf{X} \rightarrow \mathbf{Y}$, where $\mathbf{X}$ denotes feature vectors and $\mathbf{Y} \in \{0, 1\}$ indicates the presence or absence of mineralization. This task is framed as minimizing expected binary cross-entropy loss over the data distribution $\mathcal{D}$:

$$\arg\min_{f} \mathbb{E}_{(\mathbf{X},\mathbf{Y})\sim\mathcal{D}}[\mathcal{L}(f(\mathbf{X}), \mathbf{Y})] \tag{1}$$

Conventional MPM models struggle with high-dimensional, non-stationary, and spatially correlated data, often leading to overfitting and limited generalization. These limitations necessitate the integration of causal reasoning and uncertainty quantification to improve reliability (Fig. 1).

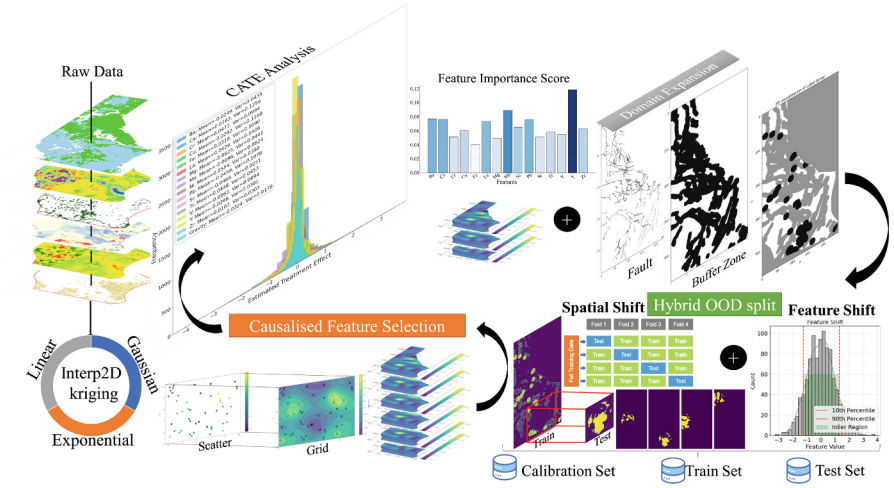


Fig. 1. Causalized Data Preprocessing workflow integrates adaptive interpolation, causal feature selection, sample balancing, and OOD-specific data splitting

To address these issues, this study proposes the Conformalized Causal Learning (CCL) framework, a modular architecture comprising two components:

- **Causalized Data Preprocessing Module:** Handles geological heterogeneity, applies adaptive interpolation, and selects causally linked features to reduce spurious correlations.
- **Quantified Prediction Module:** Integrates spatial, temporal, and causal learning with uncertainty quantification to generate confidence-calibrated predictions.

The overall optimization objective of CCL is:

$$\arg\min_{f} \mathbb{E}[\mathcal{L}(f(\mathbf{X}_c), \mathbf{Y})] + \lambda \mathcal{U}(f(\mathbf{X}_c)), \tag{2}$$

where $\mathbf{X}_c$ is the selected causal feature set, $\mathcal{U}$ is the uncertainty penalty, and λ balances accuracy and uncertainty.

2.1 Causalized Data Preprocessing Module

Adaptive Interpolation. MPM datasets often contain missing values due to inaccessible terrain or inconsistent historical records. Adaptive interpolation techniques such as Pykrige-linear, Pykrige-Gaussian, and Pykrige-Exponential are evaluated, with the optimal method selected based on mean squared error (MSE). Rigorous QA/QC measures are applied to minimize error propagation.

Causal Feature Selection. To reduce spurious associations [20], CCL selects features $\mathbf{X}_c \subseteq \mathbf{X}$ with true causal influence on mineralization, formalized via do-calculus:

$$P(\mathbf{Y} \mid \mathbf{X}_c) = P(\mathbf{Y} \mid do(\mathbf{X}_c)) \tag{3}$$

Causal Forests [16] estimate the Conditional Average Treatment Effect (CATE):

$$\tau(x) = \mathbb{E}[Y(1) - Y(0) \mid X = x], \tag{4}$$

Each feature is evaluated as a treatment by estimating its marginal effect on mineralization. Features with treatment effects exceeding a threshold (mean + one standard deviation) are retained.

Alternative causal selection methods were also tested. The PC algorithm, a constraint-based discovery method, failed to identify useful features across datasets, likely due to its sensitivity to multicollinearity and high-dimensional noise [13]. SHAP and permutation-based methods provide predictive importance but lack formal causal guarantees and depend on expert interpretation [18]. In contrast, Causal Forests yield direct, continuous treatment-effect estimates, are robust to heterogeneity, and scale effectively for geospatial modeling—making them well-suited for MPM.

Handling Sample Imbalance. MPM datasets often exhibit class imbalance, with sparse positive samples (known deposits) and numerous unmeasured regions [22]. This challenge is addressed through:

- Geological Masking: Removes non-prospective areas (e.g., distant from faults) based on geological indicators.
- Spatial Expansion: Augments positives within a spatial buffer:

$$R = \{s' \mid \|s - s'\| \leq r\}, \tag{5}$$

where $r = 2{-}3\,\text{km}$ captures local geological context.

OOD Data Splitting. To simulate real-world exploration, a hybrid OOD split combines spatial and feature-based shifts. K-Means is applied to spatial coordinates (x_i, y_i), generating clusters $c_i \in \{1, \ldots, K\}$. A subset $\mathcal{C}_{\text{train}}$ is used for training, while $\mathcal{C}_{\text{test}} = \{1, \ldots, K\} \setminus \mathcal{C}_{\text{train}}$ is held out for testing, ensuring spatial disjointness.

Within training clusters, inlier samples are filtered using interdecile constraints on selected causal features $\mathcal{F}_c$:

$$\text{Inlier}(f_j) = \{x \mid Q_{10}(f_j) \leq x \leq Q_{90}(f_j)\}\,, \tag{6}$$

where Q_{10} and Q_{90} are the 10th and 90th percentiles, respectively. Samples satisfying $x_i^{(f_j)} \in \text{Inlier}(f_j)$ for all $f_j \in \mathcal{F}_c$ are retained. A random 25% subset is reserved for validation. The OOD test set includes spatially and geochemically novel samples:

$$\mathcal{T}_{\text{OOD}} = \{x_i \mid c_i \in \mathcal{C}_{\text{test}}\} \cup \left\{x_i \mid \exists f_j \in \mathcal{F}_c,\ x_i^{(f_j)} \notin \text{Inlier}(f_j)\right\} \tag{7}$$

This setup ensures evaluation under both geographic and geochemical distribution shifts.

2.2 Quantified Prediction Module

This module integrates spatial, temporal, and causal learning to generate well-calibrated confidence estimates. Local geological features are extracted using convolutional layers (CNNs) to capture spatial dependencies. These features are then processed by a bidirectional Long Short-Term Memory (BiLSTM) network to model temporal relationships in geological sequences. An attention mechanism is applied to prioritize features that are strongly indicative of mineralization, enhancing both interpretability and model reliability. The final architecture layer quantify predictive uncertainty by incorporating conformal prediction [15], which transforms probabilistic outputs into statistically valid confidence intervals (Fig. 2). This is especially valuable in geospatial contexts where traditional uncertainty quantification methods such as those based on Bayesian model variance are limited by their reliance on the IID assumption.

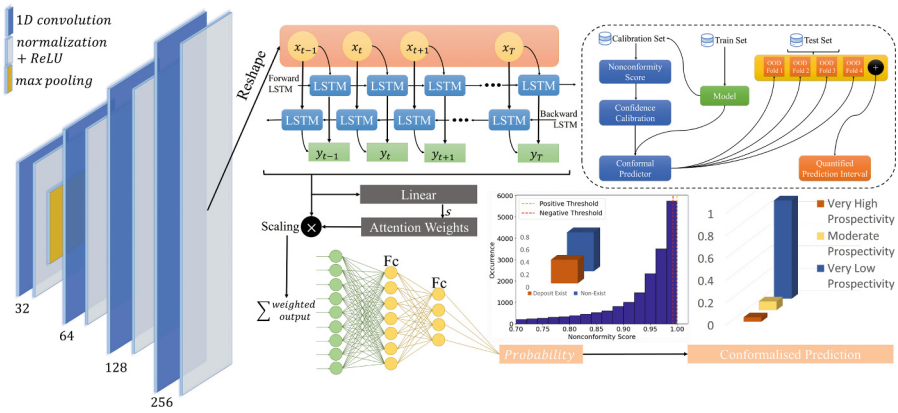


Fig. 2. Schematic workflow of Quantified Prediction Module

Conformal prediction addresses this by constructing adaptive, distribution-free prediction intervals that remain valid under weak assumptions. For a given input x_{n+1}, the prediction interval is defined as:

$$\Pi(x_{n+1}) = \{y : p(y \mid x_{n+1}) > \alpha\}, \tag{8}$$

where $p(y \mid x)$ is the p-value for outcome y, and α is the significance level. Nonconformity scores $A(x, y)$ are calculated to quantify the alignment of a prediction with the training distribution:

$$A(x, y) = |f(x) - y|, \tag{9}$$

where $f(x)$ is the model's predicted probability for x. The significance level α^* is optimized to minimize misclassification while maintaining a 95% confidence level:

$$\alpha^* = \min \left\{ \alpha : \frac{|\{i : p(y_i \mid x_i) \leq \alpha\}|}{N} \leq 0.05 \right\}, \tag{10}$$

where N represents the total number of instances.

To support actionable exploration decisions, predictions are categorized into three classes based on the derived confidence intervals: Class 0 represents no prospectivity, Class 0.5 indicates uncertain prospectivity, and Class 1 corresponds to high prospectivity. This classification framework enables nuanced interpretation of the model's predictions, effectively balancing predictive accuracy with quantified uncertainty. By providing clear distinctions among confidence levels, this approach supports robust decision-making in high-stakes exploration scenarios where reliability is critical.

3 Experimental Setup

3.1 Datasets

Four diverse geological datasets were used to evaluate the proposed CCL framework, each representing distinct mineralization styles and regional geology. The Nefb Dataset [2] covers the Northeast Fairbanks and Circle regions, targeting gold and base metals within greenschist-facies terranes, supported by geophysical and geochemical data. The Idaho Dataset [8,17] represents the Coeur d'Alene district, notable for silver, lead, and zinc deposits, with lithological, structural, and geochemical layers. The Tok Dataset [2] focuses on placer gold deposits shaped by glacial and alluvial processes in the Tok and Ladue regions. The Washington Dataset [17] encompasses volcanic-hosted systems in southern Washington, emphasizing epithermal gold and porphyry copper occurrences. These datasets present shared challenges, including severe class imbalance (positive samples <1%) and complex, heterogeneous geological features (Fig. 3).

3.2 Evaluation Protocol

CCL was benchmarked against several state-of-the-art models commonly used in MPM, including Support Vector Machines (SVM) [21], Random Forests (RF) [19], Graph Convolutional Networks (GCN) [23], and Convolutional Neural Networks (CNN) [9]. To broaden architectural diversity, the baseline set also included Long Short-Term Memory (LSTM) networks and a standalone

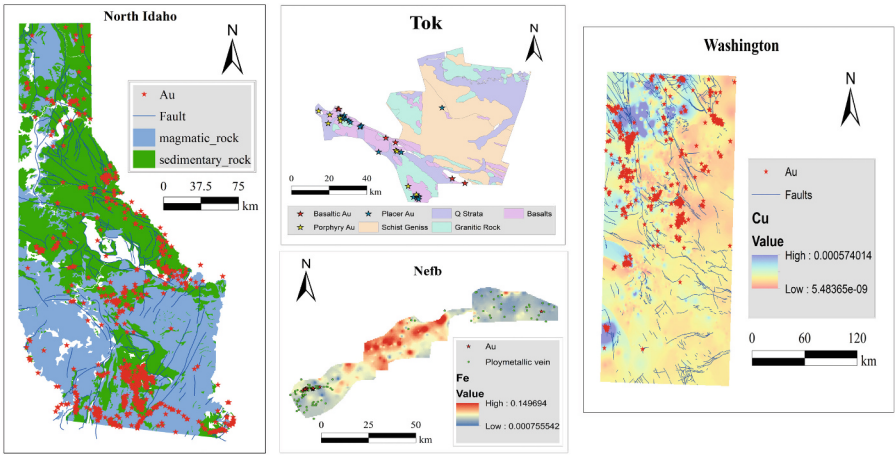


Fig. 3. Visualization of datasets. (a) Idaho, (b) Tok, (c) Nefb, (d) Washington. Each sub-graph presents mineral sites and data boundaries, with x and y axes representing transformed longitude and latitude coordinates

attention-based neural network (Att). Classical models such as Logistic Regression (LR) and Extra Trees (ET) were evaluated for comparison. A variant of the proposed model without conformal prediction (CCL-NoCP) was included to isolate the contribution of the uncertainty quantification module. All models were trained and evaluated using consistent data splits, preprocessing pipelines, and hyperparameter optimization via grid search. Performance was assessed using the AUROC, F1 score, and Predictive Uncertainty (PU) to quantify model confidence and calibration under OOD conditions.

4 Result Analysis

4.1 Performance Analysis

In the IID setting, CCL achieved competitive AUROC scores and produced localized, geologically coherent predictions, aligning well with known mineralization zones (Fig. 4(c)). Compared to other baselines, CCL demonstrated a stronger balance between classification performance and spatial precision. While CCL-NoCP attained slightly higher AUROC on the Nefb dataset, it lacked the calibrated uncertainty estimates of CCL, reinforcing the importance of confidence-aware predictions in high-stakes exploration scenarios [3].

OOD scenarios present a more challenging evaluation, where training and testing datasets are partitioned using a hybrid split strategy that combines spatial disjointness (via geospatial clustering) with feature-based novelty (using interdecile filtering on selected causal attributes). This approach introduces both geographic and geochemical shifts. In these settings, CCL consistently outperformed other models, achieving higher AUROC and lower predictive uncertainty

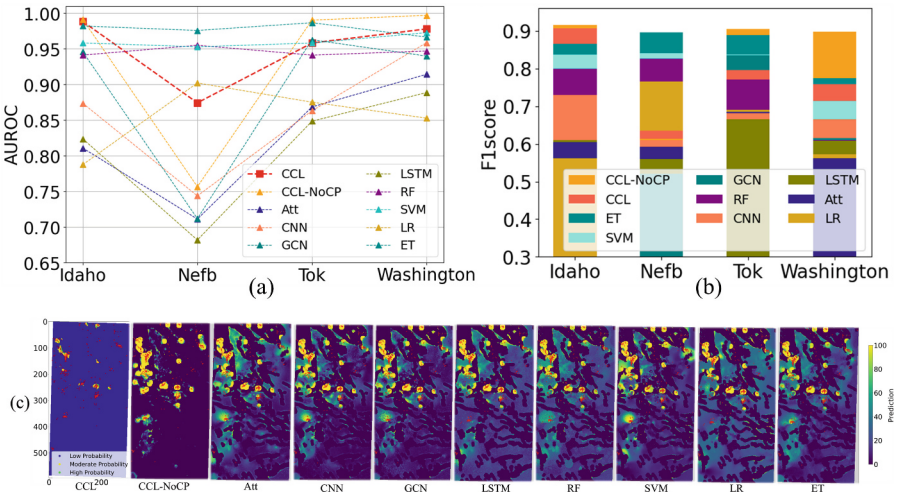


Fig. 4. Results comparison in the IID setting: (a) AUROC scores; (b) F1 scores; (c) Spatial probability maps from the Washington dataset where warmer colors indicate higher predicted probabilities of mineralization

(PU) across datasets (Fig. 5, Table 1). While all models experienced performance drops in OOD settings, CCL showed greater adaptability, with causal feature selection and conformal prediction enhancing its generalization. CCL localized high-confidence predictions to geologically significant areas and flagged uncertain regions for further investigation, making it ideal for high-cost mineral exploration. In contrast, other baselines displayed higher PU and lower AUROC, especially in more complex datasets like Tok and Nefb, highlighting CCL's superior robustness in OOD scenarios.

Table 1. AUROC and PU Results in OOD scenarios

Dataset	Tok		Washington		Idaho		Nefb	
Metrics	AUROC	PU	AUROC	PU	AUROC	PU	AUROC	PU
CCL	**0.7924**	**0.009**	**0.9062**	**0.014**	**0.9213**	**0.028**	**0.7810**	**0.032**
CCL-NoCP	0.7765	0.402	0.8910	0.372	0.9003	0.3629	0.6872	0.186
Att	0.6016	0.1371	0.7335	0.2107	0.6631	0.1353	0.3939	0.1851
CNN	0.6708	0.222	0.6989	0.244	0.6515	0.168	0.6944	0.072
GCN	0.6533	0.232	0.7283	0.138	0.7222	0.164	0.7321	0.178
LSTM	0.5896	0.2607	0.6829	0.2520	0.5266	0.2731	0.5676	0.0893
RF	0.6105	0.312	0.6939	0.252	0.6511	0.198	0.692	0.272
SVM	0.6059	0.354	0.6709	0.366	0.5723	0.312	0.6019	0.3964
LR	0.6538	0.3348	0.7526	0.2723	0.6474	0.2198	0.7034	0.3409
ET	0.5923	0.2551	0.6887	0.2643	0.4251	0.1836	0.6542	0.3139

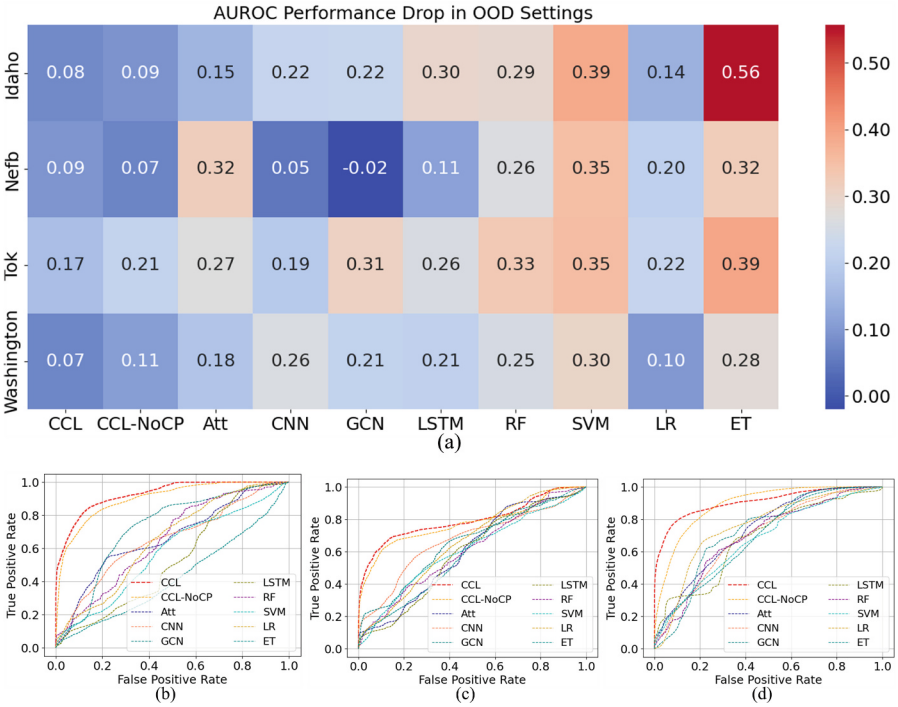


Fig. 5. Results in the OOD setting: (a) Heatmap of AUROC performance drops from IID to OOD settings, where higher values indicate greater degradation; (b-d) ROC curves for Idaho, Tok, and Washington datasets

4.2 Interpretable Analysis

The integration of causal learning within the CCL framework enhances interpretability by identifying features that are both statistically robust and geologically meaningful. Using CATE analysis, CCL isolates the causal influence of input features on mineral prospectivity, ensuring predictions align with well-established geoscientific principles [24]. Prominent causal features identified across datasets include fault zones, gravity and magnetic anomalies, and geochemical elements such as arsenic (As), copper (Cu), and nickel (Ni). These are key indicators of metallogenic processes: As contributes to arsenian pyrite formation in hydrothermal systems [5], while Cu and Ni are linked to porphyry and ultramafic-hosted deposits [12]. Structural features such as fault zones also serve as critical conduits for hydrothermal fluid flow [4].

A particularly insightful result, shown in Fig. 6(a), is the higher predictive importance of Calcium (Ca) over Gold (Au) in the Idaho dataset. While counterintuitive, this is consistent with geochemical understanding: Au often occurs in trace amounts and exhibits the nugget effect, making it difficult to detect directly. By contrast, Ca-rich carbonate host rocks are more geochemically con-

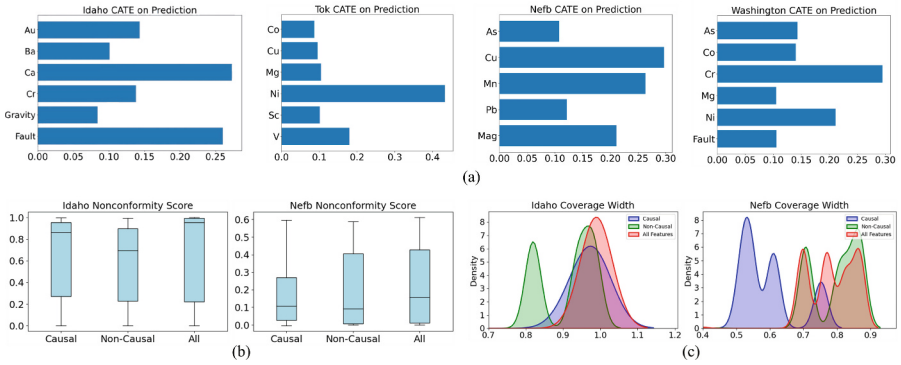


Fig. 6. (a) CATE analysis of the influence of selected features on gold (Au) prospectivity; (b) Boxplots of nonconformity scores for causal, non-causal, and all features in Idaho and Nefb datasets; (c) Kernel density estimates of coverage width, where narrower intervals for causal features indicate higher predictive precision

tinuous and serve as reliable proxies for hydrothermal alteration zones. Similar patterns are seen in the Tok and Washington datasets, where transition metals and structural features dominate CATE scores, reinforcing domain-valid patterns.

CCL also assesses prediction reliability through nonconformity scores, which measure feature alignment with the model's learned distribution. As illustrated in Fig. 6(b), causal features typically show narrower distributions, indicating more stable predictions. In the Nefb dataset, causal features have slightly higher median scores but reduced spread, suggesting stability despite geological variance. In contrast, the Idaho dataset exhibits both higher median and wider distributions for causal features, reflecting their dominant and variable role in mineralization. Importantly, lower spread—rather than median alone—signals greater predictive consistency. Figure 6(c) reinforces this, showing narrower coverage widths for causal features under conformal prediction, emphasizing CCL's strength in delivering precise, uncertainty-aware outputs suitable for high-stakes mineral exploration.

4.3 Trade-Off Between AUROC and Uncertainty

The results from both IID and OOD scenarios highlight a critical trade-off between AUROC and predictive uncertainty. While CCL-NoCP occasionally achieved higher AUROC scores than CCL, it incurred significantly higher uncertainty, resulting in less reliable predictions. This trade-off underscores the risks of models that optimize purely for accuracy without accounting for uncertainty, particularly in high-stakes applications like MPM. The integration of conformal prediction in CCL addresses this by providing confidence-quantified outputs that are spatially precise and geologically meaningful. As shown in Fig. 7, CCL effectively isolates sparse positive samples and produces focused, actionable predic-

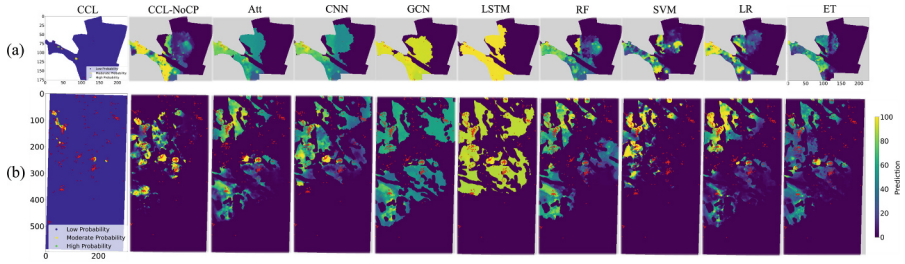


Fig. 7. Spatial prediction map in OOD scenario: (a) Tok, (b) Washington

tions. In contrast, other models yield broader, more diffuse outputs with greater spatial noise, increasing the likelihood of false positives and reducing practical utility. These differences highlight CCL's strength in balancing accuracy with uncertainty awareness, resulting in trustworthy predictions suited for real-world exploration.

5 Conclusion

This study introduced the Conformalized Causal Learning (CCL) framework, a novel approach to MPM that combines causal learning with conformal prediction to enhance uncertainty quantification, interpretability, and OOD generalization. CCL balances predictive accuracy and uncertainty, consistently outperforming baseline models by providing localized, geologically coherent predictions and reducing false positives. Causal feature selection mitigates spurious correlations, while conformal prediction ensures well-calibrated confidence intervals, supporting risk-aware decision-making. With its modular design, CCL is adaptable to broader geospatial and AI-driven domains, such as environmental monitoring and disaster risk assessment. Future work will expand its capabilities with multi-objective optimization, integration of hyperspectral imagery, and scalable computing, positioning CCL as a versatile tool for complex geospatial challenges.

Acknowledgments. We thank the anonymous reviewers for providing insightful comments and feedback. This study was funded by National Natural Science Foundation of China under Grant (No.62106139).

References

1. An, Z., Zhao, Y., Zhang, Y.: Mineral exploration and the green transition: opportunities and challenges for the mining industry. Resour. Policy **86**, 104263 (2023)
2. D.G.G.S. Staff: Alaska merged geophysical data grids. Alaska Division of Geological & Geophysical Surveys (2016)

3. Ding, Y., Liu, J., Xiong, J., Shi, Y.: Revisiting the evaluation of uncertainty estimation and its application to explore model complexity-uncertainty trade-off. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 4–5 (2020)
4. Goldfarb, R.J., Groves, D.I., Gardoll, S.: Orogenic gold and geologic time: a global synthesis. Ore Geol. Rev. **18**(1–2), 1–75 (2001)
5. Groves, D.I., Goldfarb, R.J., Robert, F., Hart, C.J.: Gold deposits in metamorphic belts: overview of current understanding, outstanding problems, future research, and exploration significance. Econ. Geol. **98**(1), 1–29 (2003)
6. Harris, J., Grunsky, E., Behnia, P., Corrigan, D.: Data-and knowledge-driven mineral prospectivity maps for Canada's north. Ore Geol. Rev. **71**, 788–803 (2015)
7. Hosseini, S.A., Abedi, M.: Data envelopment analysis: a knowledge-driven method for mineral prospectivity mapping. Comput. Geosci. **82**, 111–119 (2015)
8. Idaho Geological Survey Staff: Idaho geological survey. Online (2022). https://www.idahogeology.org
9. Li, Q., Chen, G., Luo, L.: Mineral prospectivity mapping using attention-based convolutional neural network. Ore Geol. Rev. **156**, 105381 (2023)
10. Mahboob, M.A., Celik, T., Genc, B.: Predictive modeling and comparative evaluation of geostatistical models for geochemical exploration through stream sediments. Arab. J. Geosci. **13**(20), 1–21 (2020)
11. Parsa, M., Harris, J., Sherlock, R.: Improving mineral prospectivity model generalization: an example from orogenic gold mineralization of the sturgeon lake transect, ontario, canada. Math. Geosci. **55**(7), 943–961 (2023)
12. Phillips, G.N., Powell, R.: Hydrothermal alteration in the witwatersrand goldfields. Ore Geol. Rev. **65**, 245–273 (2015)
13. Rama, B., Praveen, P., Sinha, H., Choudhury, T.: A study on causal rule discovery with pc algorithm. In: 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions)(ICTUS), pp. 616–621. IEEE (2017)
14. Sadeghi, B., Grunsky, E., Pawlowsky-Glahn, V.: Uncertainty quantification. In: Encyclopedia of Mathematical Geosciences, pp. 1583–1589. Springer (2023)
15. Shafer, G., Vovk, V.: A tutorial on conformal prediction. J. Mach. Learn. Res. **9**(3) (2008)
16. Syrgkanis, V., et al.: Causal inference and machine learning in practice with EconML and CausalML: industrial use cases at Microsoft, TripAdvisor, Uber. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 4072–4073 (2021)
17. U.S. Geological Survey Staff: U.s. geological survey. Online (2022). https://www.usgs.gov
18. Wang, H., Liang, Q., Hancock, J.T., Khoshgoftaar, T.M.: Feature selection strategies: a comparative analysis of shap-value and importance-based methods. J. Big Data **11**(1), 44 (2024)
19. Wang, J., Zuo, R., Xiong, Y.: Mapping mineral prospectivity via semi-supervised random forest. Nat. Resour. Res. **29**, 189–202 (2020)
20. Wellmer, F.W.: Statistical evaluations in exploration for mineral deposits. Springer Science & Business Media (2012)
21. Xiong, Y., Zuo, R.: Recognizing multivariate geochemical anomalies for mineral exploration by combining deep learning and one-class support vector machine. Comput. Geosci. **140**, 104484 (2020)

22. Yang, N., Zhang, Z., Yang, J., Hong, Z.: Applications of data augmentation in mineral prospectivity prediction based on convolutional neural networks. Comput. Geosci. **161**, 105075 (2022)
23. Zhang, S., Tong, H., Xu, J., Maciejewski, R.: Graph convolutional networks: a comprehensive review. Comput. Soc. Netw. **6**(1), 1–23 (2019)
24. Zuo, R., et al.: Explainable artificial intelligence models for mineral prospectivity mapping. Sci. China Earth Sci. **67**(9), 2864–2875 (2024)



PhysMamba: Synergistic State Space Duality Model for Remote Physiological Measurement

Zhixin Yan[1], Yan Zhong[2], Hongbin Xu[3], Wenjun Zhang[2], Shangru Yi[1], Wenxiong Kang[3], and Lin Shu[1(✉)]

[1] School of Future Technology, South China University of Technology, Guangzhou, China
shul@scut.edu.cn

[2] School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China

[3] School of Automation Science and Engineering, South China University of Technology, Guangzhou, China

Abstract. Remote Photoplethysmography (rPPG) enables non-contact physiological signal extraction from facial videos, offering applications in psychological state analysis, medical assistance, and anti-face spoofing. However, challenges such as motion artifacts, lighting variations, and noise limit its real-world applicability. To address these issues, we propose PhysMamba, a novel dual-pathway time-frequency interaction model based on Synergistic State Space Duality (SSSD), which for the first time integrates state space models with attention mechanisms in a dual-branch framework. Combined with a Multi-Scale Query (MQ) mechanism, PhysMamba achieves efficient information exchange and enhanced feature representation, ensuring robustness under noisy and dynamic conditions. Experiments on PURE, UBFC-rPPG, and MMPD datasets demonstrate that PhysMamba outperforms state-of-the-art methods, offering superior accuracy and generalization. This work lays a strong foundation for practical applications in non-contact health monitoring, including real-time remote patient care. The code is available at https://github.com/ZhiXinChasingLight/PhysMamba.git.

Keywords: Remote Photoplethysmography · Heart Rate Estimation · Mamba · State Space Duality

1 Introduction

Photoplethysmography (PPG) is a widely-used technique for estimating vital physiological indicators such as heart rate, heart rate variability, and respiratory rate, which are crucial for applications like sleep monitoring, medical assistance, and fatigue assessment. While traditional PPG relies on contact-based sensors like ECG monitors, its limited scalability and user inconvenience have driven the

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 128–140, 2026.
https://doi.org/10.1007/978-3-032-04555-3_11

development of non-contact alternatives. Remote photoplethysmography (rPPG) leverages cameras to estimate physiological signals by capturing subtle blood flow changes in the skin [1], unlocking potential applications in remote health monitoring and psychological state analysis.

Despite its promise, deploying rPPG in real-world environments faces significant challenges, including motion artifacts, lighting variations, and environmental noise. These factors disrupt signal consistency and hinder the generalization of rPPG systems, limiting their practical adoption in scenarios such as telemedicine and stress monitoring.

Traditional rPPG methods, including signal processing techniques like Independent Component Analysis (ICA) and chrominance-based projections [2,3], perform well in controlled environments but struggle under dynamic conditions due to their reliance on handcrafted features. Recent deep learning approaches, such as CNN-based models [4], have improved motion artifact handling, while Transformer-based architectures like PhysFormer [5] enhance temporal dependency modeling. However, these models suffer from high computational complexity, making them less practical for real-time applications. State Space Models (SSMs) have emerged as a computationally efficient alternative, as demonstrated by RhythmMamba [6], which captures quasi-periodic patterns in physiological signals with low computational cost. Yet, existing SSM-based methods face limitations in generalization and parallel processing, particularly in cross-dataset scenarios (Fig. 1).

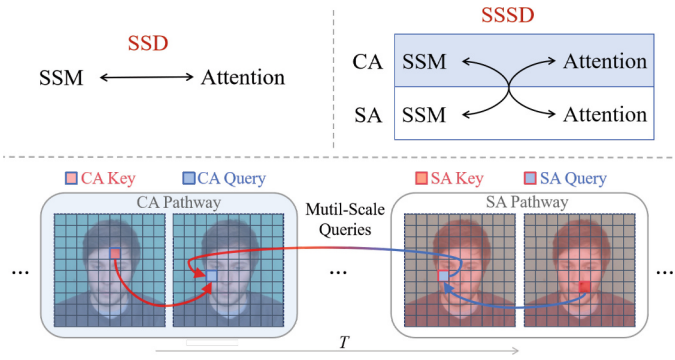


Fig. 1. Comparison between SSD and SSSD. The traditional SSD combines SSM and attention mechanisms, while our enhanced SSSD introduces Multi-Scale Queries for efficient information interaction between the two pathways.

Building on these advancements, we propose **PhysMamba**, a novel dual-pathway time-frequency interaction model that integrates **Synergistic State Space Duality (SSSD)** for rPPG estimation. This is the first framework to combine the efficiency of state space models with the feature extraction capabilities of attention mechanisms in a dual-pathway architecture. By introducing a

Multi-Scale Query (MQ) mechanism, PhysMamba achieves efficient information sharing between pathways, enabling robust feature representation under noisy and dynamic conditions.

The main contributions of this work are summarized as follows:

- We propose PhysMamba, an efficient rPPG estimation method via State Space Duality in a dual-pathway time-frequency interactive network. This is the first attempt handling rPPG estimation via SSD, creating technological breakthroughs.
- To enhance information exchange in dual-pathway model, we develop a simple state space fusion module using Multi-Scale Queries for effective lateral information sharing.
- Extensive experiments on three public rPPG datasets (PURE, UBFC-rPPG, and MMPD) validate the superior performance of PhysMamba, achieving state-of-the-art results in both controlled and complex real-world scenarios.

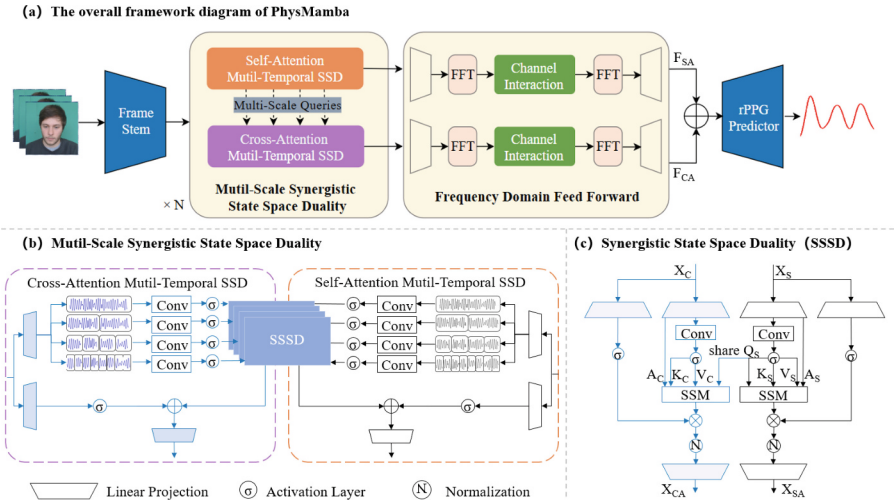


Fig. 2. Overall framework diagram of the PhysMamba model. The model comprises the Frame Stem, Multi-Scale Synergistic State Space Duality (SSSD), Frequency Domain Feed-Forward (FDF), and rPPG Predictor, integrated into two pathways: the Self-Attention Pathway and the Cross-Attention Pathway.

2 Related Work

2.1 rPPG Estimation

Traditional rPPG methods, like ICA and chrominance subspace projection [2,7], perform well in controlled environments but struggle in dynamic settings due to motion artifacts and lighting variations [8]. They also rely on handcrafted

features, limiting adaptability. Deep learning has advanced rPPG estimation with end-to-end frameworks. Models like DeepPhys [4] improved motion artifact handling using CNNs but struggled with long-term dependencies. Transformer-based methods, such as PhysFormer [5], used self-attention for global dependency modeling but faced high computational complexity, hindering real-time use.

2.2 State Space Model

SSMs have gained significant interest for capturing contextual information in long sequences. The Mamba SSM [9] preserves linear time complexity while effectively capturing long-term dependencies, applied in areas such as time series analysis [10], medical image segmentation [11], and video understanding [12]. It has also shown promising results in rPPG estimation with low computational complexity. RhythmMamba [6] robustly detects quasi-periodic patterns in rPPG signals through a multi-phase learning framework and frequency-domain feed-forward mechanism. However, Mamba-1's reliance on previous states limited its efficiency and generalization. Mamba-2 [13], introducing the State Space Duality framework, enhanced linear time complexity and efficiency by integrating attention mechanisms, improving performance in complex physiological signal estimation like rPPG.

3 Methodology

3.1 Overall Framework

PhysMamba is an end-to-end model designed to process raw facial video data and estimate rPPG signals directly. The framework, illustrated in Fig. 2, consists of two pathways: the Self-Attention (SA) Pathway and the Cross-Attention (CA) Pathway. These pathways incorporate three core modules: the Frame Stem for temporal feature extraction, the Multi-Scale Synergistic State Space Duality (SSSD) module for robust temporal representation, and the Frequency Domain Feed-Forward (FDF) network for frequency enhancement. Outputs from these pathways are fused in the rPPG Predictor to generate final signal estimations. The dual-pathway architecture enables one pathway to focus on internal temporal consistency (SA), while the other facilitates cross-feature interactions (CA), ensuring robustness in diverse conditions.

3.2 Frame Stem

The Frame Stem extracts temporal features from the input video $X \in \mathbb{R}^{3\times T\times H\times W}$ while reducing spatial noise, where T, W, and H denote the length (number of frames), width, and height of the video sequence, respectively. To enhance temporal dynamics, we define the frame difference feature X_{diff} as the concatenation of multi-scale temporal differences:

$$X_{\text{diff}} = \text{Stem}(\text{Concat}\,(D_{t-2}, D_{t-1}, D_t + 1, D_t + 2)), \tag{1}$$

Here, $D_\tau = X_{t+\tau} - X_t$, where $\tau \in \{-2, -1, 1, 2\}$. The module Stem refers to a shared feature extractor composed of 2D convolutional neural networks (CNNs), batch normalization layers, ReLU activation functions, and max-pooling operations. This operation is designed to capture subtle temporal variations that are critical for the estimation of physiological signals. The processed features are then computed as:

$$X_{\text{Stem}} = \text{Stem}(X_{origin}) + \text{Stem}(X_{origin} + X_{\text{diff}}), \tag{2}$$

where $X_{\text{origin}} = \text{Stem}(X_t)$. Furthermore, a 5×5 convolutional layer combined with a self-attention mechanism adaptively focuses on informative facial regions, thereby enhancing the quality of features for subsequent modules.

3.3 Multi-scale Synergistic State Space Duality (SSSD)

The SSSD module integrates State Space Models (SSMs) with attention mechanisms to capture both long-term dependencies and short-term variations in temporal data. A core element of SSSD is the Multi-Scale Query (MQ) mechanism, which divides input sequences X_{Stem} into different temporal windows and maps them as queries to capture features at multiple scales. This mechanism ensures that the model can effectively process diverse temporal sequences, helping it focus on varying sequence lengths and adapt to dynamic conditions such as motion artifacts and lighting changes. This design enables PhysMamba to excel in both time-domain and frequency-domain tasks, ensuring high performance in real-world scenarios.

SSMs operate as "memory systems" that track evolving patterns over time, mathematically described as:

$$h_t = Ah_{t-1} + Bx_t, \tag{3}$$

$$y_t = C^\top h_t, \tag{4}$$

where A, B, and C are learnable matrices controlling state transitions and feature outputs.

The State Space Duality (SSD) framework introduces structured attention to efficiently encode temporal dependencies:

$$\text{SSD} = (L \circ QK^\top) \cdot V, \tag{5}$$

where Q, K, and V are the query, key, and value matrices, respectively mapped from X_{Stem}, and L is a structured mask matrix that replaces traditional softmax operations. The mask matrix L ensures causal attention, such that:

$$L_{ij} = \begin{cases} 1 & \text{if } i \geq j \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

The SSSD module extends the standard attention mechanism by introducing two pathways: Self-Attention (SA) and Cross-Attention (CA). Let Q_S, K_S, V_S,

K_C, V_C denote the query, key, and value matrices derived from X_{Stem} through learnable linear projections in different pathways. The SA Pathway focuses on self-dependencies within individual features:

$$X_{\text{SA}} = (L_S \circ Q_S K_S^\top) \cdot V_S. \tag{7}$$

The CA Pathway receives multi-scale Q_S from the SA pathway, promoting cross-feature refinement:

$$X_{\text{CA}} = (L_C \circ Q_S K_C^\top) \cdot V_C. \tag{8}$$

These pathways collaborate like two analysts: one delves into the temporal consistency of individual features (SA), while the other identifies interactions across features (CA), ensuring robust learning even under noisy conditions.

3.4 Frequency Domain Feed-Forward (FDF)

To enhance periodic physiological patterns, the FDF network applies Fast Fourier Transform (FFT) to convert temporal signals $h(t)$ into the frequency domain:

$$H(f) = \int_{-\infty}^{+\infty} h(t) e^{-j2\pi ft} dt, \tag{9}$$

which emphasizes periodic components like heart rate signals. The inverse FFT restores the frequency-enhanced signals to the temporal domain, preserving these enhancements. Outputs from the SA and CA pathways, denoted as F_{SA} and F_{CA}, are then combined to optimize the rPPG signal representation. Both F_{SA} and F_{CA} belong to $X \in \mathbb{R}^{T \times C}$, where C indicate channel.

Table 1. Intra-dataset results on UBFC, PURE, and MMPD datasets.

Method	UBFC				PURE				MMPD				
	MAE	RMSE	MAPE	r	MAE	RMSE	MAPE	r	MAE	RMSE	MAPE	r	SNR
GREEN [14]	19.73	31.00	18.72	0.37	10.09	23.85	10.28	0.34	21.68	27.69	24.39	-0.01	-14.34
ICA [15]	16.00	25.65	15.35	0.44	4.77	16.07	4.47	0.72	18.60	24.30	20.88	0.01	-13.84
CMROM [16]	4.06	8.83	3.34	0.89	5.77	14.93	11.52	0.81	13.66	18.76	16.00	0.08	-11.74
LGI [17]	15.80	28.55	14.70	0.36	4.61	15.38	4.96	0.77	17.08	23.32	18.98	0.04	-13.15
PBV [18]	15.90	26.40	15.17	0.48	3.92	12.99	4.84	0.84	17.95	23.58	20.18	0.09	-13.88
POS [7]	4.08	7.72	3.93	0.92	3.67	11.82	7.25	0.88	12.36	17.71	14.43	0.18	-11.53
OMIT [19]	15.79	28.54	14.69	0.36	4.65	15.81	4.96	0.75	7.80	12.00	10.55	0.13	-8.68
DeepPhys [4]	0.76	1.09	0.79	0.99	3.33	14.45	2.91	0.90	23.73	28.25	25.63	-0.06	-15.45
PhysNet [20]	0.58	0.83	0.61	0.99	0.54	0.93	0.58	0.99	4.81	11.83	4.84	0.60	1.51
TS-CAN [21]	0.81	1.10	0.84	0.99	0.40	0.73	0.44	0.99	8.97	16.58	9.43	0.44	-6.92
PhysFormer [5]	0.63	0.98	0.65	0.99	0.25	0.37	0.34	0.99	13.64	19.39	14.42	0.15	-11.02
EfficientPhys [22]	0.72	1.01	0.75	0.99	5.10	16.61	4.19	0.87	12.79	21.12	13.48	0.24	-9.23
RhythmMamba [6]	0.54	0.79	0.54	0.99	0.29	0.39	0.36	0.99	3.16	7.27	3.37	0.84	4.74
PhysMamba (Ours)	**0.45**	**0.76**	**0.45**	**0.99**	**0.24**	**0.36**	**0.31**	**0.99**	**2.84**	**6.41**	**3.04**	**0.88**	**5.20**

3.5 rPPG Predictor

The outputs from the SA and CA pathways are concatenated along the channel dimension:

$$X_{\text{fusion}} = \text{Concat}(F_{CA}, F_{SA}). \tag{10}$$

The fused feature $X_{\text{fusion}} \in \mathbb{R}^{T \times 2C}$ is processed by a 1D convolution to yield the final rPPG predictions. By integrating both time-domain and frequency-domain features, the rPPG Predictor ensures accurate heart rate estimation even under challenging conditions, such as varying lighting and motion artifacts.

4 Experiments

4.1 Datasets

To rigorously evaluate PhysMamba's performance, we selected three widely-used rPPG datasets with varying levels of complexity: UBFC-rPPG [23], PURE [24], and MMPD [25]. These datasets present distinct challenges, ranging from controlled laboratory conditions to highly complex real-world scenarios involving significant noise and motion artifacts.

UBFC-rPPG: This dataset consists of videos from 42 subjects performing mathematical tasks to induce heart rate variability. The relatively clean facial videos with minimal noise make it a benchmark for assessing model accuracy under simple conditions.

PURE: Comprising videos of 10 subjects under six controlled head movement scenarios—including stillness, talking, and slow/fast rotations—PURE introduces moderate motion artifacts, testing the model's robustness to dynamic conditions.

MMPD: A highly challenging dataset featuring 33 subjects under varying lighting conditions, skin tones, and head movement complexities. Capturing over 11 h of video with diverse environmental factors, MMPD includes significant noise and motion artifacts, simulating real-world complexities. Due to computational constraints, we utilized the mini-MMPD version for experimentation.

We evaluated PhysMamba using standard metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and Pearson correlation coefficient (r). For MMPD, we also computed the Signal-to-Noise Ratio (SNR) to assess robustness in high-noise environments.

4.2 Experimental Details

All experiments were conducted using the open-source *rppg-toolbox* [26] in PyTorch. Videos were preprocessed by cropping and resizing facial regions to 128×128 pixels. The model was trained using a learning rate of 3×10^{-4}, a batch size of 16, and for 30 epochs. The SSD model's dimension and state space size were set to 64, with a head dimension of 16. All computations were performed on an NVIDIA RTX 4090 GPU.

The overall loss function combines time-domain (L_{Time}) and frequency-domain (L_{Freq}) components for robust learning:

$$\mathcal{L}_{\text{overall}} = L_{\text{Time}} + L_{\text{Freq}},$$

where L_{Time} minimizes the negative Pearson correlation between predicted and ground-truth rPPG signals, and L_{Freq} aligns their dominant frequency components. Specifically, L_{Freq} is defined as:

$$\mathcal{L}_{\text{Freq}} = \text{CE}\left(\arg\max_k \text{PSD}(rPPG_{\text{gt}})_k, \text{PSD}(rPPG_{\text{pred}})\right),$$

PSD$(\cdot)$ computes the power spectral density via Fast Fourier Transform (FFT), and $\arg\max_k$ identifies the frequency bin with the highest amplitude.

5 Experimental Evaluation

We evaluated PhysMamba's effectiveness and superiority in rPPG heart rate estimation through both intra-dataset and cross-dataset assessments. Comparisons were made with unsupervised methods (e.g., GREEN [14], ICA [15], CHROM [16]) and state-of-the-art deep learning methods (e.g., DeepPhys [4], PhysNet [20], RhythmMamba [6]). Further details are available in the supplementary materials.

5.1 Intra-Dataset Testing

PhysMamba was trained and tested on the same dataset to assess its performance under consistent conditions. As shown in Table 1, it achieved near-optimal results on the UBFC-rPPG and PURE datasets (MAEs of 0.45 bpm and 0.24 bpm, respectively), outperforming existing methods. On the challenging MMPD dataset, PhysMamba demonstrated superior robustness, with an MAE of 2.84 bpm and the highest SNR of 5.20. These results confirm PhysMamba's ability to handle noise, motion artifacts, and varying lighting conditions. Its low computational complexity, enabled by the SSSD framework, makes it suitable for mobile and real-time monitoring systems. The robustness on MMPD underscores its adaptability in complex, noisy environments.

5.2 Cross-Dataset Testing

For cross-dataset evaluation, PhysMamba was trained and tested on the PURE and MMPD datasets, reflecting real-world variations. As shown in Table 2, PhysMamba outperformed existing methods across datasets. When trained on MMPD and tested on PURE, it achieved an MAE of 5.32 bpm, surpassing RhythmMamba. When trained on PURE and tested on MMPD, it achieved an MAE of 9.87 bpm, significantly outperforming other methods. PhysMamba's strong generalization comes from its SSSD design, combining state space models and attention mechanisms, along with multi-scale queries (MQ), which together enable robust performance under data shifts like lighting, skin tone, and motion changes.

Table 2. Cross-dataset results. MMPD → PURE: train on MMPD, test on PURE; PURE → MMPD: train on PURE, test on MMPD.

Method	MMPD → PURE				PURE → MMPD			
	MAE	RMSE	MAPE	r	MAE	RMSE	MAPE	r
DeepPhys [4]	15.86	25.72	20.76	-0.01	16.94	24.66	18.51	0.05
PhysNet [20]	10.96	20.24	18.34	0.54	13.94	21.61	15.15	0.20
TS-CAN [21]	7.24	16.80	9.76	0.68	13.92	21.60	15.12	0.21
PhysFormer [5]	17.60	24.75	26.93	0.00	14.60	20.75	16.76	0.16
EfficientPhys [22]	5.68	16.59	**6.26**	0.70	14.13	21.65	15.33	0.17
RhythmMamba [6]	6.07	14.43	9.66	0.78	10.45	16.49	12.20	0.37
PhysMamba (Ours)	**5.32**	**12.98**	9.56	**0.84**	**9.87**	**15.47**	**11.76**	**0.40**

5.3 Ablation Study

Table 3. Ablation study results on the MMPD dataset. MQ: Multi-Scale Queries, FDF: Frequency Domain Feed-Forward, DP: Dual-Pathway, SSD: State Space Duality, CA: Cross-Attention, SF: SlowFast.

MQ	FDF	DP	SSD	CA	SF	MAE	RMSE	MAPE	r	SNR
✓	✓					4.40	8.81	4.66	0.76	3.36
✓	✓		✓			3.46	7.21	3.69	0.84	3.94
✓	✓	✓				3.13	6.55	3.31	0.87	4.63
✓	✓	✓	✓			3.72	8.06	3.95	0.80	4.15
✓		✓	✓	✓		3.53	7.46	3.83	0.82	1.50
	✓	✓	✓	✓		3.49	7.35	3.69	0.83	4.17
✓	✓		✓	✓	✓	3.47	7.67	3.67	0.82	4.05
✓	✓	✓	✓	✓		**2.84**	**6.41**	**3.04**	**0.88**	**5.20**

To evaluate the contribution of each key component in PhysMamba, we conducted ablation studies on the MMPD dataset. The results, summarized in Table 3, demonstrate the critical role of the proposed modules in achieving state-of-the-art performance. Firstly, the Multi-Scale Queries (MQ) module enhances the model's ability to capture multi-scale dependencies, reducing the MAE from 4.40 bpm to 3.46 bpm. Additionally, the Frequency Domain Feed-Forward (FDF) module improves the detection of periodic patterns, increasing the SNR from 1.50 to 5.20. We further compared Dual-Pathway (DP) with the popular two-branch strategy SlowFast in [27], and the results show that DP achieves a 22% improvement in MAE, demonstrating its superior performance through efficient redundancy reduction and enhanced feature learning. Finally, the State Space Duality (SSD) framework enables robust temporal learning, ensuring the best

Table 4. Comparison of different methods in terms of Parameters (M), MACs (M), Throughput, and GPU Memory Usage (M).

Method	Para.	MACs	Throughput	Memory
DeepPhys [4]	1.98	744.45	11.32	37.43
PhysNet [20]	0.77	438.24	23.52	11.23
TS-CAN [21]	1.98	744.45	10.89	38.66
PhysFormer [5]	7.38	316.29	17.50	28.78
EfficientPhys [22]	1.91	373.72	16.63	26.38
RhythmMamba [6]	1.07	80.90	40.10	6.66
PhysMamba(Ours)	1.85	159.78	32.12	10.23

overall performance with an MAE of 2.84 bpm and an SNR of 5.20. These results collectively confirm the effectiveness of each component in optimizing both accuracy and robustness.

5.4 Computational Efficiency

We tested all methods with a 30-second inference. Table 4 presents the Parameters, average MACs per frame, average throughput per frame, and average GPU Memory Usage per frame. It is evident that PhysMamba achieves higher throughput and lower GPU memory consumption, demonstrating its potential for application in edge devices.

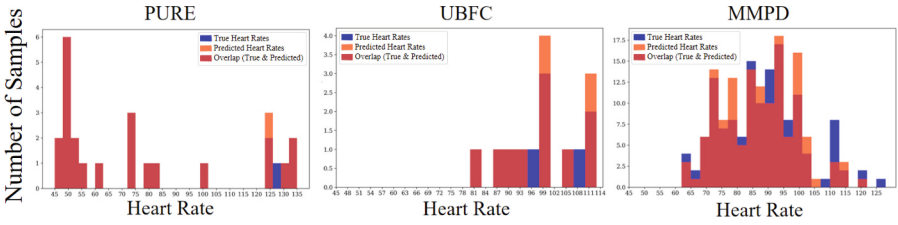


Fig. 3. Comparison of predicted and ground-truth heart rate distributions across UBFC-rPPG, PURE, and MMPD datasets.

5.5 Visualization and Further Analysis

To demonstrate the excellent performance of PhysMamba, we present the visualized results of the experiments. Figure 3 compares predicted and ground-truth heart rate distributions across three datasets, showing strong alignment, even under challenging conditions like MMPD. Figure 4 visualizes MMPD results, highlighting a strong correlation between predicted and ground-truth heart rates. A second-order Butterworth filter (cutoff: 0.75–2.5 Hz) was applied to the predicted PPG waveform, confirming PhysMamba's accuracy and robustness in heart rate estimation.

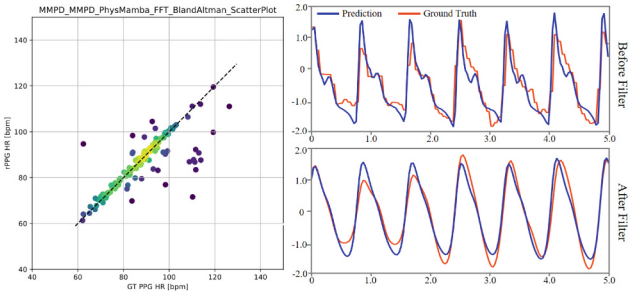


Fig. 4. Visualization of the results from the MMPD dataset. (a) the Bland-Altman plot. (b) the waveform of a sample before and after filtering.

6 Conclusion

This paper proposes PhysMamba, a dual-pathway rPPG framework integrating State Space Duality with attention mechanisms. It combines long-term dependency modeling with a full-sequence attention mechanism. We also propose a simple yet innovative Synergistic State Space Duality algorithm with Multi-Scale Queries, facilitating effective information interaction within the dual-pathway network. Both internal and cross-dataset tests show that PhysMamba achieves state-of-the-art performance, with high accuracy and robustness. The method demonstrates practical potential for non-contact health monitoring, with future extensions targeting real-time psychological state analysis in dynamic environments.

Acknowledgments. This work was supported by grants from the National Key Research and Development Program of China (2022YFC3601004), the Fundamental Research Funds for the Central Universities (2024ZYGXZR077), and the Taihu Lake Innovation Fund for the School of Future Technology at South China University of Technology (2024B105611003).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Choi, J.H., Kang, K.B., Kim, K.T.: Fusion-vital: Video-rf fusion transformer for advanced remote physiological measurement. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 1344–1352 (2024)
2. Li, X., Chen, J., Zhao, G., Pietikainen, M.: Remote heart rate measurement from face videos under realistic situations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4264–4271 (2014)

3. Tulyakov, S., Alameda-Pineda, X., Ricci, E., Yin, L., Cohn, J.F., Sebe, N.: Self-adaptive matrix completion for heart rate estimation from face videos under realistic conditions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2396–2404 (2016)
4. Chen, W., McDuff, D.: Deepphys: video-based physiological measurement using convolutional attention networks. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 349–365 (2018)
5. Yu, Z., Shen, Y., Shi, J., Zhao, H., Torr, P.H., Zhao, G.: Physformer: facial video-based physiological measurement with temporal difference transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4186–4196 (2022)
6. Zou, B., Guo, Z., Hu, X., Ma, H.: Rhythmmamba: fast remote physiological measurement with arbitrary length videos. arXiv preprint arXiv:2404.06483 (2024)
7. Wang, W., Den Brinker, A.C., Stuijk, S., De Haan, G.: Algorithmic principles of remote ppg. IEEE Trans. Biomed. Eng. **64**(7), 1479–1491 (2016)
8. Niu, X., Han, H., Shan, S., Chen, X.: Synrhythm: learning a deep heart rate estimator from general to specific. In: 2018 24th International Conference on Pattern Recognition (ICPR), pp. 3580–3585. IEEE (2018)
9. Gu, A., Dao, T.: Mamba: linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752 (2023)
10. Ma, S., Kang, Y., Bai, P., Zhao, Y.B.: Fmamba: mamba based on fast-attention for multivariate time-series forecasting. arXiv preprint arXiv:2407.14814 (2024)
11. Ma, J., Li, F., Wang, B.: U-mamba: enhancing long-range dependency for biomedical image segmentation. arXiv preprint arXiv:2401.04722 (2024)
12. Li, K., et al.: Videomamba: state space model for efficient video understanding. arXiv preprint arXiv:2403.06977 (2024)
13. Dao, T., Gu, A.: Transformers are ssms: generalized models and efficient algorithms through structured state space duality. arXiv preprint arXiv:2405.21060 (2024)
14. Verkruysse, W., Svaasand, L.O., Nelson, J.S.: Remote plethysmographic imaging using ambient light. Opt. Express **16**(26), 21434–21445 (2008)
15. Poh, M.Z., McDuff, D.J., Picard, R.W.: Non-contact, automated cardiac pulse measurements using video imaging and blind source separation. Opt. Express **18**(10), 10762–10774 (2010)
16. De Haan, G., Jeanne, V.: Robust pulse rate from chrominance-based rppg. IEEE Trans. Biomed. Eng. **60**(10), 2878–2886 (2013)
17. Pilz, C.S., Zaunseder, S., Krajewski, J., Blazek, V.: Local group invariance for heart rate estimation from face videos in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1254–1262 (2018)
18. De Haan, G., Van Leest, A.: Improved motion robustness of remote-ppg by using the blood volume pulse signature. Physiol. Meas. **35**(9), 1913 (2014)
19. Casado, C.A., López, M.B.: Face2ppg: an unsupervised pipeline for blood volume pulse extraction from faces. IEEE J. Biomed. Health Inf. (2023)
20. Yu, Z., Li, X., Zhao, G.: Remote photoplethysmograph signal measurement from facial videos using spatio-temporal networks. arXiv preprint arXiv:1905.02419 (2019)
21. Liu, X., Fromm, J., Patel, S., McDuff, D.: Multi-task temporal shift attention networks for on-device contactless vitals measurement. Adv. Neural. Inf. Process. Syst. **33**, 19400–19411 (2020)

22. Liu, X., Hill, B., Jiang, Z., Patel, S., McDuff, D.: Efficientphys: enabling simple, fast and accurate camera-based cardiac measurement. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 5008–5017 (2023)
23. Bobbia, S., Macwan, R., Benezeth, Y., Mansouri, A., Dubois, J.: Unsupervised skin tissue segmentation for remote photoplethysmography. Pattern Recogn. Lett. **124**, 82–90 (2019)
24. Stricker, R., Müller, S., Gross, H.M.: Non-contact video-based pulse rate measurement on a mobile service robot. In: The 23rd IEEE International Symposium on Robot and Human Interactive Communication, pp. 1056–1062. IEEE (2014)
25. Tang, J., et al.: Mmpd: multi-domain mobile video physiology dataset. In: 2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), pp. 1–5. IEEE (2023)
26. Liu, X., et al.: rppg-toolbox: Deep remote ppg toolbox. In: Advances in Neural Information Processing Systems, vol. 36 (2024)
27. Yu, Z., et al.: Physformer++: facial video-based physiological measurement with slowfast temporal difference transformer. Int. J. Comput. Vision **131**(6), 1307–1330 (2023)

Proactive Depot Discovery: A Generative DRL Framework for Adaptive Location-Routing

Site Qu and Guoqiang Hu(✉)

Nanyang Technological University, 639798 Singapore, Singapore
site001@e.ntu.edu.sg gqhu@ntu.edu.sg

Abstract. The Location-Routing Problem (LRP), which combines the challenges of facility (depot) locating and vehicle routing, is critically constrained by the reliance on predefined depot candidates, limiting the solution space and potentially leading to suboptimal outcomes. Existing research on LRP without predefined depots is scant and predominantly relies on heuristic algorithms that iteratively attempt depot placements across a planar area. Such approaches lack the ability to proactively generate depot locations that meet specific geographic requirements, revealing a notable gap in current research landscape. To address this gap, we propose a generative Deep Reinforcement Learning (DRL) framework that proactively generates depot locations directly from customer geographic and demand information. By extracting depots' geographic pattern from customer requests, our approach can dynamically respond to logistical needs, identifying high-quality depot locations that further reduce total routing costs compared to traditional methods. Extensive experiments demonstrate that, for a same group of customer requests, compared with those depots identified through random attempts, our framework can proactively generate depots that lead to superior solution routes with lower routing cost. Its adaptability makes it particularly suitable for urgent real-world scenarios, such as emergency medical rescue and disaster-relief logistics, where rapid, dynamic depot establishment and adjustment is essential.

Keywords: Deep Reinforcement Learning · Generative Model · Location-Routing · Depot Generation

1 Introduction

The Location-Routing Problem (LRP) is a critical optimization challenge in the urban logistics industry, combining two interdependent decisions: selecting depot locations where vehicles commence and conclude their tasks, and planning vehicle routes for serving customers. This integration is crucial, as the depot locations can directly influence the vehicle route planning, thereby impacting overall operational costs [19]. The LRP can be formally defined as [12]: *Given*

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 141–153, 2026.
https://doi.org/10.1007/978-3-032-04555-3_12

a set of customers with specific location and quantity of demands, and a set of potential depot candidates each with a fleet of vehicles featuring fixed capacity, aiming to properly select a subset of depots and plan routes for vehicles departing from these chosen depots to meet customers' demands, while minimizing both depot-related and route-related costs, without violating specific constraints.

In this traditional configuration, solving LRP relies heavily on predefined depot candidates, significantly restricting the solution space and potentially leading to suboptimal results [6,13,14,23]. This limitation becomes critical in scenarios, such as emergency medical rescue or disaster relief, where rapid and flexible depot establishment is essential due to existing infrastructure disruptions.

Research addressing this extended LRP scenario without predefined candidates remains limited. Existing approaches predominantly utilize iterative heuristic algorithms, incrementally attempting depot placements across a continuous planar area, typically limited to one or two depots [18,21]. Such methods tend to exhibit limited flexibility, low efficiency, and struggle with realistic location constraints for depots (e.g., the required specific distance range among depots). In this situation, aimlessly expanding candidate depots significantly increases problem complexity, computation cost, and time consumption. Consequently, devising a proactive method to generate optimal depots that satisfy practical constraints is an essential and open research challenge for LRP scenario.

To bridge this gap, we propose a generative deep reinforcement learning (DRL) framework, uniquely crafted to address LRP in depot-generating fashion. Leveraging customers' logistical requests data, which encompass geographic locations and specific demands, our framework generates high-quality depot locations and plans efficient vehicle routes from these depots to serve customer needs. Specially, our framework encompasses two models: *(1) Depot Generative Model (DGM)*, a deep generative model capable of generating depots in two distinct modes: direct generation of exact depot locations *or* production of a multivariate Gaussian distribution for flexible depots sampling. The exact mode ensures precision when necessary, while the Gaussian mode introduces sampling variability, enhancing the model's generalization and robustness to diverse customer distributions. *(2) Multi-depot Location-Routing Attention Model (MDLRAM)*, an end-to-end DRL model focusing on providing an efficient LRP solution for serving customers based on the generated depots, with minimized objective including both route-related and depot-related cost.

1.1 Related Work

Methods for LRP with Predefined Depot Candidates: Traditional approaches for the LRP with predefined depot candidates mainly include exact and heuristic methods. Exact approaches [2,5,6] provide precise solutions but suffer from poor scalability due to computational complexity. This limitation motivates heuristic approaches, categorized into matheuristics [7,8,17], learning-aided heuristics [13,15], and pure meta-heuristics [4,14,18]. However, these

approaches face limitations in handling complex constraints (e.g., customer time-windows) and demonstrate insufficient integration between depot selection and route planning. Additionally, each minor alteration in the problem instance typically necessitates initiating an entirely new search, resulting in inefficiencies.

Recent advancements in DRL have shown potential in solving various routing problems, categorized as "learn-to-construct/generalize" [9,10,25] and "learn-to-improve/decompose" [11,24]. Nevertheless, DRL applications for integrated depot selection and routing (LRP) remain underexplored due to inherent challenges in modeling multi-depot scenarios and integrating depot-selecting with route-planning. Existing DRL methods for multi-depot scenarios [3,16,23] typically neglect depot-related costs, reducing the problem to multi-depot VRP rather than LRP, or adopts a two-stage clustering-and-routing approach, separating depot selection from routing, thereby neglecting their interdependencies.

Exploration of LRP without Predefined Depot Candidates: Only limited studies consider LRP without predefined depot candidates, predominantly using heuristic methods [1,18,20,21] that iteratively attempt new depots across a planar area devoid of predefined depot choices, yet typically handle only single or dual-depot scenarios. These heuristic attempts iteratively trial-and-error depot placements, subsequently evaluating improvements through re-planning routes, and lack the ability to directly incorporate depot location constraints, thereby underscoring the need for more proactive and flexible solutions.

2 Methodology

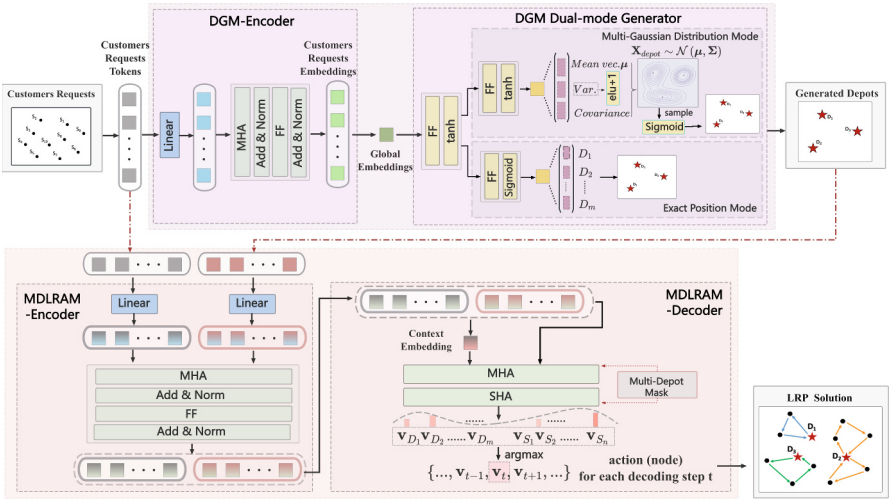


Fig. 1. Overview of the Generative DRL framework for Depot Generation in LRP.

2.1 Critic Model: MDLRAM

Following the conventional setup [5], the configuration is defined on an undirected graph where the vertices set $V = \{\mathbf{v}_{D_1}, \ldots, \mathbf{v}_{D_m}, \mathbf{v}_{S_1}, \ldots, \mathbf{v}_{S_n}\}$ comprises n customers and m depots. Each customer $\mathbf{v}_{S_e}$ has specific demands q_e, fulfilled entirely from exactly one depot by one vehicle (no order splitting or load transfers allowed). Each depot $\mathbf{v}_{D_k}$ is characterized by two attributes: (i) the maximum supply M_k (soft constraint), indicating the desired limit on total goods dispatched from this depot; (ii) the fixed opening cost O_k, indicating the expense for using this depot facility. Regarding the vehicles, we operate a homogeneous fleet, each vehicle with identical maximum capacity Q (hard constraint) indicating the maximum vehicle load during service, and a setup cost U for dispatching.

MDLRAM-Objective Function: In the LRP scenario, a feasible solution consists of multiple routes executed simultaneously by vehicles, each route starting and ending at a designated depot. MDLRAM generates these routes through sequential decision-making formulated as a Markov Decision Process (MDP). To mathematically formulate the LRP solution into MDP, with each decision step t associated with a tuple $(s_t, a_t, p_t, r_t, \gamma_t)$. The s_t represent current state, depicting overall task completion status; The action a_t denotes the next visit point, subject to the vehicle's remaining load; The p_t and the r_t correspond to the transition probability and cost associated with action a_t, respectively.

Following this construction, a feasible LRP solution is formulated, associated with an objective (cost) function expressed as Eq. (1). Apart from the step-wisely accumulated transit distance length $\sum_t r_t$ along with the MDP generating process, other costs, which depict solution's overall performance are also integrated into the total cost with a respective discount, including: *(i) the opening cost for used depots; (ii) the setup cost for dispatched vehicles; (iii) penalty of exceeding depot desired maximum supply.* In Eq. (1), $\eta_k \in \{0, 1\}$ represents whether depot D_k is opened, χ_k records the number of vehicles dispatched from depot D_k, and α, β, δ are coefficients.

$$L_{\text{Sel}}(\mathbf{A}) = \sum_t r_t + \alpha \cdot \sum_{k=1}^{m} O_k \cdot \eta_k + \beta \cdot \sum_{k=1}^{m} U \cdot \chi_k + \delta \cdot \sum_{k=1}^{m} \max[(\sum_e q_e)_k - M_k, 0] \quad (1)$$

MDLRAM aims to minimize the expectation of this loss associated with the LRP solution, defined as $\mathbb{E}[L_{\text{Sel}}(\mathbf{A})]$, where $L_{\text{Sel}}(\mathbf{A})$ is expressed in Eq. (1).

MDLRAM-Encoder: As is shown in the red block of Fig. 1, MDLRAM receives two streams of input: the depot candidates' locations and the customer requests. For each depot candidate $\mathbf{v}_{D_k}$ where $k \in \{1, \ldots, m\}$, it is represented by its coordinates $\mathbf{g}_{D_k} = [x_{D_k}, y_{D_k}]^T$. For each customer $\mathbf{v}_{S_e}$ where $e \in \{1, \ldots, n\}$, it is depicted by a vector concatenating its coordinates and specific demands, $\mathbf{g}_{S_e} = [x_{S_e}, y_{S_e}, q_e]^T$. These vectors are encoded as node embeddings through separate linear projections and N attention modules, for downstream decoding.

MDLRAM-Decoder: With the encoded node embeddings, the decoder operates iteratively to construct feasible solution routes in form of vertices' permutation as an MDP. At each decoding step, two elements are updated:

(i) Context embedding $\mathbf{h}_c^t$: Represents the current decoding state, which is designed as $\mathbf{h}_c^t = W^c[\mathbf{h}_a \| \mathbf{h}_{(t)} \| \mathbf{h}_{D(t)} \| Q_t] + \mathbf{b}^c$, concatenating four elements: $\mathbf{h}_a$ averages all node embeddings as global information; $\mathbf{h}_{(t)}$ is the node embedding of the current point; $\mathbf{h}_{D(t)}$ is the node embedding of the depot where the current route belongs to; Q_t is the remained load on the current vehicle.

(ii) Mask mechanism: In each decoding step, to determine the selection domain of feasible points, a boolean mask is engaged to dynamically exclude infeasible points based on vehicle remained load and tasks completion status (served customers), allowing for batch-wise manipulation on selection domains for a batch of instances, avoiding repeated operation for each individual.

Upon finalizing the boolean mask for current decoding step, the context embedding performs Multi-head Attention (MHA) with the masked node embeddings, generating an intermediate context embedding $\hat{\mathbf{h}}_c^t$ that captures glimpse information of feasible points. Then, $\hat{\mathbf{h}}_c^t$ undergoes Single-head Attention (SHA) with the filtered node embeddings, producing selection probabilities for each feasible point, from which the next action a_t can be chosen with probability p_t.

2.2 Dual-Mode Depot Generation: DGM

The configuration for depot generation is also defined on an undirected graph where the vertices set $V = \{\mathbf{v}_{S_1}, \ldots, \mathbf{v}_{S_n}\}$ only includes customer requests. A solution set incorporating m depots is pending to be generated. During depot generation, the distances among generated depots are expected to be within the range $[l_{\min}, l_{\max}]$, which means the depots being excessively close or distant with each other will both incur violation penalty.

DGM-Objective Function: As the main task of depot generation, the depots with desired properties are expected to be generated. According to the problem configuration for DGM, for the solution set of generated depots, denoted as $\mathcal{D}$, its loss can be defined as Eq. (2), where L_{MDLR} is the route length derived by MDLRAM based on the DGM generated depots, λ, ε are coefficients for penalty of the depots being too distant or close with each other:

$$L_{\text{Gen}}(\mathcal{D}) = L_{\text{MDLR}} + \sum_{i=1}^{m} \sum_{j=i}^{m} [\lambda \cdot \max(d_{ij} - l_{\max}, 0) + \varepsilon \cdot \max(l_{\min} - d_{ij}, 0)] \tag{2}$$

DGM aims to minimize the expectation of this loss associated with generated depot set, defined as $\mathbb{E}[L_{\text{Gen}}(\mathcal{D})]$, where $L_{\text{Gen}}(\mathcal{D})$ is formed as Eq. (2).

DGM-Encoder: The DGM solely processes the customer requests, each characterized by a vector $\mathbf{g}_{S_e} = [x_{S_e}, y_{S_e}, q_e]^T$ concatenating location and demands. These requests are encoded as node embeddings, jointly forming a global embedding: $\mathbf{h}_{serve} = \frac{1}{n} \sum_{i=1}^{n} \tilde{\mathbf{h}}_{S_i}^{(N)}$ for downstream depot generation.

DGM-Generator in Multivariate Gaussian Distribution Mode: In this mode, the DGM aims to generate a multivariate Gaussian distribution where

depots can be flexibly sampled. For identifying m depots, the generated multivariate Gaussian $\mathbf{X}_{depot} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ should exhibit $2m$ dimension, where any randomly sampled vector can be denoted as $\mathbf{X}_{depot} = (X_1, X_2, \ldots, X_{2m})^T$, with each pair of dimensions representing coordinates for a depot.

To construct such distribution, two essential components are: the mean vector $\boldsymbol{\mu} \in \mathbb{R}^{2m}$ and covariance matrix $\Sigma \in \mathbb{R}^{2m \times 2m}$. Hence, the output of DGM should be a vector $\mathbf{h}_{depot}$ structured as below, including the elements in mean vector, variance vector, and covariance matrix:

$$\mathbf{h}_{depot} = (\overbrace{h_1, \ldots, h_{2m}}^{mean}, \overbrace{\ldots, h_{4m}}^{variance}, \overbrace{\ldots, h_{4m+C_{2m}^2}}^{covariance})^T \tag{3}$$

This vector is obtained by applying two fully-connected layers to the global embedding $\mathbf{h}_{serve}$ derived by encoder.

DGM-Generator in Exact Position Mode: In this mode, DGM directly outputs exact depot coordinates. For m depots, the output vector is structured as below, where each pair of adjacent dimensions represents the coordinates (x_{D_k}, y_{D_k}) for a depot D_k:

$$\mathbf{h}_{depot} = (\overbrace{h_1, h_2}^{D_1}, \ldots, \overbrace{h_{2m-1}, h_{2m}}^{D_m})^T \tag{4}$$

Similar to Gaussian mode, this output is generated through two fully-connected layers applied to the global embedding $\mathbf{h}_{serve}$ derived by encoder.

2.3 Training Strategy

Step I: Pre-training of MDLRAM: MDLRAM concurrently processes a batch of problem instances randomly sampled from the configuration, thereby generating a batch of corresponding MDPs as their respective feasible solutions. Each MDP involves a permutation of actions, denoted as $\text{MDP}(\mathbf{A}) = \{a_1, a_2, \ldots\}$, associated with its probability: $p_{\boldsymbol{\theta}_\text{I}}(\mathbf{A}) = \prod_t p_t = \prod_t p(s_{t+1}|s_t, a_t)$, which is parameterized by $\boldsymbol{\theta}_\text{I}$, denoting the MDLRAM's parameters that require training. Based on a batch of such MDPs, each with a probability $p_{\boldsymbol{\theta}_\text{I}}(\mathbf{A})$ and a cost $L_\text{Sel}(\mathbf{A})$, the MDLRAM is trained by REINFORCE gradient estimator with greedy rollout baseline [9] to minimize the expectation of cost, as in Eq. (5), where the baseline $\bar{\mathcal{B}}$ is obtained from a parallel baseline network mirroring MDLRAM's structure, maintaining the best-found parameters.

$$\nabla \mathcal{L}(\boldsymbol{\theta}_\text{I}) = \mathbb{E}_{p_{\boldsymbol{\theta}_\text{I}}(\mathbf{A})}[(L_\text{Sel}(\mathbf{A}) - \bar{\mathcal{B}}) \nabla \log p_{\boldsymbol{\theta}_\text{I}}(\mathbf{A})] \tag{5}$$

Stability, Optimality and Necessity of Pre-trained MDLRAM: The batch-wise training of DGM relies on a fixed and pre-trained MDLRAM to provide consistent and stable routing solutions in each training iteration, enabling DGM to concentrate solely on learning effective depot-generation patterns. Since

this setting introduces dependency on MDLRAM's routing quality, we verified MDLRAM's performance rigorously against best-known and state-of-the-art results on standard real-world benchmarks, confirming its near-optimal batch-wise routing capability with significantly reduced and stable inference times. This efficient batch-processing ability is essential. Unlike exact methods, while optimal, are computationally prohibitive for batch inference due to the unstable inference times and case-by-case manner, the MDLRAM efficiently generates instant, stable, near-optimal routing solutions for a batch of cases simultaneously, providing consistent, high-quality batch-wise routing assessments necessary for effectively training DGM. It strikes a balanced trade-off between stable, efficient training conditions and solution quality, justifying its practical suitability and reliability as the foundation for the DGM training phase.

Step II: Dual-mode training of DGM: DGM can be trained in two modes, with the pre-trained MDLRAM serving as a fixed sub-solver. **Notably**, the $\boldsymbol{\theta}_{\mathrm{II}}$, denoting DGM's parameters that requires training, is appended as footnote only to those variables parameterized by DGM.

(i) Multivariate Gaussian distribution mode: In this mode, the DGM takes in a main-batch (B_{main}) of randomly sampled graphs: $\{G_b|b = 1, 2, ..., B_{\mathrm{main}}\}$, each with a group of customer requests, and then outputs a main-batch of corresponding multivariate Gaussian distributions: $\{\mathcal{N}_b(\boldsymbol{\mu}, \Sigma)|b = 1, 2, ..., B_{\mathrm{main}}\}$. Thus, training DGM involves ensuring that the depots sampled from these distributions yield favorable expectations for the cost $L_{\mathrm{Gen}}(\mathcal{D})$. To achieve this, from each distribution $\mathcal{N}_b(\boldsymbol{\mu}, \Sigma)$ within the main-batch, we sample a sub-batch (B_{sub}) sets of depots. Each depot set is represented as $\mathcal{D}_{\mathrm{multiG}}$, associated with their probabilities $p_{\boldsymbol{\theta}_{\mathrm{II}}}(\mathcal{D}_{\mathrm{multiG}})$ and costs $L_{\mathrm{Gen}}(\mathcal{D}_{\mathrm{multiG}})$ in Eq. (2). In this way, a main-batch (B_{main}) of cost expectations, each corresponding to a multivariate Gaussian distribution, can be derived. We employ following optimizer to train the DGM in distribution mode:

$$\nabla\mathcal{L}(\boldsymbol{\theta}_{\mathrm{II}}) = \frac{1}{B_{\mathrm{main}}} \sum_{b=1}^{B_{\mathrm{main}}} \mathbb{E}^{(b)}_{p_{\boldsymbol{\theta}_{\mathrm{II}}}(\mathcal{D}_{\mathrm{multiG}})}[L_{\mathrm{Gen}}(\mathcal{D}_{\mathrm{multiG}}, G_b) \cdot \nabla \log p_{\boldsymbol{\theta}_{\mathrm{II}}}(\mathcal{D}_{\mathrm{multiG}})] \tag{6}$$

(ii) Exact position mode: In this mode, the DGM still ingests a main-batch (B_{main}) of randomly sampled graphs: $\{G_b|b = 1, 2, ..., B_{\mathrm{main}}\}$, each with a group of customer requests, but directly generates the corresponding sets of depots: $\{\mathcal{D}^{(b)}_{\mathrm{exactP}}|b = 1, 2, ..., B_{\mathrm{main}}\}$. For each set of depots $\mathcal{D}_{\mathrm{exactP}}$, the cost $L_{\mathrm{Gen}}(\mathcal{D}_{\mathrm{exactP}})$ can be derived by Eq. (2) whose first part is obtained by pre-trained MDLRAM. Below optimizer guides the DGM's training in exact mode:

$$\nabla\mathcal{L}(\boldsymbol{\theta}_{\mathrm{II}}) = \frac{1}{B_{\mathrm{main}}} \sum_{b=1}^{B_{\mathrm{main}}} \nabla L_{\mathrm{Gen}}((\mathcal{D}^{(b)}_{\mathrm{exactP}})_{\boldsymbol{\theta}_{\mathrm{II}}}, G_b) \tag{7}$$

Adaptability to Variable Depot Counts: To ensure stable reinforcement learning convergence on the scale-specific expectation of total cost, the current DGM generates a fixed number of depots during training. However, we recognize real-world scenarios often demand variable depot counts due to fluctuating

customer demands and spatial constraints. The dual-generation-mode design offer adaptability to this potential scenario. Specifically, the Multivariate Gaussian Distribution Mode provides flexibility by generating depot locations probabilistically, enabling multiple candidate sets to accommodate varying operational requirements. Meanwhile, for large-scale deployments requiring numerous depots, applying proper regional decomposition strategies allows independent depot generation for each sub-region, thus enhancing scalability and adaptability without substantial retraining efforts.

3 Experimental Results and Discussion

Training Setup: To ensure comparability, the experimental dataset follows standard practices from prior routing studies [9], with depot configurations consistent with widely-used LRP benchmarks [5,15]. Each problem instance is defined on a unit square $[0,1] \times [0,1]$, where customer locations are uniformly scattered and demands are uniformly sampled from $[0,10]$.

The *problem instances* for MDLRAM's pre-training are from three problem scales: $n = 20, 50, 100$ customers, respectively coupled with $m = 3, 6, 9$ depot candidates. ***Corresponding to each scale:*** (1) The vehicle's maximum capacity Q is selected as $30, 40, 50$ respectively; (2) The vehicle's setup cost U is set as 0.3; (3) The depot's desired maximum supply M_k is uniformly selected from $[50, 80]$, $[80, 120]$, $[120, 170]$; (4) The depot's opening cost O_k is uniformly selected from $[2, 5]$, $[2, 5]$, $[12, 19]$; The coefficients in objective function Eq. (1) are defined as $\alpha = 1, \beta = 1, \delta = 2$; For each scale, we train MDLRAM on one A40 GPU for 100 epochs with 1,280,000 *problem instances* generated on the fly as training dataset, which can be split into 2,500 batches with batchsize of 512.

As for DGM's *problem instance*, only including customer requests, we also consider three problem scales: $n = 20, 50, 100$ customers. The expected distance among the generated depots ranges within $[0.2, 0.7]$. The coefficients in objective function Eq. (2) are specified as $\lambda = 10, \varepsilon = 10$. Training occurs for 100 epochs per scale (one A40 GPU), with 2,500 main-batches generated on the fly per epoch. ***In multivariate Gaussian distribution mode***, the main-batch size B_{main} is set as 32, and the sub-batch size B_{sub} for sampling in each distribution is selected as 128, 64, 32 for scale $20, 50, 100$ respectively. ***In exact position mode***, the main-batch size is set as 512, and no sub-batch sampling occurs.

3.1 Results Analysis for Critic Model MDLRAM

As critic model, MDLRAM is expected to instantly provide optimized LRP solution for serving customers based on generated depots, in batch-wise manner. Therefore, we test it on both ***Synthetic Dataset*** to highlight the instant batch-wise solving ability, and ***Real-world Benchmark Dataset*** to evaluate its generalization performance by comparing with SOTA results, which, so far, are all achieved by specifically designed heuristic methods.

Table 1. Batch-wise Testing Performance of MDLRAM on Synthetic Dataset. ("Ttl.C.: total cost in Eq.(1)"; "Len.: total length"; "Dpt.C. (Nb.): depot opening cost (opened depot number)"; "Veh.C. (Nb.): vehicle setup cost (used vehicle numbers)"; "Dpt.P.: penalty for exceeding depot desired maximum supply")

	n	m	Mtd.	Ttl.C.	Len.	Dpt.C.	(Nb.)	Veh.C.	(Nb.)	Dpt.P.	Inf.T.
Scale 20	20	3	MDLR(S)	**13.17***	5.61	6.26	(2.14)	1.23	(4.08)	0.07	0.16s
	20	3	MDLR(G)	13.81	5.86	6.45	(2.15)	1.25	(4.17)	0.24	0.10s
	20	3	ALNS	13.23	6.56	5.40	(2.01)	1.16	(3.88)	0.10	4.12s
	20	3	GA	13.72	6.87	5.61	(2.01)	1.17	(3.90)	0.07	6.63s
	20	3	TS	16.26	9.13	5.79	(2.00)	1.16	(3.88)	0.18	0.16s
Scale 50	50	6	MDLR(S)	**20.85***	8.65	10.13	(3.52)	2.05	(6.82)	0.03	0.43s
	50	6	MDLR(G)	22.05	9.20	10.66	(3.55)	2.09	(6.97)	0.10	0.31s
	50	6	ALNS	25.10	14.71	8.33	(3.03)	2.05	(6.84)	0.01	47.35s
	50	6	GA	29.65	18.53	9.03	(3.12)	2.09	(6.98)	0	12.69s
	50	6	TS	33.26	22.26	8.91	(3.16)	2.10	(6.99)	0	1.57s
Scale 100	100	9	MDLR(S)	**86.88***	14.79	68.77	(5.00)	3.32	(11.06)	0	1.04s
	100	9	MDLR(G)	94.94	16.29	75.00	(5.00)	3.37	(11.23)	0.28	0.55s
	100	9	ALNS	89.75	29.58	56.82	(3.97)	3.21	(10.69)	0.14	412.83s
	100	9	GA	108.24	43.79	61.03	(4.14)	3.28	(10.92)	0.14	19.44s
	100	9	TS	104.39	41.80	59.31	(4.08)	3.27	(10.89)	0.01	16.44s

(i) Testing on Synthetic dataset: For each problem scale, synthetic datasets of 10,000 instances are generated, identical to the training setup, capable of facilitating batch-wise testing. Due to a lack of comparable DRL-based LRP methods, we use classic heuristic algorithms, tuned to report their best performance under standard LRP constraints, even though heuristic methods are not suitable for batch-wise usage considering its solving manner and unstable inference time. Table 1 compares total costs, detailed component costs, and inference times averaged across all instances per scale. Specifically, MDLRAM results employ two decoding strategies: **(a) Greedy test:** Selects the highest-probability action at each step, deriving one deterministic solution per instance. **(b) Sampling test:** Generates 1,280 random solutions per instance by stochastic action selection, choosing the lowest-cost solution. The results indicate that MDLRAM's sampling test consistently achieves the lowest total cost across all scales. The greedy test also outperforms heuristic methods at larger scales $n = 50, 100$ with comparable results at scale $n = 20$. Notably, MDLRAM maintains inference times within 1 s for all scales, significantly outperforming heuristic methods whose runtime dramatically increases with scale.

(ii) Testing on Real-world dataset: To further assess the MDLRAM's generalization performance on real-world problem instances, We conduct individual comparison on instances from four widely-used real-world datasets: Prodhon [15], Acka [2], Tuzun [22], and Barreto [4], comparing results against best-known solutions (BKS) and the SOTA results derived by existing methods. As detailed in Table 2, these instances diverse significantly, with customer amount n ranging from 12 to 100 and depot candidate amount m from 2 to 20. MDLRAM consistently provides solutions close to BKS and existing methods while significantly reducing inference time. This speed advantage becomes more pronounced with larger scales, validating MDLRAM's efficiency, robustness and suitability as a critic model for depot generation tasks.

Table 2. MDLRAM's Performance on cases from Real-world Dataset ("*" represents an optimal solution; "Ttl.C.": Total Cost; "Inf.T.": Inference Time).

				GRASP [15]			**MDLRAM (ours)**		
Case name	n	m	BKS	Ttl C.	Gap	Inf.T.	Ttl C.	Gap	Inf.T.
P111122	100	20	14492	15269.0	5.36%	40.7s	15554.5	7.33%	0.75s
P111222	100	20	14323	14822.9	3.49%	36.2s	15154.43	5.80%	0.74s
P111112	100	10	14676.8	15252.5	3.92%	32.4s	15516.6	5.72%	0.64s
P113122	100	20	12463	12729.4	2.14%	36.0s	13081.49	4.96%	0.72s
P111212	100	10	13948	14235.4	2.06%	27.6s	14529.15	4.17%	0.58s
50-5-1a	50	5	90111	90632	0.57%	1.8s	95072	5.51%	0.25s
50-5-2b	50	5	67340	68042	1.04%	2.5s	70941	5.35%	0.23s
50-5-3b	50	5	61830	61890	0.10%	2.0s	66258	7.16%	0.23s
G67-21-5	21	5	424.9*	429.6	1.1%	0.2s	425.66	0.18%	0.12s
20-5-1a	20	5	54793*	55021	0.42%	0.2s	57005	4.04%	0.11s
20-5-2a	20	5	48908*	48908	0.00%	0.1s	50029	2.29%	0.12s
20-5-2b	20	5	37542*	37542	0.00%	0.2s	38893	3.60%	0.11s
				HBP [2]			**MDLRAM (ours)**		
Case name	n	m	BKS	Ttl C.	Gap	Inf.T.	Ttl C.	Gap	Inf.T.
P183-12-2	12	2	204*	204.0	0.00%	0.2s	204.00	0.00%	0.07s
P183-55-15	55	15	1112.06	1121.8	0.88%	10800s	1151.91	3.58%	0.38s
P183-85-7	85	7	1622.5	1668.2	2.82%	10813.8s	1676.89	3.35%	0.46s
				B&C [5]			**MDLRAM (ours)**		
Case name	n	m	BKS	Ttl C.	Gap	Inf.T.	Ttl C.	Gap	Inf.T.
30-5a-1	30	5	819.52*	819.60	0.00%	50.22s	849.33	3.64%	0.143s
30-5a-2	30	5	821.50*	823.50	0.00%	53.89s	884.29	7.64%	0.144s
40-5a-1	40	5	928.10*	928.20	0.00%	305.25s	988.80	6.54%	0.189s
40-5b-1	40	5	1052.04*	1052.07	0.00%	3694.45s	1107.54	5.28%	0.195s

(iii) Discussion on Benchmarking for LRP Configuration: It is worth noting that existing DRL methods addressing multi-depot routing scenarios [3,16] often neglect depot-related cost considerations to simplify the problem configuration, treating these problems as Multi-Depot VRP (MDVRP), rather than genuine Location Routing Problems (LRP) where depot-placement decisions can carry significant implications on route planning as well as total cost. Such simplifications, alongside a lack of validation on established real-world LRP benchmarks, diminish the suitability as robust baselines for direct comparison. In contrast, our proposed framework explicitly integrates depot opening and operational costs into the downstream route planning, and validates on well-established real-world LRP benchmarks, where the MDLRAM consistently achieve near-optimal, and in several cases, even optimal performance, highlighting the competitive advantage for providing instant and near-optimal batch-wise solutions for the depot generation process, which is the key target of MDLRAM as an efficient and reliable sub-solver for DGM's batch-wise training.

3.2 Results Analysis for DGM

For DGM evaluation, given a set of customer requests without predefined depots, it is expected to generate a depot set $\mathcal{D}$ which can lead to lower total cost than the randomly attempted depots. Thus, we compare following three strategies: (i) DGM-Exact mode; (ii) DGM-Gaussian mode; (iii) Randomly attempting in batches. The quality of the depot set is evaluated by $L_{\text{Gen}}(\mathcal{D})$ in Eq. (2). ***For each scale***, testing involves 8,000 problem instances, each only comprising a group of customer requests. The evaluation includes two performance metrics: **(i) Average test:** Each method generates 512 depot sets per instance; the average cost of these 512 sets is calculated for each instance. Final results average these costs across all 8,000 instances. **(ii) Sampling test:** Similarly, 512 depot sets per instance are generated, but only the best (lowest cost) depot set per instance is considered as the result. Final results average these best-case costs across all 8,000 instances.

Table 3. Comparison of solution depot set generated by different strategies. ("Ttl.C.: total cost in Eq.(2)"; "Len.: total length"; "Ex.P. or Ls.P.: penalty for exceeding Upper bound or Lower bound of the distance among the depots";)

	n	Method	Ttl C.	Len.	Ex. P.	Ls. P.	(Dpt C.	Veh C.	Dpt P.)
Average	20	Rdm.	7.671	5.857	1.121	0.693	(6.448	0.246	1.249)
		DGM-G.	**5.724**	**5.370**	0.354	0.000	(6.427	0.234	1.252)
		DGM-E.	**5.096***	**5.089***	0.005	0.002	(6.449	0.242	1.251)
	50	Rdm.	18.261	9.192	5.600	3.469	(10.650	0.101	2.090)
		DGM-G.	**10.921**	**8.660**	0.339	1.922	(10.649	0.090	2.092)
		DGM-E.	**8.531***	**8.496***	0.018	0.017	(10.637	0.082	2.083)
	100	Rdm.	38.040	16.281	13.434	8.325	(74.956	0.281	3.371)
		DGM-G.	**23.305**	**14.616**	0.896	7.793	(74.933	0.271	3.372)
		DGM-E.	**15.394***	**13.690***	0.026	1.678	(74.940	0.274	3.385)
Sample	20	Rdm.	**5.045***	**5.021***	0.014	0.010	(6.465	0.270	1.247)
		DGM-G.	5.137	5.115	0.022	0.000	(6.428	0.249	1.249)
		DGM-E.	5.096	5.089	0.005	0.002	(6.449	0.242	1.251)
	50	Rdm.	10.684	8.570	0.610	1.504	(10.679	0.083	2.086)
		DGM-G.	**8.769**	**8.259***	0.135	0.375	(10.676	0.085	2.086)
		DGM-E.	**8.531***	**8.496**	0.018	0.017	(10.637	0.082	2.083)
	100	Rdm.	24.056	14.910	2.491	6.655	(74.928	0.281	3.369)
		DGM-G.	**19.538**	**13.828**	0.667	5.043	(74.942	0.250	3.367)
		DGM-E.	**15.394***	**13.690***	0.026	1.678	(74.940	0.274	3.385)

As shown in Table 3, observations reveal that: **(1) Average test (*average level*):** Both DGM's two modes can identify superior solution depot set than randomly attempting, with Exact mode surpassing Gaussian mode; **(2) Sampling test (*best level*):** DGM-Gaussian mode generally yields better depot sets than "randomly attempting" within equivalent sampling efforts and timeframe. As for the DGM's Exact mode, its specific solution set consistently outperform those of the Gaussian mode, whereas the Gaussian mode can offer more flexibility with its sampling ability. Only one exception is observed in small scale ($n = 20$), where "randomly attempting" achieves better solution set without the aid of DGM, which may not be replicable at larger scales.

4 Conclusion and Future Work

In this study, we proposed a generative DRL framework for proactive depot generation without predefined candidates. Based on customer requests, the DGM generates depots while MDLRAM plans efficient routes, offering enhanced flexibility and adaptability, particularly valuable in scenarios demanding rapid depot setup and dynamic adjustments. Currently, DGM generates a fixed number of depots during training. Future work will explore adaptive depot-number generation based on customer demand variability, incorporate additional real-world

constraints (e.g., restricted zones), and extend the modular framework for simultaneous optimization across multiple routing tasks, thereby broadening applicability in complex logistics scenarios.

Acknowledgments. This research is supported by the National Research Foundation, Singapore, under the NRF Medium Sized Centre scheme (CARTIN). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

References

1. Manzour-al Ajdad, S., Torabi, S.A., Salhi, S.: A hierarchical algorithm for the planar single-facility location routing problem. Comput. Operat. Res. **39**(2), 461–470 (2012)
2. Akca, Z., Berger, R., Ralphs, T.: A branch-and-price algorithm for combined location and routing problems under capacity restrictions. In: Operations research and cyber-infrastructure, pp. 309–330. Springer (2009). https://doi.org/10.1007/978-0-387-88843-9_16
3. Arishi, A., Krishnan, K.: A multi-agent deep reinforcement learning approach for solving the multi-depot vehicle routing problem. J. Manag. Analy. **10**(3), 493–515 (2023)
4. Barreto, S., Ferreira, C., Paixao, J., Santos, B.S.: Using clustering analysis in a capacitated location-routing problem. Eur. J. Oper. Res. **179**(3), 968–977 (2007)
5. Belenguer, J.M., Benavent, E., Prins, C., Prodhon, C., Calvo, R.W.: A branch-and-cut method for the capacitated location-routing problem. Comput. Operat. Res. **38**(6), 931–941 (2011)
6. Contardo, C., Cordeau, J.F., Gendron, B.: An exact algorithm based on cut-and-column generation for the capacitated location-routing problem. INFORMS J. Comput. **26**(1), 88–102 (2014)
7. Danach, K., Gelareh, S., Monemi, R.N.: The capacitated single-allocation p-hub location routing problem: a lagrangian relaxation and a hyper-heuristic approach. EURO J. Trans. Logist. **8**(5), 597–631 (2019)
8. Ghasemi, P., Hemmaty, H., Pourghader Chobar, A., Heidari, M.R., Keramati, M.: A multi-objective and multi-level model for location-routing problem in the supply chain based on the customer's time window. J. Appli. Res. Indust. Eng. (2022)
9. Kool, W., van Hoof, H., Welling, M.: Attention, learn to solve routing problems! In: International Conference on Learning Representations (2019). https://openreview.net/forum?id=ByxBFsRqYm
10. Lin, Z., et al.: Cross-problem learning for solving vehicle routing problems. In: The 33rd International Joint Conference on Artificial Intelligence (IJCAI 2024) (2024)
11. Ma, Y., et al.: Learning to iteratively solve routing problems with dual-aspect collaborative transformer. Adv. Neural. Inf. Process. Syst. **34**, 11096–11107 (2021)
12. Nagy, G., Salhi, S.: Location-routing: Issues, models and methods. Eur. J. Oper. Res. **177**(2), 649–672 (2007)
13. Nguyen, V.P., Prins, C., Prodhon, C.: Solving the two-echelon location routing problem by a grasp reinforced by a learning process and path relinking. Eur. J. Oper. Res. **216**(1), 113–126 (2012)

14. Pourghader Chobar, A., Sabk Ara, M., Moradi Pirbalouti, S., Khadem, M., Bahrami, S.: A multi-objective location-routing problem model for multi-device relief logistics under uncertainty using meta-heuristic algorithm. J. Appli. Res. Indust. Eng. **9**(3), 354–373 (2022)
15. Prins, C., Prodhon, C., Calvo, R.W.: Solving the capacitated location-routing problem by a grasp complemented by a learning process and a path relinking. 4OR **4**(3), 221–238 (2006)
16. Rabbanian, S.S., Wang, H., Knapp, G.M.: Analysis of decoding strategies for transformer-based solution of multi-depot vehicle routing problems. In: IISE Annual Conference and Expo, vol. 2023, p. 3575. IISE (2023)
17. Rath, S., Gutjahr, W.J.: A math-heuristic for the warehouse location-routing problem in disaster relief. Comput. Operat. Res. **42**, 25–39 (2014)
18. Salhi, S., Nagy, G.: Local improvement in planar facility location using vehicle routing. Ann. Oper. Res. **167**(1), 287–296 (2009)
19. Salhi, S., Rand, G.K.: The effect of ignoring routes when locating depots. Eur. J. Oper. Res. **39**(2), 150–156 (1989)
20. Schwardt, M., Dethloff, J.: Solving a continuous location-routing problem by use of a self-organizing map. Inter. J. Phys. Distribut. Logist. Manag. **35**(6), 390–408 (2005)
21. Schwardt, M., Fischer, K.: Combined location-routing problems–a neural network approach. Ann. Oper. Res. **167**, 253–269 (2009)
22. Tuzun, D., Burke, L.I.: A two-phase tabu search approach to the location routing problem. Eur. J. Oper. Res. **116**(1), 87–99 (1999)
23. Wang, S., Zhou, J., Liang, H., Wang, Z., Su, C., Li, X.: A new approach for solving location routing problems with deep reinforcement learning of emergency medical facility. In: Proceedings of the 8th ACM SIGSPATIAL International Workshop on Security Response using GIS, pp. 50–53 (2023)
24. Ye, H., Wang, J., Liang, H., Cao, Z., Li, Y., Li, F.: Glop: learning global partition and local construction for solving large-scale routing problems in real-time. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 20284–20292 (2024)
25. Zhou, J., et al.: Mvmoe: multi-task vehicle routing solver with mixture-of-experts. In: International Conference on Machine Learning (ICML) (2024)



Learning Joint General and Specific Representation with Masked Auto-Encoder for Radiology Report Generation

Tongfei Shen, Zixuan Teng, and Dong Zhang(✉)

School of Computer Science and Technology, Soochow University, Suzhou, China
dzhang@suda.edu.cn

Abstract. As an essential cross-modal task, *radiology report generation* (RRG) has drawn much attention recently. The existing approaches normally rely on large-scale annotated data (coupled image-report pairs) to generate satisfying radiology reports. As we know, it is time-consuming and labor-intensive to build these large-scale annotated datasets and the data demand of the growing model increases simultaneously. Thanks to the global understanding and proper unlabeled exploitation of *masked auto-encoder* (MAE), we attempt to extend MAE to learn both global and specific features for RRG, probably with unlabeled image information. In this way, we can alleviate the annotation burden and leverage less labeled data to achieve competitive performance. Therefore, we propose a novel approach, namely *joint MAE* (JMAE), to simultaneously mine the global and specific representation for RRG. Moreover, we also intend to inject potential unlabeled Chest X-ray images to augment our joint model. Extensive experiments on two widely used datasets demonstrate that our approach can outperform many representative baselines, and detailed analysis further shows the feasibility and effectiveness of our approach.

Keywords: Radiology report generation · Masked auto-encoder · Joint learning · General and specific representation

1 Introduction

Radiology report generation simulates the radiologist writing a chest diagnostic report with a clinical radiograph [1,2]. Radiologists observe abnormal areas in a chest radiograph to analyze the type of disease and write descriptions to generate a diagnostic report. As health care concerns rise, more chest radiographs need radiologists to be written. Statistically, the growth rate of medical radiology imaging physicians is one-tenth of the growth rate of patient demand [3,4]. However, there exist differences in the distribution of health care resources by region. Due to that, not all hospitals have enough radiologists available to carry out the heavy workload in time [5,6]. To this end, RRG in medical research area becomes more imperative to explore. Specifically, they can find abnormalities more quickly, avoid missing diagnoses and even correct misdiagnoses [7].

T. Shen and Z. Teng—Equal contribution.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 154–165, 2026.
https://doi.org/10.1007/978-3-032-04555-3_13

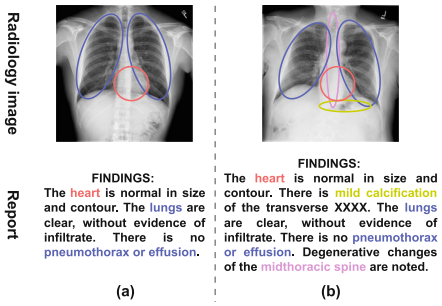


Fig. 1. Two examples of the clinical radiograph with circled areas and the corresponding diagnostic report with colored terminologies.

Despite recent achievements in radiology report generation, there still exist several apparent issues: **1)** previous studies normally rely on large-scale annotated image-report pairs and adopt supervised approaches to learn limited specific visual representations for several special regions and yield the possible corresponding words in the report. If there are only a few labeled samples, they may lead to a lack of general feature learning for some correlated images and reports. For example, in Fig. 1, the samples (a) and (b) share the same symptom, such as the region with the red circles in the images referring to the same phrase "The heart is normal in size and contour" in the reports. These general states are usually omitted by the existing approaches if with limited annotated data [8]. Therefore, we believe that it is important for our model to understand general features. **2)** The large-scale labeled image-report pairs require a substantial amount of labor and huge costs. This undoubtedly affects the enthusiasm for the development of RRG [9]. Therefore, we believe that it is necessary for us to properly leverage unlabeled data in our model.

To address the above issues, in this paper, we only depend on a few labeled samples and propose a novel generation approach, namely *joint masked auto-encoder* (JMAE), thanks to the great ability of generalization and unlabeled exploitation in *masked auto-encoder* (MAE). Specifically, JMAE consists of a self-supervised module with MAE and a supervised module with a few annotated image-report pairs by joint learning. On the one hand, the former module reconstructs all images (a few with reports and a large number without reports potentially) by self-supervised learning to capture the general information for RRG. On the other hand, the latter module resorts to labeled reports to jointly augment the latent visual representation by capturing the specific information in the reports. Overall, the contributions of this paper can be summarized as:

• We only depend on a small number of image-report pairs and propose a novel radiology report generation (RRG) approach, namely joint masked auto-encoder (JMAE). This approach can not only capture the general features, but also attend to the specific features simultaneously for RRG.

• We attempt to utilize potential large-scale unlabeled radiology images to further mine the latent information by bridging them and several labeled pairs via joint learning for better generation.

• Extensive experiments demonstrate that our proposed JMAE performs much better than many competitive baselines.

2 Related Work

Although many studies in RRG are indeed derived from image-based text generation [11] in the early stage, the generated text is not appropriate for the expression of specialized diseases in the medical report. Therefore, we mainly overview the recent approaches for RRG from supervised and unsupervised learning, as well as the MAE-based approaches most related to our model.

Supervised RRG. With the emergence of large-scale image-based and deep models (e.g., imagenet), supervised training has achieved significant performance. Therefore, it is natural to improve supervised learning approaches from different perspectives, such as visual features extractor [12], text generator [?], [13], alignment mechanism between two modalities [14], external resources or knowledge [15] and recent large language models (LLMs) based approaches [16,17].

Dissimilar to the above studies, we only leverage a few labeled image-report pairs yet utilize a large number of unlabeled images. To this end, we also propose a novel joint learning approach to simultaneously mine the proper information in both labeled and unlabeled data for RRG.

Unsupervised RRG. The data limit issue in the low resource area has existed for a long time, but there are not many effective approaches for RRG. The approach proposed in [20] explores and distills posterior and prior knowledge by first examining the abnormal regions, and then relying on the prior medical knowledge and prior working experience to write accurate radiology reports. Although their approach can also perform unsupervised or semi-supervised training, it also requires a relatively large percentage of labeled data to obtain better performance.

Unlike the above studies, we employ a combining paradigm with fully-supervised and self-supervised learning. To this end, we only use a few annotated report samples and jointly utilize MAE to mine both general and specific information for RRG.

Masked Auto-encoders. Due to the impressive representation learning ability of MAE [21], it has attracted unprecedented attention. To our best knowledge, the most related work [22] to our study uses large-scale medical images and texts and perform vision-language pre-training with MAE to visual question answering, image-text classification and image-caption retrieval.

Different from the above studies, we focus on radiology report generation with a small number of labeled samples. Besides, we employ a joint training paradigm with MAE to learn joint general and specific representation for RRG, instead of a pipeline structure as the previous vision-language pre-training.

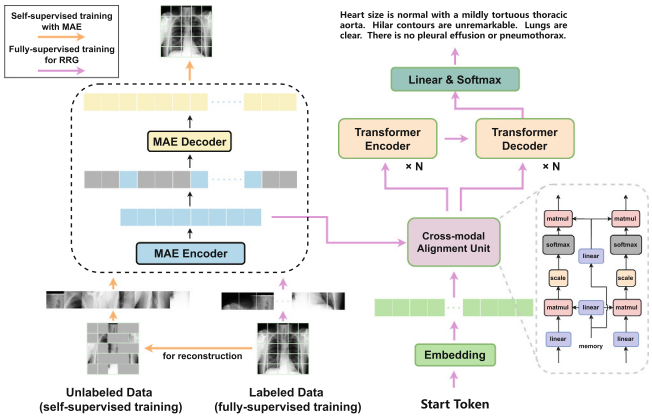


Fig. 2. The overall architecture of our JMAE, which consists of a self-supervised and a supervised module for RRG.

3 Methodology

To better understand the RRG task, we first formulate this task and then present our approach JMAE. The overall architecture of our JMAE is illustrated in Fig. 2.

3.1 Task Definition

In RRG, a report in text format is generated after the radiology image is inputted. Thus, RRG is a cross-modal generation task, which is similar to but different from image caption, text-to-image generation or other image-text-related tasks [?,?]. Formally, given a radiology image $X = \{x_1, x_2, \cdots, x_m\}$, we can obtain the target sentence $Y = \{y_1, y_2, \cdots, y_n\}$ by generating, where m, n represent the number of representation for patches from the original image and the number of words in the generated sentence respectively.

The goal of RRG aims to maximize the conditional probability as a usual generation task.

$$p(Y|X) = \prod_{i=1}^{m} p(y_i|y_1, \cdots, y_{i-1}, X) \tag{1}$$

3.2 Joint Masked Auto-encoder

Our proposed JMAE consists of a self-supervised module with MAE and a supervised module with a few annotated image-report pairs by joint learning.

First, we introduce the self-supervised module with MAE in detail, which can capture the general features. Then, we comprehensively present the supervised module, which can obtain the specific features for RRG. Finally, we describe the joint training procedure for our approach.

Self-supervised Module with MAE. For extending MAE, there are two reasons: **1)** MAE can reconstruct the whole image only relying on the sub-set of all patches. This is very similar to the ability of a doctor, which can restore the entire image based on several key areas of the chest X-ray image. Moreover, the doctor can judge the general disease of the human body (including some potential accompanying symptoms in other parts) based on these key areas. Therefore, MAE can learn the general features to facilitate generating medical reports. **2)** To alleviate the limited data of annotated image-report pairs in RRG, we seek to add the radiology images without annotation to help improve the model's performance. It is fortunate that MAE owns excellent generalization ability for any radiology images, including those without report annotation. Moreover, it can be easy to joint train with a supervised generation branch.

In the encoder of this module, we employ Vision Transformer (ViT). Given an image $X \in \mathbb{R}^{H \times W \times C}$, it is first segmented into patches $\{p_1, p_2, \cdots, p_N\}$, where $H \times W$ represents the image resolution, C denotes the number of channels, $p_n \in \mathbb{R}^{P^2 \times C}$ and $P \times P$ means the patch resolution. Note that we adopt the random sampling (e.g., 75%) strategy in [21], i.e., our encoder only accepts a small subset of image patches (e.g., K observed patches) without mask tokens.

Then, these patches are flattened and linearly projected into patch embeddings through a linear transformation $E \in \mathbb{R}^{P^2 C \times D}$ with a special learnable token embedding $p_I \in \mathbb{R}^D$ following [22]. Therefore, the input representations are obtained via summing up the patch embeddings, learnable 1D position embeddings $E_{pos} \in \mathbb{R}^{(N+1) \times D}$.

Finally, we obtain the contextual image representation: $H = \{h_1, h_2, \cdots, h_K\}$. To achieve the joint learning with the next introduced supervised module, we make the two main components share this encoder. But, for supervised learning to generate reports, the input image does not adopt masking, i.e., the final contextual image representation by this encoder is $H = \{h_1, h_2, \cdots, h_N\}$.

In the decoder of this module, besides the unremoved patches, we input to the decoder with the full set of tokens $H = \{h_1, h_2, \cdots, h_K, [MASK], \cdots, [MASK]\}$, consisting of encoded visible patches and mask tokens in [21]. Note that it is only used during the self-supervised branch to perform the image reconstruction task (i.e., it is not used for RRG). Therefore, the decoder architecture can be flexibly designed and won't bear much learning burden. While, to recover the complete image, our decoder can also augment the visual latent representations with global observation by back propagation. The above process can be formulated as

$$\hat{X} = f^d_{MAE}(f^e_{MAE}(X_{mask})) \tag{2}$$

where f^e_{MAE} and f^d_{MAE} refer to the encoder and decoder for reconstruction, X_{mask} refers to the masked image. It is illustrated in the left part of Fig. 2.

Supervised Module for RRG. For the supervised module, we aim to obtain specific features jointly for RRG. We utilize the cross-modal alignment unit (CAU) in [18] as illustrated in the right part of Fig. 2.

In CAU, we align image features and text features in the same representation space by calculating their distance with the memory matrix respectively. Given image features H extracted from MAE encoder and text embeddings

$E_Y \in \mathbb{R}^{D_Y \times d_Y}$, they are first converted through linear transformation, where D_Y denotes the numbers of embedding vectors, d_Y denotes the dimension of embedding vectors.

Then, the distances of two representations are calculated respectively with memory matrix $M \in \mathbb{R}^{D_M \times d_M}$ which is initialised randomly and converted by linear projection, where D_M denotes the numbers of memory vectors, d_M denotes the dimension of memory vectors.

Finally, we obtain cross-modal alignment representations O_e for followed encoder and O_d for followed decoder by weighting the transferred memory matrix by the normalized distances. The above process can be formulated as

$$O_e = \text{softmax}(\frac{HM^T}{\sqrt{d_M}})M,\ O_d = \text{softmax}(\frac{E_Y M^T}{\sqrt{d_M}})M \tag{3}$$

In the encoder and decoder of supervised module, we employ the vanilla transformer. Given cross-modal alignment representations O_e and O_d, the current word y_t is generated by $y_t = f^d_{Gen}(f^e_{Gen}(O_e), O_d)$, where f^e_{Gen} and f^d_{Gen} refer to the encoder and decoder in generation.

Joint Training. To jointly learn the general and specific information for RRG, we combine these two modules. To this end, the goal of our RRG task becomes maximizing the two conditional probabilities simultaneously: $p(Y|X) + p(\hat{X}|X_{mask})$.

Particularly, all data are trained under the loss function computed between reconstructed and original images with mean squared error as the first term,

$$\mathcal{L}_{unlabeled} = \sum_{i=1}^{H \times W} (\hat{x_i} - x_i)^2 \tag{4}$$

where $\hat{x_i}$ refers to the reconstructed pixels and x_i refers to the original pixels.

Besides, labeled data are trained under the cross entropy loss between generated report and the ground truth as the second term,

$$\mathcal{L}_{labeled} = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{V} y_{i,j} \log p_{i,j} \tag{5}$$

where V refers to the size of the lexicon, $p_{i,j}$ represents the probability that the i-th token is predicted to be the j-th word in lexicon, $y_{i,j}$ denotes that the i-th token is j-th word in lexicon. Finally, $\mathcal{L}_{joint} = \lambda_1 \cdot \mathcal{L}_{unlabeled} + \lambda_2 \cdot \mathcal{L}_{labeled}$, where λ_1 and λ_2 represent the hyper-parameters to balance two losses.

4 Experimentation

4.1 Experimental Settings

Datasets: For the sake of proper experimentation and fait comparison with other studies, we utilized **IU X-Ray** [23] and **MIMIC-CXR** [24] as our

Table 1. The details of our used dataset.

Dataset	Split	Image	Report	Patient	Avg.Len.
IU X-Ray	Unlabeled	200k	200k	44041	51.5
	Train	2069	2069	2069	31.8
	Val	296	296	296	31.1
	Test	590	590	590	28.2
MIMIC -CXR	Unlabeled	200k	200k	44041	51.5
	Train	5k	5k	1129	52.3
	Val	1k	1k	208	51.3
	Test	1k	1k	85	66.7

datasets. **1) IU X-Ray** is a small-scale publicly available dataset from Indiana Network for Patient Care, which contains 3996 radiology reports and 8,121 associated images from the hospitals' picture archiving systems. The images and reports were de-identified automatically and then the automatic de-identification was manually verified. **2) MIMIC-CXR** is the largest publicly available radiography Chest X-ray radiography dataset with 377,110 chest x-rays associated with 227,827 imaging studies sourced from the Beth Israel Deaconess Medical Center between 2011 - 2016. And images are provided with 14 labels derived from two natural language processing tools applied to the corresponding free-text radiology reports.

Data Split: In order to reserve some unlabeled samples, we re-divided the original dataset **IU X-Ray** and **MIMIC-CXR**, as shown in Table 1. Due to the small amount of samples in **IU X-Ray**, the validation and test sets are the same as the fully-supervised studies [18], and the unlabeled data are derived from the large-scale dataset **MIMIC-CXR** in our experiments.

Following previous studies [12,28], we take the findings section in each report as the generated target rather than the impression section, and more information about each division is shown in Table 1, including the number of patients and the average length of the reports. Meanwhile, in order to validate the factual consistency of our method, we label the diagnostic reports predicted by the model with the disease tag. Then we compare these tags with the tags of ground truth and calculate the accuracy of multi-label classification (CE in Table 2).

Implementation Details: For the sake of experimental fairness, we modified the experimental settings in the reproduction of several baselines: the change of the visual extractor (from CLIP [28] to ViT-base [26]), which keeps the consistence with our model.

In the process of report generation, the beam size is set to 3 to reduce the time of generation and obtain better sentence then greedy strategy in some case. Besides, the experiments are conducted on a single Geforce RTX 3090 machine with and hyper-parameters is selected by evaluating with validation split on two datasets.

Evaluation Metrics: Following [17], we use these common NLG metrics, including **BLEU**(BL), **METEOR**(MTR), **ROUGE-L**(RG-L). Besides, the

Table 2. The performance comparisons between previous studies and our JMAE on IU X-Ray and MIMIC-CXR. JMAE denotes our approach without unlabeled images, and JMAE+Unlabel means our approach with unlabeled images.

Dataset	Model	NLG Metrics						CE Metrics		
		BL-1	BL-2	BL-3	BL-4	MTR	RG-L	P	R	F1
	R2Gen [12]	0.434	0.280	0.205	0.160	0.181	0.359	–	–	–
	CMN [18]	0.242	0.147	0.106	0.081	0.120	0.312	–	–	–
	PD [28]	0.435	0.276	0.204	0.156	0.162	0.342	–	–	–
	PromptMRG [16]	0.454	0.297	0.212	0.151	0.159	0.332	–	–	–
	JMAE (Ours)	0.414	0.266	0.191	0.147	0.170	0.352	–	–	–
	JMAE + Unlabel (Ours)	**0.462**	**0.299**	**0.216**	**0.165**	**0.183**	**0.357**	–	–	–
MIMIC -CXR	R2Gen [12]	0.223	0.137	0.089	0.061	0.092	0.242	0.043	0.017	0.024
	CMN [18]	0.289	0.180	0.117	0.081	0.117	0.250	0.326	0.239	0.276
	PD [28]	0.300	0.176	0.106	0.079	0.119	0.260	0.342	0.228	0.274
	PromptMRG [16]	0.298	0.181	0.118	0.082	0.119	0.266	0.405	0.276	0.328
	JMAE (Ours)	0.300	**0.193**	**0.135**	**0.100**	**0.120**	**0.294**	**0.456**	**0.308**	**0.368**
	JMAE+Unlabel (Ours)	**0.327**	0.181	0.109	0.072	0.119	0.238	0.358	0.308	0.332

clinical efficacy (CE) metrics are adopted, which contains the precision, recall and F1-score of the diseases labeled by the CheXpert [27] in the generated report, according to the true categories within 14 diseases.

Baselines. To fairly compare our proposed approach, we set many representative baselines on two datasets. Following previous studies [12,28], we compare the most representative, competitive and recent approaches: 1) **R2GEN** [12], 2) **CMN** [18], 3) **PD** [28] (2023), 4) **PromptMRG** [16] (2024).

4.2 Main Results

Table 2 shows the performance comparison of different approaches for RRG on two datasets. From this table, we can see that: **1)** Most baselines exhibit very unstable, i.e., perform well on some metrics but poorly on others. This is mainly because the existing approaches for RRG either rely entirely on labeled samples (which may ignore general features), or use unsupervised learning (often ignore specific features). Therefore, it is imperative to develop a new approach, which can jointly capture both general and specific information for RRG. **2)** Although our JMAE with the unlabeled image (**JMAE+Unlabel**) outperforms the pure **JMAE** in IU X-Ray dataset, it performs a little worse than **JMAE** in MIMIC-CXR. We conjecture that the reports in MIMIC-CXR are usually longer (much longer than those in IU X-Ray dataset), and simply adding unlabeled images may bring noise information. However, our approach, which can learn both general and specific features jointly even without unlabeled samples, outperforms most baselines apparently.

Table 3. The performance comparison of our JMAE with different amounts of unlabeled data in train sets and the pipeline approach on IU X-RAY and MIMIC-CXR.

Dataset	Model	Unlabeled	NLG Metrics						CE Metrics		
			BL-1	BL-2	BL-3	BL-4	MTR	RG-L	P	R	F1
IU X-Ray	CMN	0	0.242	0.147	0.106	0.081	0.120	0.312	–	–	–
	JMAE	0	0.414	0.266	0.191	0.147	0.170	0.352	–	–	–
		10k	0.404	0.260	0.186	0.140	0.165	0.341	–	–	–
		50k	0.411	0.265	0.192	0.147	0.168	0.348	–	–	–
		100k	0.452	0.287	0.202	0.150	0.175	0.357	–	–	–
		150k	**0.462**	**0.299**	**0.216**	**0.165**	**0.183**	**0.357**	–	–	–
		200k	0.446	0.283	0.199	0.148	0.176	0.349	–	–	–
	Pipeline	150k	0.252	0.151	0.108	0.083	0.116	0.312	–	–	–
MIMIC -CXR	CMN	0	0.289	0.180	0.117	0.081	0.117	0.250	0.326	0.239	0.276
	JMAE	0	0.300	**0.193**	**0.135**	**0.100**	**0.120**	**0.294**	**0.456**	**0.308**	**0.368**
		10k	0.182	0.110	0.073	0.051	0.090	0.224	0.309	0.143	0.195
		50k	0.215	0.128	0.084	0.060	0.096	0.237	0.418	0.262	0.322
		100k	**0.327**	0.181	0.109	0.072	0.119	0.238	0.358	0.308	0.332
		150k	0.251	0.151	0.099	0.070	0.102	0.249	0.391	0.256	0.309
		200k	0.227	0.136	0.091	0.065	0.096	0.244	0.402	0.246	0.305
	Pipeline	100k	0.182	0.105	0.065	0.044	0.080	0.223	0.338	0.207	0.257

4.3 Ablation Study

Impact of Self-supervised Module. To observe the impact of self-supervised module in our JMAE, we conduct a series of experiments with unlabeled data at various quantities. Under the situation where only the number-of-unlabeled-data factor varies, the results are summarized in Table 3. From this table, we can see that: **1)** Adding a small number of unlabeled samples (e.g., 10k) into our JMAE will bring about a decrease in performance. However, in general, with the increase of unlabeled samples, the performance of our JAME shows a trend of first high and then low on both datasets. Especially, on IU X-Ray, with 150K unlabeled radiology images, our JMAE can reach the best performance. This suggests that our self-supervised module could effectively mine the knowledge from unlabeled data for RRG. **2)** On MIMIC-CXR dataset, in terms of the NLG metrics, our JMAE adding unlabeled samples does not seem to bring about performance improvement. What's worse, our JMAE with unlabeled samples performs poorer than CMN. This is mainly because the generation of long texts requires a high amount of input information, blindly adding unlabeled chest images will only cause a learning burden (reports that need to be generated on MIMIC-CXR are generally longer). But for the CE metrics, even if unlabeled samples do not help disease classification, they still yield better performance than CMN. This indicates that our self-supervised module also facilitates disease inference for radiology reports.

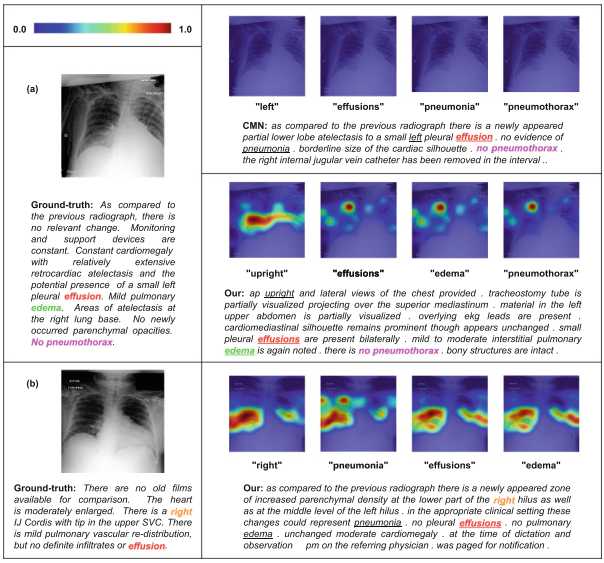


Fig. 3. Visualizations of image-text mappings between particular regions (indicated by colored weights) of two radiology images and words from their reports generated by CMN and our JMAE, respectively.

Impact of Joint Training. To verify the effectiveness of our joint training paradigm, we also implement the pipeline schema for our self-supervised and fully-supervised modules, as shown in Table 3. From this table, we can see that our JMAE with joint training (regardless of adding unlabeled data or not) indeed performs much better than with pipeline training. This shows that it is necessary to employ the joint paradigm to simultaneously learn the general and specific information for RRG.

4.4 Case Study

To further investigate how our approach learns reasonable image information via our **JMAE**, we provide a case study on the generated reports from the clinical images in MIMIC-CXR by **CMN** (easily access to the attention map for better analysis compared to other baselines) and our **JMAE**. Figure 3 visualizes the attention on the original image with regards to keywords, and colorizes the words in generated report. In particular, we obtain the attention scores of each word in the report from the multi-head attention with the largest difference in the last layer of the decoder.

From Fig. 3, we can observe that: **1)** In case (a), our approach can generate the essential descriptions which appear in the diagnosis report written by radiologists with the correct attention mappings. However, **CMN** only generates several words and unreliable attention mappings. For instance, in this case, the necessary medical terms (e.g., "effusion", "edema" and "No pneumothorax"), which are both included in the ground-truth, appear in the prediction of our approach, while CMN does not perform well, e.g., the key term "edema"

is missing in the predicted report. What's worse, CMN cannot attend to the effective regions in the image. On the contrary, our JMAE can track the correct regions and even position information (e.g., "upright") for better generating the corresponding words. This suggests the obvious advantages of our JMAE. **2)** Regarding cases (a) and (b) with our **JMAE** approach, they can obtain the similar mapping between image regions and generated words (e.g., "effusions", "edema"), and can also capture the characteristics of the respective images to generate unique reports (e.g., no pneumothorax for case (a), and right hilus for case (b)). This indicates that our JMAE indeed jointly learns the general and specific information for RRG. Therefore, it explains why our JMAE can produce more satisfactory reports than traditional approaches.

5 Conclusion

To further explore the similarity and dissimilarity between different images and between different reports, in this work, we focus on mining the general and specific information for RRG simultaneously. On this basis, we propose a novel joint approach called JMAE, which only rely on a small number of image-report pairs. Extensive experiments and deep analysis on two public datasets demonstrate effectiveness of our approach.

References

1. Kurisinkel, L.J., Aw, A.T., Chen, N.F.: Coherent and concise radiology report generation via context specific image representations and orthogonal sentence states. In: NAACL, pp. 246–254 (2021)
2. Serra, F.D., et al.: O'Neil, Multimodal generation of radiology reports using knowledge-grounded extraction of entities and relations. In: AACL, pp. 615–624 (2022)
3. Li, J., Li, S., Hu, Y., Tao, H.: A self-guided framework for radiology report generation. In: MICCAI, pp. 588–598 (2022)
4. Wang, L., Ning, M., Lu, D., Wei, D., Zheng, Y., Chen, J.: An inclusive task-aware framework for radiology report generation. In: MICCAI, pp. 568–577 (2022)
5. Zhang, Y., Wang, X., Xu, Z., Yu, Q., Yuille, A.L., Xu, D.: When radiology report generation meets knowledge graph. In: AAAI, pp. 12910–12917 (2020)
6. You, J., Li, D., Okumura, M., Suzuki, K.: JPG - jointly learn to align: automated disease prediction and radiology report generation. In: COLING, pp. 5989–6001 (2022)
7. Wu, X., et al.: DeltaNet: conditional medical report generation for COVID-19 diagnosis. In: COLING, pp. 2952–2961 (2022)
8. Jia, X., et al.: Radiology report generation for rare diseases via few-shot transformer. In: BIBM, pp. 1347–1352 (2021)
9. Yan, A., et al.: Weakly supervised contrastive learning for chest x-ray report generation. In: EMNLP Findings, pp. 4009–4015 (2021)
10. Cornia, M., Stefanini, M., Baraldi, L., Cucchiara, R.: Meshed-memory transformer for image captioning. In: CVPR, pp. 10575–10584 (2020)

11. Krojer, B., Adlakha, V., Vineet, V., Goyal, Y., Ponti, E.M., Reddy, S.: Image retrieval from contextual descriptions. In: ACL, pp. 3426–3440 (2022)
12. Chen, Z., Song, Y., Chang, T., Wan, X.: Generating radiology reports via memory-driven transformer. In: EMNLP, pp. 1439–1449 (2020)
13. Wang, Z., Tang, M., Wang, L., Li, X., Zhou, L.: A medical semantic-assisted transformer for radiographic report generation. In: MICCAI, pp. 655–664 (2022)
14. Qin, H., Song, Y.: Reinforced cross-modal alignment for radiology report generation. In: ACL Findings, pp. 448–458 (2022)
15. Liu, F., Wu, X., Ge, S., Fan, W., Zou, Y.: Exploring and distilling posterior and prior knowledge for radiology report generation. In: CVPR, pp. 13753–13762 (2021)
16. Jin, H., Che1, H., Lin, Y., Chen, H.: PromptMRG: diagnosis-driven prompts for medical report generation. In: AAAI, pp. 2607–2615 (2024)
17. Liu, C., Tian, Y., Chen, W., Song, Y., Zhang, Y.: Bootstrapping large language models for radiology report generation. In: AAAI, pp. 18635–18643 (2024)
18. Chen, Z., Shen, Y., Song, Y., Wan, X.: Cross-modal memory networks for radiology report generation. In: ACL, pp. 5904–5914 (2021)
19. Jing, B., Xie, P., Xing, E.P.: On the automatic generation of medical imaging reports. In: ACL, pp. 2577–2586 (2018)
20. Liu, F., You, C., Wu, X., Ge, S., Wang, S., Sun, X.: Auto-encoding knowledge graph for unsupervised medical report generation. In: NeurIPS, pp. 16266–16279 (2021)
21. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.B.: Masked autoencoders are scalable vision learners. In: CVPR, pp. 15979–15988 (2022)
22. Chen, Z., et al.: Multi-modal masked autoencoders for medical vision-and-language pre-training. In: MICCAI, pp. 679–689 (2022)
23. Demner-Fushman, D., et al.: Preparing a collection of radiology examinations for distribution and retrieval. J. Am. Medical Informatics Assoc. 304–310 (2016)
24. Johnson, A.E.W., et al.: MIMIC-CXR: A large publicly available database of labeled chest radiographs, CoRR (2019)
25. Rennie, S.J., Marcheret, E., Mroueh, Y., Ross, J., Goel, V.: Self-critical sequence training for image captioning. In: CVPR, pp. 1179–1195 (2017)
26. Dosovitskiy, A., et al.: An image is worth 16×16 words: tansformers for image recognition at scale. In: ICLR (2021)
27. Irvin, J., et al.: CheXpert: a large chest radiograph dataset with uncertainty labels and expert comparison. In: AAAI, pp. 590–597 (2019)
28. Kale, K., Bhattacharyya, P., Jadhav, K.: Replace and report: NLP assisted radiology report generation. In: EMNLP Findings, pp. 10731–10742 (2023)



Process Adaptive Learning for Visual-Language Navigation

Chaoqi Gao, Boyuan Zhang, and Yahong Han(✉)

College of Intelligence and Computing, Tianjin University, Tianjin 300354, China
{gaochaoqi,yahong}@tju.edu.cn

Abstract. Visual-language navigation (VLN) requires the agent to understand language instructions and navigate in unseen environments. The agent needs to align language instructions with real-time and historical features to achieve goal grounding. Existing methods struggle to align instructions with the environment and rely on rigid navigation processes, limiting performance in unseen environments. This stems from the disconnect between instructions and navigation processes, as well as the weak correlation between the endpoint and instructions, which interferes with navigation. Thus, we propose a Process-Adaptive cRoss-modal Transformer (PART), which dynamically adjusts the navigation history based on instructions and links real-time action prediction with memory reasoning. This approach enhances the alignment between instructions and environments while mitigating the adverse effects of overlapping trajectories. Additionally, PART uses a distance-adaptive loss function to reduce reliance on specific trajectories while reinforcing goal-directed learning, enhancing generalization in unseen environments. On the goal-oriented VLN benchmark REVERIE and the step-by-step VLN benchmark R2R, PART surpasses previous state-of-the-art methods.

Keywords: Vision and language · Embodied AI · Agent · Navigation

1 Introduction

Visual-language navigation (VLN) [1] requires agents to understand language instructions and visual information from unseen environments [9] to navigate to specific locations or identify target objects. Its practical utility has attracted considerable attention. In particular, with the use of goal-oriented instructions [6,8] such as "Go to the restaurant, give me an apple", the interaction with the agent becomes more practical and convenient. However, due to the brevity of language instructions and the complexity of unknown environments, it places higher demands on the agent's ability to ground long-range goals and generalize across different spatial contexts.

To support long-term exploration, many existing methods [2,17,18] leverage topological graphs [14] for storing navigation history and actions, while utilizing dual-scale graph transformers [4] to aid long-term action planning and

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 166–178, 2026.
https://doi.org/10.1007/978-3-032-04555-3_14

fine-grained cross-modal understanding. Additionally, to improve the agent's generalization across diverse environments, VLN has been enhanced in the following ways: **i**) Data augmentation techniques generate enriched environments [15] and pseudo-instructions [5], or by collecting more trajectory and instruction pairs [25]; **ii**) Enhance scene understanding by improving visual embeddings and incorporating 3D comprehension [17,18]; **iii**) Improve decision accuracy by processing cross-modal features prior to fusion [2,16]; and **iv**) Combine techniques such as knowledge prompting [20], map construction [3], imitation learning [1], and reinforcement learning [7] to create more effective action planning and learning strategies.

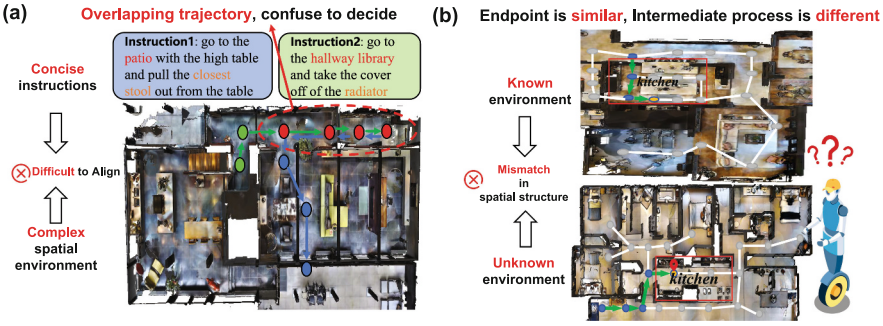


Fig. 1. Limitations of VLN tasks. (a) Visual language alignment: Concise instructions are difficult to align with complex environments, so different instructions can easily cause decision-making difficulties when trajectories overlap (b) Spatial structure mismatch: The spatial structure mismatch between the known environment and the unknown environment leads to a decrease in the effectiveness of the navigation intermediate process learned in the known environment.

However, existing methods still exhibit two main limitations. As shown in Fig. 1, first, short descriptions of instructions do not align well with the complexity of environmental spatial structures and the diversity of environmental features. Different navigation instructions can lead to overlapping portions of trajectories, and the use of identical visual memory features may negatively impact the agent's decision-making. Second, in navigation tasks, particularly goal-oriented navigation, success depends more on the target than on the process itself, particularly in unknown environments. Agents may become overly adapted to the intermediate stages of the training environment, resulting in diminished performance when locating the target in unknown environments.

To address these issues, we propose an **p**rocess-**a**daptive c**r**oss-modal **t**ransfor- mer (PART) method. This VLN model dynamically adjusts memory features based on the environment and adaptively reduces dependence on specific navigation processes, while reinforcing the learning of the target. Firstly, we propose an instruction-guided real-time feature topological mapping method, which integrates real-time panoramic features with textual instructions and maps

them into the memory feature space. This enables the agent to adaptively adjust its memory representations based on instructions and panoramic features, effectively mitigating decision confusion caused by trajectory overlap under different instructions. Secondly, to address the challenge of aligning concise language instructions with complex spatial structures, we combine real-time feature topological mapping with real-time feature reasoning and design a model architecture that integrates serial reasoning with parallel decision-making. This architecture enhances the alignment between the spatial environment and language instructions through serial reasoning based on real-time and historical memory, thereby improving the accuracy of navigation decisions. Finally, we introduce a distance-adaptive loss function, which adaptively balances the importance of the navigation process and the navigation target based on the distance between the agent and the target. This enhances the agent's ability to learn goal-directed behaviors while reducing its dependence on specific trajectories, thus improving its generalization capability in unknown environments. In summary, our main contributions are as follows:

1. We propose a process-adaptive cross-modal transformer (PART), which dynamically adjusts the navigation history memory based on instructions and links real-time feature decisions with memory reasoning.
2. We employ a distance-adaptive loss that reduces the dependency on specific trajectories while reinforcing goal learning, thereby enhancing the agent's generalization capabilities in unseen environments.
3. PART achieves state-of-the-art performance on the goal-oriented VLN benchmark REVERIE [6], with a 2.74% increase on success rate (SR). It also generalizes to the step-by-step VLN task R2R [1], increasing SR by 1.15%.

2 Proposed Method

Problem Formulation. The agent receives natural language instructions $\mathcal{I} = \{w_i\}_{i=1}^{L}$, comprising L words, and uses them to traverse a predefined undirected graph $G = \{V, E\}$ to reach the target location, where $V = \{V_i\}_{i=1}^{K}$ represents K navigable nodes, and E denotes the edges that indicate connectivity between these nodes. At step t, the agent observes the panoramic view along with its corresponding positional coordinates. The panoramic view is divided into n sub-images, denoted as $\mathcal{V}_t = \{v_i\}_{i=1}^{n}$, each paired with corresponding camera pose information $\mathcal{P}_t$. The goal of the VLN agent is to learn a policy $\pi(a_t|\mathcal{I}, \mathcal{V}_t, \mathcal{P}_t)$ to predict the action a_t. Possible actions include navigating to candidate viewpoints or stopping at the current viewpoint. In the goal-oriented VLN task REVERIE, the agent utilizes annotated object bounding boxes or an automatic object detector to extract m objects $\mathcal{O}_t = \{o_i\}_{i=1}^{m}$. The agent must identify and predict the target object from the viewpoint where it stops.

Network Architecture. Previous VLN agents [2,4,17] were designed with a parallel reasoning framework, integrating local fine-grained modules and global

coarse-grained modules. These modules dynamically fused their respective action predictions during the decision-making process. However, the direct compression of local fine-grained features into nodes within the topological graph by the coarse-grained module results in a substantial loss of detailed information. Moreover, focusing solely on compressing visual information overlooks the critical association between textual instructions and local fine-grained features.

As shown in Fig. 2, we propose a dual-scale agent framework for VLN that employs serial reasoning and parallel decision-making. This framework includes four modules: Real-Time Cross-modal Reasoning (RCR), Memory Topological Mapping (MTM), Memory Cross-modal Reasoning (MCR), and Fusion Action Prediction (FAP). Notably, RCR, MTM, and MCR perform serial reasoning on real-time and memory features.

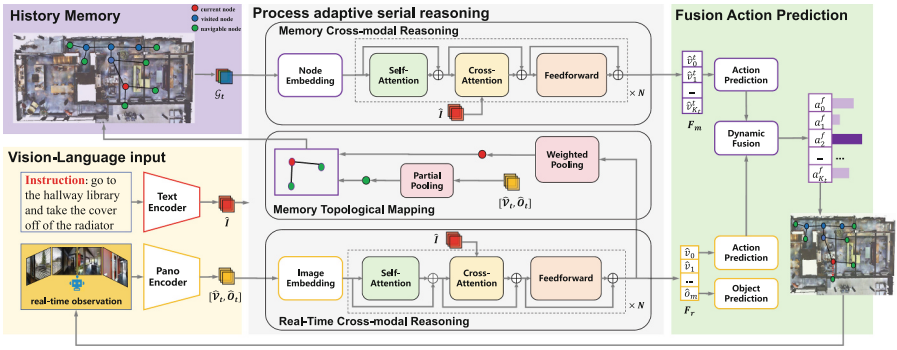


Fig. 2. Main structure of PART. Real-Time Cross-modal Reasoning performs cross-modal reasoning on real-time features based on the language instructions and then applies topological mapping to update memory features. These memory features integrate language instructions, real-time features, and real-time reasoning. Memory Cross-modal Reasoning applies cross-modal reasoning to historical features based on the language instructions. PART combines real-time reasoning features and historical reasoning features to predict actions.

2.1 Process Adaptive Serial Reasoning

Real-Time Cross-Modal Reasoning. Each word embedding in the instruction $\mathcal{I}$ is added to a positional embedding and a textual type embedding. We use the pre-trained RoBERTa [22] to obtain word feature representations $\widehat{\mathcal{I}}$ in context, as follows:

$$\widehat{\mathcal{I}} = \text{RoBERTa}(\psi(\mathcal{I}) + \psi(\mathcal{P}_{\mathcal{I}}) + \psi(\mathcal{C}_{\mathcal{I}})), \tag{1}$$

where $\psi(\cdot)$ denotes a learnable embedding layer, $\mathcal{P}_{\mathcal{I}}$ represents the positional encoding of words in the sentence [19], and $\mathcal{C}_{\mathcal{I}}$ indicates the type [10] of each sentence in the text. To model the relationship between panoramic views and target objects, we utilize a multi-layer transformer and include a special *stop*

token v_0 for stop actions. Furthermore, a positional embedding of the current location is incorporated to facilitate the fusion of position-related information with subsequent instructions:

$$[\widehat{\mathcal{V}}_t, \widehat{\mathcal{O}}_t] = \text{SelfAttn}([\mathcal{V}_t, \mathcal{O}_t]), \tag{2}$$

$$f_r = [v_0, \widehat{\mathcal{V}}_t, \widehat{\mathcal{O}}_t] + \psi(\mathcal{P}_t). \tag{3}$$

The Cross-Attention Encoder from METER [23] is then used to fuse the instructions with the real-time features, yielding the real-time environmental features F_r:

$$\begin{aligned} F_r &= \text{CrossAttn}(f_r, \widehat{\mathcal{I}}, \widehat{\mathcal{I}}) \\ &= \{\hat{v}_0, \hat{v}_1, \ldots, \hat{v}_n, \hat{o}_1, \ldots, \hat{o}_m\}. \end{aligned} \tag{4}$$

Memory Topological Mapping. The environment, represented as an undirected graph $G = \{V, E\}$, is initially unknown to the agent. Our model incrementally constructs a memory topological graph $\mathcal{G}_t = \{V_t, E_t\}$ by mapping the features encountered along the trajectory into nodes, where $\mathcal{G}_t \subset G$. The node set V_t is divided into three types: visited nodes, navigable but unexplored nodes, and the current node. The agent has access to panoramic memory features for both the current and previously visited nodes. However, navigable nodes, whose panoramas have not been observed, only possess fragmented memory features. At time step t, the current node and the newly reachable but unexplored nodes are added to the memory graph from the previous time step, $\mathcal{G}_{t-1} = \{V_{t-1}, E_{t-1}\}$, updating it to form $\mathcal{G}_t = \{V_t, E_t\}$.

At time step t, the memory features of the current node and navigable nodes in the topological graph are updated using two strategies: **i**) Employing two separate cross-attention encoders as shown in (4) to obtain fused panorama features F_{r1} for memory topological mapping, and real-time features F_{r2} for fused action prediction. **ii**) Utilizing a unified cross-attention encoder, where F_r simultaneously serves for memory topological mapping and fusion action prediction. PART adopts the latter strategy, as we posit that serial reasoning between real-time features and memory features establishes a stronger correlation between real-time predictions and memory-based predictions, thereby benefiting the subsequent parallel fused action prediction.

Specifically, for the current node, since the agent can acquire real-time environmental features, its memory representation is updated by applying weighted pooling to F_r through a simple feed-forward neural network, yielding the instruction enriched topological node representation. For navigable but unexplored nodes, since their panoramas have not been fully observed, we avoid associating them with instruction-enriched features. Instead, we aggregate their visual representations using the corresponding view embeddings from f_r observed from visited nodes. If a navigable node is observed from multiple locations, the averaged embeddings of its partial views are used as its visual representation. We denote the feature representation of a node V_i^t in the memory topological graph $\mathcal{G}_t$ as v_i^t.

By employing instruction-enriched topological mapping, the agent establishes associations between trajectory nodes and textual instructions, allowing it to adapt effectively to diverse environmental contexts.

Memory Cross-Modal Reasoning. In the memory topological graph $\mathcal{G}_t$, a special *stop* token v_0^t is added to the K_t node feature representations $\{v_1^t, v_2^t, \ldots, v_{K_t}^t\}$. Positional embeddings and temporal step embeddings are then incorporated to encode visited nodes with distinct navigation histories, thereby enhancing alignment with the instruction:

$$f_m = [v_0^t, v_1^t, \ldots, v_{K_t}^t] + \psi(\mathcal{P}_m) + \psi(T_m), \tag{5}$$

where $\mathcal{P}_m$ represents the direction and distance of other viewpoints in the topological graph relative to the current node, and T_m encodes the time step at which each visited node was last accessed, assigning a value of 0 to unexplored nodes. Using the instruction representation $\widehat{\mathcal{I}}$ from (1) and the cross-attention encoder from METER, the instruction and memory features are fused to generate the memory environment features F_m:

$$\begin{aligned} F_r &= \text{CrossAttn}(f_m, \widehat{\mathcal{I}}, \widehat{\mathcal{I}}) \\ &= \{\hat{v}_0^t, \hat{v}_1^t, \ldots, \hat{v}_{K_t}^t\}. \end{aligned} \tag{6}$$

2.2 Fusion Action Prediction

We perform parallel predictions of real-time actions a_i and memory actions a_i^t as follows:

$$a_i = \text{FFN}(\hat{v}_i), \quad a_i^t = \text{FFN}(\hat{v}_i^t), \tag{7}$$

where FFN denotes a two-layer feed-forward network. a_0 and a_0^t represent the stop scores in the real-time and memory action spaces, respectively.

To fuse the memory prediction scores and real-time prediction scores, we adopt a backtracking strategy [4], mapping the real-time action space scores a_i to the memory action space scores a_i^t. Specifically, when navigating to a node unconnected to the current node, the backtracking score for the unconnected node is computed by summing the decision scores of the visited candidate nodes. Finally, the fusion of real-time and memory action prediction scores a_f^t is achieved using a weight W_r:

$$a_i^r = \begin{cases} a_{back}, & \text{if backtrack,} \\ a_i, & \text{otherwise.} \end{cases} \tag{8}$$

$$a_i^f = W_r a_i^r + (1 - W_r) a_i^t. \tag{9}$$

2.3 Distance Adaptive Loss

The distributional bias between training and test environments can significantly impact learning efficiency and performance. To address this limitation, various

approaches have been proposed, such as DAgger and reinforcement learning. Most methods combine behavior cloning with an interactive demonstrator to provide supervision during training. However, unlike other sequence prediction tasks, navigation tasks involve two distinct behaviors: **exploration** during the navigation process and **stopping** at the destination. The training loss function for such tasks can be defined as:

$$L = \sum_{t=0}^{T-1} L_t + L_{stop}, \tag{10}$$

where T denotes the total number of navigation steps, L_t represents the loss incurred during exploration, and L_{stop} is the loss associated with the stopping action at the destination. The total loss is the sum of losses over T steps, which causes the agent to prioritize minimizing the number of steps during training while learning the stopping action. However, stopping at an incorrect location results in a complete navigation failure, particularly in goal-oriented tasks.

To mitigate this issue, we introduce a penalty factor based on the agent's distance to the destination within our proposed process adaptive learning loss function L_{pal}. This factor increases the penalty for failed stopping actions while reducing the weight of exploration losses during navigation. Specifically, for the loss incurred by predicting a stop action at an incorrect location or failing to stop at the correct destination, we augment the original loss with a distance-related reinforcement factor λ. Similarly, for the exploration loss incurred during navigation, we introduce a distance-related decay factor γ:

$$L_{pal} = \sum_{t=0}^{T-1} \gamma L_t + (1+\lambda) L_{stop}, \tag{11}$$

$$\lambda = \frac{d_t}{D_{dist}}, \tag{12}$$

$$\gamma = p^{\lambda}, \tag{13}$$

where d_t denotes the distance to the destination at step t, and D_{dist} represents the quantile of the distance distribution between nodes in the current environment. $p \in (0, 1)$ is the decay factor. When the agent fails to stop at the destination, d_t is set to D_{dist}, thereby amplifying the loss. Conversely, when the distance d exceeds D_{dist}, the decay factor γ decreases rapidly, allowing the agent to explore more thoroughly.

3 Experiments

3.1 Datasets

Our approach primarily focuses on the goal-oriented navigation task, REVERIE. REVERIE instructions have an average length of 21 words and are characterized by providing only coarse-grained, goal-directed guidance, including a brief

description of the target location and object. This task emphasizes grounding remote target objects, requiring agents to possess advanced exploration capabilities to locate distant targets and achieve target grounding. Additionally, we evaluate our model on the widely used VLN benchmark R2R, where each trajectory is paired with three step-by-step instructions, with an average instruction length of 32 words. The dataset is divided into train, val seen, val unseen, and test unseen splits.

3.2 Evaluation Metric

As in [1,24], we adopt the following evaluation metrics: Navigation Error (NE), defined as the average distance (in meters) between the agent's final position and the target location; Success Rate (SR), which measures the proportion of paths where NE is less than 3 m; Oracle Success Rate (OSR), calculated as SR under an oracle stopping strategy; and Success weighted by Path Length (SPL), which accounts for the efficiency of successful paths. Additionally, Remote Grounding Success (RGS) and Remote Grounding Success weighted by Path Length (RGSPL) are used to evaluate object grounding.

3.3 Implementation Details

We use 6, 2, 3, and 3 Transformer layers in the text encoder, panoramic encoder, real-time cross-modal encoder, and memory cross-modal encoder, respectively. CLIP-B/16 is utilized for image feature extraction, and the network weights are initialized with METER. During pretraining, we adopt Masked Language Modeling (MLM), Single-step Action Prediction (SAP), Cross-modal Feature Pooling (CFP), and Object Grounding (OG) when object annotations are available. We apply EnvEdit to enhance environmental features. Synthetic augmented datasets are used for R2R and REVERIE, respectively. Pretraining is conducted on four RTX 2080Ti GPUs for up to 225K iterations with a batch size of 64, using the AdamW optimizer and a learning rate of 5×10^{-5}. For fine-tuning, the R2R and REVERIE tasks are trained on an RTX 4090 GPU with a batch size of 8, a learning rate of 2×10^{-5}, and a maximum of 125K iterations. We apply distance-adaptive loss to enhance the agent's generalization in unseen environments, where the decay factor p is set to 0.5. The distance distribution quantile D_{dist} is set to 0.7.

3.4 Performence

Performance on REVERIE. Table 1 compare our model with recent state-of-the-art methods on REVERIE dataset, Compared to GOAT [2], PART achieves improvements of 0.8% in SPL, 2.74% in SR, 2.84% in RGS, and 1.14% in RGSPL. This demonstrates the effectiveness of our architecture design.

Table 1. Comparison with the SOTA methods on REVERIE dataset.

Methods	Val Unseen				Test Unseen			
	SR↑	SPL↑	RGS↑	RGSPL↑	SR↑	SPL↑	RGS↑	RGSPL↑
HAMT [24]	32.95	30.20	18.92	17.28	30.40	26.67	14.88	13.08
HOP+ [11]	36.07	31.13	22.49	19.33	33.82	28.24	20.20	16.86
DUET [4]	46.98	33.73	32.15	23.03	52.51	36.06	31.88	22.06
DSRG [21]	47.83	34.02	32.69	23.37	54.04	37.09	32.49	22.18
GridMM [13]	51.37	36.47	34.57	24.56	53.13	36.60	34.87	23.45
BEVBert [18]	51.78	36.37	34.71	24.44	52.81	36.41	32.06	22.09
BSG [17]	52.12	35.59	35.36	24.24	56.45	38.70	33.15	22.34
GOAT [2]	53.37	36.70	38.43	26.09	57.72	40.53	38.32	26.70
PART	**57.34**	**39.00**	**40.56**	**27.16**	**60.46**	**41.33**	**41.16**	**27.84**

Performance on R2R. Table 2 presents the comparison results on R2R dataset, our method achieves improvement of 1.15% in SR and 0.35% in SPL compared to GOAT. In addition, it also shows significant performance gains in terms of NE(i.e., 3.04 → 2.89).

Table 2. Comparison with the SOTA methods on R2R dataset.

Methods	Val Unseen				Test Unseen			
	SR↑	SPL↑	NE↓	OSR↑	SR↑	SPL↑	NE↓	OSR↑
HAMT [24]	66	61	3.29	73	65	60	3.93	72
DUET [4]	72	60	3.31	81	69	59	3.65	76
DSRG [21]	73	62	3.00	81	72	61	3.33	78
GridMM [13]	75	64	2.83	–	73	62	3.35	–
GELA [12]	71	65	3.11	–	67	62	3.59	–
EnvEdit [15]	69	64	3.24	–	68	64	3.59	–
BSG [17]	74	62	2.89	–	73	62	3.19	–
BEVBert [18]	75	64	2.81	84	73	62	3.13	81
GOAT [2]	77.82	**68.13**	2.40	84.72	74.57	64.94	3.04	80.35
PART	**78.16**	67.75	**2.35**	**85.87**	**75.72**	**65.29**	**2.89**	**81.14**

3.5 Ablation Study

We conducted an ablation study of our approach on the REVERIE dataset. All results in this section are reported on the unseen validation split.

Feature Types for Reasoning. We first evaluate the topological mapping performance of the RCR and MCR modules using three different types of visual features in the REVERIE navigation task in Table 3. The examined features include panoramic features, as well as two types of features mentioned in Sect. 2.1—instruction-guided fused panoramic features and real-time features. Panoramic features refer to utilizing only the visual modality as historical memory. Fused panoramic features integrate instructions with visual information, allowing historical memory to be associated with both visual features and instructions while remaining independent of real-time action prediction. In contrast, real-time features incorporate visual features, instructions, and real-time action prediction simultaneously in the construction of historical memory. The experimental results indicate that inference based solely on panoramic features within the RCR module is inferior to inference driven by MCR memory features, whereas the combination of both leads to further performance improvements. In terms of SPL and RGSPL metrics, fused panoramic features outperform standalone panoramic features, suggesting that incorporating instruction information into historical memory enhances trajectory-instruction alignment and improves exploration efficiency. However, due to the lack of integration with real-time prediction, fused panoramic features demonstrate limited improvements in navigation accuracy and object grounding. In contrast, real-time features achieve the best overall performance, demonstrating that the joint modeling of real-time prediction and instructional information effectively enhances historical memory, enabling adaptive navigation adjustments across different environments.

Table 3. Ablation of Modules and topological mapping feature on REVERIE val unseen split.

Module	MTM Feature	SR↑	SPL↑	RGS↑	RGSPL↑
RCR	–	30.11	23.60	21.42	17.05
MCR	panorama	39.98	26.01	–	–
MCR	fused panorama	40.67	26.62	–	–
MCR	real-time	**41.37**	**26.85**	–	–
RCR+MCR	panorama	54.30	32.97	38.37	23.02
RCR+MCR	fused panorama	54.16	37.10	38.31	25.95
RCR+MCR	real-time	**55.27**	**37.83**	**40.44**	**27.14**

Distance Adaptive Loss. In Fig. 3, we analyze the impact of different distance distribution quantiles D_{dist} and decay factor p on the Distance Adaptive Loss. The results show that the distance adaptive loss helps improve navigation accuracy and efficiency. For D_{dist}, A smaller value D_{dist} increases the influence of distance on the loss function, amplifying the penalty for stopping at the wrong location or failing to stop at the target, while reducing the impact of losses during intermediate navigation. This makes the final navigation result more influential

on the overall process, with intermediate steps playing a smaller role. For p, A smaller value p further reduces the impact of intermediate navigation losses, decreasing their influence on the overall navigation process. It can be observed that overly emphasizing or neglecting the process is detrimental to navigation accuracy.

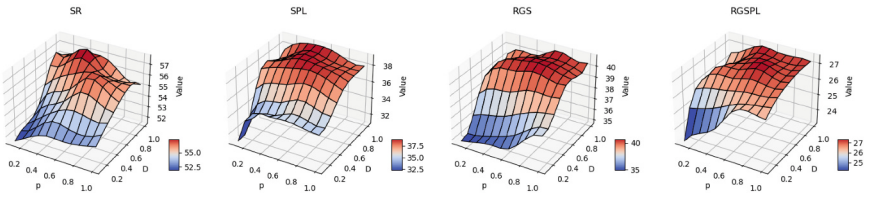


Fig. 3. Ablation study of the distance distribution quantile D_{dist} and decay factor p range on val unseen of REVERIE.

Qualitative Analysis. In Fig. 4, we visualize the trajectories predicted by GOAT [2] and PART in an unseen environment. The goal-oriented navigation task requires the agent to accurately identify the target's "region-location-object" combination(e.g. "kitchen-counter-coffee maker"). In this example, GOAT made errors in some target judgments, causing the navigation to stop at incorrect locations or even overlook the correct one. In contrast, PART adaptively reinforced the learning of the target while weakening the influence of irrelevant features, demonstrating greater robustness and accuracy.

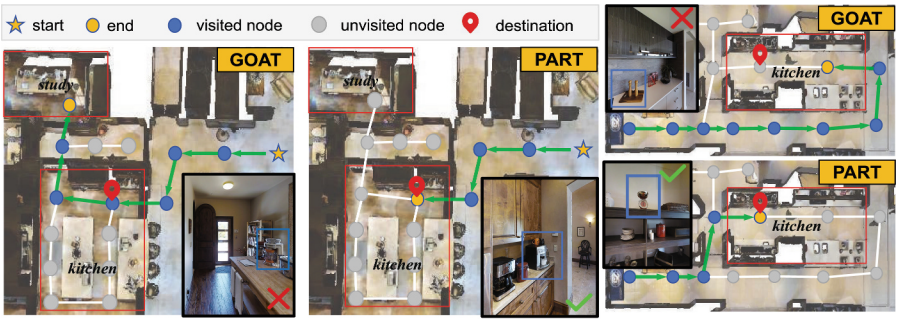


Fig. 4. Visualized Trajectories of GOAT [2] and PART on REVERIE unseen. Black circles denote the predicted target at endpoints.

4 Conclusion

We propose PART (Process-Adaptive Cross-Modal Transformer), which dynamically adjusts navigation history memory based on instructions, linking real-time action prediction with memory reasoning. This enhances the alignment between concise language instructions and complex navigation environments while mitigating the impact of overlapping trajectories on decision-making. The effectiveness of this architecture is validated through comparative experiments on different modules and topological mapping features. Furthermore, PART employs a distance-adaptive loss function to reduce dependence on specific trajectories and enhance goal-directed learning, thereby improving generalization in unknown environments. Through extensive ablation studies on the process decay factor p and the distance distribution quantile D_{dist}, we determine the optimal hyperparameters. PART achieves state-of-the-art performance on the VLN benchmarks R2R and REVERIE.

References

1. Anderson, P., et al.: Vision-and-language navigation: interpreting visually-grounded navigation instructions in real environments. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3674–3683 (2018)
2. Wang, L., He, Z., Dang, R., Shen, M., Liu, C., Chen, Q.: Vision-and-language navigation via causal learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13139–13150 (2024)
3. Chen, K., Chen, J.K., Chuang, J., Vázquez, M., Savarese, S.: Topological planning with transformers for vision-and-language navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11276–11286 (2021)
4. Chen, S., Guhur, P.-L., Tapaswi, M., Schmid, C., Laptev, I.: Think global, act local: dual-scale graph transformer for vision-and-language navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16537–16547 (2022)
5. Fried, D., et al.: Speaker-follower models for vision-and-language navigation. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
6. Qi, Y., et al.: Reverie: remote embodied visual referring expression in real indoor environments. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9982–9991 (2020)
7. Wang, X., et al.: Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6629–6638 (2019)
8. Yu, L., Chen, X., Gkioxari, G., Bansal, M., Berg, T.L., Batra, D.: Multi-target embodied question answering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6309–6318 (2019)
9. Chang, A., et al.: Matterport3d: learning from rgb-d data in indoor environments, arXiv preprint arXiv:1709.06158 (2017)
10. Tan, H., Bansal, M.: Lxmert: learning cross-modality encoder representations from transformers, arXiv preprint arXiv:1908.07490 (2019)

11. Qiao, Y., Qi, Y., Hong, Y., Yu, Z., Wang, P., Wu, Q.: Hop+: history-enhanced and order-aware pre-training for vision-and-language navigation. IEEE Trans. Pattern Anal. Mach. Intell. **45**(7), 8524–8537 (2023)
12. Cui, Y., et al.: Grounded entity-landmark adaptive pre-training for vision-and-language navigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 12043–12053 (2023)
13. Wang, Z., Li, X., Yang, J., Liu, Y., Jiang, S.: Gridmm: grid memory map for vision-and-language navigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 15625–15636 (2023)
14. Deng, Z., Narasimhan, K., Russakovsky, O.: Evolving graphical planner: contextual global planning for vision-and-language navigation. Adv. Neural. Inf. Process. Syst. **33**, 20660–20672 (2020)
15. Li, J., Tan, H., Bansal, M.: Envedit: environment editing for vision-and-language navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15407–15417 (2022)
16. Du, M., et al.: DELAN: dual-level alignment for vision-and-language navigation by cross-modal contrastive learning, arXiv preprint arXiv:2404.01994 (2024)
17. Liu, R., Wang, X., Wang, W., Yang, Y.: Bird's-eye-view scene graph for vision-language navigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10968–10980 (2023)
18. An, D., et al.: Bevbert: multimodal map pre-training for language-guided navigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2737–2748 (2023)
19. Devlin, J.: Bert: pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018)
20. Lin, B., Zhu, Y., Chen, Z., Liang, X., Liu, J., Liang, X.: Adapt: vision-language navigation with modality-aligned action prompts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15396–15406 (2022)
21. Wang, L., et al.: A dual semantic-aware recurrent global-adaptive network for vision-and-language navigation, arXiv preprint arXiv:2305.03602 (2023)
22. Liu, Y.: Roberta: a robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692, vol. 364 (2019)
23. Dou, Z.-Y., et al.: An empirical study of training end-to-end vision-and-language transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18166–18176 (2022)
24. Chen, S., Guhur, P.-L., Schmid, C., Laptev, I.: History aware multimodal transformer for vision-and-language navigation. Adv. Neural. Inf. Process. Syst. **34**, 5834–5847 (2021)
25. Lin, K., Chen, P., Huang, D., Li, T.H., Tan, M., Gan, C.: Learning vision-and-language navigation from youtube videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8317–8326 (2023)



Audio-Driven Talking Head Generation with Emotion Based on FLAME Geometry Model

Yahui Li, Liejun Wang(✉), Yinfeng Yu, and Shengjie Shen

School of Computer Science and Technology, Xinjiang University, Urumqi 830017, China
wljxju@xju.edu.cn

Abstract. Nowadays, audio-driven talking head generation has gained attention from various fields, yet there are limited studies on generating talking heads with emotions. Existing methods are either limited to generating simple facial expressions only through discrete emotion labels, or rely on complex emotion disentanglement processes of emotion-driven videos. Additionally, due to the inherent limitations of 2D generation models, capturing facial details such as foreheads, wrinkles, and teeth remains challenging. To address these challenges, this paper proposes an emotional talking head generation method based on the FLAME geometric model, which utilizes the geometric structure of the source image to provide 3D priors for the synthesis process. To overcome the emotional expression constraints imposed by emotion labels, the emotional text generated by LLaVA is used to supplement the deficiency of emotions, and a loss function is introduced to calculate the similarity between emotional text and generated images. Meanwhile, the Mamba module is introduced in the decoder to dynamically connect facial details and expressions. Experiments demonstrate that the proposed framework achieves impressive results in image generation details and the expression of complex emotions.

Keywords: Emotional Talking Head Generation · FLAME Geometric Model · Emotional Complexity

1 Introduction

In recent years, the growing adoption of digital humans across industries such as education and entertainment has significantly increased research interest in emotionally expressive talking head generation. This field has advanced rapidly with the development of generative models, particularly Generative Adversarial Networks (GANs) [9] and diffusion models [6]. A key focus within this domain is audio-driven emotional talking head generation, which aims to generate realistic talking head videos with corresponding emotions synchronized with speech through external emotional prompts. This research is particularly important because human communication inherently involves rich emotion. Therefore, in the process of simulating real human conversations, the integration of emotional elements is crucial, which can give language vividness and authenticity.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 179–191, 2026.
https://doi.org/10.1007/978-3-032-04555-3_15

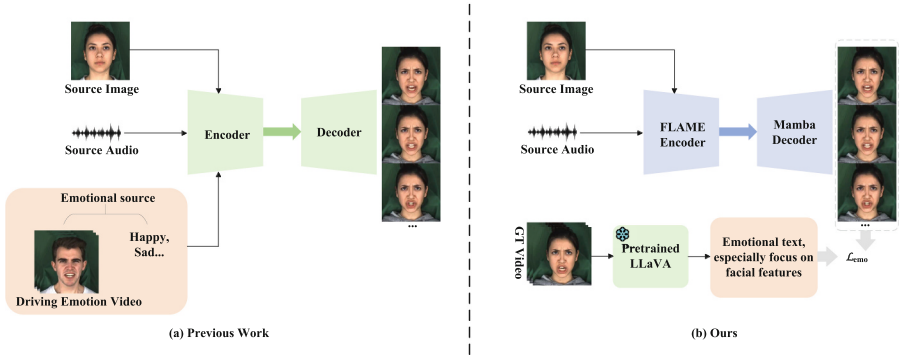


Fig. 1. Emotional talking-head generation. (a) Previous works use the GAN architectures to synthesize emotional talking head videos from a given driving emotion source (driving emotion video or emotion label). (b) Our method converts emotion-agnostic talking-head models into models with face emotions via rich detailed description.

In terms of emotional source, Ji et al. [14] decouple the emotion vector from speech as the source of emotion. Tan et al. and Sun et al. [22,23] extract emotional information from the driving video. Zhang et al. [30] consider the intensity of emotion that EAT [8] model does not consider. Sinha et al. [21] avoid the complicated operation of extracting emotion vectors from the driving video and focus on a single emotional prompt, such as happy, sad, etc. Regarding the quality of synthesized videos, the biggest drawback of the two-dimensional generative model is that it lacks accuracy in generating facial details and cannot fully capture and reproduce the subtle emotional changes of the speaker. Among the current advanced methods, the GAN architecture represented by EAT [8] still needs to improve its quality of generated video.

To effectively address the aforementioned challenges, this study proposes a novel method termed audio-driven talking head generation with **E**motion based on the FLAME [15] **G**eometry **M**odel (EGM), which achieves detailed faces and more complex emotional expressions, as illustrated in Fig. 1. Our proposed approach has two key advantages. Firstly, our method for obtaining emotional sources is more lightweight. Considering the source of emotions, the operation of extracting emotions from driving videos is too cumbersome, while relying solely on emotional labels is obviously insufficient. To address this, we use the dataset labels as prompts for the Large Language and Vision Assistant (LLaVA) [16] to generate emotional text for the ground-truth video. Subsequently, the CLIP loss [26] is used during the training process to constrain the semantic consistency between the generated video and the emotional text. Secondly, our facial representation is more comprehensive. To preserve facial details, considering that the FLAME 3D model can maintain relatively consistent facial details when reconstructing facial geometry, FLAME is used to reconstruct facial geometry from the source image. Due to the coupled relationship between facial details and expressions, we also integrate the Mamba module into the decoder architecture

to dynamically model the relationship between the two. During inference, our system requires a source image and an emotional prompt to generate emotion-controllable talking head videos.

We conduct extensive experiments to evaluate the effectiveness of EGM for emotional talking head generation, achieving state-of-the-art (SOTA) performance with 8 h of training on one GPU compared to baseline competitors.

In summary, the main contributions of our work are listed below:

- We propose a geometry-based approach for emotional talking head generation, utilizing geometric priors derived from FLAME to enhance the overall 2D generation process.
- During the training stage, we use LLaVA to generate fine-grained emotional descriptions to guide emotion generation and design an emotion-related loss to ensure the semantic consistency of emotional descriptions with images.
- We integrate Mamba into the decoder to effectively capture global facial structures and local details. Experimental results demonstrate that our method achieves notable performance in generated video quality.

2 Related Work

2.1 Audio-Driven Talking Head Generation

Audio-driven talking-head generation [18,19,25,27,31,32] based on deep learning has attracted widespread attention in recent years. Prajwal et al. [19] extract lip regions from raw video frames and use the GAN architecture to generate lip movements that are highly matched to the input audio. However, Xu et al. [27] introduce a hierarchical audio-driven visual synthesis module to enhance the alignment accuracy between audio input and visual output. Zhou et al. [32] decouple audio into two parts, using facial keypoints as intermediate motion representation to generate two facial representations respectively: audio content-driven facial representation and speaker-driven facial representation. Zhou et al. [31] also consider head posture and can synthesize talking head videos with any specified posture. Zhang et al. [30] use a diffusion model with ControlNet [12] and generate FLAME-based expression vectors based on a given emotion label, combined with identity images and audio to generate target videos. In this work, we augment facial representation by combining keypoints with FLAME-extracted geometry, and integrate Mamba [10] for realistic talking head generation through its state-space modeling of facial dynamics.

2.2 Emotion-Aware Talking Head Generation

Few works consider emotional factors in talking-head generation, and those that do typically rely on GAN architectures. Ji et al. [14] take emotion into account in talking head video generation for the first time, decoupling emotion and content from audio and using Edge-to-Video network to generate emotional talking

head videos. Ji et al. [13] directly use emotional videos as the source of emotion and enhance these videos to generate one-shot emotional head animations. Gan et al. [8] introduce an emotional adaptation module and an emotional deformation network to achieve effective conversion from emotionless talking heads into emotional ones. Among diffusion-based approaches, Zhang et al. [29] use diffusion model to generate high-quality talking head videos. They incorporate the intensity of emotions based on previous methods, using texts as the source of emotions and distinguishing emotion, content and intensity in text. Zhang et al. [30] use the diffusion model to extract expression vectors based on DECA [7] to generate emotional face conditions and preserve identity in the denoising process of the diffusion model. In our work, we use the pretrained LLaVA [16] to generate emotional texts according to labels and introduce a text-image alignment loss to generate talking head videos with emotion.

3 Methods

We propose a two-stage training paradigm for audio-driven talking head generation with emotions. First, we introduce latent keypoint representation using the Audio-to-Expression Transformer (A2ET) proposed in EAT [8] and supplement the missing geometric information using FLAME. Second, we utilize emotional text generated by LLaVA [16] to guide emotion synthesis and refine expression details through the Emotion Deformation Network (EDN) and Emotion Adaptation Module (EAM) proposed in EAT. To further capture the mutually coupled relationship between facial actions, we introduce Mamba, which allows the model to simulate changes in facial expressions in a more detailed way.

3.1 Extract Geometric Features

Considering that using keypoints as an intermediate representation may result in significant loss of facial information, then 3D facial representation is introduced to supplement the missing facial details. Inspired by the DECA [7] method, we first parameterize the input source image using FLAME to coarsely model the image in the FLAME model space, enabling the reconstruction of facial details such as expression-dependent wrinkles. Specifically, given a 2D image as input, encode it using an encoder E_c consisting of a ResNet50 and a fully connected layer to predict the latent code of the image. The generated latent code contains FLAME parameters $\boldsymbol{\beta}$, $\boldsymbol{\psi}$, $\boldsymbol{\theta}$, among which the first 100 FLAME shape parameters ($\boldsymbol{\beta}$), 50 FLAME expression parameters ($\boldsymbol{\psi}$) and 15 FLAME pose parameters ($\boldsymbol{\theta}$) are processed by FLAME to obtain a rough face mesh representation for rough geometric modeling, as shown in the top branch in Fig. 2(a). The other branch in Fig. 2(a) is based on the keypoint operation. After detecting keypoints in the source image using the keypoint detector D_k, the detected keypoints are combined with an emotion token and fed into the emotion deformation network to predict ΔE_i. The facial expression E_i is then deformed using linear summation as defined in Eq. 1. In addition, the detected image

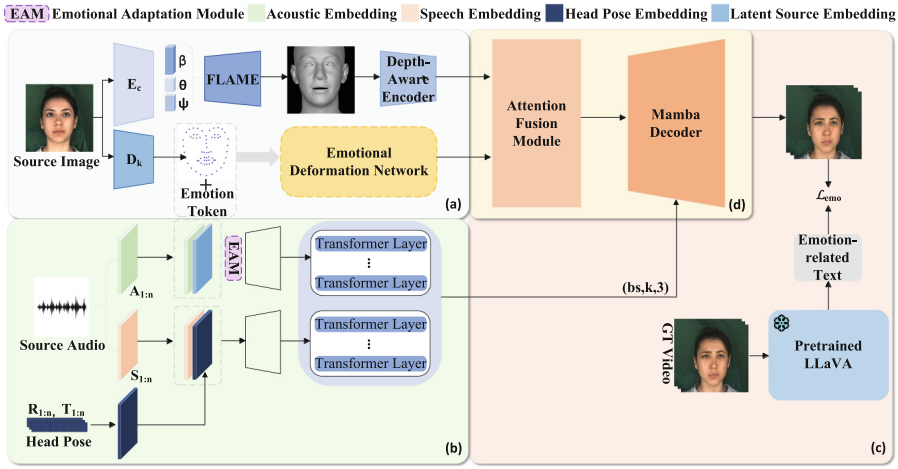


Fig. 2. Overview of EGM. In the first stage, an emotion-agnostic talking head model is trained. (a) FLAME parameterizes the source image. (b) The A2ET converts the source image, source audio and head pose sequence into discrete 3D facial expression deformation. In the second stage, emotions are injected into A2ET, EDN and EAM. (c) Additionally, the ground-truth videos are processed by LLaVA to generate emotional text. The generated videos and the corresponding emotional text are jointly utilized in the CLIP loss. (d) The generator combines all components to generate the final video.

keypoints are fused with the acoustic vector $A_{1:n}$ in Fig. 2(b) to obtain discrete acoustic tokens for A2ET decoding.

$$E'_i = E_i + \Delta E_i, \tag{1}$$

where E'_i represents the emotional expression deformation.

3.2 Mamba Decoder

The attention fusion module weightedly fuses FLAME geometric features, emotion deformation features E'_i to generate a comprehensive facial feature representation, then inputs the fused facial features into the Mamba decoder for decoding combined with audio features, head pose features and latent source features.

In the field of talking head generation, existing methods [8,13] are constrained by limited spatial attention fields, which restricts their ability to capture sufficient global features and recover detailed facial expressions. To address this limitation, we introduce the Mamba module, a State Space Model (SSM) [11] that can effectively model long-range dependencies while maintaining linear computational complexity. Specifically, leveraging the capability of SSMs to correlate distant regions in an image, we insert the Mamba module after the first Upsample and SPADEResnetBlock channel reduction step to process 256-dimensional features, as shown in Fig. 3. At this stage, the feature map restores partial spatial resolution but the number of channels is reduced. It is easy to introduce

low-frequency structural noise due to information compression, while Mamba can filter this redundant information through its selective mechanisms. The core component of Mamba module is the Visual State Space (VSS) block, which consists of two parallel branches, as illustrated in Fig. 3. In this structure, the 2D-Selective-Scan (SS2D) block comprehensively captures spatial features through multi-directional scanning and further feature extraction, then finally merges the scanning results to keep the spatial size unchanged.

3.3 Expand Emotional Sources

To make the visual content contain richer emotional semantic information, we employ the LLaVA [16] large model to generate captions for ground-truth emotional videos, enhancing the emotional information through descriptive text. We use the CLIP [20] model to align the generated video with the emotional text. However, given that the CLIP model has a maximum input length of 77 tokens, while LLaVA supports longer contexts (typically 2048 or 4096 tokens), CLIP's text encoder cannot directly process LLaVA's lengthy outputs. Therefore, we utilize longCLIP [28] as an alternative. Specifically, we use the text encoder of longCLIP-L to encode eight types of emotional texts, while proportionally sampling images according to the batch size for encoding through longCLIP-L's image encoder. We compute the Hadamard product between the image features f_v and the eight text features f_t, yielding the maximum probability distribution in Eq. 5. Finally, we minimize the cross-entropy loss between the predicted and ground-truth distributions.

4 Training Objectives

In supervised learning training, the total loss is calculated as follows:

$$\mathcal{L}_{total} = \lambda_{lat}\mathcal{L}_{lat} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{sync}\mathcal{L}_{sync} + \lambda_{emo}\mathcal{L}_{emo} \tag{2}$$

where λ_{lat}, λ_{rec}, λ_{sync}, λ_{emo} weight the corresponding terms.

Latent Loss. We use latent keypoints loss to optimize latent keypoints during training. Specifically, it can be expressed as follows:

$$\mathcal{L}_{lat} = \frac{1}{n}\sum_{i=1}^{n}(\left\|PE_i - \hat{PE}_i\right\|_2^2 + \left\|K_i - \hat{K}_i\right\|_2^2) \tag{3}$$

where n represents the length of audio frames in each batch, PE_i represents the predicted PCA of expression deformation in i frame. K_i is the latent keypoint, which can be expressed as $K_i = R_i K_c + T_i + E_i$ (the canonical keypoint K_c, the rotation matrix R_i, the translation T_i and the expression deformation E_i). $\hat{PE}_i$ and $\hat{K}_i$ are the ground-truth values corresponding to frame i.

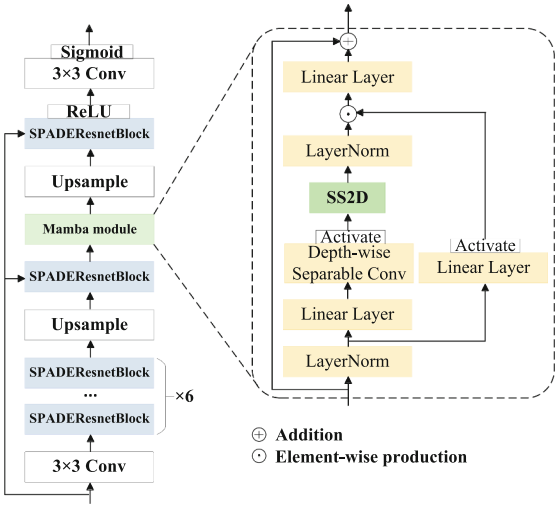


Fig. 3. The architecture of Mamba decoder.

Reconstruction Loss. To measure the expression reconstruction effect between the generated image and the ground truth, the $\mathcal{L}_1$ Loss is used to calculate the average of them only in the facial region. In addition, we apply the perceptual loss $\mathcal{L}_{per}$ to measure the similarity at the semantic level over the entire frame using a pretrained VGG19 model.

Sync Loss. We employ the pretrained SyncNet to process the generated video and the sampled audio frame, calculating the synchronization loss between them for each batch as follows:

$$\mathcal{L}_{\text{sync}} = -\log\left(\frac{v \cdot a}{max(\|v\|_2 \cdot \|a\|_2, \epsilon)}\right) \tag{4}$$

where v represents generated video embedding and a represents input speech embedding, they are respectively extracted by the speech encoder and image encoder in SyncNet.

Emotion Loss. $\mathcal{L}_{emo}$ uses CLIP loss to measure the similarity between emotional text and generated video. The text and image encoded by longCLIP are combined through Hadamard products. The cross-entropy loss is used to measure the difference between the predicted distribution p and the true distribution y. In summary, $\mathcal{L}_{emo}$ is calculated as follows:

$$p = softmax(f_v \circ f_t^T), \tag{5}$$

$$\mathcal{L}_{emo} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m} y_{i,j}\log(softmax(p_{i,j})) \tag{6}$$

where n represents the total number of samples in a batch, m represents the number of emotion categories, $p_{i,j}$ represents the output score (i.e., similarity)

of the i-th sample in the j-th category, and $y_{i,j}$ corresponds to the ground-truth label of the i-th sample in the j-th category, encoded using one-hot encoding.

5 Experiments

5.1 Experimental Setup

Implementation Details. The videos are sampled to 25 FPS, while the audios are sampled to 16 KHz. The videos are cropped to 256×256. To synchronize the audio with the video, we extract the mel spectrogram by configuring the window length and hop length to 640. The number of keypoints k is 15. Our work is based on one NVIDIA A40 GPU with 48G memory.

Datasets. The videos from LRW [4] are used to train the emotion-agnostic talking head, and videos from VoxCeleb2 [3] and MEAD [24] are used to train the emotional talking head. We split the MEAD into train and test sets based on ten distinct identities. To learn large head pose variations, we select about 5,000 emotion videos from VoxCeleb2 using a fine-tuned Emotion-FAN [17]. The test dataset consists of MEAD and LRW.

Metric. In terms of image quality, PSNR, SSIM and FID [1] are used to measure the image quality of synthesized videos. In terms of audio-visual synchronization, we use SyncNet confidence score [5]. The distance between the mouth landmarks (M-LMD) [2] is to measure the speech content consistency, while the distance between the facial landmarks (F-LMD) measures the accuracy of posture and expression. Acc_{emo} is to evaluate the accuracy of the generated emotions.

Table 1. Quantitative comparisons with SOTA methods on MEAD and LRW.

	MEAD					LRW			
	PSNR/SSIM↑	FID↓	SyncNet↑	M/F-LMD↓	Acc_{emo} ↑	PSNR/SSIM↑	FID↓	SyncNet↑	M/F-LMD↓
MakeItTalk [32]	20.39/**0.72**	64.18	1.59	3.10/2.53	52.10	21.67/0.68	20.45	2.70	2.17/2.90
EAMM [13]	15.32/0.52	108.48	2.72	3.34/4.16	44.14	17.24/0.53	55.82	2.47	2.48/3.77
EAT [8]	20.49/0.62	47.70	**7.68**	2.24/2.46	66.67	21.66/0.69	24.23	5.23	2.19/2.88
Hallo [27]	21.63/0.71	**13.65**	6.19	2.23/2.44	68.12	21.72/0.70	**14.21**	**5.82**	**1.50**/2.46
EGM (Ours)	**21.79**/0.69	44.41	7.66	**2.21**/**2.43**	**72.48**	**21.95**/**0.71**	22.71	4.71	2.03/**2.45**
Ground Truth	∞/1.00	0	7.76	0.00/0.00	84.37	∞/1.00	0	7.06	0.00/0.00

5.2 Emotion-Agnostic Talking-Head Generation

For one-shot emotion-agnostic talking head generation, we focus on the performance on the LRW dataset. We select 100 videos from LRW [4] as test set. In the self-driven generation task, we use the first frame of each video as the source image for testing. The results are shown in Table 1. Our method performs better visual generation quality, particularly at the pixel level (PSNR and SSIM), outperforming other baseline methods. Figure 4 shows intuitively

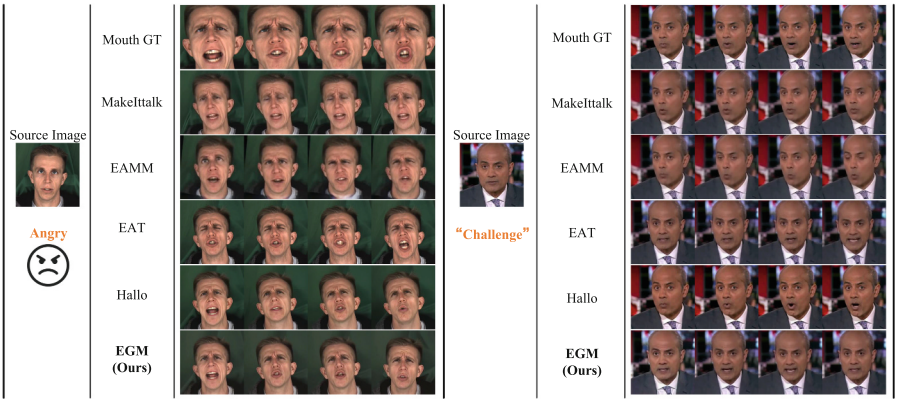


Fig. 4. The qualitative results of self-driven one-shot talking head generation. On the MEAD dataset (left), the generated consecutive frames are shown while using an image and an emotional prompt. On the LRW dataset (right), the consecutive frames when reading "challenge" are shown. The top row shows identity and driving emotion.

that the facial wrinkles, teeth and other details generated by EGM on LRW dataset are very clear. In terms of audio-visual synchronization, we focus on the SyncNet metric in Table 1 and the lip shape differences between the methods and the ground truth in Fig. 4. Our method has a gap with the true lip shape. We analyze that this is because the attention fusion module prioritizes visual effects and ignores the alignment of audio and video features, resulting in low temporal synchronization accuracy between the two modalities. However, the Hallo [27] method achieves SOTA performance in the Sync metric. We attribute this to its hierarchical mechanism, which processes audio and visual synthesis through cross-attention. This hierarchical design progressively refines the feature representations, aligning audio and visual features from coarse to fine, thereby improving the accuracy of the Sync metric to a certain extent.

5.3 Emotional Talking-Head Generation

We evaluate our method on the MEAD [24] dataset following the EAT [8] framework. We select 100 videos from 4 identities with varying emotions but consistent emotional intensity. Neutral frames are selected based on neutral emotion detection across all methods. As shown in Table 1, our method performs well in terms of pixel-level video quality and emotion accuracy (Acc_{emo}). Figure 4 intuitively shows the detailed emotional videos generated by our method in one-shot setting, i.e., the inference process. In terms of audio-visual synchronization, the experimental results on the MEAD dataset in Fig. 4 show that there is a slight difference between the lip shape generated by EGM and the ground-truth video. We analyze that this is due to the mutual constraints between the CLIP loss and the synchronization loss in the optimization process.

Table 2. Ablation study of each component on MEAD datasets.

Method	Mamba	FLAME	Emotion text	PSNR↑	M/F-LMD↓	Sync↑	Acc_{emo} ↑
(A)				20.77	2.67/2.74	7.11	68.69
(B)	✓			21.15	2.49/2.96	7.12	68.71
(C)	✓	✓		21.78	**2.19**/2.44	**7.67**	69.30
EGM	✓	✓	✓	**21.79**	2.21/**2.43**	7.66	**72.48**

Table 3. Ablation study of each loss on MEAD datasets.

Method	$\mathcal{L}_{per}$	$\mathcal{L}_{lat}$	$\mathcal{L}_{sync}$	$\mathcal{L}_1$	PSNR↑	Sync↑
(A)	✓				21.05	5.42
(B)	✓	✓			21.65	5.56
(C)	✓	✓	✓		21.12	6.97
EGM	✓	✓	✓	✓	**21.76**	**7.54**

5.4 Ablation Study

Each Component. To verify the effectiveness of the proposed module, we conduct ablation experiments, as shown in Table 2. (B) shows that Mamba can improve facial details through temporal modeling. After using FLAME and Mamba (C) together, the three-dimensional representation provided by FLAME constrains the facial geometry, while Mamba focuses on fine-grained dynamic feature modeling. The combination of the two significantly improves pixel-level quality. Meanwhile, the reduction of M-LMD indicates that (C) can keep the lip shape consistent with the speech content so that Sync can achieve better results. After adding the emotional text provided by LLaVA, the emotional accuracy is significantly improved, however, Sync slightly decreases compared to (C) method. We analyze this because the emotion-based loss weight affects the synchronization loss.

Each Loss. As shown in Table 3, we conduct an ablation study on perceptual loss, latent loss, sync loss and L1 loss. The results indicate that adding the synchronization loss $\mathcal{L}_{sync}$, as in (C) and (D), improves the Sync metric, demonstrating that $\mathcal{L}_{sync}$ helps audio-visual synchronization. Meanwhile, other losses contribute to improving the fidelity of the generated videos.

6 Conclusion

In this paper, we propose an audio-driven framework for synthesizing emotional talking face videos through a two-stage design. First, we employ the FLAME geometry model to reconstruct 3D face geometry from the source image, enhancing the limited information provided by 2D keypoints. Next, we integrate the Mamba module in the decoding stage to capture multi-directional information.

To improve emotion accuracy, we introduce an emotion-related CLIP loss based on an emotion adaptation module. Extensive experiments demonstrate the superior performance of our EGM on the MEAD and LRW datasets.

Acknowledgments. This study was funded by the Excellence Program Project of Tianshan, Xinjiang Uygur Autonomous Region, China (grant number 2022TSYCLJ0036), the Central Government Guides Local Science and Technology Development Fund Projects (grant number ZYYD2022C19), and the National Natural Science Foundation of China (grant numbers 62463029, 62472368 and 62303259).

References

1. Bynagari, N.B.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. Asian J. Appl. Sci. Eng. **8**(1), 25–34 (2019)
2. Chen, L., Maddox, R.K., Duan, Z., Xu, C.: Hierarchical cross-modal talking face generation with dynamic pixel-wise loss. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7832–7841 (2019)
3. Chung, J.S., Nagrani, A., Zisserman, A.: Voxceleb2: deep speaker recognition. arXiv preprint arXiv:1806.05622 (2018)
4. Chung, J.S., Zisserman, A.: Lip reading in the wild. In: Lai, S.-H., Lepetit, V., Nishino, K., Sato, Y. (eds.) ACCV 2016. LNCS, vol. 10112, pp. 87–103. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54184-6_6
5. Chung, J.S., Zisserman, A.: Out of time: automated lip sync in the wild. In: Chen, C.-S., Lu, J., Ma, K.-K. (eds.) ACCV 2016. LNCS, vol. 10117, pp. 251–263. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54427-4_19
6. Croitoru, F.A., Hondru, V., Ionescu, R.T., Shah, M.: Diffusion models in vision: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **45**(9), 10850–10869 (2023)
7. Feng, Y., Feng, H., Black, M.J., Bolkart, T.: Learning an animatable detailed 3D face model from in-the-wild images. ACM Trans. Graph. (ToG) **40**(4), 1–13 (2021)
8. Gan, Y., Yang, Z., Yue, X., Sun, L., Yang, Y.: Efficient emotional adaptation for audio-driven talking-head generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 22634–22645 (2023)
9. Goodfellow, I., et al.: Generative adversarial networks. Commun. ACM **63**(11), 139–144 (2020)
10. Gu, A., Dao, T.: Mamba: linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752 (2023)
11. Hamilton, J.D.: State-space models. In: Handbook of Econometrics, vol. 4, pp. 3039–3080 (1994)
12. Hu, Z., Xu, D.: Videocontrolnet: a motion-guided video-to-video translation framework by using diffusion model with controlnet. arXiv preprint arXiv:2307.14073 (2023)
13. Ji, X., et al.: EAMM: one-shot emotional talking face via audio-based emotion-aware motion model. In: ACM SIGGRAPH 2022 Conference Proceedings, pp. 1–10 (2022)

14. Ji, X., et al.: Audio-driven emotional video portraits. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14080–14089 (2021)
15. Li, T., Bolkart, T., Black, M.J., Li, H., Romero, J.: Learning a model of facial shape and expression from 4D scans. ACM Trans. Graph. **36**(6), 194-1 (2017)
16. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Adv. Neural. Inf. Process. Syst. **36**, 34892–34916 (2023)
17. Meng, D., Peng, X., Wang, K., Qiao, Y.: Frame attention networks for facial expression recognition in videos. In: 2019 IEEE International Conference on Image Processing (ICIP), pp. 3866–3870. IEEE (2019)
18. Mukhopadhyay, S., Suri, S., Gadde, R.T., Shrivastava, A.: Diff2lip: audio conditioned diffusion models for lip-synchronization. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 5292–5302 (2024)
19. Prajwal, K., Mukhopadhyay, R., Namboodiri, V.P., Jawahar, C.: A lip sync expert is all you need for speech to lip generation in the wild. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 484–492 (2020)
20. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763. PMLR (2021)
21. Sinha, S., Biswas, S., Yadav, R., Bhowmick, B.: Emotion-controllable generalized talking face generation. arXiv preprint arXiv:2205.01155 (2022)
22. Sun, Z., Xuan, Y., Liu, F., Xiang, Y.: FG-EmoTalk: talking head video generation with fine-grained controllable facial expressions. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 5043–5051 (2024)
23. Tan, S., Ji, B., Bi, M., Pan, Y.: EDTalk: efficient disentanglement for emotional talking head synthesis. In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G. (eds.) ECCV 2024. LNCS, vol. 15064, pp. 398–416. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-72658-3_23
24. Wang, K., et al.: MEAD: a large-scale audio-visual dataset for emotional talking-face generation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12366, pp. 700–717. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58589-1_42
25. Wang, S., Li, L., Ding, Y., Yu, X.: One-shot talking face generation from single-speaker audio-visual correlation learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 2531–2539 (2022)
26. Wang, Y., et al.: Cliploss and norm-based data selection methods for multimodal contrastive learning. Adv. Neural. Inf. Process. Syst. **37**, 15028–15069 (2025)
27. Xu, M., et al.: Hallo: hierarchical audio-driven visual synthesis for portrait image animation. arXiv preprint arXiv:2406.08801 (2024)
28. Zhang, B., Zhang, P., Dong, X., Zang, Y., Wang, J.: Long-clip: Unlocking the long-text capability of clip. In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G. (eds.) ECCV 2024. LNCS, vol. 15109, pp. 310–325. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-72983-6_18
29. Zhang, B., Zhang, X., Cheng, N., Yu, J., Xiao, J., Wang, J.: Emotalker: emotionally editable talking face generation via diffusion model. In: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8276–8280. IEEE (2024)
30. Zhang, J., Mai, W., Zhang, Z.: Emodiffhead: continuously emotional control in talking head generation via diffusion. arXiv preprint arXiv:2409.07255 (2024)

31. Zhou, H., Sun, Y., Wu, W., Loy, C.C., Wang, X., Liu, Z.: Pose-controllable talking face generation by implicitly modularized audio-visual representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4176–4186 (2021)
32. Zhou, Y., Han, X., Shechtman, E., Echevarria, J., Kalogerakis, E., Li, D.: Makelttalk: speaker-aware talking-head animation. ACM Trans. Graph. (TOG) **39**(6), 1–15 (2020)



Studying the Generalization Behavior of Surrogate Models for Punch-Bending by Generating Plausible Counterfactuals

Andreas Mazur[1(✉)], Henning Peters[2], André Artelt[1], Lukas Koller[3], Christoph Hartmann[4], Ansgar Trächtler[2], and Barbara Hammer[1]

[1] Center for Cognitive Interaction Technology (CITEC), Bielefeld University, Inspiration 1, 33619 Bielefeld, Germany
{amazur,aartelt,bhammer}@techfak.uni-bielefeld.de
[2] Fraunhofer Institute for Mechatronic Systems Design (IEM), Zukunftsmeile 1, 33102 Paderborn, Germany
{henning.peters,ansgar.traechtler}@iem.fraunhofer.de
[3] Professorship of Cyber Physical Systems (Prof. Althoff), Technical University of Munich, Boltzmannstr. 3, 85748 Garching b. München, Germany
lukas.koller@tum.de
[4] Chair of Metal Forming and Casting (UTG), Technical University of Munich, Walther-Meißner-Straße 4, 85748 München, Germany
christoph.hartmann@utg.de

Abstract. Punch-bending is an industrial, pivotal process that allows to rapidly manufacture large quantities of products by bending semi-finished products, such as metal sheets or -wires, into target geometries. To ensure that the desired geometry can be reliably produced, a suitable process configuration, such as the used tool geometries and material properties, is required. Unlike traditional expensive trial-and-error approaches, we propose a machine learning surrogate model for the punch-bending process and subsequent plausible counterfactual explanations, which suggest process configurations for a desired target geometry. For this to work, the surrogate model must exhibit realistic generalization behavior. We analyze the generalization behavior of a surrogate model for a punch-bending scenario by generating plausible counterfactuals, and show that these also yield better process configurations compared to a baseline.

Keywords: Machine Learning · Explainable AI · Counterfactual Explanations · Manifold Learning · Forming Technology · Punch-Bending

1 Introduction

Punch-bending processes are used to manufacture products such as brackets, electronic contacts or spring elements from semi-finished products such as metal

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 192–203, 2026.
https://doi.org/10.1007/978-3-032-04555-3_16

sheets or -wires, using automated punch-bending machines with several bending operations [20,23]. In this work, we will consider metal sheets which are bent once. The typical punch-bending process [21] starts by unwounding a sheet from a material coil and feeding it into a straightening line that minimizes the residual stresses and the curvature of the sheet [20]. After straightening, the sheet is fed into one or several punch-bending units which bend the sheet into the desired geometry. The quality of punch-bending products depends significantly on the design and adjustment of the multi-stage punch-bending process [20]. Corrections within industrial punch-bending applications are usually performed with expert knowledge, employing a time-consuming and expensive trial-and-error approach. Due to the complex interactions between the semi-finished products and the tools, this procedure does not always succeed [1,8,20] and results in a significant level of waste [20].

In the past, surrogate modeling approaches have been used to predict physical properties and outcomes of processes. They typically offer a significant speed-up compared to traditional approaches such as finite-element methods [3,12,18,25,26]. Explainable AI (XAI) methods [19] can help to gain insight into the trustworthiness of the trained surrogate by verifying the input-output behavior in comparison to the underlying physical process. One such method is represented by generating plausible counterfactual explanations [2,29], often just called plausible counterfactuals, which have proven to be useful in many cases in the past [28]. They describe an actionable change in the input to achieve a desired target outcome. Here, actionability relies on the counterfactual's plausibility which, in turn, can be achieved in various ways such as integrating variational auto-encoders (VAEs) into their computation pipeline [7,10,11,17]. Considering the necessity of adequate design choices for target geometries in punch-bending, those prove particularly useful for this context.

However, the computational overhead of generating counterfactuals needs to be justified. One traditional approach to determine adequate process configurations is to select a configuration that returns the most similar result to the desired bending-angle from a database that collects former punch-bending experiment configurations and -results. It is easy to see that this approach will fail in cases where the desired result has never been captured by the database. Consequently, to justify the overhead for computing counterfactuals, they should remain feasible process configurations that cause outcome bending-angles close to the target even when the desired target geometries lie in unobserved data regions. In order for the counterfactuals to remain physically realistic in unobserved data regions, the surrogate model needs to exhibit realistic generalization properties.

In this work, we use the punch-bending scenario to study the generalization behavior of a surrogate model by generating plausible counterfactuals and demonstrate that they produce superior results in the context of finding suitable process configurations for desired outcome bending-angles in punch-bending compared to a database look-up approach. Thereby, plausibility for the counterfactuals shall be achieved by (i) learning a meaningful latent space using a

VAE and (ii) compute counterfactuals within said latent space while using the decoder to generate a process configuration for the desired target.

We describe the necessary background information in Sect. 2 and the implementation of our approach in Sect. 3. Experiments are carried out in Sect. 4. We conclude our findings in Sect. 5.

2 Background

2.1 Counterfactual Explanations and Plausibility

A counterfactual explanation suggests changes to an input's features to change the system's output. The popularity of counterfactuals in XAI [19,28] stems from the fact that they mimic human reasoning in explanations [6]. The computation of a classic counterfactual [29] $\Delta_{\text{cf}} \in \mathbb{R}^d$ for a given instance $\boldsymbol{x}_{\text{orig}} \in \mathbb{R}^d$ is phrased as the following optimization problem:

$$\underset{\Delta_{\text{cf}} \in \mathbb{R}^d}{\arg\min} \; \ell\left(h(\boldsymbol{x}_{\text{orig}} + \Delta_{\text{cf}}), y_{\text{cf}}\right) + C \cdot \theta(\Delta_{\text{cf}}), \tag{1}$$

where $\ell(\cdot)$ penalizes deviation of the prediction $h(\boldsymbol{x}_{\text{orig}} + \Delta_{\text{cf}})$ from a given requested outcome y_{cf}, $\theta(\cdot)$ states the cost of the explanation which should be minimized, and $C > 0$ denotes the regularization strength balancing those two objectives. Note that the cost $\theta(\cdot)$ is domain-specific, but many implementations default to using the p-norm [9]. A common way [9] of solving Eq. (1) is to use a gradient-based method such as gradient descent – assuming that the model $h(\cdot)$ is differentiable. Besides those two objectives in Eq. (1), there exist additional aspects such as plausibility and actionability [2,22,27], which are crucial for their usefulness in practice [14]. Methods for ensuring plausibility and actionability range from density-based methods [2,30], over retrieval and path-based methods [22], to methods based on variational auto-encoders [7,10,11,17].

2.2 Variational Auto-encoder

Auto-encoders have been used for a long time to find lower-dimensional representations for higher dimensional inputs [24]. This learning problem has been re-framed by [15] into a Bayesian setting: The encoding of observable data $\boldsymbol{x} \in \mathbb{R}^n$ can be given by the maximum a posteriori estimate (MAP) of unobservable and typically lower-dimensional latent features $\boldsymbol{z} \in \mathbb{R}^m$. In probability distribution terms: The observable data distribution over samples $\boldsymbol{x}$ depends conditionally on non-observable latent features $\boldsymbol{z}$ and thus is described by the likelihood $p(\boldsymbol{x}|\boldsymbol{z})$. The latent features $\boldsymbol{z}$ follow a prior distribution $p(\boldsymbol{z})$, typically chosen to be Gaussian. Given the likelihood $p(\boldsymbol{x}|\boldsymbol{z})$, the prior $p(\boldsymbol{z})$ and the possibly intractable marginal likelihood $p(\boldsymbol{x})$, the posterior of $\boldsymbol{z}$ is given by $p(\boldsymbol{z}|\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{z})\, p(\boldsymbol{z})/p(\boldsymbol{x})$. A machine learning model that estimates both the likelihood $p(\boldsymbol{x}|\boldsymbol{z})$ and the posterior $p(\boldsymbol{z}|\boldsymbol{x})$ is called a ***v**ariational **a**uto-**e**ncoder* (VAE).

A VAE can be trained as a neural network [15] that consists of an *encoder network* with trainable parameters ϕ that ends in a *bottleneck-layer*, followed by a *decoder network* with separate trainable parameters θ which uses the latent features from the bottleneck layer to reconstruct the original input. It is trained by simultaneously optimizing both the encoder and decoder, minimizing the negative evidence lower bound (ELBO) [15] of the marginal log-likelihood $\log p(\boldsymbol{x})$ to circumvent $p(\boldsymbol{x})$'s potential intractability:

$$-\,\mathrm{ELBO}(\theta, \phi, \boldsymbol{x}) = D_{KL}(p_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})) - \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(\boldsymbol{x}|\boldsymbol{z}^{(l)})\,. \tag{2}$$

Here, D_{KL} describes to the Kullback-Leibler divergence between the distribution over the learned latent features $\boldsymbol{z}$ and the assumed prior while the latter term represents the reconstruction error of $\boldsymbol{x}$. The *reparametrization trick* $\boldsymbol{z}^{(l)} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\varepsilon}^{(l)}$ with $\boldsymbol{\varepsilon}^{(l)} \sim p(\boldsymbol{\varepsilon})$ is used to sample L different samples from the posterior $p_\phi(\boldsymbol{z}|\boldsymbol{x})$ to (i) estimate the expected log-likelihood $\log p_\theta(\boldsymbol{x}|\boldsymbol{z}^{(l)})$ using Monte Carlo sampling and (ii) maintain the possibility to use back-propagation for training the parameters ϕ and θ all together [15].

3 The Machine Learning Module

The purpose of the machine learning module is to return plausible process configuration suggestions to produce a desired outcome bending-angle. In our context a process configuration is described by the Young's modulus of the sheet, which describes its elasticity, the thickness of the sheet and the used bending-angle and -radius of the punch [16]. For this, we have designed it to have three components: The first part is a VAE that is supposed to learn a meaningful latent space from which plausible process configurations can be generated. The second part is a surrogate model for the bending process that learns a mapping from process configurations to outcome bending-angles in the sheet. The third element represents an algorithm to generate plausible counterfactuals for desired outcome bending-angles that operates within the learned latent space of the VAE. An overview is illustrated in Fig. 1.

3.1 VAE Architecture and Training

Our proposed VAE is implemented in the form of a neural network that consists of an encoder, a bottleneck layer and a decoder. The encoder consists of two branches $\boldsymbol{E}_1 : \mathbb{R}^4 \to \mathbb{R}^2$ and $\boldsymbol{E}_2 : \mathbb{R}^4 \to \mathbb{R}^2$ and receives the four-dimensional process configurations as input and return two-dimensional latent-means and log-variances. They share weights ϕ in the form of a multi-layer perceptron (MLP) and only differ in their used weights in the bottleneck layer. Weights Σ_1 are trained to predict the log-variance and weights Σ_2 to predict the mean:

$$\boldsymbol{E}_1(\boldsymbol{x};\, \phi, \Sigma_1) = \log \boldsymbol{\sigma}^2,\;\; \boldsymbol{E}_2(\boldsymbol{x};\, \phi, \Sigma_2) = \boldsymbol{\mu}\,. \tag{3}$$

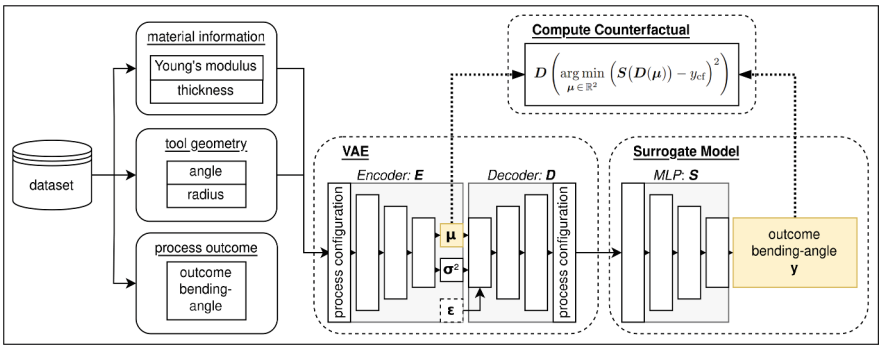


Fig. 1. The architecture of the machine learning module.

The decoder $\boldsymbol{D} : \mathbb{R}^2 \to \mathbb{R}^4$ returns process-configurations and is represented by a MLP. See Fig. 2 for a depiction of the overall VAE-architecture.

Let $\boldsymbol{x}^{(f)}$ be the f-th feature of $\boldsymbol{x}$. We have normalized each $\boldsymbol{x}^{(f)}$ by converting it to its z-score. That is, we subtract the feature's observed mean value $\bar{\boldsymbol{x}}^{(f)}$ and divide by the observed standard deviation $\boldsymbol{\sigma}^{(f)}$ to retrieve $\boldsymbol{x}^{(f)}_{\text{normalized}} = (\boldsymbol{x}^{(f)} - \bar{\boldsymbol{x}}^{(f)})/\boldsymbol{\sigma}^{(f)}$. We have done the same for the observed outcome bending-angles. The used loss function is the standard VAE loss:

$$\begin{aligned}\mathcal{L}(\boldsymbol{x};\, \beta, \phi, \Sigma_1, \Sigma_2, \theta) = \frac{\beta}{2} \sum_{l=1}^{2} \left(1 + \boldsymbol{E}_1(\boldsymbol{x};\, \phi, \Sigma_1)_l - \boldsymbol{E}_2(\boldsymbol{x};\, \phi, \Sigma_2)_l^2 - e^{\boldsymbol{E}_1(\boldsymbol{x};\, \phi, \Sigma_1)_l}\right) \\ + \frac{1}{4} \sum_{o=1}^{4} (\boldsymbol{D}(\boldsymbol{z};\, \theta)_o - \boldsymbol{x}_o)^2\end{aligned} \tag{4}$$

with $\boldsymbol{z} = \boldsymbol{E}_2(\boldsymbol{x};\, \phi, \Sigma_2) + e^{\boldsymbol{E}_1(\boldsymbol{x};\, \phi, \Sigma_1)/2} \odot \boldsymbol{\varepsilon}$ and $\boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{0}, \mathbb{1})$. The first term represents the β-weighted KL-divergence of the predicted distribution by the encoder and a standard normal distribution. Following the suggestions of [4,13], we increase β gradually over the course of the first training steps to stabilize training, improve the overall VAE performance and get disentangled latent features. The second term represents the mean-squared error (MSE) over the process configuration values. Lastly, $e^{\boldsymbol{E}_1(\boldsymbol{x};\, \phi, \Sigma_1)/2}$ is to be understood as an element-wise application of the exponential function onto the elements of $\boldsymbol{E}_1(\boldsymbol{x};\, \phi, \Sigma_1)/2$. Implementation details, such as used hyperparameters, and the dataset can be found in the repository[1].

3.2 Surrogate Model Architecture and Training

The surrogate model for the bending process is implemented in the form of a simple MLP with two fully-connected (FC) layers that use ReLU-activation and a final fully-connected linear layer that outputs the regressed bending-angle:

$$\text{Process Conf.} \to \text{FC+ReLU(32)} \to \text{FC+ReLU(32)} \to \text{FC+Linear(1)} \to \text{Bending-Angle}$$

[1] *Source code and data:* https://github.com/andreasMazur/vae_cfs.

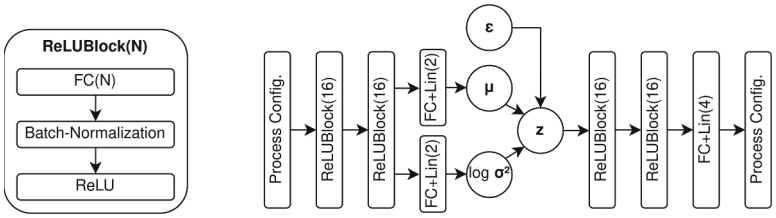


Fig. 2. [Left] A description for the ReLU block. N refers to the output-dimensionality. **[Right]** The used VAE-architecture. Here, "FC+Lin(N)" refers to a fully connected layer with linear activation.

It is trained on the original process configurations from the dataset to predict the outcome bending-angle in the sheet. Importantly, it has not been trained on the reconstructions of the VAE but on the original process configurations.

3.3 Computing Counterfactual Explanations

Generating counterfactuals represents an intuitive method for obtaining process configurations suggestions that produce desired bending-angles. As described in Eq. (1), the definition of counterfactuals comprises of (i) the deviation of a desired outcome and (ii) the cost of a counterfactual. This cost traditionally describes the deviation of the counterfactual to its original values and shall preserve the plausibility of the produced counterfactual [29]. As described in [7], alternatively to using the regularization term, plausible counterfactuals can be generated directly from the latent space of our VAE using gradient based methods that operate within the latent space. More VAE-approaches for generating counterfactuals exist [10,11,17]. They typically use the decoder of their VAE-architectures to directly generate counterfactuals. In this work, we proceed as follows: Let $\boldsymbol{\mu}$ be a mean sampled from the learned latent space and $\boldsymbol{D}$ be the decoder of our VAE. Furthermore, let $\boldsymbol{S}$ be the surrogate model. Then we search for the $\boldsymbol{\mu}$ that minimizes the squared difference between a desired bending-angle y_{cf} and the currently predicted outcome bending-angle for the decoded process configuration:

$$\widetilde{\boldsymbol{x}} = \boldsymbol{D}\left(\underset{\boldsymbol{\mu}\in\mathbb{R}^2}{\arg\min}\left(\boldsymbol{S}\big(\boldsymbol{D}(\boldsymbol{\mu})\big) - y_{\text{cf}}\right)^2\right). \tag{5}$$

The found process configuration $\widetilde{\boldsymbol{x}}$ represents the generated counterfactual for y_{cf}. We know that the composition $\boldsymbol{D}\circ\boldsymbol{S}$ is differentiable due to the fact that both $\boldsymbol{D}$ and $\boldsymbol{S}$ are differentiable neural networks. Hence, we can search for $\boldsymbol{\mu}$ via gradient based methods. In difference to [7], who compute only "one shift" in the latent space using pre-determined distances to exaggerate certain input-features, we iteratively shift $\boldsymbol{\mu}$ multiple times:

$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t - \eta\,\nabla_{\boldsymbol{\mu}_t}\big(\boldsymbol{S}\big(\boldsymbol{D}(\boldsymbol{\mu}_t)\big) - y_{\text{cf}}\big)^2, \tag{6}$$

Algorithm 1. Generating plausible counterfactuals using a VAE

Require: Decoder $\boldsymbol{D}$ of a VAE, training data $\mathcal{D}_{\text{train}}$, surrogate $\boldsymbol{S}$, target outcome y_{cf}, maximum allowed deviation ε, maximum amount of iterations $t_{\max}$; if early restart is desired, then early to achieve $\varepsilon_{\text{restart}} > \varepsilon$ and $t_{\text{restart}} < t_{\max}$

$\boldsymbol{\mu}_{\text{init}} = \mathbf{0}$
for each latent feature $\boldsymbol{\mu}_i$ in $\boldsymbol{\mu}_{\text{init}}$ **do** ▷ Sample initial $\boldsymbol{\mu}$
  $\boldsymbol{\mu}_i \sim \mathcal{U}\left(\max_{\boldsymbol{x}\in\mathcal{D}_{\text{train}}} \boldsymbol{E}_2(\boldsymbol{x};\,\phi,\Sigma_2)_i, \min_{\boldsymbol{x}\in\mathcal{D}_{\text{train}}} \boldsymbol{E}_2(\boldsymbol{x};\,\phi,\Sigma_2)_i\right)$
end for
$t = 0$, $\boldsymbol{\mu}_t = \boldsymbol{\mu}_{\text{init}}$
while $\left|\boldsymbol{S}\left(\boldsymbol{D}(\boldsymbol{\mu}_t)\right) - y_{\text{cf}}\right| \geq \varepsilon \wedge t < t_{\max}$ **do** ▷ Generate plausible counterfactual
  $\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t - \eta\,\nabla_{\boldsymbol{\mu}_t}\left(\boldsymbol{S}\left(\boldsymbol{D}(\boldsymbol{\mu}_t)\right) - y_{\text{cf}}\right)^2$
  $t = t + 1$
  if check for restart **then** ▷ Boolean hyperparameter
    if $\left|\boldsymbol{S}\left(\boldsymbol{D}(\boldsymbol{\mu}_t)\right) - y_{\text{cf}}\right| \geq \varepsilon_{\text{restart}} \wedge t \geq t_{\text{restart}}$ **then**
      Restart process from line 1
    end if
  end if
end while
return $\boldsymbol{D}(\boldsymbol{\mu}_t)$

using η as a correction rate. We stop iteration when the deviation of y_{cf} is small enough. More formally phrased, we stop when $|\boldsymbol{S}(\boldsymbol{D}(\boldsymbol{\mu}_t)) - y_{\text{cf}}| < \varepsilon$, where ε represents a hyperparameter. Eventually, the counterfactual $\widetilde{\boldsymbol{x}}$ is given by the last $\boldsymbol{D}(\boldsymbol{\mu}_t)$.

As for the determination of an initial $\boldsymbol{\mu}$, i.e., $\boldsymbol{\mu}_{\text{init}}$, we proceed as follows: We determine the minimal and maximal embedding values of each latent variable using the available training data:

$$\underline{\boldsymbol{\mu}_i} = \min_{\boldsymbol{x}\in\mathcal{D}_{\text{train}}} \boldsymbol{E}_2(\boldsymbol{x};\,\phi,\Sigma_2)_i,\ \overline{\boldsymbol{\mu}_i} = \max_{\boldsymbol{x}\in\mathcal{D}_{\text{train}}} \boldsymbol{E}_2(\boldsymbol{x};\,\phi,\Sigma_2)_i\ . \tag{7}$$

Then, for each latent feature $\boldsymbol{\mu}_i$, we randomly sample from its corresponding interval $[\underline{\boldsymbol{\mu}_i}, \overline{\boldsymbol{\mu}_i}]$ with uniform probability $\mathcal{U}$. This way, we encourage the optimization procedure to start in a region of the latent space where the VAE has seen training data. In case $\boldsymbol{\mu}_{\text{init}}$ produces process configurations that lead to outcome bending-angles which deviate too far from the desired target, we resample $\boldsymbol{\mu}_{\text{init}}$ and restart the process for some trials. If the problem persists for too many trials, we run the algorithm until a predefined maximum amount of iterations. Summarizing pseudocode for both, $\boldsymbol{\mu}$-sampling and generating the counterfactual, is shown in Algorithm 1.

4 Experiments

4.1 Experiment and Evaluation Setup

In this work, we consider a single-stage punch-bending operation that bends an input sheet vertically upwards and thus induces an outcome bending-angle into the sheet [16]. As stated before, a process configuration is represented by the Young's modulus of the sheet, the thickness of the input-sheet and the used bending-angle and -radius of the punch. This punch-bending operation

was repeatedly performed using a phenomenological model to produce 124 835 bending-angles for varying process configurations.

We split the dataset into a train-, validation and test split. The train-split of our dataset has a total of 93 627, the validation split 15 604 and the test split 200 samples. We randomly select one sample from the *test split*, considering a uniform probability distribution over all test samples. Using the absolute difference between outcome bending-angles, we subsequently remove the test sample's k-nearest neighbors in the *training split*. For each k, we train the VAE as well as the surrogate model using the downsized training splits. Hyperparameters are tuned using the static validation split.

We generate counterfactuals for each outcome of all samples in the unaltered test set using Algorithm 1 with $\varepsilon = 0.1$, $t_{\max} = 2\,000$ and a maximum 5 trials. We compute a counterfactual for each sample in the test set, using their outcome bending-angles as target y_{cf}, and compare those to the original test samples and their nearest neighbors in the dataset using the mean relative absolute difference:

Definition 1 (MRAD). *Let $\mathcal{D}$ be a dataset, f the index of the f-th feature, $\widetilde{\boldsymbol{X}} = \{\widetilde{\boldsymbol{x}}_i\}_{i=1}^N$ and $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^N \subseteq \mathcal{D}$ two feature sets which we want to compare and $\boldsymbol{X}_{nn} = \{\boldsymbol{x}_{i,nn}\}_{i=1}^N \subseteq \mathcal{D}$ the set of nearest-neighbors to elements in $\boldsymbol{X}$. We define **m**ean **r**elative **a**bsolute **d**ifference as:*

$$MRAD_{\mathcal{D}}^{f}\big(\widetilde{\boldsymbol{X}},\ \boldsymbol{X}\big) = d\big(\widetilde{\boldsymbol{X}},\ \boldsymbol{X}\big) - d\big(\boldsymbol{X},\ \boldsymbol{X}_{nn}\big),$$

$$\text{where}\quad d\left(\{\boldsymbol{a}_i\}_{i=1}^N,\ \{\boldsymbol{b}_i,\}_{i=1}^N\right) = \frac{\sum_{i=1}^N |\boldsymbol{a}_i^{(f)} - \boldsymbol{b}_i^{(f)}|}{N \cdot \left(\underset{\boldsymbol{x}\in\mathcal{D}}{max}\ \boldsymbol{x}^{(f)} - \underset{\boldsymbol{x}\in\mathcal{D}}{min}\ \boldsymbol{x}^{(f)}\right)}\ . \tag{8}$$

The first term measures the equality of the f-th feature from elements in $\widetilde{\boldsymbol{X}}$ to $\boldsymbol{X}$ while paying attention to the intra-dataset nearest neighbor feature difference with the second term. MRAD relies on the function d, which compares the mean absolute distance and normalizes it according to the observed feature range. In case MRAD becomes zero for f, the average difference of $\widetilde{\boldsymbol{X}}$ to $\boldsymbol{X}$ for feature f equals the average difference for feature f of nearest neighbor pairs in $\mathcal{D}$. In other words, $\widetilde{\boldsymbol{X}}$ behaves like nearest neighbors to $\boldsymbol{X}$ in $\mathcal{D}$ considering feature f.

In addition to MRAD, we also calculate the permutation feature importance [5] (PFI) for each trained surrogate model. Furthermore, we compare the absolute differences between the predicted outcome bending-angle $S(\widetilde{\boldsymbol{x}}_i)$ and target $y_{cf,i}$, between the closest available outcome bending-angle from the training data, i.e., the baseline-suggestion, and target $y_{cf,i}$ as well as between $S(\widetilde{\boldsymbol{x}}_i)$ and the ground-truth outcome bending-angle of the counterfactual's closest neighbor in the entire dataset. The experiment results are shown in Fig. 3.

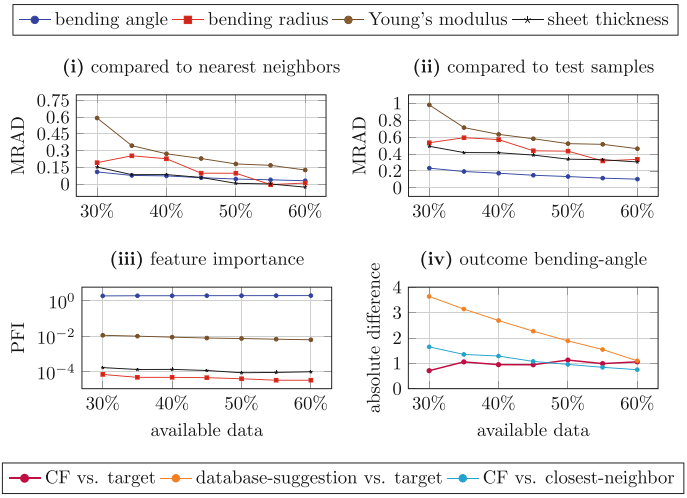


Fig. 3. Mean statistics over 125 repetitions. **(i,ii)** MRAD scores of counterfactuals (CF) to $\boldsymbol{X}$ = "nearest neighbors" and $\boldsymbol{X}$ = "original test samples". **(iii)** Feature importance scores. **(iv)** Absolute differences between outcome bending-angles.

4.2 Evaluation

First, we want to analyze the *plausibility* of the counterfactuals by investigating how the counterfactuals differ compared to their nearest neighbors in the dataset. Figure 3 **(i)** illustrates that given only 55% if the available training data, the feature difference of counterfactuals to their nearest neighbors in $\mathcal{D}$ appears very similar to feature differences among original nearest neighbor pairs in $\mathcal{D}$. Besides the Young's modulus, which is off by $\approx 1/7$-th of its feature range, all MRAD scores are nearly zero. The blue curve in Fig. 3 **(iv)** shows that the predictions of the counterfactuals differ by only 1° to the ground-truth outcome bending-angles of their nearest neighbors. Interestingly, Fig. 3 **(i)** and **(ii)** shows the counterfactual tool bending-angles align earlier closely with the data distribution compared to other features. The reason for that may lie in the fact that it is significantly more important for the outcome bending-angle, as depicted by Fig. 3 **(iii)**.

Next, we investigate *how the surrogate generalizes* by analyzing how the counterfactual values differ from the original test-sample values, whose outcome bending-angles we have selected to be the target values y_{cf} and whose k-nearest neighbors we have removed from the training data. Figure 3 **(ii)** indicates that the counterfactual process configurations differ from the original process configurations. Considering that the test samples and the counterfactuals cause *similar outcome bending-angles* although their *features differ* implies that one outcome bending-angle can be achieved using different process configurations. The counterfactuals therefore reveal that the surrogate model exploits a many-

to-one characteristic of punch-bending to explain outcome bending-angles which it did not see during training.

Eventually, we *compare the counterfactual's practicality to the baseline*. Considering the orange and purple curve in Fig. 3 **(iv)**, we can see that when only 50% of the training data is available, the outcome bending-angles of the counterfactuals deviate only by half as much from the target value compared to the deviations of the baseline, database-suggestions.

In conclusion, by generating plausible counterfactuals, i.e., counterfactuals that behave similar to the observable data-distribution, our experiments indicate that the surrogate model exploits the many-to-one characteristic of the underlying process to associate adequate process configurations to unseen outcome bending-angles. This signals evidence for generalizing behavior. Our counterfactual process configurations cause outcome bending-angles that typically lie closer to the target compared to simply selecting the closest outcome bending-angle from a database that might lack data, demonstrating their practicality.

4.3 Limitations and Future Work

Future work results as a cause of limitations of the conducted studies. For instance, the studied data did not contain information about usual former processes to punch-bending, such as unwinding the wire from its coil or the straightening process. Neither did we consider the situation of multiple subsequent punch-bending operations. For real practice, it would be interesting to study the nature of plausible counterfactuals given a more wholistic process in the form of a more comprehensive dataset.

Another issue is shown by our evaluation: The chosen model has problems in understanding local changes in the Young's modulus. Other data-formats than tabular data, such as 3D data that is being used in simulation, might incorporate more details that reveal process information that is not captured by tabular data. Hence, diving into the subject of representation learning with the help of more sophisticated architectures, could help in generating better counterfactuals.

5 Conclusion

In this work, we have investigated the generalization behavior of a surrogate model for punch-bending by examining generated counterfactuals and developed a method for obtaining suitable process configurations for non-observed outcome bending-angles. Key component in the introduced machine learning module is an algorithm that leverages the latent space of a VAE to produce plausible counterfactuals. We studied the generated counterfactuals to investigate how the surrogate model generalizes. Doing so, we have not only shown that the surrogate model exploits the many-to-one characteristic of the punch-bending process to explain unseen outcome bending-angles, but also found out that the counterfactuals represent more suitable process configurations for desired bending-angles compared to the suggestions of a database look-up approach.

Acknowledgement. The authors gratefully acknowledge the funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 520459685; 520459543.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Amor, A., Rachik, M., Sfar, H.: Combination of finite-element and semi-analytical models for sheet metal leveling simulation. KEM **473** (2011). https://doi.org/10.4028/www.scientific.net/KEM.473.182
2. Artelt, A., Hammer, B.: Convex density constraints for computing plausible counterfactual explanations. In: Farkaš, I., Masulli, P., Wermter, S. (eds.) ICANN 2020. LNCS, vol. 12396, pp. 353–365. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-61609-0_28
3. Ashraf, I., Strotherm, J., Hermes, L., Hammer, B.: Physics-informed graph neural networks for water distribution systems. In: AAAI, vol. 38 (2024). https://doi.org/10.1609/aaai.v38i20.30192
4. Bowman, S.R., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R., Bengio, S.: Generating sentences from a continuous space. In: CoNLL. ACL (2016). https://doi.org/10.18653/v1/K16-1002
5. Breiman, L.: Random forests. Mach. Learn. **45** (2001). https://doi.org/10.1023/A:1010933404324
6. Byrne, R.M.J.: Counterfactuals in explainable artificial intelligence (XAI): evidence from human reasoning. In: IJCAI (2019). https://doi.org/10.24963/ijcai.2019/876
7. Cohen, J.P., et al.: Gifsplanation via latent shift: a simple autoencoder approach to counterfactual generation for chest x-rays. In: MIDL, vol. 143. Proceedings of Machine Learning Research (2021). https://proceedings.mlr.press/v143/cohen21a.html
8. Gräler, M., Springer, R., Henke, C., Trächtler, A., Homberg, W.: Assisted setup of forming processes: compensation of initial stochastic disturbances. Procedia Manuf. **25** (2018). https://doi.org/10.1016/j.promfg.2018.06.104
9. Guidotti, R.: Counterfactual explanations and how to find them: literature review and benchmarking. Data Min. Knowl. Discov. **38** (2024). https://doi.org/10.1007/S10618-022-00831-6
10. Guo, H., Nguyen, T.H., Yadav, A.: Counternet: end-to-end training of prediction aware counterfactual explanations. In: KDD. Association for Computing Machinery (2023). https://doi.org/10.1145/3580305.3599290
11. Guyomard, V., Fessant, F., Guyet, T., Bouadi, T., Termier, A.: VCNet: a self-explaining model for realistic counterfactual generation. In: Amini, M.R., Canu, S., Fischer, A., Guns, T., Kralj Novak, P., Tsoumakas, G. (eds.) ECML PKDD 2022. LNCS, vol. 13713. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-26387-3_27
12. Haghighat, E., Raissi, M., Moure, A., Gomez, H., Juanes, R.: A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. Comput. Methods Appl. Mech. Eng. **379** (2021). https://doi.org/10.1016/j.cma.2021.113741
13. Higgins, I., et al.: beta-VAE: learning basic visual concepts with a constrained variational framework. In: ICLR (2017). https://openreview.net/forum?id=Sy2fzU9gl

14. Keane, M.T., Smyth, B.: Good counterfactuals and where to find them: a case-based technique for generating counterfactuals for explainable AI (XAI). In: Watson, I., Weber, R. (eds.) ICCBR 2020. LNCS (LNAI), vol. 12311, pp. 163–178. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58342-2_11
15. Kingma, D.P.: Auto-encoding variational bayes. arXiv (2013). https://arxiv.org/abs/1312.6114
16. Koller, L., Hartmann, C., Martinitz, L., Volk, W., Althoff, M.: Training robust neural networks for uncertainty prediction in stamping technology. AUTO **73** (2025). https://doi.org/10.1515/auto-2024-0132
17. Mahajan, D., Tan, C., Sharma, A.: Preserving causal constraints in counterfactual explanations for machine learning classifiers. arXiv (2019). http://arxiv.org/abs/1912.03277
18. Markmann, T., Straat, M., Hammer, B.: Koopman-based surrogate modelling of turbulent rayleigh-bénard convection. In: IJCNN. IEEE (2024). https://doi.org/10.1109/IJCNN60899.2024.10651496
19. Molnar, C.: Interpretable Machine Learning, 3 edn (2025). https://christophm.github.io/interpretable-ml-book
20. Paech, M.: Advanced semi-automatic straightening technology. WJI **41** (2008)
21. Peters, H., et al.: Novel approach for data-driven modelling of multi-stage straightening and bending processes. MRP **41** (2024). https://doi.org/10.21741/9781644903131-252
22. Poyiadzi, R., Sokol, K., Santos-Rodriguez, R., De Bie, T., Flach, P.: Face: feasible and actionable counterfactual explanations. In: AIES. Association for Computing Machinery (2020). https://doi.org/10.1145/3375627.3375850
23. Richter, K., Reuther, F., Müller, R., Landgrebe, D.: Investigating the influence of bending parameters on the springback behavior of ultra-high strength spring strips. In: MSF, vol. 918. Trans Tech Publ (2018). https://doi.org/10.4028/www.scientific.net/MSF.918.125
24. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation, vol. 1. MIT Press (1986)
25. Straat, M., Markmann, T., Hammer, B.: Solving turbulent rayleigh-bénard convection using fourier neural operators. ESANN (2025). https://www.esann.org/sites/default/files/proceedings/2025/ES2025-131.pdf
26. Tripathy, R.K., Bilionis, I.: Deep UQ: learning deep neural network surrogate models for high dimensional uncertainty quantification. J. Comput. Phys. **375** (2018). https://doi.org/10.1016/j.jcp.2018.08.036
27. Van Looveren, A., Klaise, J.: Interpretable counterfactual explanations guided by prototypes. In: Oliver, N., Pérez-Cruz, F., Kramer, S., Read, J., Lozano, J.A. (eds.) ECML PKDD 2021. LNCS (LNAI), vol. 12976, pp. 650–665. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86520-7_40
28. Verma, S., Boonsanong, V., Hoang, M., Hines, K., Dickerson, J., Shah, C.: Counterfactual explanations and algorithmic recourses for machine learning: a review. ACM Comput. Surv. **56** (2024). https://doi.org/10.1145/3677119
29. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: automated decisions and the GDPR. Harv. JL Tech. **31** (2017). https://arxiv.org/abs/1711.00399
30. Wielopolski, P., Furman, O., Stefanowski, J., Zięba, M.: Probabilistically plausible counterfactual explanations with normalizing flows. arXiv (2024). https://arxiv.org/abs/2405.17640

NGAT: A Node-Level Graph Attention Network for Long-Term Stock Prediction

Yingjie Niu[1,2(✉)], Mingchuan Zhao[3], Valerio Poti[4], and Ruihai Dong[1,2]

[1] School of Computer Science, University College Dublin, Dublin, Ireland
yingjie.niu@ucdconnect.ie, ruihai.dong@ucd.ie
[2] SFI Centre for Research Training in Machine Learning, Dublin, Ireland
[3] School of Mathematical Sciences, Dublin City University, Dublin, Ireland
mingchuan.zhao@dcu.ie
[4] Michael Smurfit Business School, University College Dublin, Dublin, Ireland
valerio.poti@ucd.ie

Abstract. Graph representation learning methods have been widely adopted in financial applications to enhance company representations by leveraging inter-firm relationships. However, current approaches face three key challenges: (1) The advantages of relational information are obscured by limitations in downstream task designs; (2) Existing graph models specifically designed for stock prediction often suffer from excessive complexity and poor generalization; (3) Experience-based construction of corporate relationship graphs lacks effective comparison of different graph structures. To address these limitations, we propose a long-term stock prediction task and develop a Node-level Graph Attention Network (NGAT) specifically tailored for corporate relationship graphs. Furthermore, we experimentally demonstrate the limitations of existing graph comparison methods based on model downstream task performance. Experimental results across two datasets consistently demonstrate the effectiveness of our proposed task and model. The project is publicly available on GitHub to encourage reproducibility and future research. (https://github.com/FreddieNIU/NGAT).

Keywords: Graph Neural Network · Stock Prediction

1 Introduction

Graph Neural Networks (GNNs) have become the dominant approach for graph representation learning due to their message-passing mechanisms, which aggregate neighboring nodes to learn contextual embeddings reflecting graph structures. This success has spurred interest in applying graph representation learning to stock prediction tasks, where companies are modeled as nodes with inter-firm relationships as edges [20,30] to capture momentum spillover effects. Existing studies typically follow a common paradigm: 1) modeling corporate relationships to construct edges, 2) applying GNNs to Corporate Relationship Graphs

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 204–215, 2026.
https://doi.org/10.1007/978-3-032-04555-3_17

(CRGs) for node embedding aggregation, and 3) evaluating performance through downstream tasks. However, this paradigm presents three key challenges:

First, momentum spillover often exhibits lagged effects, where historical trends of one stock influence another's future trends over uncertain periods (potentially exceeding one day). Existing graph-based stock prediction studies predominantly focus on next-day trend prediction [26,31] with single-day prediction horizons, which may fail to fully exploit graph models' capabilities. Furthermore, most models emphasize stock returns while neglecting volatilityâĂŤa critical financial risk indicator essential for robust market analysis. To bridge these gaps, we introduce a long-term stock return/volatility prediction task that forecasts average returns or volatility over future periods using historical data and relational graphs. This formulation better aligns with momentum spillover characteristics and provides actionable insights for long-term investors.

Second, general GNN architectures like Graph Convolutional Networks (GCNs) and Graph Attention Networks (GATs), despite their generalization capabilities, may not directly suit CRGs. In conventional homogeneous graphs (e.g., Cora dataset [2]), nodes share similar feature distributions within categories, making shared attention mechanisms effective. However, in CRGs where each node represents a distinct company with unique stock behavior patterns, assuming identical distributions across nodes becomes untenable. While specialized architectures for stock prediction exist [6,12], their complexity often compromises generalizability. To address this, we propose a Node-level Graph Attention Network (NGAT) that assigns each company a unique attention mechanism, enabling tailored modeling of CRGs. Crucially, NGAT operates as a portable graph convolution layer adaptable to diverse graph-based tasks and datasets.

Third, the construction of CRGs predominantly relies on experience driven approaches, with no consensus on optimal methodologies. We compare the performance of identical GNN models under different graph constructions, demonstrating the limitations of relying solely on downstream task performance for graph quality assessment.

Our main contributions are summarized as follows:

- Designed a novel long-term stock prediction task that better captures momentum spillover effects;
- Developed a highly generalizable Node-level Graph Attention Network (NGAT) that better suits corporate relationship graphs (CRGs);
- Demonstrated empirically the limitations of model-based graph comparison;
- Validated by experimental results the practical utility of our task and demonstrate NGAT's superior performance over baselines across multiple tasks.

2 Related Work

Existing research on stock prediction evolves along two key dimensions: (1) task formulation progressing from traditional econometric modelling to machine learning paradigms, and (2) architectural innovation transitioning from generic models to financial-specific adaptations.

The foundation of stock prediction rests on econometric models including factor analysis [4,11], time-series approaches [21,24], and volatility modelling via GARCH variants [10,22] and HAR-RV [8]. Machine learning advancements introduced hybrid models combining GARCH with neural networks [9,15] and sequence models like LSTM [14,25,27]. While these methods initially focused on next-day prediction [31], recent works explore ranking tasks [12], overnight trend prediction [20], and price-limit hitting forecasting [30]. Our proposed long-term return/volatility prediction task overcomes the limitation that next-day prediction cannot fully exploit the benefit of learning momentum spillovers by jointly modeling multi-period spillovers and risk factors through corporate relationship graphs.

The integration of alternative data sources—particularly textual information [1,3]—motivated graph-based approaches using GCN [17] and GAT [28] to model inter-firm relationships via price correlations [20], industry taxonomies [12], or news co-occurrences [23]. Subsequent domain-specific adaptations introduced temporal-GNNs [29], AD-GAT [6], and hierarchical GNN architectures [30], but at the cost of increased complexity and reduced generalizability. Our NGAT architecture resolves this trade-off through node-specific attention mechanisms while enabling portable deployment.

3 Problem Formulation

To overcome the limitations of the next-day trend prediction task, we introduce the long-term return mean/volatility prediction tasks in this paper. The objectives of these tasks are to forecast the stock return trend and volatility over the future T business days. The return trend prediction is framed as a classification task, aiming to predict whether the average return over the upcoming period will be higher or lower than in the previous period. The daily log return of stock s on day d is calculated as: $r_d^s = log(p_d^s/p_{d-1}^s)$ where p_d^s denotes the adjusted closing price of stock s at day d. The classification label $y_{d,c}^s$ is calculated as:

$$y_{d,c}^s = \mathbf{1}(\frac{1}{T}\sum_{i=d+1}^{d+T} r_i^s > \frac{1}{T}\sum_{i=d-T+1}^{d} r_i^s), \tag{1}$$

where T is the fixed period length. This task allows for the categorization of return trends into the following four scenarios:

- (LP, N+): **L**ast period **P**ositive return (LP) followed by an even **higher** return in the next period (N+), indicating a "surge."
- (LN, N+): **L**ast period **N**egative return (LN) followed by a **higher** return in the next period (N+), indicating a "rebound."
- (LP, N−): **L**ast period **P**ositive return (LP) followed by a **lower** return in the next period (N-), indicating a "pullback."
- (LN, N−): **L**ast period **N**egative return (LN) followed by an even **lower** return in the next period (N-), indicating a "plunge."

On the other hand, volatility prediction is treated as a regression task, focused on forecasting the exact value of the volatility over the upcoming period. In our experiment, we use the realized sample standard deviation to estimate the volatility, which is calculated as follows:

$$y_{d,r}^{s} = \sqrt{\frac{1}{T}\sum_{i=d+1}^{d+T}(r_i^s - \mu)^2}, \quad \mu = \frac{1}{T}\sum_{i=d+1}^{d+T} r_i^s, \tag{2}$$

where μ is the mean of log returns in the period $[d+1, d+T]$.

The N+ scenarios suggest market opportunities, while N− scenarios signal downside risks. Specifically, identifying (LP, N+) points to a likely surge, recommending a "hold" or "increase holdings" strategy. Conversely, (LN, N−) serves as a warning for potential loss, advising "sell" or "short sell" strategies. By understanding these scenarios, investors can more accurately assess future stock price trends. Furthermore, integrating predictions of long-term trends with volatility forecasts enables a more robust evaluation of the reliability and associated risks of purported "opportunities". Together, these predictions provide a more comprehensive understanding of market dynamics, thereby enhancing long-term investment strategy formulation.

4 Model Architecture

In this section, we describe the proposed model architecture, including the sequential embedding module, relational embedding module, and output layer.

4.1 Sequential Embedding

To capture the temporal dependencies in stock movements, researchers commonly encode historical trading data using RNNs to generate sequential embeddings for stock prediction [5,7,12,18,19]. Following the common practice, LSTM is selected as the sequential embedding module in this paper. Given a set of assets $\mathbf{S}$, to generate the sequential embedding for stock $s \in \mathbf{S}$ on day d, we fed a time-series feature matrix X_d^s into the LSTM module.

$$h_d^s = LSTM(X_d^s), \tag{3}$$

where h_d^s is the sequential embedding of stock s on day d and $X \in \mathbb{R}^{\Delta d \times F}$. Δd is a fixed lag size, and F is the number of features. We use the output of the last time step as the sequential embeddings.

4.2 Relational Embedding

Beyond temporal dependencies on historical transaction data, a stock's trend is influenced by related stocks. Thus, we fuse the sequential embeddings using stock relationships to generate the corresponding relational embedding.

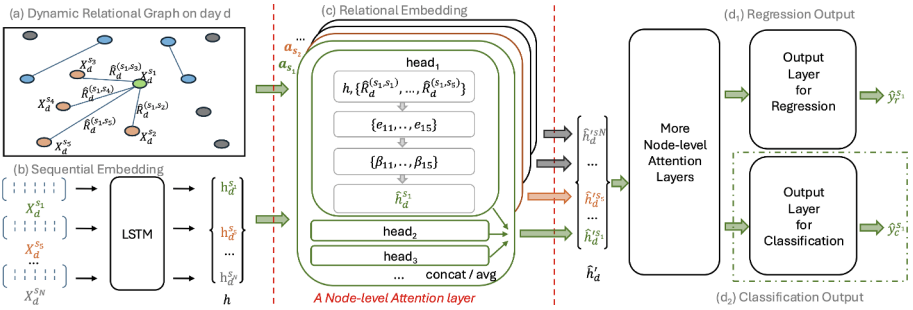


Fig. 1. Node-level Graph Attention Network model architecture and calculating pipeline.

Relation Building. Edges are constructed based on firm co-occurrences in news/tweets. For each article n_d^k (the k-th news on day d), we extract its mentioned firm set S, then assign $R_d^{(s_i,s_j),k} = 1$ for all $(s_i, s_j) \in S \times S$. Daily aggregation yields unconditional relation attributes: $R_d^{(s_i,s_j)} = \sum_{k=1}^{K} R_d^{(s_i,s_j),k}$ where K is the daily news count. Each node receives a self-loop with the attribute: $R_d^{(s_i,s_i)} = \max_{j \in \mathbb{N}_i} R_d^{(s_i,s_j)}$ where $\mathbb{N}_i$ is the neighbors of s_i. To model persistent news impacts, we compute *conditional relations* via temporal weighted aggregation:

$$\hat{R}_d^{(s_i,s_j)} = \sum_{m=0}^{\delta-1} w_m \cdot R_{d-m}^{(s_i,s_j)}, \quad w_m = \frac{\delta - m}{\sum_{n=1}^{\delta} n}, \tag{4}$$

where δ is the memory window size, and w_m is the aggregated weight to apply on the unconditional attribute on the day $d - m$, implementing a forgetting mechanism that prioritizes recent days.

Node-level Graph Attention Layer. As illustrated in Fig. 1, our NGAT layer transforms inputs (the sequential embeddings) to outputs through the following steps. Let $\mathbf{h} = \{h_d^1, ..., h_d^N\}$ $(h_d^i \in \mathbb{R}^F)$ denote the sequential embeddings of N nodes, with time subscript d omitted when unambiguous. First, each node undergoes three linear projections parameterized by $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{F' \times F}$. We then compute *masked attention* coefficients for connected nodes:

$$e_{ij} = \hat{R}_d^{(i,j)} \cdot a_i(\mathbf{W}_q h^i, \mathbf{W}_k h^j), \quad j \in \mathbb{N}_i, \tag{5}$$

where a_i denotes node i's attention mechanism and $\hat{R}_d^{(i,j)}$ the conditional relation attribute. We normalize attention weights via:

$$\beta_{ij} = \text{Softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{m \in \mathbb{N}_i} \exp(e_{im})}. \tag{6}$$

The attention mechanism a employs node-specific parameters $\mathbf{a}_i \in \mathbb{R}^{2F' \times F'}$ (aggregated as $\mathbf{a} \in \mathbb{R}^{N \times 2F' \times F'}$), enabling unique computation per node. Incorpo-

rating Eqs. 5-6 with LeakyReLU (slope=0.2), the final coefficients derive from:

$$\beta_{ij} = \frac{\exp(\hat{R}_d^{(i,j)} LeakeyReLU(\mathbf{a}_i^T[\mathbf{W}_q h^i || \mathbf{W}_k h^j]))}{\sum_{m \in \mathbb{N}_i} \exp(\hat{R}_d^{(i,m)} LeakeyReLU(\mathbf{a}_i^T[\mathbf{W}_q h^i || \mathbf{W}_k h^m]))}, \quad (7)$$

where $.^T$ represents transposition, $||$ represents the concatenation operation, and $\mathbf{a}_i \in \mathbb{R}^{2F' \times F'}$ represents the attention matrix corresponding to node i.

The attention coefficients generate *relational embeddings* through neighborhood aggregation. While the Softmax normalization (Eq. 6) ensures comparable coefficients, its smoothing effect may cause over-smoothing by over-emphasizing neighborhood information. To preserve node-specific patterns, we concatenate the transformed sequential embedding $\mathbf{W}_v h^i$ with the relational embedding $\sum_{j \in \mathbb{N}_i} \beta_{ij} \mathbf{W}_v h^j$, followed by a nonlinear projection:

$$\hat{h}^i = \sigma\left(\mathbf{W}\left[\mathbf{W}_v h^i \parallel \sum_{j \in \mathbb{N}_i} \beta_{ij} \mathbf{W}_v h^j\right]\right), \quad \mathbf{W} \in \mathbb{R}^{F' \times 2F'}, \quad (8)$$

where $\parallel$ denotes concatenation. Following [28], multi-head attention (M independent heads) stabilizes learning via concatenated/averaged outputs. If concatenate:

$$\hat{h}'^i = \|_{m=1}^{M} \sigma(\mathbf{W}^{\mathbf{m}}[\mathbf{W}_v^m h^i || \sum_{j \in \mathbb{N}_i} \beta_{ij}^m \mathbf{W}_v^m h^j]), \quad (9)$$

where β_{ij}^m is the attention coefficient of the m-th attention head. If average:

$$\hat{h}'^i = \sigma(\frac{1}{M} \sum_{m=1}^{M} \mathbf{W}^{\mathbf{m}}[\mathbf{W}_v^m h^i || \sum_{j \in \mathbb{N}_i} \beta_{ij}^m \mathbf{W}_v^m h^j]). \quad (10)$$

4.3 Output Layer

The output layer parameterized by a weight matrix $\mathbf{W}_o$ is implemented as follows:

$$\hat{y}^i = \begin{cases} \mathbf{W}_o \hat{h}'^i & \text{(regression)} \\ \text{Sigmoid}(\mathbf{W}_o \hat{h}'^i) & \text{(classification).} \end{cases} \quad (11)$$

Loss Function. For regression task, the Mean Square Error loss between y_r^i and $\hat{y}_r^i$ is used to back-propagate and learn the model parameters. The Binary Cross Entropy loss between y_c^i and $\hat{y}_c^i$ is used instead for the classification task.

5 Experiments

In this section, we detail our datasets, training setup, baseline models, and evaluation metrics.

5.1 Data

We use two publicly available daily based datasets. The SPNews Dataset [23] includes news and transaction data for S&P500 companies, with daily bulletins listing the latest eight news items per company. To construct accurate relational graphs, redundant repetitions across days and companies are removed, ensuring each news item is recorded once for all involved companies. The ACL2018 Dataset [31] contains tweets and transaction data spanning two years. Text data is derived from Twitter posts, where edges between companies are defined by the co-occurrence of ticker tags (e.g. #aapl, #goog). Repeated mentions of the same company within a tweet, often used for emphasis, are removed to ensure accurate co-occurrence measurement.

The numerical data in both datasets include daily trading information for the companies involved, such as open (O), high (H), low (L), close (C), adjusted close (A), and volume (V). A three-step regularization process is implemented: (a) Open prices are standardized via $O' = \frac{O-\mathbb{E}(O)}{\sigma(O)}$, where $\mathbb{E}(\cdot)$ and $\sigma(\cdot)$ denote the mean and standard deviation, calculated individually for each stock. (b) High, low, and close prices are expressed as rates of change relative to open prices: $H' = \frac{H-O}{O}$, $L' = \frac{L-O}{O}$, $C' = \frac{C-O}{O}$. (c) Adjusted close prices are converted into stock returns. This process standardizes prices across stocks and reduces information redundancy due to collinearity between open prices and other variables.

5.2 Training Setup

All parameters are selected via grid search on validation performance: *Graph Construction*: Feature dimensions $F \in [8, 12]$; memory window $\delta \in [1, 2, 3, 5, 7, 10]$ (optimal $\delta = 5$); relation threshold $\in [0, 1]$; forecasting horizon $T \in [1, 5, 10, 21]$ (business days: 1d/1w/2w/1mo). *Model Architecture*: Attention heads $\in [1, 4, 8]$; hidden size $F' \in [16, 32, 64]$; dropout $\in [0, 0.2, 0.4, 0.6]$; fixed graph layers $= 1$. *Training*: Adam optimizer ($lr = 10^{-4}$, weight decay 5×10^{-4}) with Glorot initialization [13,16]. Fixed estimation window $\Delta d = 21$.

5.3 Baselines

We compare the proposed NGAT model with both non-graph-based models and graph-based models.

LSTM: A single-layer Long Short-term Memory (LSTM) model.

GCN [17]: with one sequential embedding layer and one convolutional layer.

GAT [28]: with one sequential embedding layer and self-attention layer.

LSTM+GCN [5]: The results of GCN and LSTM are concatenated for prediction.

TGC [12]: A Temporal Graph Convolutional Network (TGC).

AD-GAT [6]: Attribute-Driven Graph Attention Network captures the attribute-sensitive momentum spillovers of the neighbour nodes. We use the implementation of AD-GAT from the original paper.

5.4 Evaluation Metrics

For classification models predicting return trends, we use overall accuracy (ACC), the Matthews Correlation Coefficient (MCC), and the area under the ROC curve (AUC). ACC and AUC are commonly used in classification tasks, while MCC provides a fair evaluation when class distributions are imbalanced. Given the confusion matrix $\begin{pmatrix} TP & TN \\ FP & FN \end{pmatrix}$, where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively, MCC is computed as:

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN))}}. \tag{12}$$

For regression models predicting future volatility, we use out-of-sample R^2 and Mean Squared Error (MSE), both of which are commonly used evaluation metrics. To ensure comprehensive evaluation, we compute the average metrics across all companies in each dataset: 76 companies in ACL2018 and 268 companies in SPNews.

6 Results and Discussion

Return Trend. Table 1 demonstrates NGAT's superior performance in return mean classification across both datasets. Notably, NGAT achieves significant improvements over GAT, with relative gains of 4.03%, 5.90%, and 2.07% for $T = 21$ on ACL2018, and 2.08%, 8.12%, and 1.41% for $T = 10$ on SPNews. These results validate the universal superiority of our node-level attention mechanism in financial modeling. Crucially, all models exceed 70% accuracy âĂŞ substantially outperforming traditional next-day prediction benchmarks (typically $50 - 60\%$) âĂŞ confirming the practical value of long-term forecasting. Additionally, high MCC and AUC values indicate balanced performance across classes, further validating effectiveness of prediction. The underperformance of AD-GAT stems from its exclusion of textual embeddings, unlike its original design.

Our four market scenarios (Sect. 3) reveal strategic insights: While NGAT shows slightly weaker "rebound" (LN, N+) prediction on ACL2018, it excels in identifying "surge" (LP, N+) and "plunge" (LN, N−) opportunities âĂŞ critical for crisis avoidance and profit capture. On SPNews, NGAT achieves balanced performance with 3.2% N+ accuracy over LSTM, despite minor N− tradeoffs, demonstrating robust pattern recognition across market phases.

Volatility Analysis. As shown in Table 3, NGAT consistently outperforms all baselines in volatility forecasting across both datasets. For ACL2018, graph-based models demonstrate clear advantages over sequential models, validating the importance of relational information. On SPNews, while text-derived graphs show limited improvements due to potential noise in news-to-graph translation, NGAT maintains robust performance, confirming its ability to extract reliable

Table 1. Return Mean Prediction Performance (Five-run average). Notations: *r-mean*$_X$ denotes T = X classification task (e.g., *r-mean1*: T = 1).

Dataset	Model	r-mean1			r-mean5			r-mean10			r-mean21		
		ACC	MCC	AUC	ACC	MCC	AUC	ACC	MCC	AUC	ACC	MCC	AUC
ACL2018	LSTM	0.7345	0.3734	0.8291	0.7445	0.3763	0.7933	0.7358	0.3271	0.7643	0.6963	0.3609	0.7994
	GCN	0.7211	0.3184	0.8018	0.7466	0.3447	0.7734	0.7516	0.2992	0.7415	0.7045	0.3289	0.7510
	GAT	0.7361	0.3792	0.8357	0.7611	0.4095	0.8111	0.7521	0.3586	0.7822	0.6789	0.3579	0.7918
	LSTM+GCN	0.7379	**0.3852**	0.8317	0.7555	0.3928	0.7986	0.7476	0.3465	0.7648	0.7000	0.3727	0.8030
	TGC	0.7018	0.2788	0.7806	0.7550	0.3480	0.7786	0.7513	0.3134	0.7486	0.6971	**0.3846**	0.7992
	AD-GAT	0.6571	0.2105	0.7147	0.7313	0.2996	0.7497	0.7211	0.2757	0.7046	0.6553	0.2921	0.7375
	NGAT (Our)	**0.7395**	0.3825	**0.8370**	**0.7774**	**0.4150**	**0.8172**	**0.7745**	**0.3652**	**0.7849**	**0.7063**	0.3790	**0.8082**
SPNews	LSTM	0.7569	0.4754	0.8345	0.7449	0.4643	0.8263	0.7195	0.4098	0.8151	0.7287	0.3920	0.7999
	GCN	0.7571	0.4738	0.8341	0.7418	0.4594	0.8232	0.7230	0.4175	0.8161	0.7348	0.4126	0.8083
	GAT	0.7582	0.4764	0.8388	0.7421	0.4589	0.8256	0.7233	0.4127	0.8210	0.7336	0.4039	0.8053
	LSTM+GCN	0.7589	0.4797	0.8382	0.7381	0.4535	0.8219	0.7210	0.4078	0.8147	0.7374	0.4162	0.8016
	TGC	0.7593	0.4727	0.8391	0.7363	0.4500	0.8220	0.7221	0.4137	0.8193	0.7358	0.3921	0.8015
	AD-GAT	0.6861	0.3003	0.7397	0.6870	0.3485	0.7676	0.6911	0.3621	0.7886	0.7151	0.3421	0.7537
	NGAT (Our)	**0.7627**	**0.4835**	**0.8407**	**0.7504**	**0.4711**	**0.8296**	**0.7384**	**0.4462**	**0.8326**	**0.7397**	**0.4168**	**0.8142**

Table 2. Accuracy of predicting return trends in different scenario groups.

Dataset	Model	r-mean5		r-mean10		r-mean21	
ACL2018	LSTM		**LN**	**LP**	**LN**	**LP**	**LN**	**LP**
		N+	611/801 (0.7627)	719/893 (0.8051)	**N+**	487/630 (0.7730)	734/911 (0.8057)	**N+**	369/467 (0.7901)	495/633 (0.7819)
		N-	764/1039 (0.7353)	735/1067 (0.6884)	**N-**	724/996 (0.7269)	851/1263 (0.6737)	**N-**	713/1027 (0.6942)	1069/1673 (0.6389)
	NGAT		**LN**	**LP**	**LN**	**LP**	**LN**	**LP**
		N+	622/801 (0.7765)	731/893 (0.8186)	**N+**	465/630 (0.7380)	750/911 (0.8232)	**N+**	355/467 (0.7601)	491/633 (0.7756)
		N-	831/1039 (0.7998)	770/1067 (0.7216)	**N-**	794/996 (0.7971)	934/1263 (0.7395)	**N-**	711/1027 (0.6923)	1091/1673 (0.6512)
SPNews	LSTM		**LN**	**LP**	**LN**	**LP**	**LN**	**LP**
		N+	1592/2379 (0.6692)	1511/2433 (0.6210)	**N+**	1504/2461 (0.6111)	1076/1906 (0.5645)	**N+**	1678/2667 (0.6291)	648/1106 (0.5858)
		N-	2124/2537 (0.8372)	2359/2835 (0.8321)	**N-**	2121/2610 (0.8126)	2626/3207 (0.8188)	**N-**	2714/3445 (0.7956)	2354/2966 (0.7936)
	NGAT		**LN**	**LP**	**LN**	**LP**	**LN**	**LP**
		N+	1734/2379 (0.7288)	1648/2433 (0.6773)	**N+**	1694/2461 (0.6888)	1269/1906 (0.6657)	**N+**	1844/2667 (0.6914)	624/1106 (0.5804)
		N-	2040/2537 (0.8040)	2220/2835 (0.7830)	**N-**	2035/2610 (0.7796)	2522/3207 (0.7864)	**N-**	2667/3445 (0.7741)	2380/2966 (0.8024)

Table 3. Return Std Forecasting Performance (Five-run average). Notations: *std*$_X$ denotes T = X.

Model	ACL 2018						SP News					
	std5		std10		std21		std5		std10		std21	
	R^2	MSE	R^2	MSE	R^2	MSE	R^2	MSE	R^2	MSE	R^2	MSE
LSTM	0.04	0.39	0.26	0.24	0.35	0.22	0.16	0.66	0.27	0.44	0.36	0.31
GCN	0.05	0.38	0.27	0.23	0.39	0.20	0.15	0.66	0.26	0.45	0.33	0.33
GAT	0.06	0.38	0.30	0.22	0.40	0.19	0.13	0.67	0.24	0.46	0.33	0.33
LSTM+GCN	0.03	0.39	0.26	0.24	0.36	0.21	0.15	0.66	0.26	0.44	0.34	0.32
TGC	0.01	0.40	0.23	0.25	0.35	0.22	0.14	0.68	0.24	0.47	0.28	0.36
AD-GAT	-0.04	0.42	0.20	0.25	0.36	0.19	0.04	0.75	0.15	0.52	0.23	0.38
NGAT (Our)	**0.07**	**0.38**	**0.31**	**0.22**	**0.45**	**0.17**	**0.19**	**0.63**	**0.31**	**0.42**	**0.38**	**0.30**

signals even from imperfect relational data. The superiority of NGAT over other graph architectures highlights the effectiveness of its node-level attention mechanisms in interpreting financial relationships. Notably, monthly volatility prediction (T = 21) achieves the highest performance across all three settings.

Table 4. Performances comparison of different graph construction methods on SPNews Dataset. Number in parathesis means memory window δ

Graph Construction Method	GAT						NGAT					
	std5		std10		std21		std5		std10		std21	
	R^2	MSE	R^2	MSE	R^2	MSE	R^2	MSE	R^2	MSE	R^2	MSE
Static Graph	0.1383	0.6799	0.2453	0.4616	0.3350	0.3334	0.1923	0.6448	0.3102	0.4258	0.3746	0.3150
Correlation (5 day)	0.1008	0.7081	0.1878	0.4955	0.2645	0.3668	0.1836	0.6514	0.3014	0.4317	0.3607	0.3221
Cooccurence (5 day)	**0.1550**	**0.6703**	**0.2704**	**0.4492**	**0.3505**	**0.3271**	**0.1999**	**0.6382**	**0.3128**	0.4239	**0.3771**	**0.3132**
Correlation (21 day)	0.0603	0.7434	0.1133	0.5402	0.1773	0.4113	0.1869	0.6491	0.3069	**0.4181**	0.3695	0.3170
Cooccurence (21 day)	0.1175	0.6944	0.2109	0.4814	0.2856	0.3579	0.1872	0.6489	0.2991	0.4332	0.3591	0.3218

This superiority aligns with financial practicality: monthly volatility (T = 21) holds broader portfolio applications than noisy short-term measures (T = 5/10). Combining the results from Table 1 and Table 3, aside from the NGAT model, no graph-based model (including both generic graph architectures and finance-specific SOTA variants) has demonstrated comprehensive superiority over the standalone LSTM sequence model across all forecast horizons in both tasks. This phenomenon arises because different prediction horizons require capturing distinct momentum spillover effects. For instance, highly connected companies (those with numerous edges) may experience varying influential counterparts across forecast horizons, whereas sparsely connected entities might show minimal horizon-dependent variations. This renders shared feature aggregation mechanisms inefficient, ultimately proving the limitations of inter-node shared feature aggregation while validating the effectiveness of node-level attention mechanisms in adapting to temporal dynamics. It is also noteworthy that, as the forecast horizon T increases, performance on return trend prediction slightly declines, while volatility prediction improves significantly. Thus, setting $T = 21$ provides a balance between performance on both tasks. Investors can make decisions based on predicted return trends and volatility, tailored to their risk preferences over the selected forecast horizon (Table 2).

Graph Construction Methods. We compare text-derived dynamic graphs (news/tweets co-occurrence) against conventional return correlation graphs and static baselines (Table 4), revealing three findings: 1) NGAT consistently surpasses GAT across all graph types, demonstrating the architectural advantage of node-specific attention mechanisms; 2) Memory length optimization proves critical, with 5-day windows maximizing performance âĂŞ particularly for news-based graphs where outdated content diminishes market relevance; 3) Downstream evaluation alone provides incomplete graph quality assessment: While GAT favours co-occurrence graphs over correlation-based ones (suggesting better graph quality), this distinction diminishes under NGAT due to its ability to mitigate graph construction imperfections âĂŞ demonstrating how advanced architectures can obscure inherent graph deficiencies.

7 Conclusion

This paper proposes the Node-level Graph Attention Network (NGAT) for modelling corporate relationship graphs, along with novel stock return/volatility prediction tasks extensible to long-term horizons. Extensive experiments validate NGAT's performance advantages and the practical utility of long-term forecasting. Though NGAT incurs higher computational complexity, this remains acceptable under fixed asset universe settings common in institutional portfolios.

Acknowledgments. This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 18/CRT/6183. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

1. Atkins, A., Niranjan, M., Gerding, E.: Financial news predicts stock market volatility better than close price. J. Finance Data Sci. **4**(2), 120–137 (2018)
2. Bojchevski, A., Günnemann, S.: Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. arXiv preprint arXiv:1707.03815 (2017)
3. Bollen, J., Mao, H., Zeng, X.: Twitter mood predicts the stock market. J. Comput. Sci. **2**(1), 1–8 (2011)
4. Carhart, M.M.: On persistence in mutual fund performance. J. Financ. **52**(1), 57–82 (1997)
5. Chen, Y., Wei, Z., Huang, X.: Incorporating corporation relationship via graph convolutional neural networks for stock price prediction. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 1655–1658 (2018)
6. Cheng, R., Li, Q.: Modeling the momentum spillover effect for stock prediction via attribute-driven graph attention networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 55–62 (2021)
7. Cho, K., et al.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
8. Corsi, F.: A simple approximate long-memory model of realized volatility. J. Financ. Economet. **7**(2), 174–196 (2009)
9. Di Persio, L., Garbelli, M., Wallbaum, K.: Forward-looking volatility estimation for risk-managed investment strategies during the covid-19 crisis. Risks **9**(2) (2021)
10. Engle, R.F., Ng, V.K.: Measuring and testing the impact of news on volatility. J. Financ. **48**(5), 1749–1778 (1993)
11. Fama, E.F., French, K.R.: A five-factor asset pricing model. J. Financ. Econ. **116**(1), 1–22 (2015)
12. Feng, F., He, X., Wang, X., Luo, C., Liu, Y., Chua, T.S.: Temporal relational ranking for stock prediction. ACM Trans. Inf. Syst. (TOIS) **37**(2), 1–30 (2019)
13. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–256. JMLR Workshop and Conference Proceedings (2010)

14. Jin, Z., Yang, Y., Liu, Y.: Stock closing price prediction based on sentiment analysis and lstm. Neural Comput. Appl. **32**, 9713–9729 (2020)
15. Kim, H.Y., Won, C.H.: Forecasting the volatility of stock price index: a hybrid model integrating LSTM with multiple garch-type models. Expert Syst. Appl. **103**, 25–37 (2018)
16. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
17. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
18. Li, Q., Chen, Y., Jiang, L.L., Li, P., Chen, H.: A tensor-based information framework for predicting the stock market. ACM Trans. Inf. Syst. (TOIS) **34**(2), 1–30 (2016)
19. Li, Q., Tan, J., Wang, J., Chen, H.: A multimodal event-driven lstm model for stock prediction using online news. IEEE Trans. Knowl. Data Eng. **33**(10), 3323–3337 (2020)
20. Li, W., Bao, R., Harimoto, K., Chen, D., Xu, J., Su, Q.: Modeling the stock relation with graph network for overnight stock movement prediction. In: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, pp. 4541–4547 (2021)
21. Lo, A.W., MacKinlay, A.C.: When are contrarian profits due to stock market overreaction? Rev. Financ. Stud. **3**(2), 175–205 (1990)
22. Nelson, D.B.: Conditional heteroskedasticity in asset returns: a new approach. Econometrica **59**(2), 347–370 (1991)
23. Niu, Y., Lu, L., Dolphin, R., Poti, V., Dong, R.: Evaluating financial relational graphs: interpretation before prediction. In: Proceedings of the 5th ACM International Conference on AI in Finance, pp. 564–572 (2024)
24. Poterba, J.M., Summers, L.H.: Mean reversion in stock prices: evidence and implications. J. Financ. Econ. **22**(1), 27–59 (1988)
25. Selvin, S., Vinayakumar, R., Gopalakrishnan, E., Menon, V.K., Soman, K.: Stock price prediction using lstm, rnn and cnn-sliding window model. In: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1643–1647. IEEE (2017)
26. Soun, Y., Yoo, J., Cho, M., Jeon, J., Kang, U.: Accurate stock movement prediction with self-supervised learning from sparse noisy tweets. In: 2022 IEEE International Conference on Big Data (Big Data), pp. 1691–1700. IEEE (2022)
27. Sunny, M.A.I., Maswood, M.M.S., Alharbi, A.G.: Deep learning-based stock price prediction using lstm and bi-directional lstm model. In: 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES), pp. 87–92. IEEE (2020)
28. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
29. Xiang, S., Cheng, D., Shang, C., Zhang, Y., Liang, Y.: Temporal and heterogeneous graph neural network for financial time series prediction. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, pp. 3584–3593 (2022)
30. Xu, C., Huang, H., Ying, X., Gao, J., Li, Z., Zhang, P., Xiao, J., Zhang, J., Luo, J.: Hgnn: hierarchical graph neural network for predicting the classification of price-limit-hitting stocks. Inf. Sci. **607**, 783–798 (2022)
31. Xu, Y., Cohen, S.B.: Stock movement prediction from tweets and historical prices. In: Gurevych, I., Miyao, Y. (eds.) Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1970–1979. Association for Computational Linguistics, Melbourne, Australia (2018)



CSM: Corn Instance Segmentation Model Fusing Dilated Residual Networks and Low-Rank Adaptation

Deli Zhu, Xinjie Li(✉), and Liang He

Chongqing Normal University, Chongqing, China
923468074@qq.com

Abstract. Corn ear phenotypic characteristics are important predictors of corn quality, yield, and growth. The Corn Instance Segmentation Model (CSM) is a high-precision instance segmentation framework that was created in this study to meet the need for automated phenotypic extraction and crop monitoring. Using accurate localization supplied by an RT-DETR-based model to direct segmentation through a SAM network, CSM combines sophisticated object detection and instance segmentation techniques. In order to maximize multi-scale feature fusion, the framework includes context and spatial feature calibration modules (CFC and SFC) in addition to a dilated residual network (DWRP3) for feature extraction. LoRA fine-tuning is also used to increase model adaptability and computational efficiency. This method improves operating efficiency by drastically lowering manual involvement while simultaneously increasing segmentation accuracy. According to experimental results on a custom corn ear dataset, CSM outperforms Mask R-CNN, Mask2Former, and YOLACT in precision by 3.5%, 4.3%, and 12.7%, respectively, achieving 95.5% precision, 84.8% mean intersection-over-union (mIoU), and 88.4% recall. With great potential for widespread use, this research offers crucial technological assistance for automated crop monitoring in precision agriculture.

Keywords: Instance segmentation · Corn Phenotyping · Dilated Residual Network · Low-Rank Adaptation · Feature Calibration

1 Introduction

Food security and agricultural economic growth are tightly linked to corn, a vital staple crop whose phenotypic characteristics directly affect yield and quality [1,2]. Accurate and efficient collection of corn phenotypic data is essential for regulating crop growth, improving variety, and preparing for climatic change [3]. As smart agriculture has developed, automated crop monitoring has emerged as a key tool for accurate, large-scale analysis [4,5]. Corn farming is increasingly

D. Zhu—Supported by Chongqing Technology Innovation and Application Development Project (CSTB2024TIAD-CYKJCXX0023) and Research Project of Chongqing Normal University (25XLB006).

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 216–227, 2026.
https://doi.org/10.1007/978-3-032-04555-3_18

using field inspection robots to track crop growth, and one of the main functions of the visual system is the accurate segmentation of corn ears.

A key technology in computer vision, instance segmentation aims to give high-precision segmentation results by precisely recognizing and segmenting each individual item in an image, particularly when addressing problems like object overlap and morphological variation [6]. Conventional approaches to fruit segmentation usually use methods like region growth, edge detection, and thresholding for recognition and segmentation. It is difficult to guarantee effective and precise segmentation, nevertheless, because these techniques frequently perform poorly in complicated situations, such as changes in lighting and fruit occlusion [7–9]. As deep learning has advanced, more and more deep learning techniques have been used extensively for agricultural fruit identification and segmentation, greatly increasing segmentation efficiency and accuracy [10–15]. Nevertheless, current models continue to fall short of the accuracy and efficiency needed for automatic harvesting operations and corn ear monitoring.

To overcome the shortcomings of the existing maize ear instance segmentation methodologies, this work proposes the CSM, an automated and high-precision instance segmentation method. Traditional segmentation methods concentrate on fixed features, which often leads to poor accuracy in complex scenarios. Conversely, CSM combines object identification and instance segmentation techniques to optimize the segmentation process. In order to correctly locate target locations, it automatically creates bounding boxes. SAM (Segment Anything Model) [16] carries out high-precision segmentation based on the precise bounding boxes produced by RT-DETR (Real-Time Detection Transformer) [17] within the CSM framework. The DWRP3 dilated residual module, the LoRA fine-tuning strategy, and the CFC and SFC feature calibration modules allow CSM to function consistently in a variety of complex environments, particularly for the precise monitoring and management of agricultural crops like maize ears, exhibiting remarkable performance and efficiency. This approach significantly reduces manual intervention, increases segmentation efficiency, and demonstrates exceptional adaptability and robustness in complex agricultural imagery.

2 Materials and Methods

2.1 Data Collection and Dataset Creation

The corn ear picture collection used in this study was collected in Yangshuo County, Guilin City, Guangxi Zhuang Autonomous Region, in May and June of 2024 from natural field settings. The data was collected using a 12-megapixel rear camera on an iPhone 13 (12 MP, 4032 × 3024 resolution, 24-bit color image format) with 512 × 512 pixel image resolution. In order to maximize the diversity of the data, a total of 6540 pictures were collected, as shown in Fig. 1. These images featured samples with varying lighting conditions, fruit counts, and maturation phases, as well as challenging scenarios such as cluttered backgrounds, shifting illumination, and overlapping fruits. The open-source LabelMe tool [18]

was used to annotate the images, and the dataset was generated in COCO [19] format.

The dataset was expanded to 19,620 images using data augmentation techniques (such as flipping, translation, noise addition, and contrast adjustment) to increase the model's capacity for generalization. Figure 2 displays the augmented samples after the dataset was split into training, validation, and test sets at a 7:2:1 ratio.



a. Milk stage b. Wax stage c. Full stage d. Back lighting e. Fruit obscured by blade f. Fruit obscured each other

Fig. 1. Example of maize ear image samples



a. Original Image b. Decreased Brightness c. Added Noise d. Horizontal Flip + Contrast Enhancement e. Vertical Flip + Added Noise f. Vertical Flip + Translation

Fig. 2. Example of enhanced image samples

2.2 Experimental Setup

Experiments were performed on an NVIDIA GeForce RTX A6000 GPU with 48GB of RAM, powered by an Intel Xeon W-255 processor, and running Ubuntu 20.04 with PyTorch and CUDA 11.1.

2.3 Corn Instance Segmentation Model

By creatively combining object detection and instance segmentation, CSM overcomes the drawbacks of conventional techniques in intricate agricultural situations. Because of its adaptable input prompt mechanism (text, points, and bounding boxes), SAM, a deep learning-based segmentation model that is a fundamental part of CSM, has demonstrated impressive performance in a variety of image segmentation tasks. However, the ambiguity in text or point cues might result in unstable segmentation in complicated scenes with multiple or overlapping items, which limits their practical utility.

CSM combines the segmentation capabilities of SAM with the effectiveness of the RT-DETR object detection paradigm to address these issues. By producing

accurate bounding boxes, RT-DETR minimizes the need for user involvement and fixes the localization problems that SAM encounters in intricate situations. Bounding boxes offer more solid and distinct target zones than text or point cues, which increases segmentation accuracy. RT-DETR clearly outlines the target region in agricultural imagery, like maize ears, which helps SAM identify and segment the target more precisely.

Through the use of this two-stage methodology, CSM effectively overcomes the drawbacks of conventional single-stage models, providing a precise and effective solution that is especially well-suited for agricultural crop monitoring and management. Figure 3 depicts the CSM architecture.

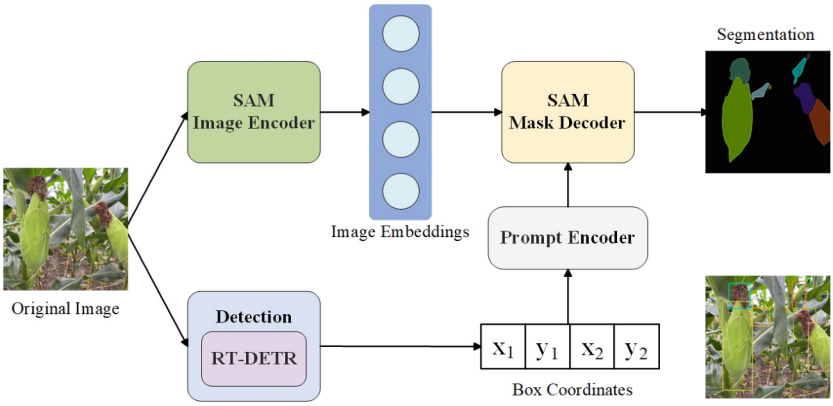


Fig. 3. Architecture of the proposed CSM framework.

Object Detection. As a core component of CSM, RT-DETR is an end-to-end real-time object detection model based on the Transformer architecture, consisting of the Backbone, Efficient Hybrid Encoder, and Transformer Decoder.

The ResNet18 backbone network is used in CSM by RT-DETR to extract image features, and the Efficient Hybrid Encoder receives the P3, P4, and P5 feature maps. The two modules that make up this encoder are CCFF (CNN-based Cross-scale Feature Fusion) and AIFI (Attention-based Intra-scale Feature Interaction). In order to effectively fuse multi-scale information, CCFF improves multi-scale feature representation, and AIFI improves semantic information. To guarantee query quality and prediction accuracy, the decoder implements an uncertainty-minimizing query selection mechanism and optimizes the original queries as part of its IoU-aware query mechanism. The object bounding boxes and their confidence scores are then produced by the decoder after repeated refinement.

Feature Extraction Network Combined with Dilated Wide Residual Module. The foundation of CSM's RT-DETR is the traditional ResNet18 architecture, which uses stacked convolutional layers to extract visual information. ResNet18 uses just four BasicBlock layers to minimize computational overhead and strike a balance between model complexity and computational efficiency. However, elements

including background interference, leaf occlusion, and changing illumination frequently result in blurred object boundaries and a loss of local clarity in field photos of maize ears. Furthermore, feature extraction performance is impacted by ResNet18's reduced receptive field, which makes it difficult to fully capture the structure of objects in complicated backgrounds.

To address this issue, CSM optimizes ResNet18 by introducing the Dilated Wide Residual (DWR) module [20] and constructing an enhanced DWRP3 structure. This structure expands the receptive field through dilated convolutions and residual connections, improving structural perception of targets in complex backgrounds and significantly enhancing maize ear recognition accuracy.

The feature extraction method used by the DWR module involves two steps: first, it processes the input feature map with 3×3 convolutions, batch normalization, and ReLU activation to extract basic features; then, it applies multi-branch dilated convolutions (with dilation rates d=1, 3, 5) to process features at different scales, further enhancing the receptive field. After feature fusion, point-wise convolution is used to compress the channel dimensions and integrate multi-scale information. Finally, the fused feature map is added element-wise to the input feature map via residual connections, producing a feature map with global feature representation.

Figure 4 depicts the DWRP3 module's structure, which consists of N DWR blocks and incorporates a dual-path fusion mechanism to compute and combine feature representations from two paths.

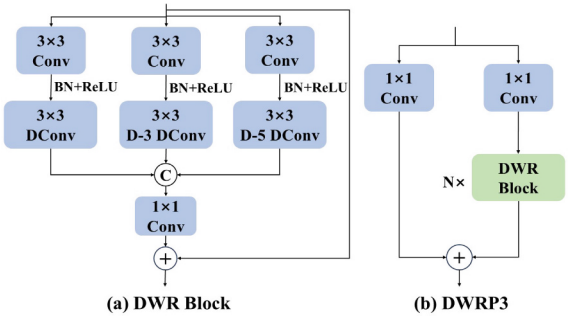


Fig. 4. DWR Block and DWRP3 Structure

Feature Fusion Based on Context and Spatial Feature Calibration. In the CSM framework, feature map P5 has a large receptive field, enabling it to capture rich global contextual information. However, its relatively low resolution makes it susceptible to background noise interference. In contrast, feature map P3 has a higher resolution, allowing it to capture finer details, which is crucial for target boundary localization. In field environments, maize ears are often occluded by neighboring plant leaves, which share similar colors with the ears and stems, leading to confusion between target features and background information. To effectively distinguish these similar features and improve maize ear recognition

accuracy, CSM optimizes the neck network of RT-DETR by introducing context feature calibration (CFC) and spatial feature calibration (SFC) modules [21] to enhance global contextual perception and spatial alignment capabilities.

In particular, the CFC module integrates multi-scale information, performs multi-scale pooling on feature map P5 using pyramid scene parsing (PSP), and adds a local attention mechanism to highlight the target region while reducing background noise. Local details in feature map P3 are optimized using the SFC module. An initial spatially aligned feature map is produced by first processing the feature map by convolution. In order to ensure that features from various scales can successfully fuse in the same spatial dimension, the low-resolution feature map is then adjusted to a higher resolution via bilinear interpolation. Lastly, SFC generates spatial offsets using an offset learning method and then resamples the feature map using grid sampling to remove spatial inconsistencies between scales, improving the multi-scale feature fusion effect. Figure 5 displays both the CFC and SFC module structures.

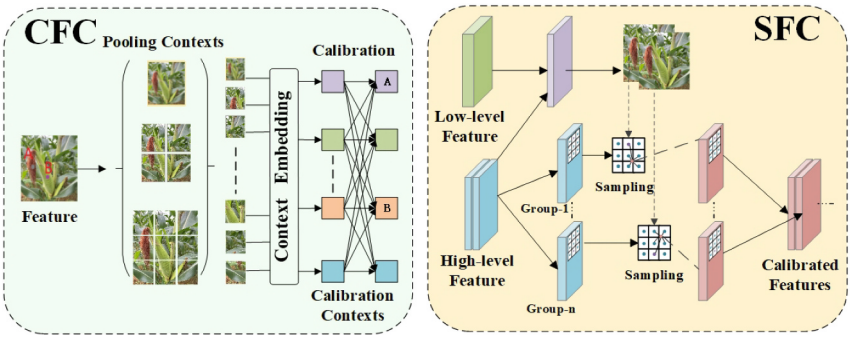


Fig. 5. CFC and SFC module structures

Segmentation. CSM leverages SAM as the fundamental component for image segmentation, with SAM consisting of three main modules: the image encoder, the prompt encoder, and the mask decoder. The pre-trained Vision Transformer (ViT) [22] architecture is used by the image encoder, which divides the image into blocks for global feature learning and uses self-attention techniques to extract features. The final image embedding is produced by convolutionally compressing the image information to increase efficiency. The model is guided to focus on the target region by the prompt encoder, which incorporates external prompt information into the picture characteristics. The final segmentation mask is then produced by the mask decoder by combining the features produced by the prompt and image encoders and applying deconvolution and upsampling techniques.

In field settings, complex backgrounds, occlusions, and small target instances frequently affect maize ear images, lowering segmentation accuracy. The short distance between maize plants, the leaves' ability to obstruct fruit, and the color

resemblance between the ears of maize and the leaves or stem internodes all contribute to the segmentation's difficulty.

In order to solve this problem, CSM uses the Low-Rank Adaptation (LoRA) [23] technique to improve the image encoder of SAM, which increases the model's capacity to detect small targets and capture details. LoRA optimizes the self-attention mechanism while lowering computational cost by incorporating low-rank matrix approximations, which reduce the amount of parameters that need to be trained. This method increases training efficiency by avoiding the need for extensive model fine-tuning.

In the implementation, LoRA fine-tunes the image encoder of SAM by introducing low-rank matrices into the Transformer module, based on frozen pre-trained weights. The low-rank matrices A and B are used to approximate the weight update ΔW, avoiding full model fine-tuning and preserving the original output dimension. The weight update ΔW is approximated as follows:

$$\bar{W} = W + \Delta W \approx W + AB\left(\frac{\alpha}{r}\right) \tag{1}$$

The rank satisfies $r \ll \min(d, k)$, and $A \in \mathbb{R}^{d\times r}$ and $B \in \mathbb{R}^{r\times k}$. The updated weight is denoted by $\bar{W}$, the pre-trained weight by W, and the weight increment by ΔW. Adjustable optimization speed is made possible by the scaling factor α, which regulates the gradient descent rate.

LoRA is only applied to the Transformer's essential self-attention mechanism, not all layers, in order to minimize GPU memory consumption. Figure 6 shows how to fine-tune the SAM image encoder using LoRA.

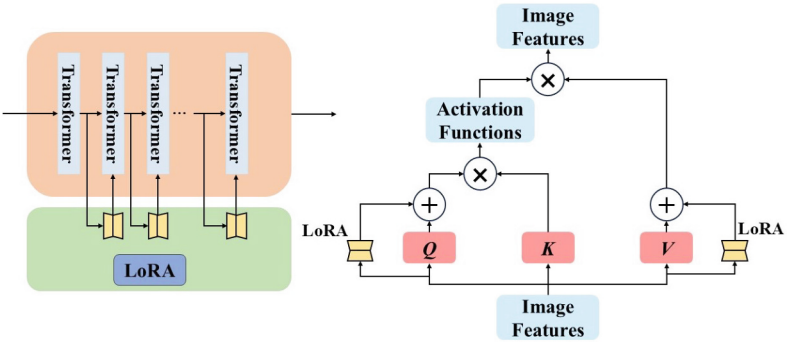


Fig. 6. LoRA fine-tuning SAMâĂŹs image encoder.

By calculating the cosine similarity between the query (Q), key (K), and value (V) vectors, the self-attention mechanism dynamically learns dependencies between places. The mechanism's expression is:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^{\top}}{\sqrt{C}}\right)V \tag{2}$$

The Transformer module introduces low-rank matrices $B_Q A_Q$ and $B_V A_V$ for Q and V, respectively, by freezing the pre-trained weights of Q, K, and V. The relevant formulations are as follows:

$$Q = \bar{w}_Q F = W_Q + \frac{\alpha}{r} B_Q A_Q, \quad V = \bar{w}_V F = W_V + \frac{\alpha}{r} B_V A_V \tag{3}$$

The input vector length is denoted by C, and the frozen pre-trained weights of Q and V are denoted by W_Q and W_V, respectively. The input vector is F, and the trainable LoRA parameters are A_Q, B_Q, A_V, and B_V. By strengthening the model's attention areas, segmentation performance is increased and adaptation to downstream tasks is improved.

3 Results and Analysis

3.1 Ablation Study Results and Analysis

An ablation study was carried out to evaluate the contribution of various optimization techniques in order to confirm the efficacy of the suggested model. Table 1 displays the findings.

Table 1. Results of ablation experiments

Number	DWRP3	CFC-SFC	LoRA	Precision(%)	mIoU(%)	Recall(%)
1	×	×	×	92.3	81.5	85.8
2	√	×	×	93.2	82.1	86.8
3	×	√	×	93.6	82.3	86.7
4	×	×	√	93	82	86.1
5	√	√	×	94.6	82.5	86.4
6	√	√	√	95.5	84.8	88.4

Table 1 unequivocally shows that the precision, mean intersection over union (mIoU), and recall of CSM in corn ear segmentation are all improved by the individual application of the DWRP3, CFC, SFC, and LoRA fine-tuning.

First, precision, mIoU, and recall improved by 0.9, 0.6, and 1.0% points, respectively, when the BasicBlock structure in RT-DETR was optimized using the DWRP3. This suggests that CSM captures finer details of corn ears because the DWRP3 aids in the retention of additional spatial information during partial feature extraction. Precision, mIoU, and recall were then further increased by 1.3, 0.8, and 0.9% points, respectively, with the introduction of the CFC and SFC. This demonstrates how well these modules optimize multi-scale feature interactions, strengthening the integration of low-level details with high-level semantic information and boosting the model's capacity to identify targets at different sizes. Lastly, precision, mIoU, and recall increased by 0.7, 0.5, and 0.3% points when SAM was fine-tuned using LoRA alone. LoRA greatly improves segmentation accuracy by reducing parameter adjustments through low-rank

adaptation, even though the gains are less pronounced than those of DWRP3 and CFC-SFC.

CSM performs best when all three improvements are combined: recall climbs to 88.1%, mIoU rises to 84.8%, and precision reaches 95.5%. This outcome shows how the three techniques work in concert to greatly improve CSM's segmentation capabilities, guaranteeing strong flexibility and stability under challenging field conditions.

3.2 Comparison of CSM with Other Segmentation Models

To validate the effectiveness of CSM, this study compares it with segmentation models such as Mask R-CNN [24], Mask2Former [25], and YOLACT [26]. All models were trained on the same dataset and evaluated using the same metrics: precision, mean intersection over union (mIoU), and recall. The segmentation results are shown in Table 2.

In accordance with the experimental results, the CSM model achieves a precision of 95.5%, an mIoU of 84.8%, and a recall of 88.4%, respectively. CSM performs 3.5 to 12.7% points better in precision, 1.5 to 5.0% points better in mIoU, and 0.3 to 8.1% points better in recall than other segmentation algorithms. Even though CSM includes `SAM_ViT-H`, which raises the overall amount of parameters, there are only 21.3 M trainable parameters, which is a lot fewer than in other models. All things considered, CSM produces outstanding segmentation results, improves segmentation accuracy, and reduces training costs by precisely segmenting maize ears in complicated backdrops.

Table 2. Comparison of the results of different segmentation models

Model	Precision (%) ↑	mIoU (%) ↑	Recall (%) ↑	Total Params (M) ↓	Trainable Params (M) ↓
Mask R-CNN	92	80.7	86	44.4	44
Mask2Former	91.2	83.3	88.1	45	45
YOLACT	82.8	79.8	80.3	29.8	29.8
CSM	95.5	84.8	88.4	656	21.3

3.3 Comparison of CSM with Other Segmentation Models Results

The distribution of maize ears, occlusion, and background can all have varying effects on the segmentation performance in the complex and variable field environment. To evaluate the performance of various models in natural field environments, CSM is compared with other segmentation models for maize ear segmentation, and the results are displayed in Fig. 7.

Figure 7 shows the segmentation results across three different maize growth stages: milk stage, wax stage, and full stage, as well as under various environmental conditions such as backlighting and fruit or silk occlusion. From the figure, it is observed that Mask R-CNN and Mask2Former exhibit false detection, while YOLACT suffers from more severe mis-segmentation and missed segmentation

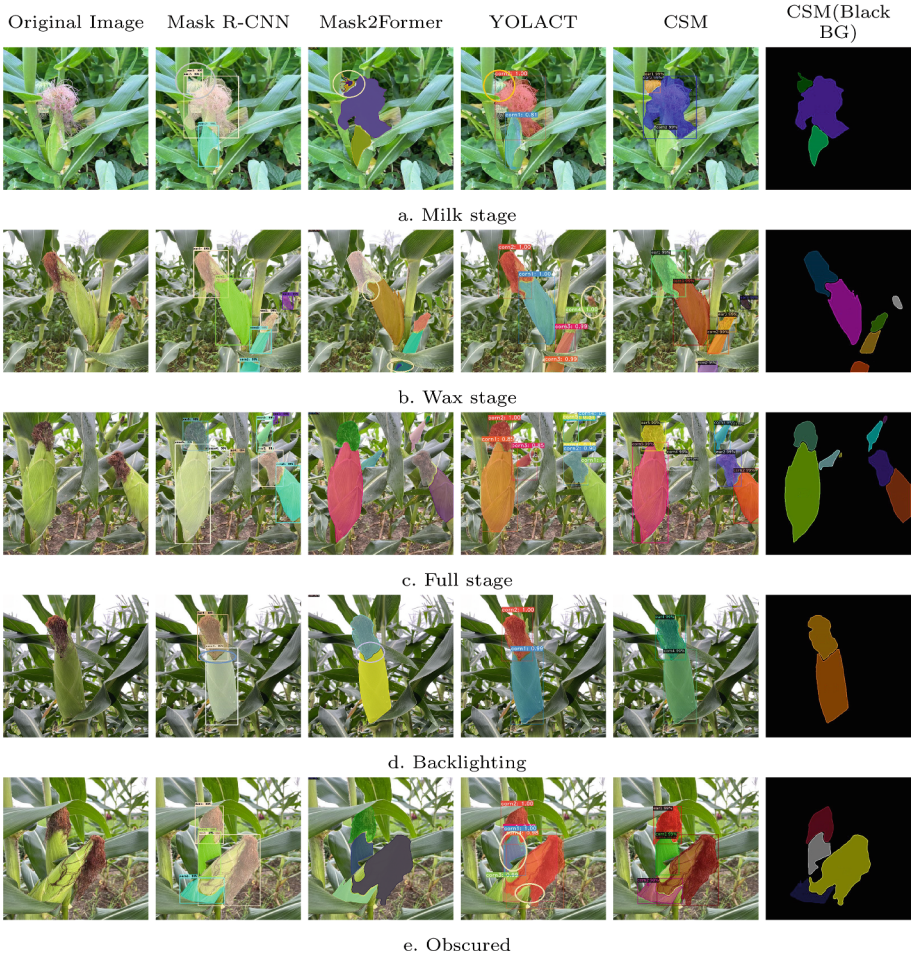


Fig. 7. Segmentation results of different models

issues. Additionally, YOLACT shows limitations in accurately segmenting maize ears and maintaining clear segmentation boundaries. In contrast, CSM achieves pixel-level precise segmentation across various growth stages and complex background conditions, significantly reducing false detections and yielding better segmentation results.

4 Conclusions

An essential part of crop growth tracking and production evaluation is the accurate segmentation of maize earheads, which also serves as the basis for automation in intelligent agriculture. This study presents an automatic instance segmentation model, CSM, to tackle the difficulties caused by complicated field sit-

uations, including target overlap, fruit occlusion, and complex backdrops. The following are the main findings and innovations:

(1) RT-DETR is introduced as the object detection network, leveraging its high efficiency in detecting maize earheads. Additionally, the SAM module is incorporated for instance-level segmentation to ensure stable performance in complex scenarios. To further enhance the model's detection capabilities, the DWRP3 module is integrated to improve detection accuracy. Furthermore, by combining the CFC and SFC modules, the modelâĂŹs performance in multi-scale feature fusion is strengthened, thereby improving its ability to segment targets of various sizes and under different backgrounds. Moreover, LoRA is used to perform lightweight fine-tuning on the SAM encoder, resulting in an overall performance enhancement, culminating in the new CSM.

(2) Experimental results on a self-constructed maize earhead dataset demonstrate that CSM achieves an accuracy of 95.5%, a mean intersection-over-union (mIoU) of 84.8%, and a recall rate of 88.4%, significantly outperforming traditional segmentation methods. Compared to other models, CSM exhibits the best overall performance, maintaining superior segmentation results when processing images with complex backgrounds, overlapping fruits, and maize ears at different stages of maturity.

In conclusion, by combining effective object recognition and instance segmentation techniques, the CSM model presented in this research successfully tackles the difficulties of segmenting maize earheads in intricate field settings. Strong technical assistance is provided for the automatic monitoring of maize growth conditions by the model, which not only enables high segmentation efficiency but also greatly increases the accuracy of crop phenotypic parameter acquisition.

References

1. Ranum, P., Peña-Rosas, J.P., Garcia-Casal, M.N.: Global maize production, utilization, and consumption. Ann. N. Y. Acad. Sci. **1312**(1), 105–112 (2014)
2. Halli, H.M., et al.: Influence of planting and irrigation levels as physical methods on maize root morphological traits, grain yield, and water productivity in semi-arid region. Agronomy **11**, 294 (2021). https://doi.org/10.3390/agronomy11020294
3. Lu, Y., Young, S.: A survey of public datasets for computer vision tasks in precision agriculture. Comput. Electron. Agric. **178**, 105760 (2020)
4. Bai, Y., Zhang, B., Xu, N., et al.: Vision-based navigation and guidance for agricultural autonomous vehicles and robots: a review. Comput. Electron. Agric. **205**, 107584 (2023)
5. Wang, T., Chen, B., Zhang, Z., et al.: Applications of machine vision in agricultural robot navigation: A review. Comput. Electron. Agric. **198**, 107085 (2022)
6. Gené-Mola, J., Sanz-Cortiella, R., Rosell-Polo, J.R., et al.: Fruit detection and 3D location using instance segmentation neural networks and structure-from-motion photogrammetry. Comput. Electron. Agric. **169**, 105165 (2020)
7. Septiarini, A., Hamdani, H., Sari, S.U., Hatta, H.R., Puspitasari, N., Hadikurniawati, W.: Image processing techniques for tomato segmentation applying K-means clustering and edge detection approach. In: 2021 International Seminar on Machine Learning, Optimization, and Data Science (ISMODE), Jakarta, Indonesia, pp. 92–96 (2022). https://doi.org/10.1109/ISMODE53584.2022.9742740

8. Xu, T., Yao, L., Xu, L., et al.: Image segmentation of cucumber seedlings based on genetic algorithm. Sustainability **15**(4), 3089 (2023)
9. Ahmad, M.T., Greenspan, M., Asif, M., Marshall, J.A.: Robust apple segmentation using fuzzy logic. In: 2018 5th International Multi-Topic ICT Conference (IMTIC), Jamshoro, Pakistan, pp. 1–5 (2018). https://doi.org/10.1109/IMTIC.2018.8467275
10. Attri, I., Awasthi, L.K., Sharma, T.P., et al.: A review of deep learning techniques used in agriculture. Eco. Inform. **77**, 102217 (2023)
11. Zu, L., Zhao, Y., Liu, J., et al.: Detection and segmentation of mature green tomatoes based on Mask R-CNN with automatic image acquisition approach. Sensors **21**(23), 7 (2021)
12. Li, Y., Feng, Q., Liu, C., et al.: MTA-YOLACT: Multitask-aware network on fruit bunch identification for cherry tomato robotic harvesting. Eur. J. Agron. **146**, 126812 (2023)
13. Wang, A., Qian, W., Li, A., et al.: NVW-YOLOv8s: An improved YOLOv8s network for real-time detection and segmentation of tomato fruits at different ripeness stages. Comput. Electron. Agric. **219**, 108833 (2024)
14. Jia, W., Zhang, Z., Shao, W., et al.: FoveaMask: A fast and accurate deep learning model for green fruit instance segmentation. Comput. Electron. Agric. **191**, 106488 (2021)
15. Li, Q., Jia, W., Sun, M., et al.: A novel green apple segmentation algorithm based on ensemble U-Net under complex orchard environment. Comput. Electron. Agric. **180**, 105900 (2021)
16. Kirillov, A., Mintun, E., Ravi, N., et al.: Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4015–4026 (2023)
17. Zhao, Y., Lv, W., Xu, S., et al.: DETRs beat YOLOs on real-time object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16965–16974 (2024)
18. Torralba, A., Russell, B.C., Yuen, J.: LabelMe: Online image annotation and applications. Proc. IEEE **98**(8), 1467–1484 (2010)
19. Lin, T.Y., Maire, M., Belongie, S., et al.: Microsoft COCO: Common objects in context. In: European Conference on Computer Vision, pp. 740–755 (2014)
20. Wei, H., Liu, X., Xu, S., et al.: DWRSeg: Rethinking efficient acquisition of multi-scale contextual information for real-time semantic segmentation. arXiv preprint arXiv:2212.01173 (2022)
21. Li, K., Geng, Q., Wan, M., et al.: Context and spatial feature calibration for real-time semantic segmentation. IEEE Trans. Image Process. **32**, 5465–5477 (2023)
22. Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
23. Hu, E.J., Shen, Y., Wallis, P., et al.: LoRA: Low-rank adaptation of large language models. In: ICLR, **1**(2), 3 (2022)
24. He, K., Gkioxari, G., Dollár, P., et al.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
25. Cheng, B., Misra, I., Schwing, A.G., et al.: Masked-attention mask transformer for universal image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1290–1299 (2022)
26. Bolya, D., Zhou, C., Xiao, F., et al.: YOLACT: Real-time instance segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9157–9166 (2019)



Sensor-Enhanced PINNs for Contaminant Dispersion Modeling

Zhijing Feng[1(✉)], Elisa Y. M. Ang[1], Tay Bee Kiat[2], Koh Wai Heng[2], and Peng Cheng Wang[1]

[1] Engineering Cluster, Singapore Institute of Technology, 11 New Punggol Rd, Singapore 828616, Singapore
zhijing.feng@singaporetech.edu.sg

[2] DSO National Laboratories, 12 Science Park Drive, Singapore 118225, Singapore

Abstract. Physics-Informed Neural Network (PINN) is a powerful, mesh-free tool for predicting contamination dispersion, but they often struggle with the unknown boundary conditions common in real-world scenarios. This work presents a novel framework that overcomes this limitation by integrating sparse sensor data with a perceptually-aware evaluation metric. We introduce a systematic sensor placement strategy that significantly reduces the required instrumentation while maintaining high predictive accuracy. To better assess performance, we utilize a perceptual LAB error metric that more closely aligns with visual fidelity than traditional MSE. Critically, we validate the framework's geometric robustness across six distinct obstacle configurations, including shifts and rotations, proving its adaptability with a single network architecture. Together, these innovations provide a unified and efficient method for real-world dispersion prediction using minimal field instrumentation.

Keywords: contamination dispersion prediction · physics informed neural network · sensors feedback · physics based model

1 Introduction

Predicting the dispersion of contaminants is critical for managing public health risks from sources like airborne diseases and toxic chemical releases [1,2]. While Computational Fluid Dynamics (CFD) is the traditional tool for these predictions due to its ability to handle complex geometries [3], it suffers from two major drawbacks: it is computationally intensive [4] and requires precisely defined boundary and initial conditions [5,6]. These limitations restrict its use in time-sensitive applications like first-response missions, where conditions are often unknown or changing dynamically [7,8].

Physics-Informed Neural Networks (PINNs), introduced in 2017 [9,10], offer a promising mesh-free alternative by embedding physical laws directly into the neural network's loss function [11]. This allows PINNs to act as surrogate CFD models without being limited by a training dataset. However, recent studies

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 228–238, 2026.
https://doi.org/10.1007/978-3-032-04555-3_19

consistently highlight that the high cost and long duration of training PINNs limit their practical application [12].

To overcome these challenges, a hybrid approach combining PINNs with sparse sensor data has been shown to accelerate convergence and improve accuracy [13]. This method allows for the rapid reconstruction of a flow field using a few strategically placed sensors, removing the need for expensive CFD solvers. To the best of our knowledge, the integration of sensor data with PINNs for contamination dispersion prediction has not yet been reported in the literature and is the focus of this work.

Building on our prior work [14] that established the feasibility of this method for a fixed 2D scenario, this paper addresses critical gaps to enhance the model's robustness and practicality. We systematically evaluate sensor placement strategies to minimize the required measurements, and we validate the framework's adaptability across six different 2D obstacle configurations, including translations and rotations up to 30°. Furthermore, we introduce a perceptual LAB error metric that better aligns with visual fidelity than traditional MSE. Our results demonstrate that an optimized, sensor-augmented PINN can serve as a fast and reliable surrogate for traditional CFD in modeling contamination dispersion in dynamic environments.

2 Methodology

2.1 Problem Formulation

Building upon the baseline 2D bluff-body case established in prior work [14] (central rectangular obstacle at 0° orientation), this study introduces six geometric variants to systematically evaluate PINN's adaptability: (1) 0.5-unit rightward shift, (2) 0.5-unit leftward shift, (3) 5° clockwise rotation, (4) 355° rotation (equivalent to 5° counterclockwise), (5) combined 0.5-unit right shift with 5° rotation, and (6) 0.5-unit right shift with 30° rotation. All cases maintain identical non-dimensionalized parameters (Re = 68,474, Pe-scaled diffusion) and turbulence modeling (k-ε in OpenFOAM) to isolate configuration effects while generating physically distinct dispersion patterns. This progression from a single validated baseline to multiple transformed states enables quantitative assessment of how sensor-augmented PINNs generalize across flow topologies - a critical capability for field applications where obstacle geometry and source locations are often uncertain.

2.2 PINN Methodology

The Physics-Informed Neural Network (PINN) architecture employs a fully connected deep neural network to establish a direct mapping between spatial coordinates (x, y) and key flow variables (U, V, p, C). This framework, building upon our established approach in prior work [14], incorporates physical constraints through a composite loss function that simultaneously enforces: (1) adherence to

the governing Navier-Stokes and transport equations at randomly sampled collocation points (physics-based loss, MSE_f) and (2) agreement with sparse velocity and concentration measurements from strategically placed sensors (data-driven loss, MSE_{sensor}). By optimizing this dual-objective loss function during training, the network learns to satisfy both fundamental fluid dynamics principles and empirical observations, enabling accurate flow field predictions without requiring complete boundary condition specifications.

2.3 Metrics for Accuracy Evaluation

The conventional Mean Squared Error (MSE) was found to be an inadequate metric for evaluating model performance. Testing revealed that low MSE values can mask significant visual discrepancies between predicted and reference fields, as illustrated in the Supplementary Materials (Figure S1.1).

To address the limitations of conventional error metrics, we implemented a perceptual evaluation framework using the LAB67 color space [15]. This three-dimensional system decomposes images into: (1) lightness (L, 0–100 scale), (2) green-red spectrum (a-axis, negative to positive values), and (3) blue-yellow spectrum (b-axis, negative to positive values). The quantitative comparison metric LAB is calculated as the Euclidean distance between corresponding pixels in predicted and reference fields:

$$LAB = \sqrt{(L_2 - L_1)^2 + (a_2 - a_1)^2 + (b_2 - b_1)^2} \tag{1}$$

The training protocol was optimized at a fixed duration of 2,000 epochs. This decision was based on our prior convergence analysis [14], which identified this duration as an effective balance between prediction accuracy and computational efficiency, as quantitatively demonstrated in the Supplementary Materials (Figure S1.2).

2.4 Neural Network Hyperparameters Tuning

To refine performance, the model's key hyperparameters—initialLearnRate, decayRate, numNeurons, and numLayers—were optimized using a grid search. We evaluated 48 different parameter combinations, with the specific candidate values for each hyperparameter detailed in the Supplementary Materials.

Each configuration was trained for a consistent number of epochs and assessed using the LAB metrics for concentration and velocity fields (LABc, LABu, LABv). The final parameter set was selected for this study because it produced the lowest LABc value, indicating the highest prediction accuracy for the contaminant concentration field. A detailed comparison of the outcomes for all configurations is available in Figure S2.1 of the Supplementary Materials.

3 Results

3.1 Optimizing Sensor Configurations

While our prior study [14] established a 78-sensor layout (Sensor Configuration 1) as a high-performing benchmark with a LABc of 5.60, such a high number

of sensors is not always practical or cost-effective. To address this, we aimed to minimize the sensor count by developing a more efficient placement strategy.

Through extensive experimentation, we have made 3 main conclusions to help guide sensors placement for PINN usage in contamination prediction. Firstly, sensors placement should primarily be concentrated along the path of contamination diffusion (which would depend on gas release location and wind direction) and around obstacles. Second, we noted that sparse placement, when appropriately implemented, does not compromise accuracy while reducing costs. However, it is essential to avoid excessive sparsity around obstacles. Following this strategy, we tested two sparser layouts: Configuration 2 with 20 sensors and Configuration 3 with 14 sensors, as depicted in Fig. 1.

The results, shown in Fig. 2, demonstrate that a significant reduction in sensors is possible with only a minor loss of accuracy. The 20-sensor configuration yielded a LABc of just 6.78, producing a concentration map visually comparable to the 78-sensor benchmark. However, a further reduction to 14 sensors caused the LABc to increase sharply to 11.9 as the solution began to diverge. This establishes that while sparse sensor placement is effective, a balance must be struck between sensor economy and the required accuracy for the operation.

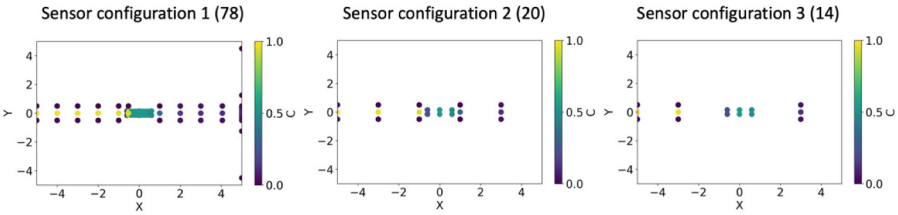


Fig. 1. Illustration of sensor configuration 1, 2 and 3. The number of sensors used for each sensor configuration are indicated in brackets.

3.2 Sensitivity of Solution to Different Configuration

To evaluate the model's generalizability, we apply the optimized methods to six new 2D cases involving geometric modifications of the obstacle (shown in Fig. 3). These cases systematically examine the PINN's response to: (i) lateral displacements (0.5 units left/right), (ii) angular rotations (5°-30° clockwise), and (iii) combined transformations. This progression enables fundamental investigation of the model's adaptability to geometric variations while maintaining identical flow parameters, providing critical insights for real-world applications where obstacle configurations may be uncertain.

The first observation that can be made was that the same sensor configuration used for baseline case in prior work [14] cannot be replicated across to all other cases. For the new case where the obstacle is shifted 0.5 units to the right along the x-axis (Case 1), the same sensor placement configuration is employed

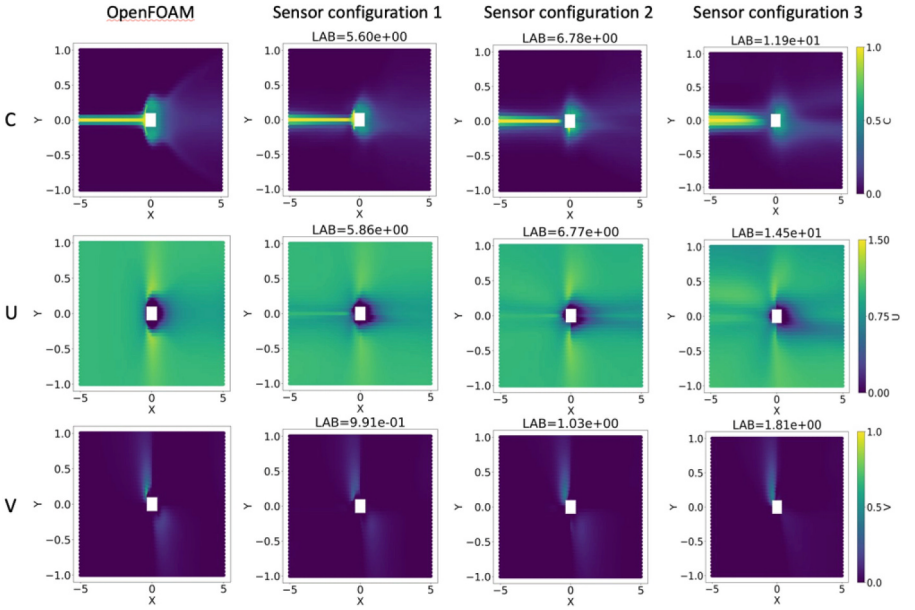


Fig. 2. Results when using sensor 1, 2, 3 as sensor data. The first column depicts images from OpenFOAM. Each row corresponds to C, U, and V, respectively.

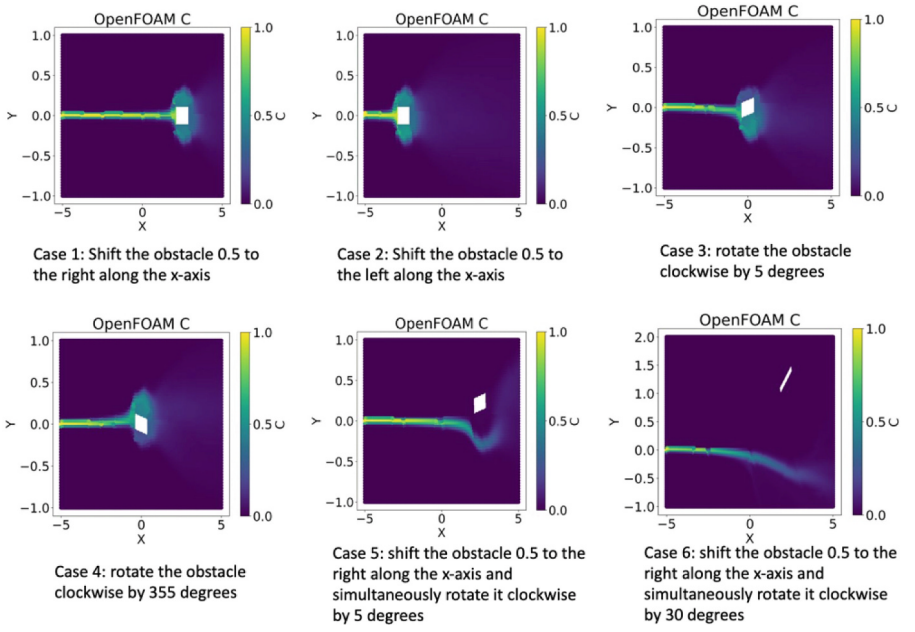


Fig. 3. More complex 2D scenarios and their corresponding OpenFOAM plots.

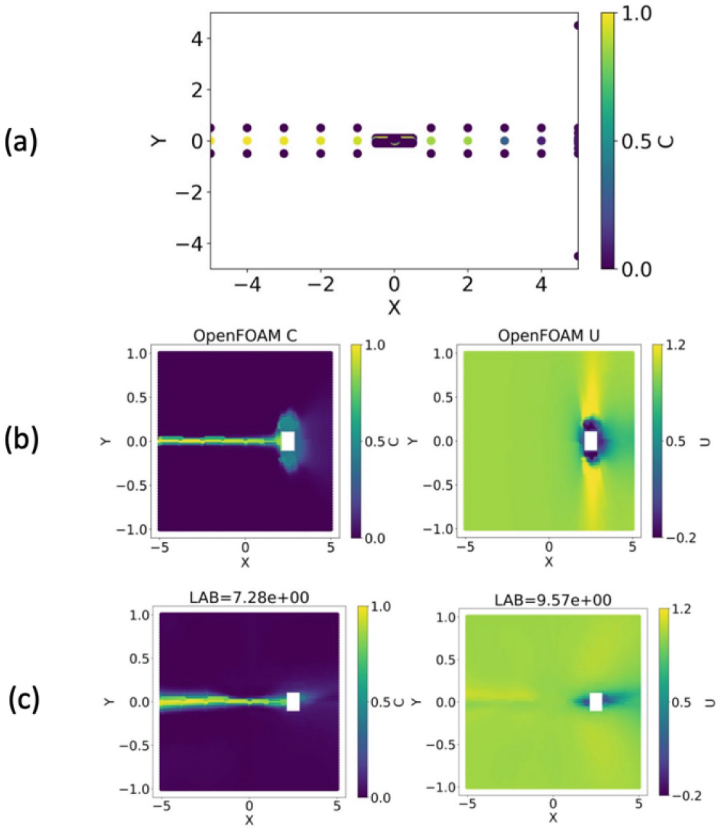


Fig. 4. Results for Case 1 with (a) sensor placement which has the same location as sensor 1, (b) OpenFOAM results, (c) PINN results.

as the benchmark sensor configuration 1. After running 2,000 epochs, it is evident that this sensor arrangement fails to yield satisfactory results for the new case scenario. Note that the same optimization parameters for baseline case in prior work [14] were applied to all the other cases. As shown in Fig. 4, this fixed sensor arrangement failed to produce an accurate prediction. The resulting concentration (C) and velocity (U) fields show significant divergence from the OpenFOAM ground truth, confirming that an adaptive sensor placement strategy is essential when dealing with varied geometries. Note that as the V velocity is small, it is omitted from this section for clarity.

Thus, it appears that for different 2D cases, selecting an appropriate sensor case is necessary to achieve favorable model performance. Therefore, we opted to adhere to the guidelines as discussed in Sect. 3.1, by altering the positions of the sensors. Sensors were strategically placed along the contamination diffusion path and around the obstacle. Subsequently, the number of sensors was systematically reduced to determine the minimum quantity required while maintain-

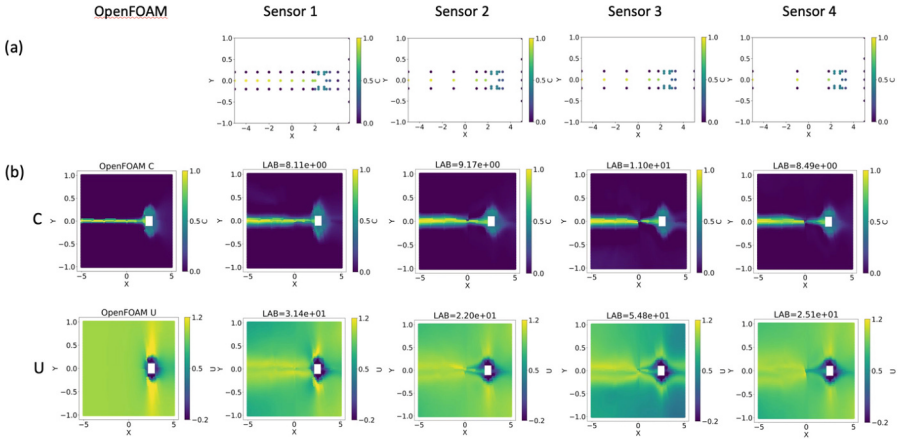


Fig. 5. (a) Sensor cases for shifting the obstacle 0.5 to the right along the x-axis case (Case 1), (b) OpenFOAM and PINN results for C and U.

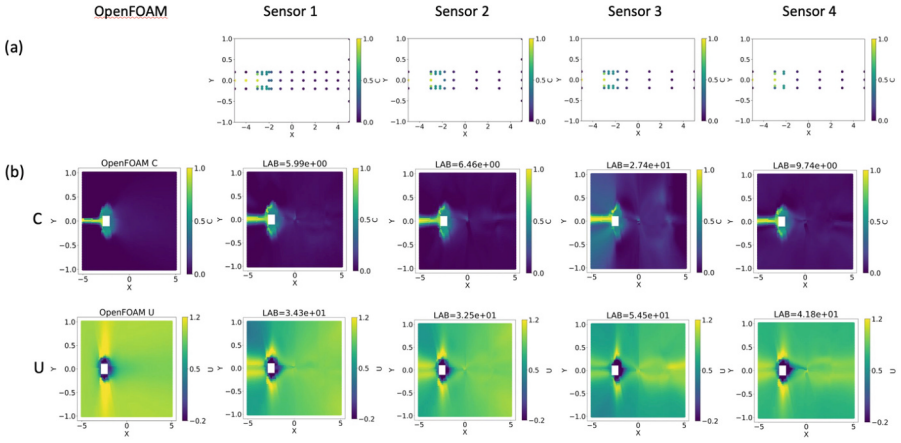


Fig. 6. (a) Sensor cases for shifting the obstacle 0.5 to the left along the x-axis case (Case 2), (b) OpenFOAM and PINN results for C and U.

ing optimal model performance. This approach yields favorable results for cases involving shifting and rotating obstacles. Figures 5 to 10 illustrate distinct sensor configurations and their corresponding model performance outcomes for six new experimental cases. It should be noted that for pure translation or rotation cases, incorporating a right-boundary sensor enhances model prediction accuracy. In the absence of this sensor, the density of sensors around obstacles and along the path must be moderated to avoid excessive concentration. However, for cases involving combined translation and rotation, the right-boundary sensor becomes unnecessary, and the guidance rules outlined in Sect. 3.1 remain fully applicable.

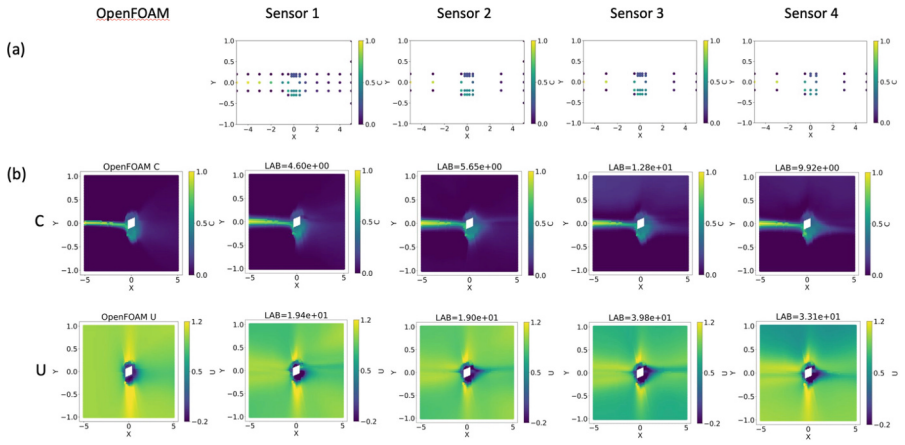


Fig. 7. (a) Sensor cases for rotating the obstacle clockwise by 5° (Case 3), (b) OpenFOAM and PINN results for C and U.

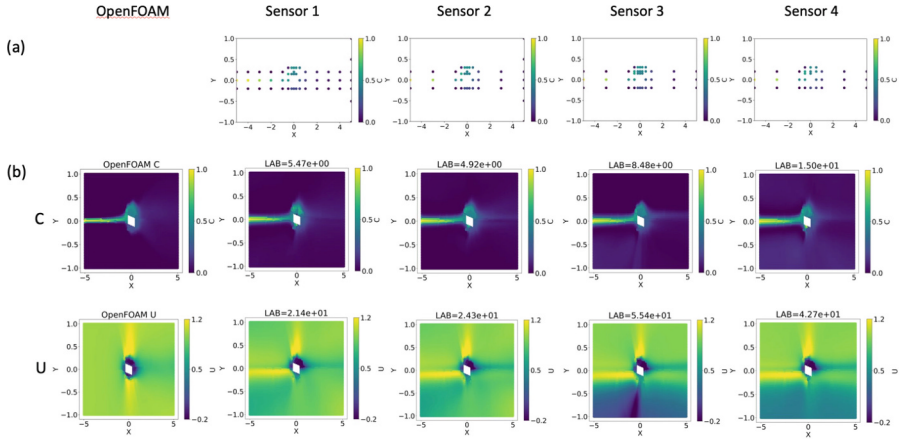


Fig. 8. (a) Sensor cases for rotating the obstacle clockwise by 355° (Case 4), (b) OpenFOAM and PINN results for C and U.

4 Identified Gaps and Future Work

To advance this methodology towards practical application, three key areas require further investigation.

First, future work should explore alternative sensor modalities. Optical gas imaging cameras, for instance, can provide dense 2D data maps, which could significantly accelerate convergence and improve the accuracy of 3D dispersion predictions.

Second, the normalization scheme for the governing equations needs refinement. The current approach results in magnitude disparities between physical

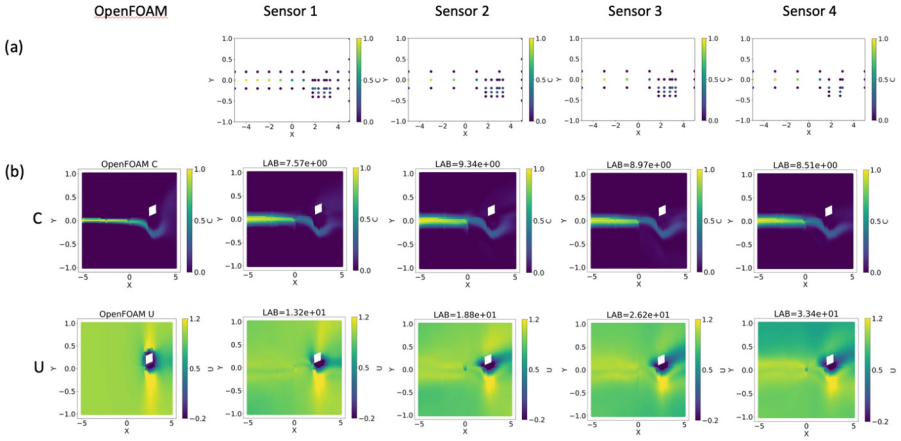


Fig. 9. (a) Sensor cases for shifting the obstacle 0.5 to the right along the x-axis and simultaneously rotating it clockwise by 5° (Case 5), (b) OpenFOAM and PINN results for C and U.

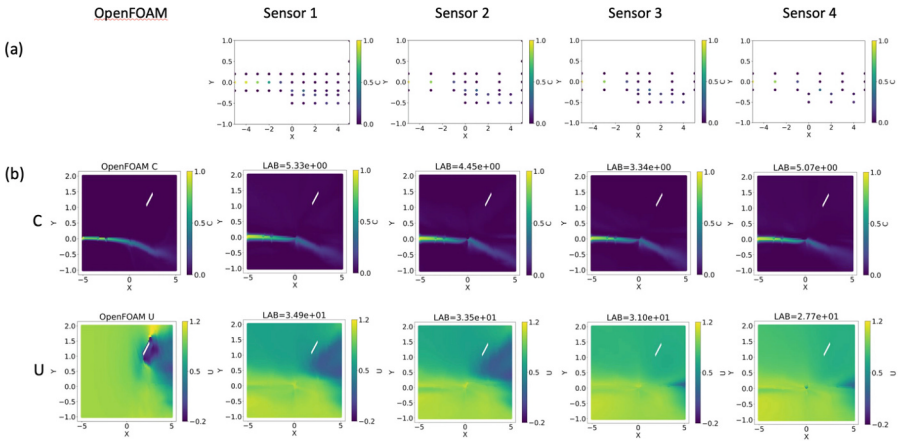


Fig. 10. (a) Sensor cases for shifting the obstacle 0.5 to the right along the x-axis and simultaneously rotate it clockwise by 30° (Case 6), (b) OpenFOAM and PINN results for C and U.

variables (e.g., concentration 'C' vs. velocity 'V'), inadvertently biasing the training. A new methodology is required to balance the loss terms effectively.

Finally, for the framework to be practically relevant, it must be scaled up from 2D steady-state models to full 3D transient simulations capable of representing complex urban scenarios.

5 Conclusion

To summarize, this work showcases how Physics-Informed Neural Networks can be employed for predicting contamination dispersion, offering a viable substitute for conventional CFD methods. CFD solvers typically depend on precise details about source release and boundary parameters like wind speed and direction; in contrast, the PINN approach relies on data from thoughtfully placed sensors within the area of interest. Thus, while CFD still has a strong place in applications such as hazard risk assessment, the work presented here shows that PINN can complement CFD by offering a solution for fast predictions without the need to specify boundary conditions that may be unknown or changing in real time. This paper presents a detailed feasibility analysis based on a fundamental 2D scenario, which was then expanded to include demonstrations of several different configurations. The findings herein indicate that by deploying sensors in optimal locations, PINN is capable of generating 2D dispersion maps that closely mirror the accuracy of those from CFD simulations.

References

1. Mei, X., Zeng, C., Gong, G.: Predicting indoor particle dispersion under dynamic ventilation modes with high-order Markov chain model. Build. Simul. **15**(7), 1243–1258 (2021). https://doi.org/10.1007/S12273-021-0855-Y
2. Li, H., et al.: Airborne dispersion of droplets during coughing: a physical model of viral transmission. Sci. Rep. **11**(1), 1–10 (2021). https://doi.org/10.1038/s41598-021-84245-2
3. Armand, P., Oldrini, O., Duchenne, C., Perdriel, S.: Topical 3D modelling and simulation of air dispersion hazards as a Keywords: 3D modelling and simulation, weather forecast, atmospheric dispersion, health (2021)
4. Kochkov, D., Smith, J.A., Alieva, A., Wang, Q., Brenner, M.P., Hoyer, S.: Machine learning–accelerated computational fluid dynamics. Proc. Natl. Acad. Sci. U S A **118**(21) (2021). https://doi.org/10.1073/PNAS.2101784118
5. Kocijan, J., et al.: Surrogate modelling for the forecast of Seveso-type atmospheric pollutant dispersion. Stochastic Environmental Research and Risk Assessment, 1–16 (2022). https://doi.org/10.1007/S00477-022-02288-X
6. Gunawardena, N., Pallotta, G., Simpson, M., Lucas, D.D.: Machine learning emulation of spatial deposition from a multi-physics ensemble of weather and atmospheric transport models. Atmosphere **12**(8), 953 (2021). https://doi.org/10.3390/ATMOS12080953
7. Coldrick, S., et al.: Modeling and experimental study of dispersion and deposition of respiratory emissions with implications for disease transmission. Indoor Air **32**(2) (2022). https://doi.org/10.1111/INA.13000
8. Mendil, M., Leirens, S., Armand, P., Duchenne, C.: Hazardous atmospheric dispersion in urban areas: a deep learning approach for emergency pollution forecast. Environ. Model. Softw. **152**, 105387 (2022). https://doi.org/10.1016/J.ENVSOFT.2022.105387
9. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics informed deep learning (Part I): data-driven solutions of nonlinear partial differential equations. J. Comput. Phys. **378**, 686–707 (2019)

10. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics informed deep learning (Part II): data-driven discovery of nonlinear partial differential equations. arXiv preprint arXiv:1711.10566 (2017)
11. Sliwinski, L., Rigas, G.: Mean flow reconstruction of unsteady flows using physics-informed neural networks. Data-Centric Eng. **4**, e4 (2023). https://doi.org/10.1017/DCE.2022.37
12. Hu, S., Liu, M., Zhang, S., Dong, S., Zheng, R.: Physics-informed neural network combined with characteristic-based split for solving forward and inverse problems involving Navier-Stokes equations. Neurocomputing **573**, 127240 (2024). https://doi.org/10.1016/J.NEUCOM.2024.127240
13. Raissi, M., Yazdani, A., Karniadakis, G.E.: Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations. Science **367**(6481), 1026–1030 (2020)
14. Feng, Z., Yun Mei, E.A., Tay, B.K., Koh, W.H., Wang, P.: Physics informed neural networks (PINNs) for rapid contamination dispersion predictions. arXiv preprint (2024). https://doi.org/10.25447/sit.26982754.v1
15. Hunter, R.S.: Photoelectric color difference meter. J. Opt. Soc. Am. **48**(12), 985–995 (1958)

Uniform Representation of Parametric CAD Models for Generative Application

Shengling Duan[1,2(✉)], Jiali Feng[2], and Yue Qi[1]

[1] The School of Computer Science and Engineering, Beihang University, Beijing 100191, China
dsl167@126.com
[2] AVIC Digital Corporation Ltd., Beijing 100028, China

Abstract. At present, most of 3D models generated by deep neural networks are non-parametric and not used for mechanical design and engineering application. To generate parametric CAD models automatically, one of the key problems is that there is no uniform structured representation strategy for parametrical CAD models. In this paper, we introduce a new feature extraction strategy for parametric CAD models and propose a new auto-encoder model, Y-type GAE, to solve the uniform representation problem. In experiments, in order to verify the quality of Y-type GAE, we design a generalization experiment, and an ablation test on some public datasets. Meanwhile, based on the uniform representation of parametric CAD models, we propose three feasible pipelines for 3D model generation, classification and retrieval. The effectiveness of the generation pipeline is evaluated on two public datasets, and the performance of the classification pipeline surpasses other representation methods of parametric CAD models. (The code is available at https://github.com/dishy313/Y-typeGAE).

Keywords: Parametric CAD Model · Uniform Representation · Graph Auto-Encoder · 3D Model Generation

1 Introduction

In the realm of computer-aided design (CAD), accurate and detailed parametric models are crucial for facilitating efficient design processes, enabling rapid prototyping, and ensuring seamless integration with manufacturing systems. Representing a parametric CAD model means encoding the design in a way that captures its geometric and topological properties. At present, the mainstream graphical representation method for parametric CAD models is BRep (Boundary Representation) [16]. BRep represents an entity by describing its boundaries, decomposing the geometric entity into basic elements and organizing these elements as a hierarchical data structure to reflect the relationships between the various elements [28]. The tree-like structure guarantees that BRep can describe parametric models accurately, but when we consider to generate BRep automatically, there are many challenges [1], for example, how to obtain an uniform

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 239–250, 2026.
https://doi.org/10.1007/978-3-032-04555-3_20

structured representation for BRep information, and how to generate a parametric CAD model based on the uniform structured representation. In this paper, we explore a strategy to represent the BRep information of parametric CAD models as a uniform structured representation, and strive to provide a feasible method for generating parametric CAD models automatically based on the uniform representation.

Because the BRep representation itself is a graph structure, using graph neural networks on BRep has garnered significant attention from researchers. Comparing with other methods, a weighted graph strategy is firstly proposed to represent a parametric CAD model in this paper. The node features extracted by the strategy focus on the "defining attributions", like focuses of ellipses, control points of NURBS surfaces or curves, rather than "measure attributes", such as area, length, bounding boxes and sampling UV point information of faces and curves [9,14,15]. "Measurement attributes" might be effective for classification or segmentation tasks, but could only build inexact or non-parametric models in generation tasks. Meanwhile, the adjacent matrices in this paper record three types of connection (face-edge, edge-vertex and face-vertex) rather than only face-edge connections, and the weight value in the adjacent matrices record the loop information and the topological directions of edges.

From a technical point of view, this paper presents a new graph auto-encoder network called Y-type GAE for encoding a weighted graph to a latent-space representation and decoding the latent-space matrix to a feature matrix and a adjacent matrix. Further, the latent-space representation of the weighted graph of a parametric CAD model is used to train DDPM (Denoising Diffusion Probabilistic Model) [7], and using the trained DDPM, we achieve the final aim, the generation of parametric CAD models.

Overall, the contributions of this paper are as follows. (1) A uniform structured representation for parametric CAD models is proposed. In this paper, we use a weighted graph to express a parametric CAD model, where the "defining attributes" of geometric elements are recorded, and the weighted adjacency matrix records complete topological information. (2) A new graph auto-encoder network Y-type GAE is proposed for processing a weighted graph to a latent-space matrix which can be decoded to a feature matrix and a weighted adjacency matrix by feeding to two branch decoders. (3) Based upon the proposed uniform representation strategy for parametrical CAD models, a model generation pipeline is proposed.

2 Related Work

2.1 Representations of Parametric CAD Models

In the area of traditional engineering, CAD models can be represented by constructive solid geometry (CSG), BRep, mesh and so on methods [5]. In the intersection of neural networks and CAD models, more and more researchers use CAD models as inputs of neural networks. Some researchers process CAD models to multi-view images [30], voxels [18] or point clouds [6], and feed the processed data

into neural networks. Although these methods are plausible on some tasks, the accuracy of the original parametric CAD models would lose inevitably. Driven by the problem, there are researchers using BRep as input of neural networks, for example, UV-Net [9], BRepNet [14], BRepGAT [15], BRep-BERT [20], BrepGen [29] and so on. These methods have been used on classification, retrieval or segmentation tasks and could generate non-parametric CAD models, but these representation methods pay more attention on "measure attributes", therefore they do not have the ability to generate accurate parametric models. Motivated by the situation, this paper propose a new feature extraction strategy where the focus will fall on "defining attributes", and a new representation network to give uniform representations for parametric CAD models expressed by BRep.

2.2 3D Model Generation

From the point of view of the representation methods of generated 3D models, most existing generation methods could be divided into the following categories: mesh models [25], voxel models [22], CSG trees in terms of geometric primitives [10,23] and CAD program in terms of parametric operation sequences [4,17]. Mesh models, point cloud models and voxel models are non-parametric, and the other two methods can only generate limited CAD models because the types of geometric primitives or operations cannot be fully covered. Few recent works try to rebuild 3D models based on BRep. BrepGen [29] uses two VAEs to encode faces and edges information which are expressed by UV sampling points and the endpoints of edges, and two diffusion modules to generate the latent-space representations of faces and edges, respectively. Decoding the latent-space representations, BrepGen finally obtain a series UV sampling points of faces and edges of 3D models. Obviously, the generated 3D models by BrepGen are non-parametric. ComplexGen [6] uses cloud points as input to search geometric primitives (like faces, edges and vertices) and the relationships between primitives, and finally obtains a unstructured chain-like representation of BRep. Comparing with aforementioned works, this paper is committed to establish a uniform structured representation for generating parametric CAD models, and the representation should be expressed as one matrix for a parametric CAD model rather than multiple matrices.

2.3 Graph Auto-Encoders for Parametric CAD Generation

Using graphs to express BRep is common, and some recent works use graph networks to get graph embeddings of BRep information [9,11]. However, the graphs processed by these models are unweighted, which should be due to the application scenarios are model classification or segmentation. For generating parametric CAD models, this paper use weighted graphs to express BRep, and use a graph auto-encoder with two decoder branches to get graph embeddings which can be decoded as feature matrices and adjacent matrices.

Graph Auto-Encoders (GAEs) has shown great potential in processing and analyzing complex graph data, supporting various research and application

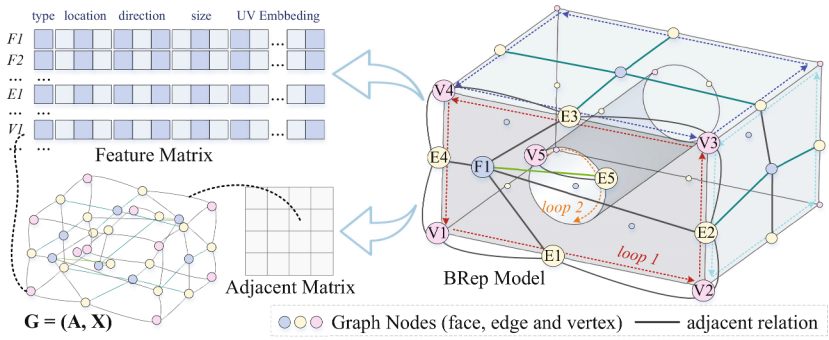


Fig. 1. Obtain a weighted graph $\mathbf{G} = (\mathbf{A}, \mathbf{X})$ from BRep model $S(F, E, V)$. A BRep model includes 3 kind of elements: faces, edges and vertices. The boundary of a face is defined by edges, which are connected to form closed loops, and the boundary of an edge is defined by vertices. A weighted graph $\mathbf{G} = (\mathbf{A}, \mathbf{X})$ consists of the 3 kind of nodes. The feature matrix $\mathbf{X}$ expresses 5 feature attributes which can be extracted from S, and the adjacent matrix $\mathbf{A}$ expresses the node relations. In the figure, different colored lines mark relations with different weights.

areas. According to the output of decoders of existing GAE architectures can be divided into 3 type: feature matrices only [24], adjacent matrices only [12,21], both feature and adjacency matrices [19,26]. The third type GAEs are similar with the Y-type GAE proposed in this paper, but most of them mainly focuses on link prediction or node classification tasks instead of generation tasks, and few of them pays attention on weighted graphs.

3 Method

3.1 BRep Information Extraction

Consider a BRep model $S(F, E, V)$ composed by faces $F = \{F_i\}$, edges $E = \{E_j\}$ and vertices $V = \{V_k\}$. The loops used to bound the face F_i are expressed by $L_{i,l}$, $l \in \mathbb{Z}_+$. In this paper, we use a weighted graph $\mathbf{G} = (\mathbf{A}, \mathbf{X})$ to describe the model S. Figure 1 illustrates the method to obtain a weighted graph from a BRep model. The detail of the 5 feature attributes are described in Table 1. The adjacent relations can be divided to 3 kinds. (1) **Face & Defining Edge:** The weight is l when F_i is bounded by $L_{i,l}$, and $E_j \in L_{i,l}$, and otherwise, it is 0. (2) **Edge & Defining Vertex:** The weight is 1 when E_j is bounded by V_k, and otherwise, it is 0. (3) **Face & Defining Vertex:** The weight is 1 when F_i is defined by V_k, and otherwise, it is 0.

3.2 The Pipeline

Through extracting BRep information from a CAD model, we obtain a weighted graph $\mathbf{G} = (\mathbf{A}, \mathbf{X})$, where $\mathbf{A}$ can be constructed directly, while $\mathbf{X}$ has to pre-process the UV point data as a fixed-size embedding in order to guarantee $\mathbf{X}$ is

Table 1. The Feature Attributes Used to Express the Feature Matrix of a Graph

Attribute	Description
type	Face: plane, cylinder, sphere, torus, Bspline surface, other; Edge: line, circle, ellipse, hyperbola, parabola, Bspline curve, other; Point: basic point, point for defining surface or curves
location[1]	the center of gravity of a node
direction	Face: the normal vector of a face; Edge: the tangent vector at the start point of a edge; Point: default value [0, 0, 0]
size[1]	the bounding box size of a node
UV points[2]	the coordinates of the sampling points of surfaces and curves in the UV coordinate system

[1]Models are scaled to the unit space and placed at the origin to alleviate translation and scaling errors.
[2]This attribute is used to reconstruct surfaces and curves that are marked as other type.

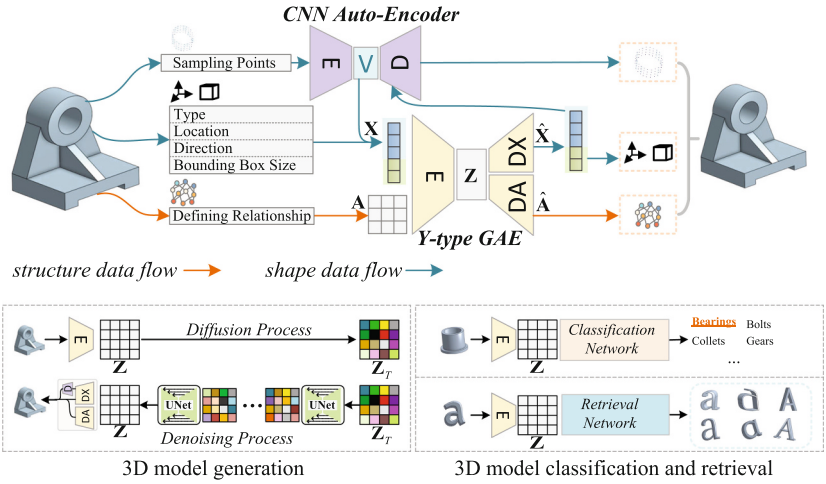


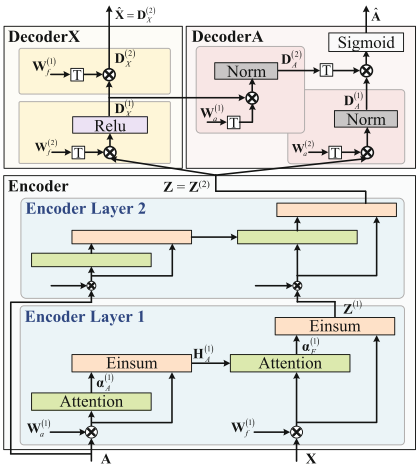
Fig. 2. The pipeline of the representation method of parametric CAD models.

structured. The preprocessing network for UV point data is an auto-encoder network based on a convolutional neural network. Figure 2 illustrates the pipeline of the representation method proposed in this paper. The representation method is to design an auto-encoder network structure that receives $\mathbf{X}$ and $\mathbf{A}$ as input data; compresses the input to a latent representation $\mathbf{Z}$, and uses two decoders to reconstruct an estimation graph $\hat{\mathbf{G}} = (\hat{\mathbf{A}}, \hat{\mathbf{X}})$. Furthermore, as shown in Fig. 2, the latent representations of parametrical CAD models can be used for 3D model generation or other applications, like 3D model classification and retrieval.

3.3 Y-type Graph Auto-Encoder (Y-type GAE)

The Encoder and Decoders of Y-type GAE all have 2 layers, which ensure that the network has strong learning ability, and the representation of graph nodes

are rich enough. Figure 3 illustrates the architecture and the forward propagation algorithm of Y-type GAE. The Y-type GAE include four hyperparameters: the default number of graph nodes N, the dimensions of node features $d^{(0)}$, the dimension of the output representation of 1st and 2nd encoder layers $d^{(1)}$ and $d^{(2)}$. Hence, $\mathbf{X}$ can be expressed as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]^T$, where $\mathbf{x}_i \in \mathbb{R}^{d^{(0)}}$, $\mathbf{X} \in \mathbb{R}^{N \times d^{(0)}}$ and means, and $\mathbf{A}$ can be expressed as $\mathbf{A} \in \mathbb{R}^{N \times N}$.



Algorithm 1 The forward propagation algorithm of Y-type GAE

Input: Graph $\mathbf{G} = (\mathbf{A}, \mathbf{X})$; Hyperparameters N, $d^{(0)}$, $d^{(1)}$ and $d^{(2)}$
Output: Graph representation $\mathbf{Z}$, Estimating Graph $\hat{\mathbf{G}} = (\hat{\mathbf{A}}, \hat{\mathbf{X}})$

Initialize $\mathbf{W}_a^{(k)}$, $\mathbf{W}_f^{(k)}$, $\mathbf{V}_0^{(k)}$ and $\mathbf{V}_1^{(k)}$, $\forall k \in \{1, 2\}$
for *epoch* $\leftarrow 1$ **to** P **do**
  Input $\{\mathbf{A}, \mathbf{X}\}$ into the Encoder Layer 1 to get $\mathbf{Z}^{(1)}$:
    Calculate the attention coefficient $\alpha_A^{(1)}$ according to (1)
    Calculate the adjacent latent representation $\mathbf{H}_A^{(1)}$ according to (2)
    Calculate the attention coefficient $\alpha_F^{(1)}$ according to (3)
    Calculate the graph latent representation $\mathbf{Z}^{(1)}$ according to (4)
  Input $\{\mathbf{A}, \mathbf{Z}^{(1)}\}$ into the Encoder Layer 2 to get $\mathbf{Z}$:
    Calculate $\alpha_A^{(2)}$ according to (1)
    Calculate $\mathbf{H}_A^{(2)}$ according to (2)
    Calculate $\alpha_F^{(2)}$ according to (3)
    Calculate $\mathbf{Z}^{(2)}$ (i.e., $\mathbf{Z}$) according to (4)
  Input $\mathbf{Z}$ into the DecoderX Layer 1 to get $\mathbf{D}_X^{(1)}$ according to (5)
  Input $\mathbf{D}_X^{(1)}$ into the DecoderX Layer 2 to get $\hat{\mathbf{X}}$ according to (5)
  Input $\mathbf{Z}$ into the DecoderA Layer 1 to get $\mathbf{D}_A^{(1)}$ according to (6)
  Input $\mathbf{D}_X^{(1)}$ into the DecoderA Layer 2 to get $\mathbf{D}_A^{(2)}$ according to (6)
  Using $\{\mathbf{D}_A^{(1)}, \mathbf{D}_A^{(2)}\}$ to get $\hat{\mathbf{A}}$ according to (7)
end for

Fig. 3. The architecture and the forward propagation algorithm of Y-type GAE model.

Every encoder layer generates a latent representation $\mathbf{Z}^{(k)}, k = 1, 2$ to express the comprehensive information of the inputs $\mathbf{X}$ and $\mathbf{A}$. In each encoder layer, we use a self-attention mechanism to find the characters of $\mathbf{A}$ firstly, and the attention coefficient of $\mathbf{A}$ can be calculated by the following function.

$$\alpha_A^{(k)} = \text{Softmax}(\mathbf{A}\mathbf{W}_a^{(k)}(\mathbf{A}\mathbf{W}_a^{(k)})^T) \tag{1}$$

where $\mathbf{W}_a^{(k)}$ is a trainable parameter, and $\mathbf{W}_a^{(k)} \in \mathbb{R}^{N \times d^{(k)}}, k = 1, 2$. Using the attention of the adjacent matrix, we can obtain a latent representation of the adjacent matrix $\mathbf{H}_A^{(k)}, k = 1, 2$.

$$(\mathbf{H}_A^{(k)})_{pq} = \textstyle\sum_r (\alpha_A^{(k)})_{pq} (\mathbf{A}\mathbf{W}_a^{(k)})_{pr} \tag{2}$$

Then we use another self-attention structure to find the relevance between the input node features, and the attention coefficient can obtain by using

$$\alpha_F^{(k)} = \text{Softmax}(\textstyle\sum_p (\mathbf{H}_A^{(k)})_{pq} (\mathbf{\Gamma}^{(k)}\mathbf{V}_0^{(k)})_{pr} + \sum_q (\mathbf{H}_A^{(k)})_{pq} (\mathbf{\Gamma}^{(k)}\mathbf{V}_1^{(k)})_{pr}) \tag{3}$$

where $\mathbf{W}_f^{(k)} \in \mathbb{R}^{d^{(k-1)} \times d^{(k)}}$, $\mathbf{V}_0^{(k)} \in \mathbb{R}^{d^{(k)} \times d^{(2)}}$, $\mathbf{V}_1^{(k)} \in \mathbb{R}^{d^{(k)} \times d^{(2)}}, k = 1, 2$ are trainable parameters, and $\mathbf{\Gamma}^{(k)} = \mathbf{Z}^{(k-1)}\mathbf{W}_f^{(k)}$. Using the attention $\alpha_F^{(k)}$, we can obtain a latent representation of the input graph $\mathbf{Z}^{(k)}, k = 1, 2$.

$$(\mathbf{Z}^{(k)})_{pr} = \textstyle\sum_q (\alpha_F^{(k)})_{pq} (\mathbf{Z}^{(k-1)}\mathbf{W}_f^{(k)})_{pr} \tag{4}$$

where $\mathbf{Z}^{(0)} = \mathbf{X}$, and the uniform code of the input graph $\mathbf{Z} = \mathbf{Z}^{(2)}$.

The 2-layer DecoderX is used to generate an estimating feature matrix $\hat{\mathbf{X}}$, which can be obtain by

$$\hat{\mathbf{X}} = \mathbf{D}_X^{(2)} = \mathbf{D}_X^{(1)}(\mathbf{W}_f^{(1)})^T = \mathrm{Relu}(\mathbf{Z}(\mathbf{W}_f^{(2)})^T)(\mathbf{W}_f^{(1)})^T \tag{5}$$

where $\mathbf{D}_X^{(1)}$ and $\mathbf{D}_X^{(2)}$ are the output matrices of the 1st and 2nd DecoderX layers, respectively.

The 2-layer DecoderA is used to generate an estimating adjacency matrix $\hat{\mathbf{A}}$. Each layer of DecoderA generates a latent adjacent representation.

$$\mathbf{D}_A^{(k)} = \mathrm{Norm}(\mathbf{D}_X^{(k-1)}(\mathbf{W}_a^{(2-k+1)})^T), k = 1, 2 \tag{6}$$

where $\mathbf{D}_X^{(0)} = \mathbf{Z}$. Combining the outputs of the two layers of DecoderA, we finally obtain the estimating adjacency matrix.

$$\hat{\mathbf{A}} = \mathrm{Sigmoid}(\mathbf{D}_A^{(1)}(\mathbf{D}_A^{(2)})^T) \tag{7}$$

The loss $\mathcal{L}$ of the network Y-type GAE consists of two parts: the feature loss $\mathcal{L}_X$ and the structure loss $\mathcal{L}_A$, i.e. $\mathcal{L} = \mathcal{L}_X + \mathcal{L}_A$. The feature loss $\mathcal{L}_X$ is calculated by the distance between the input feature matrix $\mathbf{X}$ and the estimating feature matrix $\hat{\mathbf{X}}$, i.e. $\mathcal{L}_X = ||\hat{\mathbf{X}} - \mathbf{X}||_2$. The structure loss $\mathcal{L}_A$ is calculated by the weighted cross-entropy of the input adjacent matrix $\mathbf{A}$ and the estimating adjacency matrix $\hat{\mathbf{A}}$.

$$\mathcal{L}_A = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} -\mathbf{W}_{ij} \Big[(\mathbf{A}_{ij} \log(\hat{\mathbf{A}}_{ij}) + (1 - \mathbf{A}_{ij}) \log(1 - \hat{\mathbf{A}}_{ij})) \Big] \tag{8}$$

where $\mathbf{W} \in \mathbb{R}^{N \times N}$ is the weights of the structure loss, and $\mathbf{W}_{ij} = 1$, if $\mathbf{A}_{ij} = 0$, otherwise, $\mathbf{W}_{ij} = \frac{\sum_{i,j} \mathbf{A}_{ij}}{M^2}$, where M is the truth number of graph nodes.

4 Experiments

All experiments are implemented on 2 NVIDIA H800 GPU under the Pytorch framework. For Y-type GAE, the default number of graph nodes is $N = 1024$; the dimension of node features is $d^{(0)} = 48$; the dimensions of output representations of encoder layers are $d^{(1)} = 512$, $d^{(2)} = 1024$. The learning rate is 0.0001 with Adam optimizer. The Y-type GAE has been trained 3000 epochs.

Table 2. The Generalization Experiment Results

Metrics	X: R2 Score			A: Accuracy		
Dataset	ABC	SolidLetters	FabWave	ABC	SolidLetters	FabWave
Raw input	0.98287	0.98826	0.98895	0.99369	0.95134	0.99200
Noisy input	0.97501	0.97557	0.96237	0.98788	0.95687	0.99231
Incomplete input	0.96420	0.96411	0.94526	0.98816	0.96743	0.99730

4.1 Y-type GAE

► Generalization Experiments

For evaluating the generalization performance of the Y-type GAE, we train the CNN auto-encoder and Y-type GAE networks on ABC dataset [13], and test the trained networks on ABC, SolidLetters [9] and FabWave [2] datasets. Moreover, for further evaluate the generalization performance, we test on noisy input (by adding Gaussian Noise $\mathcal{N}(0, 0.2)$) and incomplete input (by randomly missing data at a 20% proportion). The evaluation metrics of feature matrices used in generalization experiments is R2 score, and the evaluation metrics of adjacency matrices is accuracy. The experiment results (Table 2) illustrate that the trained model on ABC dataset generalize well on the other two datasets, as well as noisy and incomplete input, and the network proposed in this paper has strong generalization ability.

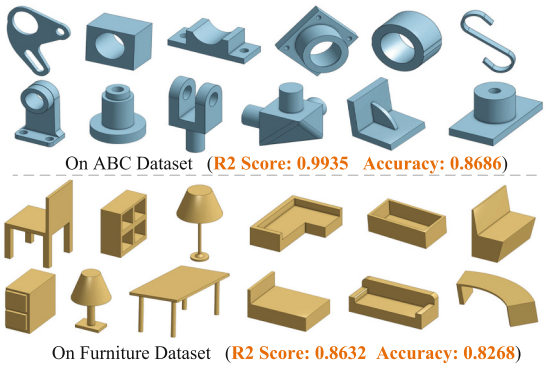
►Ablation Study

In order to find the most effective component of the Y-type GAE, we design three variants of the Y-type GAE: 1) the variant *Y-type GAE/*αA removes the attention for adjacent matrices ($\alpha_A^{(k)}$); 2) the variant *Y-type GAE/*αF removes the attention for feature matrices ($\alpha_F^{(k)}$); 3) the variant *Y-type GAE/2nd L* removes the 2nd Layers of Encoder and Decoders.

Table 3. The Ablation Test Results on node classification tasks

Metrics	R2 Score			Accuracy		
Dataset	ABC	SolidLetters	FabWave	ABC	SolidLetters	FabWave
Y-type GAE/αA	0.97671	0.93293	0.98865	0.02207	0.05886	0.01741
Y-type GAE/αF	−0.03385	−17.49834	−19.42860	0.98378	0.89562	0.98414
Y-type GAE/2nd L	0.98627	0.98954	0.99035	0.95851	0.92246	0.96576
Y-type GAE	0.98287	0.98826	0.98895	0.99369	0.95134	0.99200

The ablation test results are shown in Table 3. In generalization tasks, both the self-attention mechanisms for feature matrices and adjacent matrices have



Method	Val.*↑	MMD†↓	JSD‡↓
BRepGen	48.2%	1.35	3.69
Y-type GAE	66.6%	1.78	4.18

* Val. ↑ measures the ratio of valid models in the generated samples.
† MMD ↓ measures the averaged Chamfer Distance between the each generated sample and its nearest neighbor in the test set.
‡ JSD ↓ measures the distribution difference between the generated samples and the test set after converting point clouds into 28^3 discrete voxels.

Fig. 4. Generation experiment results. The left figure present partial qualitative results on two datasets. The right table present quantitative results on three typical metrics. The metrics are follow the definition in BRepGen.

influence on reconstruction of feature and adjacency matrices, but the self-attention mechanism for feature matrices and adjacent matrices are dominant in reconstruction of feature matrices and adjacent matrices, respectively. The 1-layer network can also reconstruct feature and adjacent matrices, but the reconstruction accuracy on adjacent matrices is lower than the 2-layer network, which would seriously affect the generation effectiveness of CAD models.

4.2 Application: 3D Model Generation

In this section, we present an application experiment of the Y-type GAE, 3D model generation. The datasets used to train the CNN auto-encoder, Y-type GAE and DDPM are ABC and Furniture BRep datasets [29]. To evaluate the quality of generated models, we adapt metrics consistent with the Y-type GAE: using the average R2 score and the average accuracy to evaluate the generated feature and adjacency matrices with their closest neighbors in the reference set. In Fig. 4, we present partial generated models and relative evaluation based on ABC and Furniture BRep Datasets. The generation pipeline proposed in this paper can generate feature and adjacency information of parametric CAD models. Although the accuracy of generated adjacency matrices is relatively low, we find that generated adjacency matrices can be corrected in the subsequent parametric CAD model reconstruction process. Further, as shown in Fig. 5, we compare the representation methods of parametric CAD models based on unweighted graphs and weighted graphs. The weighted graphs ensure the effective generation of complex models.

This experiment also implies that conditional CAD generation is feasible by training a conditional latent diffusion model, for example, we can replace the VAE module in Stable Diffusion model with the Encoder of Y-type GAE. Meanwhile, using graph compression methods and de-noising methods might provide opportunities for generating more complex and precise CAD models.

Table 4. The Comparative Experiment Results on 3D Model Classification

FabWave		SolidLetters	
Method	Accuracy (%)	Method	Accuracy (%)
UVNet [9]	$94.51 \pm 0.1\%$	UVNet [9]	$97.24 \pm 0.1\%$
PointNet(2048) [3]	$80.08 \pm 3.61\%$	PointNet(2048) [3]	$94.72 \pm 0.17\%$
DGCNN(2048) [27]	$69.95 \pm 2.37\%$	DGCNN(2048) [27]	$96.62 \pm 0.13\%$
FuS-GCN [8]	98.2%	-	-
Y-type GAE (ours)	$98.78 \pm 0.15\%$	Y-type GAE (ours)	$98.65 \pm 0.1\%$

4.3 Application: 3D Model Classification

In this section, we compare the 3D model classification pipeline presented in Fig. 2 against some state-of-the-art baselines. The classification network is ResNet. The datasets used in this part are SolidLetters and FabWave, which are same with the datasets reported in the baselines. The evaluation metrics is accuracy of classification. The experiment results are shown in Table 4 where the data of the baselines are reported in the published papers. From Table 4, we can find that the network Y-type GAE outperforms present state-of-the-art representation networks in 3D model classification tasks.

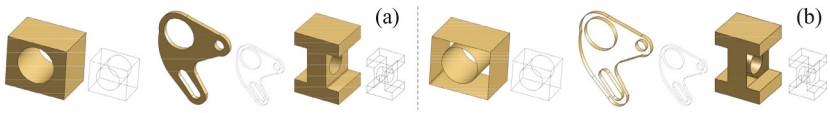


Fig. 5. Comparison of representation methods. The subfigure (a) and (b) show the generated models represented by weighed and unweighted graphs, respectively. Unweighted graphs cannot express the loop information of BRep models, as a result, using unweighted graphs cannot rebuild faces with multiple loops.

5 Conclusion

This paper presents a representation method of parametric CAD models, and applies the representation method on DDPM to achieve the automatic generation of parametric CAD models. In order to provide a uniform representation for parametric CAD models, this paper uses weighted graphs to represent both the geometrical and topological information of BRep. In weighted graphs, this paper introduces definition vertex nodes, and the feature attributes of graph nodes pay attention to "defining attributes" rather than "measurement attributes"; the weighted adjacent matrices use weights to express the bounded and definition information of topological elements. On the other hand, this paper introduces a new network structure Y-type GAE which encode a weighted graph to a latent

representation and decode it to feature and adjacency matrices. From experiment results, we can find that the Y-type GAE has strong generalization ability. The latent-space representation can be used on DDPM to generate parametric CAD models effectively. Meanwhile, in 3D model classification tasks, the representation strategy surpasses present state-of-the-art networks. In the future, we will continue to strive in the direction of generating parametric CAD models, addressing the difficulties in generating extremely large models due to the limited number of graph nodes at present, and exploring the impact of the order of graph nodes on the generation process.

References

1. Alam, M.F., Ahmed, F.: Gencad: Image-conditioned computer-aided design generation with transformer-based contrastive representation and diffusion priors. arXiv preprint arXiv:2409.16294 (2024)
2. Angrish, A., Craver, B., Starly, B.: "Fabsearch" : a 3D cad model based search engine for sourcing manufacturing services. J. Comput. Inf. Sci. Eng. **19** (2018)
3. Charles, R.Q., Su, H., Kaichun, M., Guibas, L.J.: Pointnet: deep learning on point sets for 3d classification and segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 77–85 (2017)
4. Dupont, E., et al.: Cadops-net: jointly learning cad operation types and steps from boundary-representations. In: 2022 International Conference on 3D Vision (3DV), pp. 114–123 (2022)
5. Dupont, E., Cherenkova, K., Mallis, D., Gusev, G., Kacem, A., Aouada, D.: Transcad: a hierarchical transformer for cad sequence inference from point clouds. In: European Conference on Computer Vision, pp. 19–36. Springer (2024)
6. Guo, H., Liu, S., Pan, H., Liu, Y., Tong, X., Guo, B.: Complexgen: cad reconstruction by b-rep chain complex generation **41**(4) (2022)
7. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Advances in Neural Information Processing Systems, vol. 33, pp. 6840–6851 (2020)
8. Hou, J., Luo, C., Qin, F., Shao, Y., Chen, X.: FUS-GCN: Efficient b-rep based graph convolutional networks for 3D-cad model classification and retrieval. Adv. Eng. Inform. **56**, 102008 (2023)
9. Jayaraman, P.K., Sanghi, A., Lambourne, J., Davies, T., Shayani, H.: UV-net: Learning from boundary representations. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11698–11707 (2021)
10. Kania, K., Ziundefinedba, M., Kajdanowicz, T.: UCSG-net - unsupervised discovering of constructive solid geometry tree. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS 2020, Curran Associates Inc., Red Hook (2020)
11. Kesler, S., Slama, D.: Grastnet: Graph neural networks and set transformer for machining feature recognition. In: 2024 International Conference on Control, Automation and Diagnosis (ICCAD), pp. 1–6 (2024)
12. Kipf, T.N., Welling, M.: Variational graph auto-encoders. arXiv preprint arXiv:1611.07308 (2016)
13. Koch, S., et al.: ABC: a big cad model dataset for geometric deep learning. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9593–9603 (2019)

14. Lambourne, J., Willis, K.D.D., Jayaraman, P.K., Sanghi, A., Meltzer, P., Shayani, H.: BRepNet: a topological message passing system for solid models. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12768–12777 (2021)
15. Lee, J., Yeo, C., Cheon, S.U., Park, J.H., Mun, D.: Brepgat: graph neural network to segment machining feature faces in a b-rep model. J. Comput. Des. Eng. **10**, 2384–2400 (2023)
16. Lee, S.H., Lee, K.: Partial entity structure: a compact non-manifold boundary representation based on partial topological entities. In: International Conference on Smart Media and Applications (2001)
17. Li, C., Pan, H., Bousseau, A., Mitra, N.J.: Free2cad: parsing freehand drawings into cad commands. ACM Trans. Graph. **41**(4) (2022)
18. Li, P., Guo, J., Zhang, X., Yan, D.M.: SECAD-Net: self-supervised cad reconstruction by learning sketch-extrude operations. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 16816–16826 (2023)
19. Li, X., Ye, T., Shan, C., Li, D., Gao, M.: Seegera: self-supervised semi-implicit graph variational auto-encoders with masking. In: Proceedings of the ACM Web Conference 2023, WWW 2023, pp. 143–153. Association for Computing Machinery, New York (2023)
20. Lou, Y., Li, X., Chen, H., Zhou, X.: BREP-BERT: pre-training boundary representation BERT with sub-graph node contrastive learning. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (2023)
21. Pan, S., Hu, R., Long, G., Jiang, J., Yao, L., Zhang, C.: Adversarially regularized graph autoencoder for graph embedding. arXiv preprint arXiv:1802.04407 (2018)
22. Sanghi, A., et al.: Clip-forge: towards zero-shot text-to-shape generation. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 18582–18592 (2022)
23. Sharma, G., Goyal, R., Liu, D., Kalogerakis, E., Maji, S.: CSGNET: neural shape parser for constructive solid geometry. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5515–5523 (2017)
24. Sun, C.: Hat-GAE: self-supervised graph auto-encoders with hierarchical adaptive masking and trainable corruption. Knowl. Based Syst. **279**, 110920 (2023)
25. Tang, J., et al.: Edgerunner: auto-regressive auto-encoder for artistic mesh generation. arXiv preprint arXiv:2409.18114 (2024)
26. Wang, W., et al.: HGATE: heterogeneous graph attention auto-encoders. IEEE Trans. Knowl. Data Eng. **35**(4), 3938–3951 (2023)
27. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph CNN for learning on point clouds **38**(5) (2019)
28. Weiler, K.J.: Topological structures for geometric modeling. PHD Thesis Rensselelaer Polytechnic Institute (1986)
29. Xu, X., Lambourne, J.G., Jayaraman, P.K., Wang, Z., Willis, K.D., Furukawa, Y.: Brepgen: a B-REP generative diffusion model with structured latent geometry. arXiv preprint arXiv:2401.15563 (2024)
30. You, Y., et al.: Img2CAD: reverse engineering 3d cad models from images through vlm-assisted conditional factorization. arXiv preprint arXiv:2408.01437 (2024)

Surrogate-Assisted Multi-objective Design of Complex Multibody Systems

Augustina C. Amakor[1,2(✉)], Manuel B. Berkemeier[1,2], Meike Wohlleben[3], Walter Sextro[3], and Sebastian Peitz[1,2]

[1] Department of Computer Science, TU Dortmund, Dortmund, Germany
{augustina.amakor,manuel.berkemeier,sebastian.peitz}@tu-dortmund.de
[2] Lamarr Institute for Machine Learning and Artificial Intelligence, Dortmund, Germany
[3] Faculty of Mechanical Engineering, Paderborn University, Paderborn, Germany
{meike.wohlleben,walter.sextro}@upb.de

Abstract. Optimizing large-scale multibody systems is a challenging task, particularly in the presence of multiple conflicting criteria. To prevent high simulation costs, surrogate models constructed from a small number of expensive model evaluations are very popular. However, it is difficult to ensure the optimality of the obtained solutions using a single pre-computed model. We present a back-and-forth approach between surrogate modeling and multi-objective optimization, and we compare different strategies for optimization, sampling, and surrogate modeling, to identify the most promising approach in terms of computational efficiency and solution quality.

Keywords: multi-objective optimization · surrogate modeling · evolutionary algorithms · gradient · multibody system · machine learning

1 Introduction

Multibody systems are composed of interconnected rigid or flexible bodies that interact with each other through joints, forces, or constraints [17,18]. They are described by a set of coupled ordinary (or partial) differential equations and algebraic constraints, with examples ranging from robots over vehicle suspension systems or wind turbines to human biomechanics. With increasing numbers of components, the numerical simulation becomes increasingly expensive, rendering multi-query tasks such as optimization or control very costly. This issue is further amplified in the context of *multi-objective optimization* [14]. There, we want to optimize multiple conflicting criteria at the same time, such that we need to calculate the entire *Pareto set* of optimal compromises.

As solving *multi-objective optimization problems* (*MOPs*) is computationally expensive, surrogate-assisted approaches have been studied extensively [4], where we train a surrogate function using a small number of expensive model evaluations at carefully chosen points. Modeling techniques range from polynomials over radial basis functions and Kriging models to neural networks, and there is

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 251–262, 2026.
https://doi.org/10.1007/978-3-032-04555-3_21

a distinction between global approximations and ones that are valid only locally. Besides smaller models, local surrogates may allow for error analysis through trust-region techniques [1], at the cost of requiring additional intermittent evaluations of the original objective function. Deep learning has found its entrance into this area of research as well, see [16] for a recent overview. Deep surrogate models were suggested in [3,24], and generative Kriging modeling was studied in [10]. Another generative modeling approach called GFlowNets was proposed in [11], and the usage of large language models was suggested in [13].

In this paper, we present an algorithm (in Sect. 3) for calculating the entire Pareto front as accurately as possible, while respecting a prescribed budget of expensive calculations, with a specific focus on expensive-to-simulate multibody systems (the basics are introduced in Sect. 2.1). We compare a single offline training phase with a back-and-forth procedure between surrogate modeling (Sect. 2.3) and multi-objective optimization (2.2). In the latter case, surrogate-based Pareto sets (using either radial basis functions or deep neural networks as surrogate models) can be used for more informed sampling. Our findings from experiments (Sect. 4) and results (Sect. 5) suggest that an interactive coupling between the well-known NSGA-II algorithm [6] and a moderate number of Pareto-informed sampling steps for neural network surrogates is the best way to find near-optimal solutions.

2 Preliminaries

2.1 Multibody Systems

Multibody systems (MBS) consist of three fundamental elements: bodies with mass, joints connecting these bodies, and forces or moments acting on the bodies. The bodies can be either rigid or flexible, but for simplicity, we will focus on rigid bodies. Likewise, we restrict ourselves here to translations and the conservation of impulse, although the same approach is equally valid for rotations using the conservation of momentum. Forces in the system are classified as either applied forces, such as those generated by massless springs or dampers, or reaction forces, which represent the internal interactions at the joints. Using the Newton-Euler formalism, the equations of motion are

$$M(y,t)\ddot{y}(t) + C(y,\dot{y},t) = Q(y,\dot{y},t). \tag{1}$$

Here, $M(y,t)$ represents the mass matrix, $C(y,\dot{y},t)$ is the vector of generalized Coriolis, centrifugal, and gyroscopic forces (i.e., reaction forces), and $Q(y,\dot{y},t)$ is the vector of generalized applied forces. The vector y denotes the positions, $\dot{y} = \frac{\mathrm{d}y}{\mathrm{d}t}$ the velocities, and $\ddot{y} = \frac{\mathrm{d}^2y}{\mathrm{d}t^2}$ the accelerations.

2.2 Multi-objective Optimization

Consider the situation where instead of a single loss function, we have a vector with K conflicting losses which we want to minimize at the same time,

$$\min_{x\in\mathbb{R}^N} \left[f_1(x), \cdots, f_K(x)\right]^\top. \tag{MOP}$$

If the objectives are conflicting, there does not exist a single optimal x^* that minimizes all f_k, but a *Pareto set* $\mathcal{P}$ with optimal trade-off arguments,

$$\mathcal{P} = \left\{ x \in \mathbb{R}^N \;\middle|\; \nexists \hat{x} : \begin{array}{l} f_k(\hat{x}) \leq f_k(x) \text{ for } k = 1, \ldots, K, \\ f_k(\hat{x}) < f_k(x) \text{ for at least one } k \end{array} \right\}.$$

In other words, a point $\hat{x}$ *dominates* a point x, if it is at least as good in all f_k while being strictly better with respect to at least one loss. The Pareto set $\mathcal{P}$ thus consists of all non-dominated points. The corresponding set in objective space is called the *Pareto front* $\mathcal{P}_\mathcal{F} = f(\mathcal{P})$. Under smoothness assumptions, both objects have dimension $K - 1$ i.e., $\mathcal{P}$ and $\mathcal{P}_\mathcal{F}$ are lines for two objectives, 2D surfaces for three objectives, and so on [9]. There exists a large number of conceptually very different methods to find (approximate) solutions of (MOP).

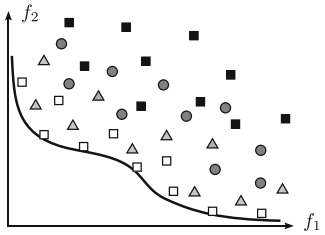


Fig. 1. MOEA: a population of individuals is improved from one generation to the next (■ → ◯ → △ → □).

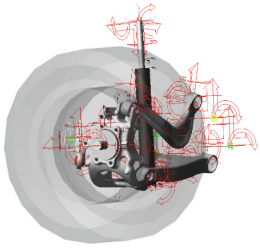


Fig. 2. Trapezoidal link rear suspension system under consideration.

To calculate single points, *multiple-gradient descent algorithms (MGDAs)* [7,22] are increasingly popular, in particular when it comes to very high-dimensional problems. If we want to postpone the decision-making to take a more informed decision, we need to calculate the entire Pareto set $\mathcal{P}$ and front $\mathcal{P}_\mathcal{F}$. Besides various extensions to MGDA or using scalarization techniques, we can directly consider a *population* of weights $\left\{x^{(j)}\right\}_{j=1}^{M}$ that we iteratively update to improve each individual's performance while also ensuring a suitable spread over the entire front $\mathcal{P}_\mathcal{F}$. Algorithms of this class are referred to as *multi-objective evolutionary algorithms (MOEAs)* [5], see Fig. 1 for an illustration. Therein, the population's fitness is increased from generation to generation by maximizing a criterion that combines optimality (or non-dominance) with a spreading criterion. The most popular and widely used algorithm in this category is likely NSGA-II [6], but there are many alternatives [25].

2.3 Surrogate Modeling

Solving MOPs is naturally more expensive than identifying a minimizer of a single-objective problem. Consequently, expensive objectives f quickly results in prohibitively large computational costs. It is thus a straightforward idea to replace $f : \mathbb{R}^N \to \mathbb{R}^K$ by a surrogate model $g : \mathbb{R}^N \times \mathbb{R}^P \to \mathbb{R}^K$ which depends on P trainable parameters $\theta \in \mathbb{R}^P$. There is a lot of research on surrogate modeling,

either directly for the objective function f (in the multi-objective context), but also for the simulation of complex dynamical systems.

Given a training set $\mathcal{D} = \left\{\left(x^{(i)}, f\left(x^{(i)}\right)\right)\right\}_{i=1}^{s}$ of size s, fitting g is a supervised learning problem:

$$\min_{\theta \in \mathbb{R}^P} \frac{1}{s} \sum_{i=1}^{s} \left\| f\left(x^{(i)}\right) - g\left(x^{(i)}; \theta\right) \right\|_2^2. \tag{2}$$

Instead of the mean squared loss, one can use various alternatives and also consider additional regularization terms such as the ℓ_1 or the ℓ_2 norm. As a result, we can replace the expensive (MOP) with a surrogate version

$$\min_{x \in \mathbb{R}^N} \left[g_1(x;\theta), \cdots, g_K(x;\theta)\right]^\top. \tag{$\widehat{\text{MOP}}$}$$

We will denote the Pareto set and front of ($\widehat{\text{MOP}}$) by $\widehat{\mathcal{P}}$ and $\widehat{\mathcal{P}}_{\mathcal{F}}$, and the central goal is to obtain $\widehat{\mathcal{P}} \approx \mathcal{P}$ (or alternatively a set of solutions such that $f(\widehat{\mathcal{P}}) \approx f(\mathcal{P})$). Central questions that need to be addressed in this context are:

- Which type of surrogate model should we use?
- What is the appropriate choice for s? (Small s results in low computational effort, at the cost of poor generalization towards points $(x, f(x)) \notin \mathcal{D}$.)
- Where should we sample?
- How can we ensure convergence or optimality?

We will address all of these points and propose an adaptive back-and-forth strategy between surrogate modeling and multi-objective optimization that helps us to choose samples in a more informed way while ensuring convergence. We will test our method using both MGDA and the popular evolutionary algorithm NSGA-II, as well as surrogate models based on radial basis functions and deep neural networks, respectively.

3 Methodology

In this work, we will focus on artificial neural networks (ANNs) and radial basis functions (RBFs) to approximate g.

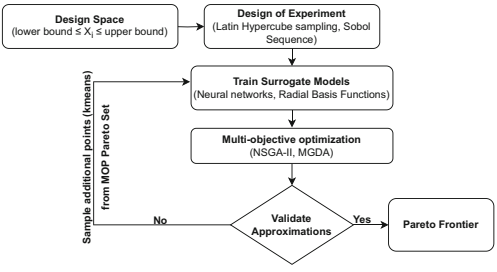


Fig. 3. Sketch of the methodology.

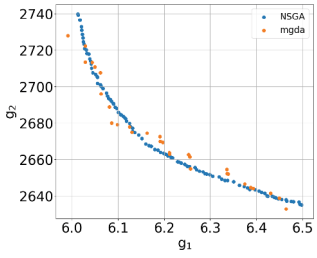


Fig. 4. NSGA versus MGDA with run times 1.24 and 352.24 secs, respectively, for one run of Algorithm 1 for the MBS considered in this work.

The overall procedure is visualized in Fig. 3 and formalized in Algorithm 1. The central idea is to intertwine surrogate modeling with optimization, as is very common for trust-region methods [1]. In a first step, i.e., where nothing is known about the solution, we collect s samples from the expensive model in a random fashion. Sampling can be performed using various techniques from Design of Experiments (DoE), such as Latin hypercube, Monte-Carlo, Sobol sequences or Halton sets [23]. In our case, we are using Latin hypercube sampling.

Having calculated the first s samples and trained the first surrogate model by solving (2), we can compute an approximation of $\mathcal{P}$ and $\mathcal{P}_\mathcal{F}$ by solving ($\widehat{\text{MOP}}$) using a suitable algorithm. In our example, this accelerates the problem from approximately 30 s for a single function evaluation—which would render optimization infeasible—to just a few seconds for solving ($\widehat{\text{MOP}}$) using NSGA-II. The central issue with this approach is that for a small number s, the solution $\widehat{\mathcal{P}}$ and $\widehat{\mathcal{P}}_\mathcal{F}$ is likely far away from the true solution $\mathcal{P}$ and $\mathcal{P}_\mathcal{F}$. To improve this, we can now compute additional samples, but in a much more informed fashion by taking s additional samples equidistantly distributed over the obtained surrogate solution $\widehat{\mathcal{P}}$. For this, we use k-means clustering. This now-enhanced data set allows us to find a better approximation, and we can repeat this process until either our budget S of expensive simulations is used up, or until we reach convergence. The latter is realized by a relative stopping criterion based on the Hausdorff distance (cf. [20]), which measures the distance of two sets:

$$d_H(\mathcal{X}, \mathcal{Y}) = \max\left\{\sup_{y\in\mathcal{Y}} \hat{d}(y, \mathcal{X}), \sup_{x\in\mathcal{X}} \hat{d}(x, \mathcal{Y})\right\}. \tag{3}$$

Here, $\hat{d}(y, \mathcal{X}) = \inf_{x\in\mathcal{X}} d(y, x)$ is the Euclidean distance between a point $y \in \mathcal{Y}$ and the closest element of the set $\mathcal{X}$. Supposing that there is a Cauchy sequence of sets $\{\mathcal{X}_n\}_n$, it is sensible to deduce existence of a limit set $\mathcal{X}^*$ [21]. E.g., if $\{\widehat{\mathcal{P}}_\mathcal{F}^{(j)}\}_j$ is Cauchy, then there is a limiting set $\widehat{\mathcal{P}}_\mathcal{F}$.

Since for stopping we use the distance of the surrogate-based Pareto fronts $\widehat{\mathcal{P}}_\mathcal{F}^{(j-1)}$ and $\widehat{\mathcal{P}}_\mathcal{F}^{(j)}$ in Algorithm 1 – which are finite – the sup and inf operators can be replaced by max and min, respectively.

4 Experiment

The MBS we consider is an independent rear suspension system of a passenger car (Fig. 2), described in detail in [19]. It consists of a spring, a shock absorber, and links connecting the wheel carrier to the subframe of the axle, enabling the transfer of forces and ultimately supporting the vehicle structure. The suspension primarily provides the wheel with a vertical degree of freedom, decoupling the body from road excitations, while ensuring a robust contact between tire and road which is key to ensure safe driving behavior. At the same time, to improve passenger comfort, movements of the vehicle body should remain moderate.

The model was implemented using the Neweul-M^2 software, which generates a system of differential equations as in (1). In addition to the elements already

Algorithm 1. Surrogate-based adaptive multi-objective optimization algorithm

Require: Maximum number of evaluations of the complex model: S, Number of samples per iteration: $s \leq S$, Hausdorff distance stopping criterion $h_{\min}$, MOP solver with population size M

***Round 1**: random sampling*

1: *Sampling* of s random points $\Rightarrow \mathcal{D} = \left\{ \left(x^{(i)}, f\left(x^{(i)} \right) \right) \right\}_{i=1}^{s}$

2: *Surrogate modeling*: Solve (2) using $\mathcal{D} \Rightarrow g^{(0)}(x;\theta)$

3: *Optimization*: Solve ($\widehat{\text{MOP}}$) using $g^{(0)}(x;\theta) \Rightarrow \widehat{\mathcal{P}}^{(0)}$ and $\widehat{\mathcal{P}}_{\mathcal{F}}^{(0)}$ with M elements

***Rounds 2, 3, ...**: Pareto-informed sampling*

4: Set loop counter $j = 1$

5: **while** $(j-1) \cdot s < S$ **do**

6: *Sampling*: Determine s sample locations via *k-means clustering* of $\widehat{\mathcal{P}}^{(j-1)}$ and collect s new samples in $\widehat{\mathcal{D}}$, then add them to the previous data $\Rightarrow \mathcal{D} = \mathcal{D} \cup \widehat{\mathcal{D}}$

7: *Surrogate modeling*: Solve (2) using $\mathcal{D} \Rightarrow g^{(j)}(x;\theta)$

8: *Optimization*: Solve ($\widehat{\text{MOP}}$) $\Rightarrow \widehat{\mathcal{P}}^{(j)}$ and $\widehat{\mathcal{P}}_{\mathcal{F}}^{(j)}$

9: *Convergence:* Compute *Hausdorff distance* $h = d_H\left(\widehat{\mathcal{P}}_{\mathcal{F}}^{(j-1)}, \widehat{\mathcal{P}}_{\mathcal{F}}^{(j)} \right)$

10: **if** $h < h_{\min}$ **then**

11: **STOP**

12: $j = j + 1$

13: Calculate the Pareto subsets $\mathcal{P}$ and $\mathcal{P}_{\mathcal{F}}$ using a non-dominance test on $\mathcal{D}$

mentioned, the model includes a test rig that imposes motion on the wheel hub. The overall model consists of 13 rigid bodies (subframe, chassis, toe link, camber link, trapezoidal link, knuckle, wheel bearing, wheel hub, damper rod, damper housing, lower spring seat, upper spring seat, test rig), three point masses (body mass, unsprung mass, brake caliper), six force elements (two spring seats, body spring, damper, rebound stop, test rig), and 22 ideal joints, resulting in a system with five remaining degrees of freedom.

For optimization, the joint centers of the links can be translated within a radius of 0.003 m around their initial position. The toe link and camber link each have two joints, and the trapezoidal link has four joints such that there are a total of $N = 24$ optimization parameters: 8 (joints) $\times$ 3 (spatial directions). To evaluate safety and comfort, the test rig can impose different displacement excitations. However, since only a single wheel suspension is considered, the choice of possible driving maneuvers is limited: lateral dynamics cannot be analyzed as no lateral forces can be absorbed, which precludes cornering maneuvers. Similarly, longitudinal dynamics such as braking and driving cannot be evaluated. We thus focus on the vertical dynamics. Two conflicting criteria are considered: a safety criterion in terms of minimizing wheel load fluctuations, and a comfort criterion in the form of low body accelerations [8,15]. The driving maneuver is a sinusoidal excitation with an amplitude of 0.001 m and a frequency of 7Hz [15]. The wheel load F_z refers to the vertical forces acting on the tire (in our case, on the wheel hub), and is therefore, an entry of the vector C in (1), which contains the reaction forces. Wheel load fluctuations are the variations in wheel load at a single wheel

caused by driving on uneven roads [15]. Therefore, the safety objective function is the amplitude of F_z, estimated over the simulated time horizon t_0 to t_e by $\hat{F}_z = \frac{1}{2}\left(\max_{t\in[t_0,t_e]} F_z(t) - \min_{t\in[t_0,t_e]} F_z(t)\right)$. The body acceleration $\ddot{z}_A$ represents the vertical acceleration of the vehicle body and is an entry of the acceleration vector $\ddot{y}$. The amplitude $\hat{\ddot{z}}_A = \frac{1}{2}\left(\max_{t\in[t_0,t_e]} \ddot{z}_A(t) - \min_{t\in[t_0,t_e]} \ddot{z}_A(t)\right)$ is selected as the comfort objective function [15]. We thus have

$$\min_{x\in\mathbb{R}^{24}} f(x) = \min_{x\in\mathbb{R}^{24}} \begin{pmatrix} \hat{F}_z \\ \hat{\ddot{z}}_A \end{pmatrix}. \tag{4}$$

The objective function f is evaluated in a black-box fashion, i.e., we set the values for x, then perform a simulation over the time horizon $[t_0, t_e]$, collect (discretized) trajectories $F_z(t)$ and $\ddot{z}_A(t)$, and finally calculate $f_1(x)$ and $f_2(x)$.

The experiments are carried out on a machine with 2.10 GHz 12th Gen Intel(R) Core(TM) i7-1260P CPU and 32 GB memory, using Python 3.11.1 and MATLAB R2024b. Our implementation is in Python but calls MATLAB to evaluate (4) in a black-box fashion. Further settings are:

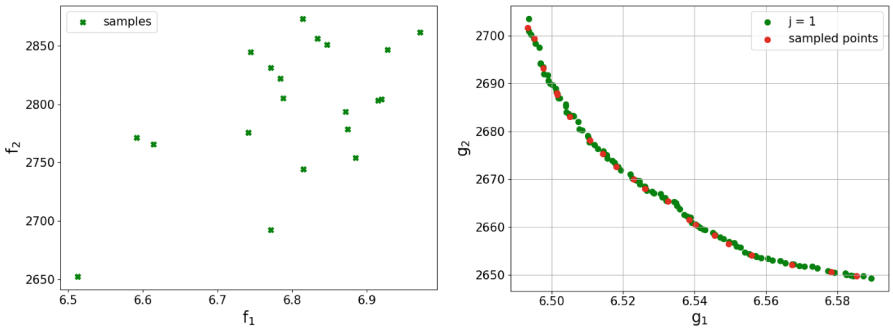
- the design space for the optimization parameters of MBS is set to the infinity norm i.e., $-0.003 \leq x_i \leq 0.003$.
- as described in Algorithm 1 the Latin hypercube is used for initial sampling and the k-means clustering for succeeding sampling.
- we found $h_{\mathsf{min}} = 2$ to be a good value for our particular application, having tried different values for h_{min} e.g., $5, 2, 1, 0.5$.

5 Results

We first show a detailed analysis over multiple iterations using NSGA-II and a neural network surrogate model. We then study different variations in terms of surrogate modeling and sample sizes s. In our experiments, we have also investigated MGDA with multi-start strategy (i.e., a set of random initial guesses $\left\{x^{(0,j)}\right\}_{j=1}^{M}$) as an alternative to NSGA-II for MOP to obtain multiple Pareto optimal points. However, we found it to be inferior both in terms of the spread across the Pareto front as well as in terms of computational times, see Fig. 4. We thus focus on NSGA-II (using the Pymoo implementation [2]) as the optimizer from now on.

5.1 NSGA-II Optimizer with ANN Surrogate Models

To analyze the performance and demonstrate the efficiency of our approach, we first use a fully connected neural network with two hidden layers of 64 neurons each as our surrogate model. For now, we set the sample size to $s = 20$. In the first loop of Algorithm 1, we begin by sampling s times from $-0.003 \leq x_i \leq 0.003$ using Latin hypercube, and compute the true values $f(x)$, cf. Fig. 5a. These 20 points are normalized and split in a 80 : 20 ratio for validation. We then



(a) (b)

Fig. 5. (a) The first $s = 20$ points computed by evaluating the complex model (4) using Latin hypercube sampling. (b) Pareto front $\widehat{\mathcal{P}}_{\mathcal{F}}$ using the ANN surrogate model (constructed using the samples in (a)) and NSGA-II. Red dots show the Pareto-informed samples, obtained via k-means clustering of $\widehat{\mathcal{P}}_{\mathcal{F}}$.

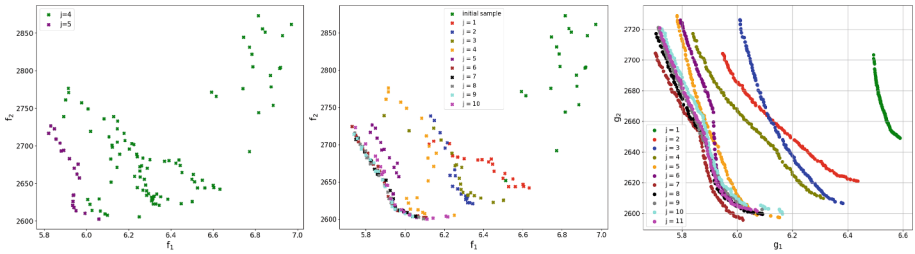
train the ANN surrogate model using the ADAM optimizer [12] and the mean squared error (cf. Equation (2)) as the error metric. We solve the surrogate-based problem ($\widehat{\text{MOP}}$) using NSGA-II with a population size of $M = 100$, $i_{\max} = 200$ generations, a simulated binary crossover of probability 0.5 and polynomial mutation with distribution index $\eta = 20$. For the sampling of new data points, we use k-means clustering. Figure 5b shows the first approximate Pareto front $\widehat{\mathcal{P}}_{\mathcal{F}}$ and the location of the 20 Pareto-informed samples.

With $s = 20$ and a convergence threshold of $h_{\min} = 2$ for the Hausdorff distance (a larger threshold resulting in premature termination of the algorithm), $j = 11$ iterations were required for convergence. Figure 6 shows a summary of the iterative steps. Note that expensive model evaluations $f(x)$ are denoted by cross signs, whereas surrogate-based points $g^{(j)}(x)$ are shown as circles.

The true objective function values of the k-means sampled points are shown in Fig. 6b in red for $j = 1$. In the next iteration, the data points are combined to retrain the surrogate model and compute $\widehat{\mathcal{P}}_{\mathcal{F}}$ with $M = 100$ points as illustrated in 6c (now $j = 2$). Since the stopping criterion $h < h_{\min}$ is not satisfied, 20 new data points are sampled again from $\widehat{\mathcal{P}}_{\mathcal{F}}^{(2)}$. Figures 6b and 6c show summaries of the $S = 220$ sampled data points and the corresponding $\widehat{\mathcal{P}}_{\mathcal{F}}^{(j)}$ computed in each iteration. The increasing closeness between iterations both in terms of the collected samples (6b) and the surrogate-based fronts (6c) nicely demonstrates the convergence criterion $h < h_{\min}$. A smaller value of $h_{\min}$ further improves the approximation, at the cost of more iterations.

5.2 Varying Sample Size

For further experimentation, we perform the same procedure as in Sect. 5.1 using an RBF model with a Gaussian kernel. A linear polynomial term for global



(a) (b) (c)

Fig. 6. A breakdown summary of the iterative steps taken to compute $\widehat{\mathcal{P}}_{\mathcal{F}}$ for the MBS. We use $s = 20$ as shown in Fig. 5a, then always sample 20 additional points (cf. Fig. 5b). We use an ANN surrogate model and NSGA-II. Sub-figure 6a illustrate the additional samples obtained at iteration $j = 4$ combined to compute $\widehat{\mathcal{P}}_{\mathcal{F}}$ at iteration $j = 5$. Sub-figures 6b show all sample points computed at each iteration (color coded) and their corresponding Pareto front Fig. 6c (same color code) computed in the next iterations, repeated until the Hausdorff convergence criterion is met.

interpolation and a 0.0001 regularization term for stability are added to the RBF model, while the other settings remain unchanged. Different parameter values for the kernel width, i.e., $\sigma \in \{0.1, 0.5, 1.0, 2.0, 5.0\}$, were tested and $\sigma = 0.5$ was observed to yield the smallest mean squared error. Using the RBF surrogate model, Algorithm 1 converges after 18 iterations which is more than the number of iterations required for convergence when using the ANN surrogate model. The final $\widehat{\mathcal{P}}_{\mathcal{F}}$ can be seen in Fig. 7e in green.

Figure 7 illustrates the different sample sizes of $s \in \{5, 10, 20, 30\}$, Algorithm 1 is repeated using both the ANN and RBF surrogate models. The threshold of $h_{\mathsf{min}} = 2$ as a convergence criterion for Algorithm 1 was maintained for all sample sizes. From Fig. 7 we observe that larger sample size does not guarantee fewer iteration needed for convergence. On the other hand, a better approximated Pareto front is obtained with larger sample size. For future research, adaptive sample sizes thus appear to be an interesting approach to allow for quick improvements in the beginning and a good final performance.

5.3 ZDT Test Problems

To validate the efficiency of our approach, we further performed experiments on well-known multi-objective optimization test problems, namely ZDT1, ZDT2 and ZDT3 from [26], with $x_i \in \mathbb{R}^{30}$, $0 \leq x_i \leq 1$ and $s = 20$. For stopping, we used smaller values of h i.e., $h_{\mathsf{min}} = 0.1$ and we observed that despite the fact that these are fairly simple problems, convergence takes more iterations. Choosing a higher value, such as $h_{\mathsf{min}} = 0.5, 2$, also results in a good $\widehat{\mathcal{P}}_{\mathcal{F}}$. The final approximated fronts using the ANN and RBF surrogate model for the ZDT problems are illustrated in Fig. 8 which further substantiates the effectiveness of our approach to approximate the Pareto front of (MOP).

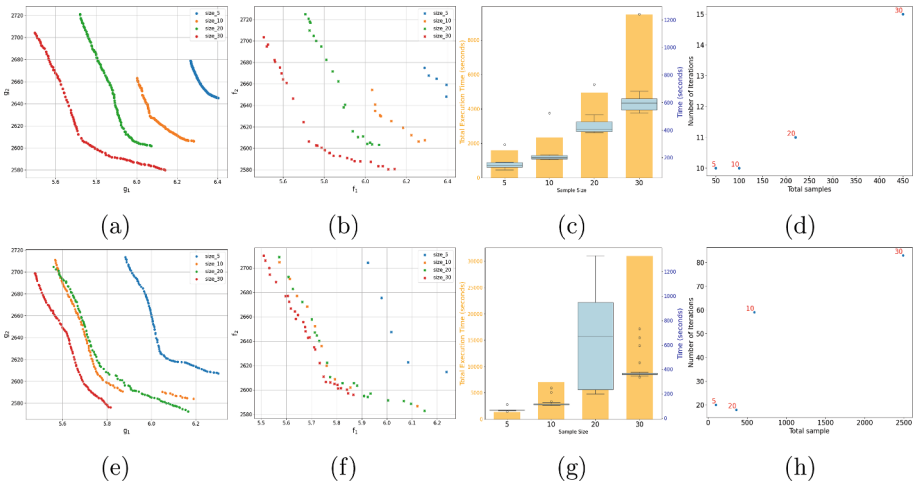


Fig. 7. (a) and (b) illustrate the final Pareto front $\widehat{\mathcal{P}}_{\mathcal{F}}$ after convergence and the last computed samples of the MBS complex model for the different sample sizes using ANN. The computation times (total and per iteration) until convergence is illustrated in (c) and (d) respectively. (c) number of iterations j versus sample size s. (e–h) same as in the top row, but for the RBF surrogate model.

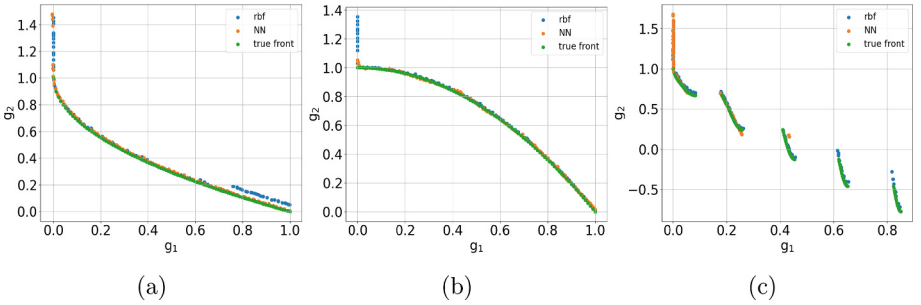


Fig. 8. (a), (b) and (c) shows the Pareto fronts $\widehat{\mathcal{P}}_{\mathcal{F}}$ for ZDT1, ZDT2 and ZD3 problems respectively where the "true front" label is the analytical front.

6 Conclusion

We have presented an adaptive surrogate-based framework for solving expensive multi-objective optimization problems. The key ingredient is a back-and-forth procedure between sampling and optimization that is controlled by a Hausdorff measure stopping criterion. We find that, as expected, trade-offs have to be made between approximation accuracy and the number of expensive model evaluations. A significant speed-up of several orders was observed with the MBS optimization task, such as in a single objective evaluation which could take 10 s or more, using a surrogate g instead of f reduces the computation time

for Pareto-optimization with a population size of 100 to only 1.24 s, whilst still giving high-quality solutions.

For future work, we will investigate a larger number of objectives as well as advanced sampling techniques. In this study, we simply used s samples from each iteration, which can result in duplicates or at least clustered sampling over multiple iterations. Moreover, we want to develop a more efficient surrogate update strategy, using only the new samples—in combination with strong regularization or early stopping—to update the model, thus allowing for more iterations and better convergence.

Acknowledgments. A. C. Amakor and S. Peitz acknowledge funding by the German Federal Ministry of Education and Research (BMBF) through the AI junior research group "Multicriteria Machine Learning" (Grant ID 01|S22064).M. B. Berkemeier, M. Wohlleben, W. Sextro and S. Peitz acknowledge funding by the DFG Priority Programme 2353 "Daring more intelligence" (Project ID 501834605).

References

1. Berkemeier, M., Peitz, S.: Derivative-free multiobjective trust region descent method using radial basis function surrogate models. Math. Comput. Appl. **26**(2), 31 (2021)
2. Blank, J., Deb, K.: Pymoo: multi-objective optimization in Python. IEEE Access **8**, 89497–89509 (2020)
3. Botache, D., et al.: Enhancing multi-objective optimisation through machine learning-supported multiphysics simulation. In: Bifet, A., Krilavičius, T., Miliou, I., Nowaczyk, S. (eds.) Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track, vol. 14950, pp. 297–312. Springer, Cham (2024)
4. Chugh, T., Sindhya, K., Hakanen, J., Miettinen, K.: Handling computationally expensive multiobjective optimization problems with evolutionary algorithms-a survey. Rep. Dep. Math. Inf. Technol. Ser. B Sci. Comput. no. B **4**, 1957 (2015)
5. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems. Springer, 2 edn. (2007)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
7. Fliege, J., Svaiter, B.F.: Steepest descent methods for multicriteria optimization. Math. Methods Oper. Res. **51**(3), 479–494 (2000)
8. Heißing, B., Ersoy, M.: Chassis Handbook: Fundamentals, Driving Dynamics, Components, Mechatronics, Perspectives. SpringerLink Bücher, Vieweg+Teubner, Wiesbaden, 1 edn. (2011)
9. Hillermeier, C.: Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach. Birkhäuser, 1 edn. (2001)
10. Hussein, R., Deb, K.: A generative Kriging surrogate model for constrained and unconstrained multi-objective optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference 2016, GECCO 2016, pp. 573—580. ACM, New York, NY, USA (2016)
11. Jain, M., et al.: Multi-objective GFlowNets. In: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (eds.) Proceedings of the 40th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 202, pp. 14631–14653. PMLR, 23–29 July 2023

12. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 **9**, 15 (2017)
13. Liu, F., et al.: Large language model for multi-objective evolutionary optimization. arXiv:2310.12541 **3**, 13 (2024)
14. Miettinen, K.: Nonlinear Multiobjective Optimization, 1st edn. Springer, New York, NY (1998)
15. Mitschke, M., Wallentowitz, H.: Kraftfahrzeug, unebenheits-einpunktanregung. In: Dynamik Der Kraftfahrzeuge, pp. 369–441. Springer Fachmedien Wiesbaden (2014)
16. Peitz, S., Hotegni, S.S.: Multi-objective deep learning: Taxonomy and survey of the state of the art. arXiv:2412.01566 **2**, 18 (2024)
17. Schiehlen, W.: Multibody system dynamics: Roots and perspectives. Multibody Sys.Dyn. **1**(2), 149–188 (1997)
18. Schiehlen, W.: Research trends in multibody system dynamics. Multibody Sys.Dyn. **18**(1), 3–13 (2007)
19. Schütte, J., Sextro, W.: Tire wear reduction based on an extended multibody rear axle model. Vehicles **3**(2), 233–256 (2021)
20. Schütze, O., Esquivel, X., Lara, A., Coello, C.A.C.: Using the averaged hausdorff distance as a performance measure in evolutionary multiobjective optimization. IEEE Trans. Evol. Comput. **16**(4), 504–522 (2012)
21. Sendov, B.: Hausdorff Approximations, 1st edn. Springer, Netherlands (1990)
22. Sener, O., Koltun, V.: Multi-task learning as multi-objective optimization. Adv. Neural. Inf. Process. Syst. **31**, 12 (2018)
23. Thombre, M.N., Preisig, H.A., Addis, M.B.: Developing Surrogate Models via Computer Based Experiments, pp. 641–646. Elsevier (2015)
24. Zhang, T., Li, F., Zhao, X., Qi, W., Liu, T.: A convolutional neural network-based surrogate model for multi-objective optimization evolutionary algorithm based on decomposition. Swarm Evol. Comput. **72**, 101081 (2022)
25. Zhou, A., Qu, B.Y., Li, H., Zhao, S.Z., Suganthan, P.N., Zhang, Q.: Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm Evol. Comput. **1**(1), 32–49 (2011)
26. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evol. Comput. **8**(2), 173–195 (2000)



KANLoc: WiFi Localization with A Lightweight KAN

Yunlong Gu[1], Meng Xu[2], Mengshan Li[1(✉)], Jiawei Li[3], Jiguang Li[4], Lixin Guan[1], and Zhao Huang[4(✉)]

[1] Gannan Normal University, Ganzhou, China
msli@gnnu.edu.cn
[2] University of International Business and Economics, Beijing, China
[3] King's College London, London, UK
[4] Northumbria University, Newcastle, UK
zhao.huang@northumbria.ac.uk

Abstract. With the development and widespread adoption of WiFi technology, WiFi-based localization has become one of the most popular indoor positioning solutions. Existing WiFi fingerprinting methods require the creation of a fingerprint database in advance, while Multi-Layer Perception (MLP)-based WiFi localization offers low computational complexity but suffers from limited accuracy. Recently, researchers have introduced Kolmogorov-Arnold Networks (KAN) to improve localization performance. However, KAN models typically have a large number of parameters, leading to increased computational complexity. To address this issue, this paper proposes a lightweight KAN model for WiFi indoor localization (KANLoc), which consists of only two KANLinear layers. KANLoc retains the core idea of KAN while reducing parameter complexity by converting parameter calculations into simple matrix multiplications through the use of basis functions, thereby enhancing computational efficiency. Extensive experiments on the UJIIndoorLoc and Tampere datasets demonstrate that the proposed lightweight KAN model outperforms both MLP and standard KAN models.

Keywords: Wifi · Indoor Localization · MLP · KAN

1 Introduction and Related Work

Indoor localization technology has attracted increasing attention in recent years due to its crucial role in smart homes, the Internet of Things, and robotic navigation, making it a hot topic of research. Among the various localization techniques, WiFi fingerprinting has become a focus of research due to its simplicity of deployment, low cost, and the advantage of not requiring additional hardware [17]. WiFi fingerprinting technology utilizes received signal strength (RSSI) as an environmental feature to construct a fingerprint database that estimates indoor locations [11]. However, traditional indoor positioning methods, such as signal fingerprint matching and geometric measurements, face challenges in accuracy and robustness, especially in multi-floor and multi-building environments [8,18,19]. WiFi

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 263–275, 2026.
https://doi.org/10.1007/978-3-032-04555-3_22

fingerprinting often performs well in single-floor settings but struggles in complex scenarios due to signal propagation effects such as multipath, attenuation, and interference [6,7].

Traditional machine learning models have been widely used in indoor positioning. Early studies, such as the RADAR system by Bahl [1], employed the KNN algorithm for location estimation and achieved good results. Similarly, algorithms like the Online Sequential Extreme Learning Machine (OS-ELM) [20] improved positioning efficiency by leveraging sequential learning techniques. However, these traditional models are limited in handling diverse and complex environments required for modern indoor positioning. Compared to machine learning algorithms, deep learning algorithms can significantly enhance performance in indoor positioning tasks by extracting data features through deep neural networks. For example, DNNs and Deep Belief Networks (DBNs) have demonstrated better generalization capabilities and the ability to capture complex nonlinear relationships in data. CNNLoc, introduced by Song et al. [16], employed Convolutional Neural Networks (CNNs) for fingerprint feature extraction and positioning, achieving good performance. Hao et al. [5] improved cross-layer positioning accuracy by integrating multi-sensor and WiFi fingerprint data using techniques such as Multidimensional Scaling (MDS) and the Unscented Kalman Filter (UKF). However, challenges such as data dependency, training complexity, and environmental sensitivity limit their practical applications. To address the above issues, Kolmogorov-Arnold Networks (KANs) have emerged as a novel alternative to improve localization accuracy and reduce computation complexity. KANs use learnable splines rather than fixed linear weight matrices , to enhance the model's expressiveness and interpretability [12]. Compared to Multi-Layer Perception (MLP), KANs have shown promise in reducing the number of parameters while maintaining performance. Bodner et al. [2] proposed combining CNNs with KANs to introduce spline-based activation functions into convolutional layers which reduced nearly half the parameters without sacrificing accuracy. Despite a significant reduction in parameters, KANs' complexity and computational costs remain high, requiring further optimization for applications in large-scale, multi-building, and multi-floor environments.

In this paper, we propose an improved lightweight KAN network for indoor localization, which not only improves the accuracy of building and floor classification but also significantly reduces the number of parameters and computational complexity. Different from traditional KAN, this lightweight KAN consists of only two simple KANLinear layers and introduces a normalization function to normalize weights. Specifically, we restructured the basis function activation to avoid large memory overhead caused by tensor expansion, where the activation of the basic function is split into two independent matrix multiplications, additionally, the SiLU and B-spline functions are separately applied to activate the input, then linearly combined the activation results to significantly improve computational efficiency. Additionally, L1 regularization was applied to the weights to ensure compatibility with the new computation framework.

2 Method

2.1 Framework of the Proposed KANLoc

The proposed KANLoc system architecture as shown in Fig. 1, which includes multi-buildings and multi-floors classification and a localization block. Firstly, the data is reprocessed and input into the lightweight-KAN network. Then, the proposed two KANLinear layers are used to perform classification and positioning tasks, and finally, the human position is refined from coarse to fine by combining building, floor, and initial position information.

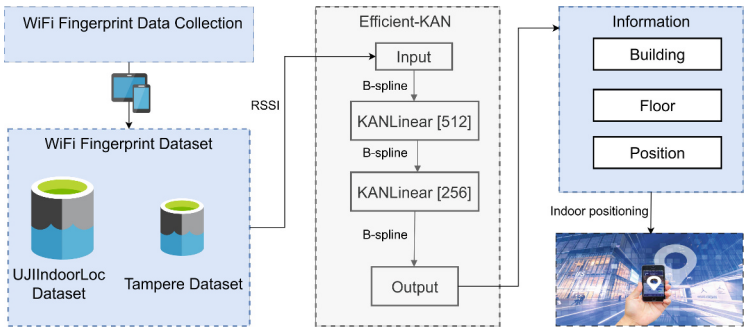


Fig. 1. Framework of the proposed KANLoc.

2.2 Kolmogorov-Arnold Representation Theorem

The Kolmogorov-Arnold Representation Theorem is a major achievement in approximation theory. It states that any multivariate continuous function defined on a bounded domain can be represented as a superposition of univariate continuous functions. In essence, this allows complex tasks to be decomposed into a series of simpler tasks, which are then aggregated to solve the original problem. This decomposition reduces the overall complexity and enhances computational efficiency. Given a multivariate continuous function f, the theorem expresses f in the following form: $f(x_1, \ldots, x_n) = \sum_{a=1}^{2n+1} \varphi_a \left(\sum_{b=1}^{n} \phi_{a,b}(x_b) \right)$. In this equation, φ_a is called the outer function, and $\phi_{a,b}$ represents the univariate inner function, which maps the input. Theoretically, this representation enables the approximation of any complex function. However, in certain applications, the smoothness of $\phi_{a,b}$ may be insufficient. To address this, in KAN networks, each $\phi_{a,b}$ is defined as a B-spline, a type of piecewise polynomial function.

The parameter `grid_size`, denoted by G, defines the number of nodes uniformly distributed within a given input range (specified by the grid_range parameter). This grid size determines the mesh density; a larger G results in a more detailed mesh, allowing the model to learn more features, but at the cost of increased computational and parameter complexity. The parameter `spline_order`, denoted by K, represents the order of the B-spline function. A higher spline order results in smoother curves, but increases computational complexity. The node vector, denoted as N, plays a key role in shaping and positioning the B-spline curve, and is defined as: $N = G + K$. This equation shows

that the number of nodes depends on both the grid size and the spline order. As G and K increase, the complexity and smoothness of the spline increase, enabling more precise function approximation.

2.3 Efficient Kolmogorov-Arnold Representation Theorem

The Efficient-KAN network consists of multiple KANLinear layers, each utilizing a combination of B-spline and SiLU (Sigmoid Linear Unit) based activation functions. The key optimization of the Efficient-KAN network is transforming the activation function computation from element-wise operations to matrix multiplication, reducing memory consumption and increasing computational speed.

KANLinear: The KANLinear layer is a fundamental building block of the Efficient-KAN network. Its operations are divided into three main parts: basis function transformation, activation function application, and linear transformation.

Basis Function Transformation: Given an input vector $\mathbf{x} \in \mathbb{R}^{\text{in_features}}$, the input is first transformed through B-spline basis functions. Each component of $\mathbf{x}$ passes through K B-spline functions, yielding the following output: $B(\mathbf{x}) = [B_1(x_1), B_2(x_1), \ldots, B_K(x_1), \ldots, B_1(x_{\text{in_features}}), \ldots, B_K(x_{\text{in_features}})]$. This results in an expanded vector $\mathbf{B}(\mathbf{x})$ with dimensions $\mathbb{R}^{\text{in_features} \times K}$, where each feature of the input vector has been transformed by K B-spline basis functions.

Activation of Functions and Linear Transformation: After the input vector $\mathbf{x}$ is passed through the SiLU activation function, it undergoes a linear transformation: $\phi_{\text{SiLU}}(\mathbf{x}) = \text{SiLU}(\mathbf{x}) \cdot \mathbf{W_b}$, where $\text{SiLU}(\mathbf{x}) \in \mathbb{R}^{\text{in_features}}$ is the output of the SiLU activation function, and $\mathbf{W_b} \in \mathbb{R}^{\text{in_features} \times \text{out_features}}$ is the weight matrix for the linear transformation. Simultaneously, the B-spline transformed output $\mathbf{B}(\mathbf{x})$ is linearly transformed using the weight matrix $\mathbf{W_s}$:$\phi_{\text{spline}}(\mathbf{x}) = \mathbf{B}(\mathbf{x}) \cdot \mathbf{W_s}$, where $\mathbf{W_s} \in \mathbb{R}^{(\text{in_features} \times K) \times \text{out_features}}$ is the weight matrix for the B-spline transformation. Finally, the combined output of the SiLU activation and the B-spline transformation is given by: $\mathbf{y} = \text{SiLU}(\mathbf{x}) \cdot \mathbf{W_b} + \mathbf{B}(\mathbf{x}) \cdot \mathbf{W_s} + \mathbf{b}$, where $\mathbf{b} \in \mathbb{R}^{\text{out_features}}$ is the bias term.

Reconstruction of the Activation Function: In the original KAN network, the activation function is a linear combination of the SiLU activation and B-spline basis functions. The overall activation function can be expressed as: $\phi(\mathbf{x}) = \mathbf{W_b} \cdot \text{SiLU}(\mathbf{x}) + \mathbf{W_s} \cdot \text{spline}(\mathbf{x})$, where the SiLU function is defined as: $\text{SiLU}(\mathbf{x}) = \mathbf{x} \cdot \sigma(\mathbf{x})$, and the B-spline function is defined as: $\text{spline}(\mathbf{x}) = \sum_{k=1}^{K} c_k \cdot B_k(\mathbf{x})$. Here, $B_k(\mathbf{x})$ represents the k-th B-spline basis function, and c_k is the corresponding coefficient. The weight matrices $\mathbf{W_b}$ and $\mathbf{W_s}$ are applied to the SiLU and B-spline activations, respectively. In the original implementation, to apply the activation function, the input required expanding into a tensor of shape $(\text{batch_size}, \text{out_features}, \text{in_features})$, leading to significant memory overhead and computational complexity. To address this issue, the Efficient-KAN network separates the computations, applying the SiLU and B-spline functions independently before combining their results. This approach allows for more efficient matrix multiplication.

Regularization Strategy: In the original KAN network, regularization is typically achieved using L1 regularization on input samples to promote sparsity and interpretability. However, in Efficient-KAN, matrix reconstruction makes this approach less effective. Instead, Efficient-KAN applies L1 regularization on the weight matrices: $L_{\text{reg}} = \lambda |W|_1$, where λ is the regularization coefficient, and $|W|_1$ is the sum of the absolute values in the weight matrix. The regularized loss function for the KANLinear layer is defined as: $L_K = \lambda_{\text{act}} \cdot \sum_{j=1}^{\text{out_f}} \sum_{i=1}^{\text{in_f}} |W_s[j,i,:]|_1 + \lambda_{\text{ent}} \cdot \mathcal{H}(p)$, where $\mathcal{H}(p)$ is an entropy term used to promote weight sparsity:$\mathcal{H}(p) = -\sum_{j,i} p_{j,i} \cdot \log(p_{j,i})$. By applying L1 regularization to the weights, Efficient-KAN maintains sparsity and interpretability while ensuring computational efficiency.

3 Experiment Results Analysis

3.1 Experimental Setting and Parameters

To validate the model's performance, the MLP [10,15] and KAN [12] are used as baselines to compare with the proposed EKAN on the UJIIndoorLoc Dataset [14] and the Tampere Dataset [13]. For MLP, we set the hidden layer structure to two layers use the ReLU activation function, and adopt AdamW as the optimizer. The learning rate is adjusted using the ReduceLROnPlateau scheduler (reducing the learning rate by a factor of 0.95 if the validation loss stagnates for five consecutive epochs). For KAN, we set the hidden layer structure to 2 layers, choose a combination of B-spline and SiLU functions as activation functions, and use the same optimizer, learning rate scheduler, and loss function as MLP. The detailed parameters are summarized in Table 1.

Table 1. Experimental Parameters for UJIIndoorLoc and Tampere Datasets

Parameter	UJIIndoorLoc	Tampere
Hidden layers	[512, 256]	[128, 64]
Input features	520	992
Output features	118	5
Batch size	10	10
Learning rate	0.01	0.06
Epochs	200	300
Optimizer	AdamW(0.001)	AdamW(0.00005)
Regulation (L1/Entropy)	0.5	–
Grid size G	5	2
Spline function order K	3	2

Parameters and FLOPS Calculation: The learnable parameters in the KANLinear layer include B-spline control points, shortcut weights, B-spline weights, and bias terms. Efficient-KAN introduces an additional learnable parameter called spline scaler (`scale_spline`). Given the spline grid size G, and the spline order K, the total number of learnable parameters is: P1 = input_features $\times$ output_features $\times (3 + G + K)$. For a simple MLP, the parameter count is: P2 = input_features $\times$ output_features + output_features. The

FLOPS for Efficient-KAN include SiLU activation, B-spline basis function calculation, and matrix multiplication. The FLOPS for computing B-spline functions is: $6 \times K \times \text{input_features} \times (G + K)$ FLOPS, and the total FLOPS for linear transformations is: $2 \times \text{input_features} \times (G+K) \times \text{output_features}$ FLOPS. The FLOPS for a single KANLinear layer can be expressed as: $\text{input_features} + (5 \times (G+K)) \times \text{input_features} \times \text{output_features} + 6 \times K \times \text{input_features} \times (G+K)$. For a network with L layers, the total FLOPS is calculated as:$\sum_{i=1}^{L} K_{\text{FLOPS}}(\text{input_features}[i], \text{output_features}[i])$.

3.2 Result Analysis

Experiments on UJIIndoorLoc Dataset: Due to the large size of the UJIIndoorLoc dataset, training the KAN network requires a significant amount of time and is inefficient on the GPU [12], making it challenging to perform experiments on large-scale datasets with many hidden layers. We conducted preliminary experiments using smaller hidden layers, setting the KANLinear layer to two layers of [32, 16], but found that the performance was unsatisfactory despite high GPU memory usage and long training times. Therefore, we focused on comparing and evaluating the performance of Efficient-KAN and MLP. Figure 2 below compares the experimental results. As shown, Efficient-KAN significantly outperforms MLP in all aspects. Specifically, Efficient-KAN achieves a building recognition accuracy of 99.91%, a floor recognition rate of over 90%, and a location error of 9.40 m. In contrast, MLP appears unsuitable for handling complex and diverse data, performing poorly in the experiment. MLP's building and floor recognition rates are both below 50%, and its location error is as high as 46.75 m. This demonstrates that Efficient-KAN is more suitable than MLP for complex and large-scale indoor localization tasks.

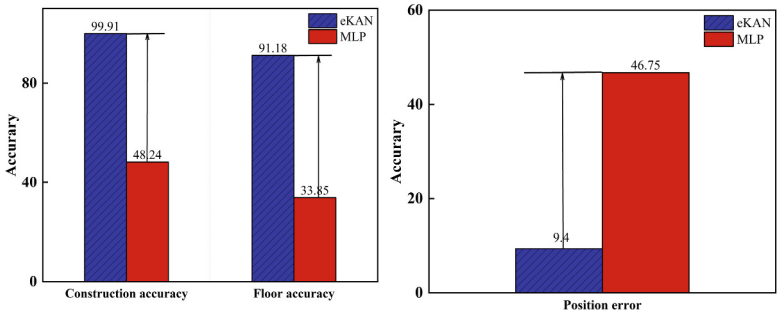


Fig. 2. Comparison of the results of Efficient-KAN and MLP in terms of building and floor recognition accuracy and location error.

Additionally, we compare our results with existing research [9], as shown in Table 2 below. The results indicate that Efficient-KAN achieves better positioning performance, with a location error 0.36 m lower than that of the DNNS method proposed in the paper. Furthermore, Compared with HADNN [3] and

Table 2. Performance comparison of different methods

Method	Efficient-KAN	DNNS	HADNN	CNNLoc
Building recognition accuracy	99.91%	99.82%	100.00%	100.00%
Floor recognition accuracy	91.18%	91.27%	93.15%	95.00%
Position error	9.40 m	9.76 m	14.93 m	11.78 m

CANNLoc [16], Efficient-KAN exhibits a lower location error by 5.53 m and 2.38 m, respectively.

Moreover, we compare the recognition performance of Efficient-KAN and MLP under similar parameter sizes and FLOPS. To achieve this, we adjust the hidden layer sizes of Efficient-KAN to [32, 16] and [64, 32], and those of MLP to [512, 256] and [256, 256], respectively. The results of the comparison are presented in Fig. 3. It is evident that variations in the number of parameters and FLOPS have minimal impact on Efficient-KAN's performance, which remains significantly superior to that of MLP. Efficient-KAN's building and floor recognition accuracy is approximately 70% higher. In contrast, MLP's performance is notably affected by changes in its parameter count and FLOPS.

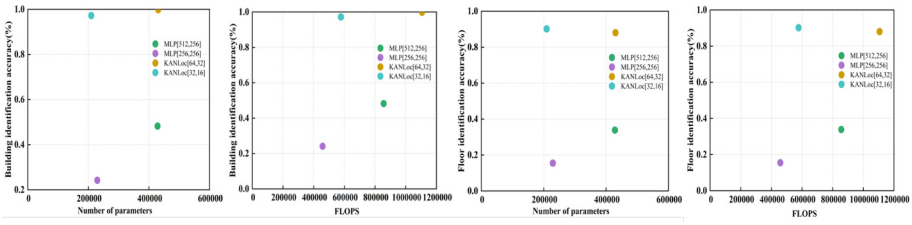


Fig. 3. Comparison of recognition performance between Efficient-KAN and MLP with similar numbers of parameters and FLOPS.

Experiments on Tampere Dataset: Given that the Tampere Dataset is significantly smaller compared to the UJIIndoorLoc Dataset, we initially explored the model's performance with a reduced number of hidden layer nodes. We configured the KANLinear layer of Efficient-KAN as a two-layer structure [128, 64], with the grid size G set to 2 and the spline function order K set to 2. The AdamW optimizer was used, along with the ReduceLROnPlateau learning rate scheduler, which adjusts the learning rate by multiplying it by a decay factor of 0.3 if the validation loss does not decrease significantly over five consecutive epochs. Additionally, a noise parameter of `scale_noise=0.1` was applied to initialize the spline function weights. The regularization parameters `regularize_activation` and `regularize_entropy` were both set to 0.1. As shown in Fig. 4, Efficient-KAN achieves optimal results in both recognition and localization performance. Specifically, Efficient-KAN attains a floor recognition accuracy of 91.65%, while KAN and MLP achieve 86.51% and 81.93%, respectively. In terms of localization, Efficient-KAN achieves the lowest localization error of 8.9 m, compared to 12.12 m for KAN and 10.42 m for MLP. This demonstrates that Efficient-KAN

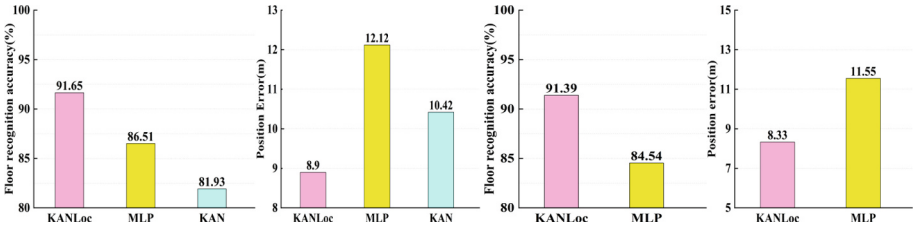


Fig. 4. Comparison of the results for Efficient-KAN, KAN, and MLP on floor recognition accuracy and location error, the left two figures show results for the hidden layer configuration [128, 64], while the right two figures show results for the hidden layer configuration [512, 256].

performs excellently in indoor localization tasks, even with smaller datasets. Furthermore, we have also expanded the model's hidden layer to [512, 256], the results show that Efficient-KAN achieves the best performance in both classification and localization tasks. Specifically, the floor recognition accuracy is 91.39%, which is 7.19% higher than that of MLP, and the location error is 8.33 m, which is 3.21 m lower than that of MLP.

Additionally, we swapped the original training data and test data to train the model and test classification accuracy. The results are shown in Fig. 5, Efficient-KAN still achieved the highest floor recognition accuracy of 93.97% while achieving 7.08 m localization error, reducing up to 1.27 m and 1.44 m compared with KAN and MLP, respectively, which indicates the strong robustness.

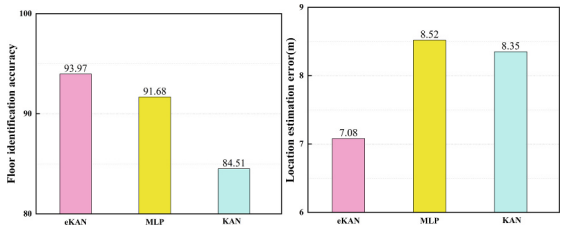


Fig. 5. Performance comparison of Efficient-KAN with other models in terms of floor recognition accuracy and positioning error after swapping training and test data.

The proposed network is also comparable to classical methods reported in the literature [3,16]. As shown in Table 3, although Efficient-KAN's floor recognition accuracy is slightly lower than that of CNNLoc [16] and HADNN [4], it achieves better localization performance, with a position error approximately 2.55 m smaller than that of CNNLoc and 0.72 m smaller than that of HADNN.

Model Inference Time Analysis: We tested the inference times across various hidden layers. As shown in Table 4, our model achieves better performance (per-sample 0.0128 ms and 0.0010 ms), rendering it highly suitable for scenarios demanding high real-time performance. Furthermore, the model sustains fast

Table 3. Performance comparison of different methods on the Tampere Dataset

Method	Efficient-KAN	CNNLoc	HADNN
Floor recognition accuracy	91.39%	94.22%	94.58%
Position error	8.33 m	10.88 m	9.05 m

Table 4. Model Inference Time and Performance Comparison

Dataset	Hidden layers	Bldg. acc (%)	Floor acc (%)	Pos. err (m)	Total time (s)	Avg. time (ms)
UJIIndoorLoc	256	100.00	88.48	14.10	0.0074	0.0067
	512	100.00	88.03	14.50	0.0086	0.0078
	256-128	99.91	91.81	11.01	0.0108	0.0097
	512-256	99.91	91.18	9.40	0.0142	0.0128
	256-128-64	100.00	91.36	10.85	0.0124	0.0111
	512-256-64	99.55	91.27	10.03	0.0168	0.0152
Tampere	256	–	91.57	12.71	0.0023	0.0006
	512	–	91.19	12.54	0.0027	0.0007
	256-128	–	91.75	8.59	0.0032	0.0008
	512-256	–	91.39	8.33	0.0038	0.0010
	256-128-64	–	90.33	8.34	0.0041	0.0010
	512-256-64	–	89.93	8.22	0.0046	0.0012

inference times and robust performance even under relatively simple architectures, making it a viable option for computationally constrained devices.

Confusion Matrix Analysis: KANLoc exhibited state-of-the-art classification accuracy across diverse experimental datasets. As illustrated in Fig. 6, the confusion matrices demonstrated exemplary building recognition accuracy with minor instances of floor-level and spatial coordinate misclassifications. Notably, there is confusion between adjacent floors (e.g. Floors 4 and 3 in Tampere, and Floors 0 and 4 in UJIndoorLoc) suggests the difficulty of achieving fine-grained floor separation using RSS features.

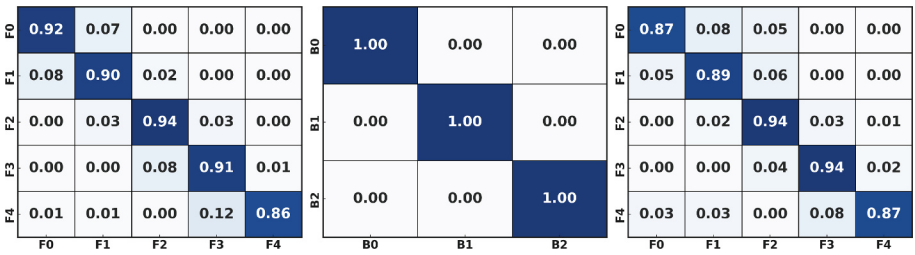


Fig. 6. Confusion Matrix on UJIIndoor and Tampere Datasets, the left one is floor classification on the Tampere dataset, and the rest are building and floor classification on the UJIIndoor dataset.

Table 5. Hyperparameter(Grid size and Spline order) Sensitivity Analysis

K	G	UJIIndoorLoc				Tampere			
		Bldg Acc (%)	Floor Acc (%)	Pos Err (m)	Train Time (s)	Floor Acc (%)	Pos Err (m)	Train Time (s)	**Δ** Time [a]
1	5	99.91	91.99	10.15	23.54	89.02	8.89	7.04	16.50
	4	99.91	92.97	10.29	20.58	90.58	8.65	5.58	15.00
	3	99.82	92.07	11.13	17.27	90.99	8.40	5.25	12.02
	2	99.37	92.35	9.91	14.38	91.09	8.34	3.89	10.49
2	5	99.91	90.91	9.86	46.33	90.23	8.45	10.26	36.07
	4	99.91	92.34	10.20	40.65	90.76	8.40	9.10	31.55
	3	100.00	92.89	10.17	35.45	91.14	8.38	8.30	27.15
	2	99.64	91.54	9.92	30.45	91.39	8.33	7.76	22.69
3	5	99.91	91.18	9.40	78.50	91.01	8.43	10.65	67.85
	4	100.00	91.70	9.61	71.37	90.86	8.47	9.52	61.85
	3	100.00	92.07	9.89	60.99	91.17	8.35	8.44	52.55
	2	99.82	91.54	9.72	53.10	91.09	8.36	6.68	46.42

[a] Δ Time = UJIIndoorLoc training time - Tampere training time

Hyperparameter Sensitivity Experiment: This section evaluates three critical hyperparameters: grid size (G), spline order (K), and regularization coefficient (λ). Systematic experiments examine configurations spanning $K \in \{1, 2, 3\}$, $G \in \{2, 3, 4, 5\}$, and $\lambda \in \{0, 0.1, 0.3, 0.5, 0.8, 1.0\}$, quantifying building/floor recognition accuracy, positioning error, and computational efficiency.

As demonstrated in Table 5, building recognition accuracy (UJIIndoorLoc) exhibits exceptional robustness ($> 99.37\%$ across configurations). Floor recognition peaks at G = 4 (92.97%) on UJIIndoorLoc and G=2 (91.39%) on Tampere. Position error minimization occurs at K = 3/G = 5 (9.40 m) for UJIIndoorLoc, while Tampere achieves optimal precision (8.33 m) at K = 2/G = 2. Computational cost escalates nonlinearly with K and G, with K dominating UJIIndoorLoc latency (78.50 s at K = 3/G = 5 vs 23.54 s at K = 1/G = 5) and G driving Tampere costs (10.26 s at G = 5 vs 3.89 s at G = 2 for K = 2).

Regularization effects diverge substantially across datasets (Table 6). On UJIIndoorLoc, *lambda* = 0.5 minimizes position error (9.40 m) while maintaining floor recognition (91.18%). Conversely, Tampere exhibits severe degradation at *lambda* = 0.8: floor accuracy declines 8.75% (91.39%→82.64%) with error inflation of 7.07 m (8.33 m→15.40 m). This bifurcation originates from dataset-specific fitting characteristics–Tampere's near-optimal baseline (error=8.33 m) indicates underfitting susceptibility, exacerbated by regularization constraints.

Ablation experiment: We performed systematic ablation studies to evaluate the contribution of B-spline functions to model performance. As demonstrated in Table 7, the B-spline implementation provides marginal yet consistent improvements in classification accuracy across both datasets. More substantially, it reduces positioning errors by 0.66 m and 2.28 m on the UJIIndoorLoc and Tampere datasets respectively.

Table 6. Transposed Regularization Coefficient (λ) Sensitivity Analysis

Dataset	Metric			λ			
		None	**0.1**	**0.3**	**0.5**	**0.8**	**1.0**
UJIIndoorLoc	Bldg Acc (%)	100.00	99.91	100	99.91	100.00	100.00
	Floor Acc (%)	91.68	90.82	90.82	91.18	91.09	91.00
	Pos Err (m)	9.95	9.76	9.56	9.40	9.66	9.60
Tampere	Floor Acc (%)	91.39	84.31	83.17	82.71	82.64	82.91
	Pos Err (m)	8.33	12.82	14.97	15.30	15.40	15.41

Table 7. B-spline Ablation Study on UJIIndoorLoc and Tampere Datasets

Dataset	Model	Bldg. acc (%)	Floor acc (%)	Pos. err (m)	Error Reduction
UJIIndoorLoc	KANLoc	99.91	91.18	9.40	0.66 m
	w/o B-spline	99.82	91.09	10.06	
Tampere	KANLoc	–	91.39	8.33	2.28 m
	w/o B-spline	–	88.59	10.61	

4 Conclusion and Future Work

In this paper, we addressed the challenges of low localization accuracy and computational inefficiency in Wi-Fi fingerprint-based indoor localization by proposing the Efficient-KAN network, which consists of two KANLinear layers. Unlike traditional MLP networks, Efficient-KAN incorporates a spline function as a learnable activation within the weight calculation, a concept inherited from the KAN network. Additionally, we optimized the computation of basis functions by converting it into an efficient matrix multiplication process, significantly improving computational efficiency. These enhancements enable Efficient-KAN to outperform conventional methods in both accuracy and efficiency, as demonstrated in our experiments on large and small datasets. Efficient-KAN achieves higher building and floor recognition accuracy while maintaining a lower localization error, making it a promising solution for real-time indoor localization tasks that demand high performance in terms of both accuracy and computational speed. In future work, we will deploy the AP in situ and gather empirical data to evaluate the model's performance under highly dynamic and noisy conditions. Furthermore, by maintaining input-signal continuity, we will undertake a formal convergence analysis and perform statistically rigorous significance tests to substantiate the superiority of our proposed enhancement.

Symbols and Notation

This appendix provides definitions of all mathematical symbols used in the paper.

Fundamental Symbols

Symbol	Description	Definition
f	Multivariate function	$f : \mathbb{R}^n \to \mathbb{R}$
x_i	Input variable	$i \in \{1, 2, \ldots, n\}$
φ_a	Outer function	$\varphi_a : \mathbb{R} \to \mathbb{R}$
$\phi_{a,b}$	Inner function	$\phi_{a,b} : \mathbb{R} \to \mathbb{R}$
G	Grid size	Node count
K	Spline order	Degree
N	Node vector	$N = G + K$
λ	Regularization	$\lambda \in [0, 1]$

Efficient-KAN Components

Symbol	Description	Dimension
$\mathbf{x}$	Input vector	$\mathbb{R}^{d_{in}}$
$\mathbf{W_b}$	SiLU weights	$\mathbb{R}^{d_{in} \times d_{out}}$
$\mathbf{W_s}$	Spline weights	$\mathbb{R}^{(d_{in}K) \times d_{out}}$
$\mathbf{B}(\mathbf{x})$	B-spline output	$\mathbb{R}^{d_{in} \times K}$
$\mathbf{b}$	Bias term	$\mathbb{R}^{d_{out}}$
$\mathbf{y}$	Output vector	$\mathbb{R}^{d_{out}}$

References

1. Bahl, P., Padmanabhan, V.N.: Radar: an in-building rf-based user location and tracking system. In: Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064), vol. 2, pp. 775–784. IEEE (2000)
2. Bodner, A.D., Tepsich, A.S., Spolski, J.N., Pourteau, S.: Convolutional kolmogorov-arnold networks. arXiv preprint arXiv:2406.13155 (2024)
3. Cha, J., Lim, E.: A hierarchical auxiliary deep neural network architecture for large-scale indoor localization based on wi-fi fingerprinting. Appl. Soft Comput. **120**, 108624 (2022)
4. Chen, L., Li, B., Zhao, K., Rizos, C., Zheng, Z.: An improved algorithm to generate a wi-fi fingerprint database for indoor positioning. Sensors **13**(8), 11085–11096 (2013)
5. Hao, Z., Dang, J., Cai, W., Duan, Y.: A multi-floor location method based on multi-sensor and wifi fingerprint fusion. IEEE Access **8**, 223765–223781 (2020)
6. Huang, Z., et al.: Deepwe: a deep bayesian active learning waypoint estimator for indoor walkers. IEEE Internet Things J. **10**(11), 9738–9752 (2023)
7. Huang, Z., Valkama, M., Zhang, J., Xu, M., Yin, C., Guan, M.: Wiloc: encoding-based wifi indoor localization. In: 2024 14th International Conference on Indoor Positioning and Indoor Navigation (IPIN), pp. 1–6. IEEE (2024)
8. Huang, Z., Zeng, Y., Poslad, S., Gu, F.: Spikepr: position regression with deep spiking neural network. IEEE Sens. J. (2024)
9. Kim, K.S., Lee, S., Huang, K.: A scalable deep neural network architecture for multi-building and multi-floor indoor localization based on wi-fi fingerprinting. Big Data Anal. **3**, 1–17 (2018)
10. Kunhoth, J., Karkar, A.G., Al-Maadeed, S., Al-Ali, A.: Indoor positioning and wayfinding systems: a survey. HCIS **10**(1), 1–41 (2020). https://doi.org/10.1186/s13673-020-00222-0
11. Lee, S., Kim, J., Moon, N.: Random forest and wifi fingerprint-based indoor location recognition system using smart watch. HCIS **9**, 1–14 (2019)
12. Liu, Z., et al.: Kan: kolmogorov-arnold networks. arXiv preprint arXiv:2404.19756 (2024)
13. Lohan, E.S., Torres-Sospedra, J., Leppäkoski, H., Richter, P., Peng, Z., Huerta, J.: Wi-fi crowdsourced fingerprinting dataset for indoor positioning. Data **2**(4), 32 (2017)
14. Nowicki, M., Wietrzykowski, J.: Low-effort place recognition with wifi fingerprints using deep learning. In: Automation 2017: Innovations in Automation, Robotics and Measurement Techniques 1, pp. 575–584. Springer (2017)

15. Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. Psychol. Rev. **65**(6), 386 (1958)
16. Song, X., et al.: A novel convolutional neural network based indoor localization framework with wifi fingerprinting. IEEE Access **7**, 110698–110709 (2019)
17. Xie, Y., Wang, Y., Nallanathan, A., Wang, L.: An improved k-nearest-neighbor indoor localization method based on spearman distance. IEEE Signal Process. Lett. **23**(3), 351–355 (2016)
18. Xu, M., Shu, Q., Huang, Z., Chen, G., Poslad, S.: Arlo: augmented reality localization optimization for real-time pose estimation and human-computer interaction. Electronics **14**(7), 1478 (2025)
19. Xu, M., et al.: A critical analysis of image-based camera pose estimation techniques. Neurocomputing **570**, 127125 (2024)
20. Zou, H., Lu, X., Jiang, H., Xie, L.: A fast and precise indoor localization algorithm based on an online sequential extreme learning machine. Sensors **15**(1), 1804–1824 (2015)



DualGF: Example-Driven Path Planning via Dual Gradient Fields

Mingdong Wu[1], Fangwei Zhong[2,3], Yulong Xia[1], Yizhou Wang[1], and Hao Dong[1(✉)]

[1] PKU-Hyperplane Lab, Center on Frontiers of Computing Studies, School of Computer Science, Peking University, Beijing, China
{hao.dong,hao.dong}@pku.edu.cn
[2] School of Artificial Intelligence, Beijing Normal University, Beijing, China
[3] State Key Laboratory of General Artificial Intelligence, BIGAI, Beijing, China

Abstract. Path planning is a crucial function for intelligent agents, with applications ranging from navigation to object tracking and rearrangement. Existing methods primarily focus on tasks with explicit goal and constraint specifications, limiting their applicability in scenarios where such specifications are difficult to define. This work introduces a novel data-driven paradigm called *example-driven planning*, which allows users to specify goals and constraints through sets of target and support examples, respectively. Formulated as a constraint optimisation problem, we propose a learning-based framework, namely *DualGF*, to address this task. Specifically, DualGF models the planning process using two gradient fields: a target gradient field, which attracts the agent towards the target examples, and a support gradient field, which repels the agent away from the support examples. Both fields are parameterized by graph neural networks and are trained using a denoising score-matching objective based on the provided examples. To strike a balance between these two fields, we introduce a gradient mixer that dynamically adjusts the mixing rate during execution. Experiments across four tasks (navigation, tracking, particle rearrangement, and room rearrangement) demonstrate our method's scalability, generality, and effectiveness.

Keywords: Generative Model · Object Rearrangement Planning · Graph Neural Network

1 Introduction

Planning a feasible path toward a goal is essential for intelligent agents [24], with applications including navigation [19], object tracking [30], and object rearrangement [13]. State-of-the-art methods effectively address complex path planning problems with high-dimensional systems and kinodynamic constraints [2,9,19]. However, these approaches typically require explicit goal and constraint specifications, limiting their applicability in tasks where such definitions are ambiguous, such as tidying a workspace [28].

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 276–288, 2026.
https://doi.org/10.1007/978-3-032-04555-3_23

Inverse Reinforcement Learning (IRL)-based approaches [5,21,28] mitigate the need for explicit goal definitions. For instance, TarGF [28] learns a target gradient field from successful examples to implicitly represent goals, but still relies on manually defined constraints or explicit assumptions about objects.

To overcome these limitations, we propose *example-driven planning*, a novel data-driven paradigm enabling task specification through target and support examples. This paradigm removes the reliance on manually engineered cost functions, explicit assumptions about object shapes, or extensive online sampling. Specifically, we introduce *DualGF*, an offline-learning-based framework that addresses example-driven planning by decomposing the problem into two distinct gradient fields trained from examples. The *target gradient field* captures the implicit goal through target examples, generating an attractive force directing the agent toward success states. Conversely, the *support gradient field* learns environmental constraints from uniformly sampled safe states, producing a repulsive force to avoid collisions. Both gradient fields are represented as graph neural networks (GNN) and are trained using denoising score-matching, which enables robust generalization across varying object configurations and visual inputs. To balance goal-oriented actions with safety constraints during execution, we integrate these gradients using an adaptive mixing strategy, enhancing DualGF's flexibility and adaptability.

Our extensive experiments validate DualGF's scalability and generalization in diverse tasks, including navigation, tracking, and object rearrangement. Empirical results show significant improvements in planning efficiency and safety compared to state-of-the-art learning-based methods. Ablation studies further highlight the effectiveness of our dual-gradient approach.

In summary, our main contributions include:

- Introducing a novel paradigm of example-driven path planning that replaces explicit goal and constraint definitions with user-provided examples.
- Developing DualGF, an offline-learning framework integrating goal-directed and constraint-aware gradient fields trained through denoising score-matching.
- Demonstrating superior scalability, efficiency, and safety of DualGF across multiple planning tasks through comprehensive experiments.

2 Related Works

Learning from Demonstration. Example-driven planning is a special case of Learning from Demonstration (LfD), where policies are typically learned from expert trajectories. LfD primarily includes Behavioural Cloning (BC) [20,22,23], which employs supervised learning, and Inverse Reinforcement Learning (IRL), which [6,14,18,31] infers a cost function to explain expert behaviors. Unlike traditional LfD, which requires full demonstrations, our method trains from two examples without interactions or offline data. Existing RL-based example-driven

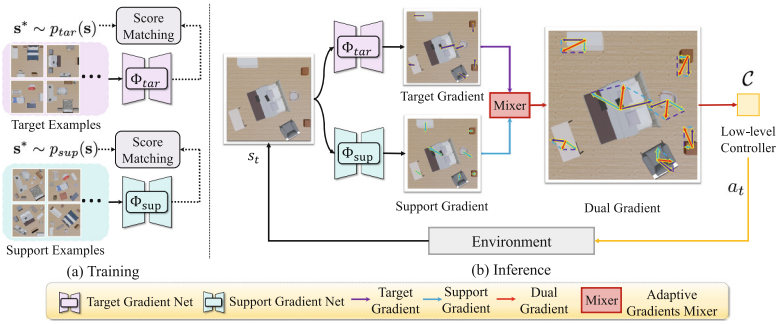


Fig. 1. Method Overview. a): A target and a support score network are trained from examples via the score-matching objective. b): DualGF computes target and support gradients at each time step. The target gradient guides task completion, such as moving objects to a tidy layout, while the support gradient directs objects to avoid collisions. The two gradients are mixed using an adaptive gradient mixer and interpreted into an action via a low-level controller.

methods like VICE [7], SQIL [21], RCE [5], and Offline-RCE [10] either require environment interaction or offline demonstrations, which our approach avoids.

Path Planning. Traditional methods include sampling-based algorithms like PRM [3,4,11,12,29] and RRT [8,15,17,26,27], while our approach is an end-to-end planner that eliminates runtime sampling, enabling efficient scaling to high-dimensional tasks. Unlike the Artificial Potential Field method [16], which uses manual fields, our method learns generalizable gradient fields for goal attraction and obstacle avoidance. TarGF [28] requires online interactions or shape assumptions, whereas ours learns offline without such limitations.

3 Problem Statement

The example-driven planning can be described by a tuple $(\mathcal{S}, \mathcal{S}_f, p_{tar}, \rho, \epsilon)$ where $\mathcal{S}$ denotes the state space (*e.g.*, configurations of objects and obstacles) while $\mathcal{S}_f \subseteq \mathcal{S}$ denotes the free space (the valid states without object collision or trans-boundary). At each time step t, our planning agent $\pi : \mathcal{S} \to \mathcal{A}$ first selects a waypoint within an epsilon-ball of the input state, *i.e.*, $\|\hat{\mathbf{s}_{t+1}} - \mathbf{s}_t\| \leq \epsilon$. Then a low-level controller (*e.g.*, PID controller) outputs actions to achieve this waypoint $\mathbf{a}_t = \mathcal{C}(\mathbf{s}_t, \hat{\mathbf{s}_{t+1}})$. During training, the agent is given a set of target examples $S^*_{tar} = \{\mathbf{s}^*, \mathbf{s}^* \sim p_{tar}(\mathbf{s})\}$ where $p_{tar} : \mathcal{S} \to \mathbb{R}^+$ is a target distribution indicating the goal specification. Our objective is to find the most efficient path to increase the likelihood of the target distribution:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\rho(\mathbf{s}_0), \tau \sim \pi} \left[\sum_{\mathbf{s}_t \in \tau} \gamma^t \log p_{tar}(\mathbf{s}_t) \right] \text{ s.t. } \forall \mathbf{s}_t \in \mathcal{S}_f \tag{1}$$

In this work, we consider the offline setting of example-driven planning, where the agent has to infer physics constraints from support examples instead of

interacting with the environment. The support examples are uniformly sampled from the free space $S^*_{sup} = \{\mathbf{s}^*, \mathbf{s}^* \sim p_{sup}(\mathbf{s})\}$ where

$$p_{sup}(\mathbf{s}) = \begin{cases} \dfrac{1}{|\mathcal{S}_f|}, & \mathbf{s} \in \mathcal{S}_f \\ 0, & \text{otherwise} \end{cases}, \ |\mathcal{S}_f| = \int_{\mathcal{S}_f} ds \tag{2}$$

The rationale for 'replacing interactions with examples' is that the support examples are enough for inferring the free space $\mathcal{S}_f$ while the online interactions are unsafe and costly. Besides, the agent should learn generalisable inference of free space instead of overfitting to a specific environment.

4 Method: DualGF

Overview: Our key idea is to learn two gradient fields from examples to tackle example-driven planning. As illustrated in Fig. 1 (a), the *target gradient field* $\mathbf{\Phi}^{\theta}_{tar}$ and the *support gradient field* $\mathbf{\Phi}^{\phi}_{sup}$ are trained from the target and support examples via score-matching respectively. The target gradient guides the task completion by providing the fastest direction to increase the task likelihood. The support gradient helps avoid collision by pointing to the inner of the free space. Finally, DualGF policy plans the next waypoint based on the mixture of the two gradients, *i.e.*, the *dual gradient*. Incorporated with a low-level controller, DualGF can interpret the dual gradient into an action.

4.1 Planning with the Dual Gradient Fields

We tackle example-driven planning in a closed-loop manner. At each timestep, we search for the next waypoint $\hat{\mathbf{s}_{t+1}}$ from the free space that greedily maximises the target likelihood:

$$\hat{\mathbf{s}_{t+1}} = \underset{\mathbf{s}_{t+1}}{\operatorname{argmax}} \log p_{tar}(\mathbf{s}_{t+1}) \ \text{ s.t. } \log p_{sup}(\mathbf{s}_{t+1}) \geq \log \frac{1}{|\mathcal{S}_f|} \tag{3}$$

However, this objective is problematic since both the $\log p_{sup}(\mathbf{s}_{t+1})$ and $\log p_{tar}(\mathbf{s}_{t+1})$ may be zero in many regions, such as non-target regions or inner of the free space. Besides, $p_{sup}(\mathbf{s}_{t+1})$ is non-differentiable on the boundary of the free space $\partial\mathcal{S}_f$. As a result, we choose to 'soften' both distributions by adding a small Gaussian noise:

$$p^{\sigma}_{sup}(\hat{\mathbf{s}}) = \int \mathcal{N}(\hat{\mathbf{s}}; \mathbf{s}, \sigma^2 I) p_{sup}(\mathbf{s}) d\mathbf{s}, \ p^{\sigma}_{tar}(\hat{\mathbf{s}}) = \int \mathcal{N}(\hat{\mathbf{s}}; \mathbf{s}, \sigma^2 I) p_{tar}(\mathbf{s}) d\mathbf{s} \tag{4}$$

In this way, we replace the objective in Eq. 3 with the following:

$$\underset{\mathbf{s}_{t+1}}{\operatorname{argmax}} \ \log p^{\sigma}_{tar}(\mathbf{s}_{t+1}), \ \text{ s.t. } \log p^{\sigma}_{sup}(\mathbf{s}_{t+1}) \geq c \tag{5}$$

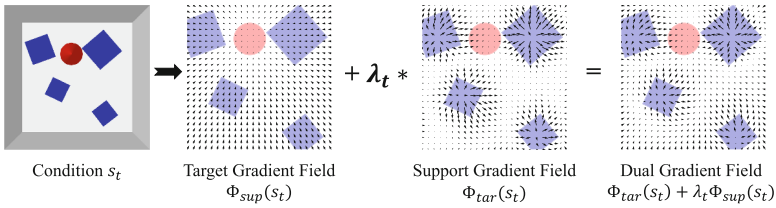


Fig. 2. Visualisation of the learned target, support, and dual gradient fields.

where $c \in \mathbb{R}$ is a threshold indicating the conservative degree of the policy. The problem Eq. 5 can be further related to an unconstrained optimisation with a penalty term by the Lagrangian method:

$$\max_{\mathbf{s}_{t+1}} \min_{\lambda \geq 0} \log p^{\sigma}_{tar}(\mathbf{s}_{t+1}) + \lambda \left(\log p^{\sigma}_{sup}(\mathbf{s}_{t+1}) - c \right) \tag{6}$$

where $\lambda > 0$ is known as a Lagrangian multiplier. The $\mathbf{s}_{t+1}$ and λ can be updated iteratively for multiple steps to obtain a solution. Here, we choose to update the $\mathbf{s}_{t+1}$ and λ **only one step** since the $\hat{\mathbf{s}_{t+1}}$ is close to $\mathbf{s}_t$ in our problem, *i.e.*, $\|\hat{\mathbf{s}_{t+1}} - \mathbf{s}_t\|_2 \leq \epsilon$:

$$\begin{aligned} \hat{\mathbf{s}_{t+1}} &= \mathbf{s}_t + \mu_s \left(\nabla_{\mathbf{s}} \log p^{\sigma}_{tar}(\mathbf{s}_t) + \lambda_t \nabla_{\mathbf{s}} \log p^{\sigma}_{sup}(\mathbf{s}_t) \right) \\ \lambda_{t+1} &= \text{ReLU} \left(\lambda_t - \mu_\lambda \left(\log p^{\sigma}_{sup}(\mathbf{s}_{t+1}) - c \right) \right) \end{aligned} \tag{7}$$

where μ_s and μ_λ are two step sizes. Empirically, the latter is set as a fixed scalar $\mu_\lambda = 0.01$ while the former is properly set to ensure $\|\mathbf{s}_{t+1} - \mathbf{s}_t\|_2 \leq \epsilon$. In this way, the selected waypoint $\hat{\mathbf{s}_{t+1}}$ is derived as follows:

$$\begin{aligned} \hat{\mathbf{s}_{t+1}} &= \mathbf{s}_t + \epsilon \frac{\mathbf{g}_{dual}}{\|\mathbf{g}_{dual}\|_2}, \ \mathbf{g}_{dual} = \underbrace{\mathbf{g}_{tar} + \lambda_t \mathbf{g}_{sup}}_{dual\ gradient}, \\ \mathbf{g}_{tar} &= \underbrace{\nabla_{\mathbf{s}} \log p^{\sigma}_{tar}(\mathbf{s}_t)}_{target\ gradient}, \ \mathbf{g}_{sup} = \underbrace{\nabla_{\mathbf{s}} \log p^{\sigma}_{sup}(\mathbf{s}_t)}_{support\ gradient} \end{aligned} \tag{8}$$

Intuitively, the waypoint is updated by the mixture of the *target gradient* $\nabla_{\mathbf{s}} \log p^{\sigma}_{tar}(\mathbf{s}_t)$ and the *support gradient* $\nabla_{\mathbf{s}} \log p^{\sigma}_{sup}(\mathbf{s}_t)$ weighted by λ_t. As shown in Fig. 2, the target gradient is a vector field that points to the goal region to guide the task completion. The support gradient fields point to the inner region of the free space. The closer a state is to the boundary of the free space, the larger the magnitudes of its support gradient will 'push' it into the inner region.

4.2 Updating Mixing Rate via Support Gradient Field

From Eq. 7, we can infer that λ_t is updated to maintain the 'safety level' $\log p^{\sigma}_{sup}(\mathbf{s})$ at a certain threshold c, since a larger λ_t will lead to an increase of $\log p^{\sigma}_{sup}(\mathbf{s})$ on the further states, vice versa.

However, updating λ_t requires the exact computation of $\log p^{\sigma}_{sup}(\mathbf{s}_{t+1}) - c$ which involves integration over the state space as shown in Eq. 4. This is

intractable when the state space is high-dimensional. Thus, we seek to develop a practical field-balancing mechanism. Firstly, we approximate $\log p^{\sigma}_{sup}(\mathbf{s}_{t+1}) - c$ via the first-order Taylor expansion:

$$\begin{aligned}\log p^{\sigma}_{sup}(\mathbf{s}_{t+1}) - c &= -\big(c - \log p^{\sigma}_{sup}(\mathbf{s}_0)\big) + \log p^{\sigma}_{sup}(\mathbf{s}_{t+1}) - \log p^{\sigma}_{sup}(\mathbf{s}_0) \\ &\approx -\big(c - \log p^{\sigma}_{sup}(\mathbf{s}_0)\big) + \sum_{k=0}^{t} \left\langle \nabla_{\mathbf{s}} \log p^{\sigma}_{sup}(\mathbf{s}_k), \mathbf{s}_{k+1} - \mathbf{s}_k \right\rangle\end{aligned} \tag{9}$$

We further assume that the initial state $\mathbf{s}_0$ is always initialised far away from the boundary of the free space so that $\mathrm{Var}_{\mathbf{s}_0 \sim \rho(\mathbf{s})}[\log p^{\sigma}_{sup}(\mathbf{s}_0) - c] \approx 0$. Thus, the remaining term $c - \log p^{\sigma}_{sup}(\mathbf{s}_0)$ can be regarded as a constant c' in the sense of expectation:

$$\mathbb{E}_{\mathbf{s}_0 \sim \rho(\mathbf{s})}\left[c - \log p^{\sigma}_{sup}(\mathbf{s}_0)\right] = c - \mathbb{E}_{\rho(\mathbf{s})}[\log p^{\sigma}_{sup}(\mathbf{s})] = c + \underbrace{H(\rho(\cdot), p^{\sigma}_{sup}(\cdot))}_{constant} \overset{def}{=} c' \tag{10}$$

where $H(\cdot, \cdot)$ denotes the cross entropy function.

4.3 Incorporating DualGF with Low-Level Controller

DualGF can also tackle the control tasks incorporated with a low-level controller $\mathcal{C} : \mathcal{S} \times \mathcal{S} \to \mathcal{A}$. Under the assumption that the system is holonomic, if the atomic action space is velocity (with speed limit v_{max} and simulation duration Δt), we can set $\epsilon = v_{max} \cdot \Delta t$ in Eq. 7 and properly set the target velocity to reach the next waypoint $\hat{\mathbf{s}_{t+1}}$ via a heuristic-based controller:

$$\mathbf{a}_t = \mathcal{C}(\hat{\mathbf{s}_{t+1}}, \mathbf{s}_t) = \frac{\hat{\mathbf{s}_{t+1}} - \mathbf{s}_t}{\Delta t} = \frac{v_{max}}{\|\mathbf{g}_{dual}\|_2} \cdot \mathbf{g}_{dual} \tag{11}$$

In this work, we conduct experiments on tasks with velocity-based action space (*e.g.*, linear or angular velocity) since the complexity of the system dynamics is not our main focus. The whole planning framework is summarised below:

Algorithm 1. Test-time Inference of DualGF

Initialisation: State $\mathbf{s}_0$, target and support gradient fields $\nabla \log p^{\sigma}_{tar}, \nabla \log \mathrm{sup}^{\sigma}$ and a low-level controller $\mathcal{C}$
for $t = 0$ **to** T **do**
  $\mathbf{g}_{dual} = \nabla_{\mathbf{s}} \log p^{\sigma}_{tar}(\mathbf{s}_t) + \lambda_t \nabla_{\mathbf{s}} \log p^{\sigma}_{sup}(\mathbf{s}_t)$ ▷ Compute the dual gradient.
  $\hat{\mathbf{s}_{t+1}} = \mathbf{s}_t + \epsilon \frac{\mathbf{g}_{dual}}{\|\mathbf{g}_{dual}\|_2}$ ▷ Set the next waypoint.
  $\mathbf{a}_t = \mathcal{C}(\hat{\mathbf{s}_{t+1}}, \mathbf{s}_t) = \frac{v_{max}}{\|\mathbf{g}_{dual}\|_2} \cdot \mathbf{g}_{dual}$ ▷ Compute the action via low-level controller.
  $\mathbf{s}_{t+1} = \mathcal{P}(\mathbf{s}_t, \mathbf{a}_t)$ ▷ Step the environment with action $\mathbf{a}_t$.
  $\nabla \lambda = \sum_{k=0}^{t} \left\langle \nabla_{\mathbf{s}} \log p^{\sigma}_{sup}(\mathbf{s}_k), \mathbf{s}_{k+1} - \mathbf{s}_k \right\rangle - c'$ ▷ Update the mixing-ratio.
  $\lambda_{t+1} = \mathrm{ReLU}\,(\lambda_t - \mu_\lambda \nabla \lambda)$
end for

4.4 Learning Gradient Fields from Examples

Finally, we seek to estimate the target and support gradient fields mentioned above to realise our planning framework. Thanks to the denoising score-matching (DSM), we can obtain a guaranteed estimation of the gradient fields from examples. With $\mathbf{s}_a \in \mathbb{R}^{d_a}$ and $\mathbf{s}_c \in \mathbb{R}^{d_c}$ denoted as agent-state (*e.g.*, agent's position and orientation) and conditional-state (*e.g.*, agent's category, obstacles' state) respectively ($\mathbf{s} = [\mathbf{s}_a, \mathbf{s}_c]$), we adopt DSM to train a *target score network* $\mathbf{\Phi}^{\theta}_{tar} : \mathbb{R}^{d_a} \times \mathbb{R}^{d_c} \to \mathbb{R}^{d_a}$ and a *support score network* $\mathbf{\Phi}^{\phi}_{sup} : \mathbb{R}^{d_a} \times \mathbb{R}^{d_c} \to \mathbb{R}^{d_a}$ from target examples S^*_{tar} and support examples S^*_{sup} respectively. The training objectives of both score networks are as follows:

$$\begin{aligned} \mathcal{L}(\theta) &= \mathbb{E}_{\substack{q_\sigma(\widetilde{\mathbf{s}}_a|\mathbf{s}_a) \\ p_{tar}(\mathbf{s})}} \left[\left\| \mathbf{\Phi}^{\theta}_{tar}(\widetilde{\mathbf{s}}_a|\mathbf{s}_c) - \frac{\mathbf{s}_a - \widetilde{\mathbf{s}}_a}{\sigma^2} \right\|_2^2 \right] \\ \mathcal{L}(\phi) &= \mathbb{E}_{\substack{q_\sigma(\widetilde{\mathbf{s}}_a|\mathbf{s}_a) \\ p_{sup}(\mathbf{s})}} \left[\left\| \mathbf{\Phi}^{\phi}_{sup}(\widetilde{\mathbf{s}}_a|\mathbf{s}_c) - \frac{\mathbf{s}_a - \widetilde{\mathbf{s}}_a}{\sigma^2} \right\|_2^2 \right] \end{aligned} \tag{12}$$

where $q_\sigma(\widetilde{\mathbf{s}}_a|\mathbf{s}_a) = \mathcal{N}(0, \sigma^2 I)$ is a Gaussian kernel. When σ is small enough, the optimal score networks hold that $\mathbf{\Phi}^*_{tar}(\mathbf{s}_a|\mathbf{s}_c) = \nabla_{\mathbf{s}_a} \log p_{tar}(\mathbf{s}_a|\mathbf{s}_c)$, $\mathbf{\Phi}^*_{sup}(\mathbf{s}_a|\mathbf{s}_c) = \nabla_{\mathbf{s}_a} \log p_{sup}(\mathbf{s}_a|\mathbf{s}_c)$. In practice, we adopt a variant of DSM [25] that simultaneously employs DSM on varying noise scales. Both networks are implemented using graph neural networks (GNNs), where the input is a fully connected graph with each node representing the state of an object. This design follows a similar architecture to that proposed in [28]. Our method (and all the baselines included in this work) are trained on a single RTX3090 GPU within one day.

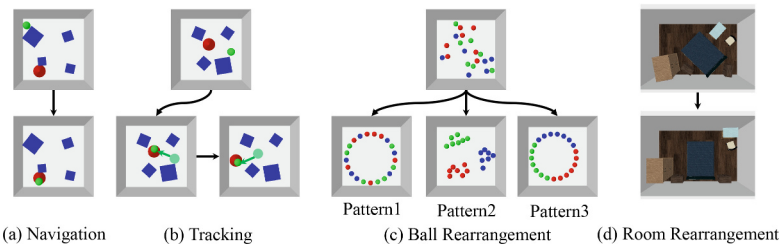


Fig. 3. Task illustrations. *Navigation* (and *Tracking*): Reaching a (dynamic) goal location while avoiding collision with obstacles. *Ball Rearrangement*: Rearranging balls into certain patterns. *Room rearrangement*: Rearranging objects into tidy layouts.

5 Experiments

5.1 Setups

Task Descriptions: We first evaluate our method on single-agent path planning tasks, *SimpleMaze*: *Navigation* requires the agent to reach the goal without colliding with the obstacles. The obstacles' size, location and orientation will be randomly initialised in each episode. *Tracking* is built upon *Navigation* and requires the agent to catch up with a dynamic goal. Except for the state-based setting, we also consider the image-based setting for *SimpleMaze*. We also evaluate our method on multi-agent path planning tasks proposed by [28]: *Ball Rearrangement* requires the agent to control 21 balls of different categories to form patterns. The agent is not given a goal in the test phase and needs to infer pattern priors from target examples. *Room Rearrangement* is a more realistic task with diverse object categories and a more complicated action space. It requires the agent to move objects to a tidy layout. The agent receives 756 rooms as target examples for training and is evaluated under 83 unseen rooms in the test phase. The task demonstrations are illustrated in Fig. 3.

Baselines: We compare our framework (*i.e.*, *DualGF*) with learning-based baselines and planning-based references.

- **Leaning-based Baselines:** *TarGF (SAC):* is the learning-based framework of TarGF [28]. *RCE:* The SOTA example-driven reinforcement learning method. *RL (SAC):* The agent is trained under the human-designed task reward and collision signals.
- **Planning-based References:** *Probabilistic Road Map (PRM):* A classical sampling-based path planning method [12]. *Potential Field:* Artificial Potential Field method [16].
- **UpperBounds:** *Oracle:* The performance is obtained by slightly perturbing the target examples with a small Gaussian noise. We use *Oracle*'s performance to normalise the PL-curves of the *Ball Rearrangement*. *TarGF (ORCA):* A model-based framework that requires the ground-truth model.

5.2 Comparative Results on *SimpleMaze*

We collect trajectories starting from the same initial states over 5 random seeds for evaluation. For *Navigation* and *Tracking*, we measure the task performance and constraint violation via the following metrics:

- **Task Return weighted by Feasibility (TRF)** is the **primary metrics** that calculates the averaged cumulative reward of an episode : With the goal position denoted as $g \in \mathbb{R}^2$, TRF $= \mathbb{E}[\mathbb{1}(\tau \text{ is feasible}) \sum_\tau r_t]$ where $r_t = \mathbb{1}(||\mathbf{s}_a^t - g||_2 < 0.4)$ in *Navigation* and $r_t = 2\sqrt{2} - ||\mathbf{s}_a^t - g||_2$ in *Tracking*. $\mathbb{1}(\tau \text{ is feasible}) = 1$ only when τ has no collision with the obstacles.
- **Feasibility Rate (FR)** calculates the percentage of **non-collision** trajectories: $\mathbb{E}[\mathbb{1}(\tau \text{ is success})]$.

Table 1. Results on *Navigation* and *Tracking*. We compare *DualGF* with learning-based baselines where TRF serves as the **primary metric** and the planning-based approaches serve as the **references**. In the image-based setting, the agent receives a 64×64 image and its two-dimensional position.

		Navigation(State)		Tracking(State)		Navigation(Image)		Tracking(Image)	
		TRF ↑	FR ↑	TRF ↑	FR ↑	TRF ↑	FR ↑	TRF ↑	FR ↑
Learning-based Baselines	RL(SAC)	$0.17_{\pm 0.01}$	$0.18_{\pm 0.02}$	$72.05_{\pm 8.26}$	$0.40_{\pm 0.04}$	$0.10_{\pm 0.05}$	$0.18_{\pm 0.03}$	$46.71_{\pm 7.10}$	$0.22_{\pm 0.03}$
	RCE	$0.04_{\pm 0.01}$	$0.22_{\pm 0.04}$	$11.59_{\pm 3.85}$	$0.05_{\pm 0.02}$	$0.01_{\pm 0.01}$	$0.53_{\pm 0.06}$	$50.62_{\pm 3.08}$	$0.53_{\pm 0.06}$
	TarGF	$0.15_{\pm 0.04}$	$0.19_{\pm 0.05}$	$40.51_{\pm 1.94}$	$0.17_{\pm 0.00}$	$0.04_{\pm 0.01}$	$0.22_{\pm 0.05}$	$32.69_{\pm 2.62}$	$0.14_{\pm 0.01}$
	DualGF(wo sup)	$0.18_{\pm 0.04}$	$0.18_{\pm 0.04}$	$22.29_{\pm 3.80}$	$0.09_{\pm 0.01}$	$0.14_{\pm 0.04}$	$0.16_{\pm 0.02}$	$20.72_{\pm 0.28}$	$0.08_{\pm 0.00}$
	DualGF	$\mathbf{0.46}_{\pm 0.03}$	$0.62_{\pm 0.02}$	$\mathbf{113.02}_{\pm 6.07}$	$0.47_{\pm 0.03}$	$\mathbf{0.22}_{\pm 0.01}$	$0.34_{\pm 0.02}$	$\mathbf{56.37}_{\pm 2.44}$	$0.22_{\pm 0.01}$
Classical Path Planners	Potential Field	$0.56_{\pm 0.03}$	$0.87_{\pm 0.01}$	$169.98_{\pm 4.13}$	$0.77_{\pm 0.03}$	– – –	– – –	– – –	– – –
	PRM	$0.65_{\pm 0.07}$	$0.65_{\pm 0.07}$	$97.01_{\pm 19.02}$	$0.38_{\pm 0.07}$	– – –	– – –	– – –	– – –

As shown in Table 1, *DualGF* demonstrates strong generalisation in unseen conditions and complex observations (image-based setting). 1) *DualGF* consistently outperforms learning-based baselines by a large margin on the primary metric TRF . In image-based experiments, we observed that *RCE* suffers from training collapse due to high-dimensional observations. This collapse leads to the policy rarely reaching the goal and consistently moving to the lower left corner, resulting in higher FR values compared to *DualGF*. 2) Planning-based references generally outperform *DualGF* since they leverage human knowledge or environment priors. However, the performance gaps between *DualGF* and the references are smaller than those between *DualGF* and the learning-based baselines. Additionally, our method directly adapts to the image-based setting without ad-hoc design, unlike the planning-based references.

5.3 Comparative Results on Rearrangement

We collect trajectories starting from the same initial states over 5 random seeds for evaluation. For *Ball Rearrangement* and *Room Rearrangement*, we use PL and CS to measure task performance and ACN to measure constraint violation:

- **Pseudo-Likelihood (PL)** measures the similarity between a given state and a target distribution by assigning a *pseudo-likelihood function* $\mathbf{F}_{proxy} : \mathcal{S} \rightarrow \mathbb{R}^+$. At each time step t, the PL-curve reports the averaged PL across all trajectories $\mathbb{E}[\mathbf{F}_{proxy}(\mathbf{s}_t)]$.
- **Coverage Score (CS):** reports the Minimal-Matching-Distance (MMD) [1] between terminal states of a method S_T and a fixed set of examples S_{gt} from p_{tar}: $\sum_{\mathbf{s}_{gt} \in S_{gt}} \min_{\mathbf{s}_T \in S_T} \|\mathbf{s}_{gt} - \mathbf{s}_T\|$.
- **Averaged Collision Number (ACN)** reports the averaged collision number $\mathbb{E}[\mathbf{c}_t]$ at each time step where $\mathbf{c}_t = \sum c^t_{i,j}$, $c^t_{i,j} = 1$ only if the i-th and j-th object collide and $c^t_{i,j} = 0$ otherwise.

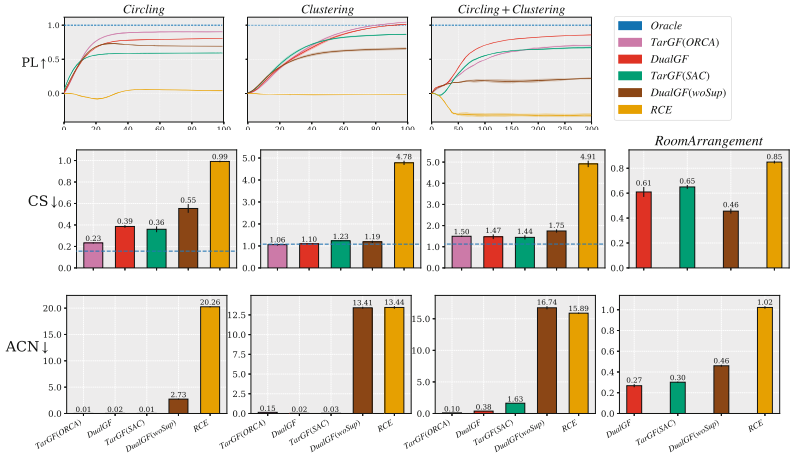


Fig. 4. Comparative Results on *Ball Rearrangement* and *Room Rearrangement*. The results on *Ball Rearrangement* are plotted in the first three columns. Results on *Room Rearrangement* are plotted in the last column. The X-axis order is fixed across CS and ACN bars. Note that ***TarGF (ORCA)*** serves as the upperbound method in *Ball Rearrangement* since it leverages the ground truth model.

On *Ball Rearrangement*: *DualGF* shows good scalability and generalisation to high-dimensional domains. 1) As Fig. 4 illustrates, *DualGF* significantly outperforms learning-based baselines in PL while achieving comparable performance with the *TarGF (SAC)* in ACN. In the hardest task *Circling + Clustering*, *DualGF* even outperforms the upperbound *TarGF (ORCA)* that plans with the ground truth model. These demonstrate the effectiveness of our method to scale to high-dimensional tasks. 2) The CS of *DualGF* is comparable with other competitive baselines such as *TarGF (SAC)*. Hence, our method is not trivially overfitting to a single mode.

On *Room Rearrangement*: *DualGF* shows good generalisation under the novel condition and generality for the complex domains. 1) *DualGF* achieves the best performance compared with baselines in both CS and ACN. This indicates that our method can generalise well to the diverse conditional attributes such as different sizes, numbers, and compositions of objects. 2) The planning-based methods suffer from either scalability or feasibility issues. *PRM* and Rapidly-exploring Random Trees (RRT) [15] cannot search for a feasible solution in 10 min. Besides, *TarGF (ORCA)* assumes the agent is circular in shape, which is infeasible in *Room Rearrangement*.

5.4 Ablation Studies and Analysis

We conduct ablation studies to analyse the effectiveness of the key components of our method, *i.e.*, the *support gradient field* and the *field-balancing mechanism*.

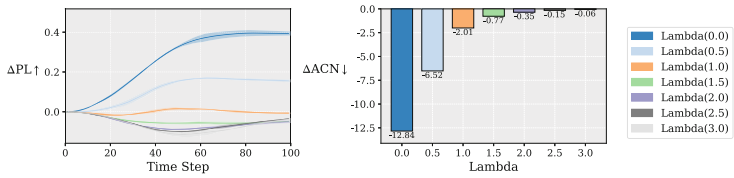


Fig. 5. Ablation results on *Clustering*, where ΔMetric = Metric(DualGF) - Metric(Ablated). As the mixing rate λ_0 decreases, the advantage of *DualGF* over the ablation grows significantly.

Effectiveness of the Support Gradient Field: *Ours wo Sup* is the ablated version with a fixed mixing rate $\lambda \equiv 0$. In all tasks (*Navigation*, *Tracking*, and *Ball Rearrangement*), *DualGF wo Sup* performs worse than *DualGF*. Additionally, *DualGF* has lower ACN than *DualGF wo Sup* (Fig. 4), and its FR in *Navigation* and *Tracking* is significantly better. These results show that the support gradient field effectively avoids collisions and improves performance. In *Room Rearrangement*, *DualGF wo Sup* achieves better CS but worse ACN than *DualGF*, which outperforms the baselines in both CS and ACN. This suggests that the support gradient field strikes a better balance between task completion and collision avoidance.

Effectiveness of the Field-Balancing Mechanism: *DualGF (Fixed)* is the ablated version of *DualGF* with mixing rate lambda fixed at an initial level. We illustrate the advantage of *DualGF* over *DualGF (Fixed)* on PL and ACN on Fig. 5. As the initial lambda decreases, the advantage of *DualGF* increases significantly in both PL and ACN. This indicates our field balancing mechanism can moderate the sensitivity to the choice of initial mixing rate λ_0.

6 Conclusion

In this study, we first reformulate the path planning problem in a novel data-driven paradigm where the agent learns to plan with two sets of examples. We propose a novel score-based planning framework (DualGF) to tackle this problem. There are two gradient fields in DualGFs: a *target gradient field* that guides task completion and a *support gradient field* that ensures moving with physical constraints. Moreover, an adaptive gradient balance mechanism is introduced to combine the merits of the two fields. Experiments demonstrate the scalability and generality of our method, and empirical results show that it significantly outperforms off-the-shelf learning-based baselines in efficiency and safety.

Limitations and Future Work: The main limitations of this work are the local minimum issue that arises from gradient-based optimization and the holonomic assumption made in Sect. 4.3. Another minor issue is the Lagrangian Relaxation's inherent instability, which can be mitigated via a small learning rate. In the future, we may incorporate our method with sampling-based approaches to overcome the local minimum issues. Besides, we may extend our method to non-holonomic scenarios by integrating model-predictive control (MPC).

Acknowlegement. We sincerely thank Yang Tian and Jiyao Zhang for their insightful discussions. This project was supported by the National Natural Science Foundation of China (62376006), National Natural Science Foundation of China (62136001), and National Youth Talent Support Program (8200800081).

References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3D point clouds. In: International Conference on Machine Learning, pp. 40–49. PMLR (2018)
2. Aggarwal, S., Kumar, N.: Path planning techniques for unmanned aerial vehicles: a review, solutions, and challenges. Comput. Commun. **149**, 270–299 (2020)
3. Chai, Q., Wang, Y., He, Y., Xu, C., Hong, Z.: Improved PRM path planning in narrow passages based on PSO. In: 2022 IEEE International Conference on Mechatronics and Automation (ICMA), pp. 41–46. IEEE (2022)
4. Dobson, A., Bekris, K.E.: Sparse roadmap spanners for asymptotically near-optimal motion planning. Int. J. Robot. Res. **33**(1), 18–47 (2014)
5. Eysenbach, B., Levine, S., Salakhutdinov, R.: Replacing rewards with examples: example-based policy search via recursive classification. arXiv preprint arXiv:2103.12656 (2021)
6. Fu, J., Luo, K., Levine, S.: Learning robust rewards with adversarial inverse reinforcement learning. arXiv preprint arXiv:1710.11248 (2017)
7. Fu, J., Singh, A., Ghosh, D., Yang, L., Levine, S.: Variational inverse control with events: a general framework for data-driven reward definition. arXiv preprint arXiv:1805.11686 (2018)
8. Gammell, J.D., Barfoot, T.D., Srinivasa, S.S.: Batch informed trees (bit*): informed asymptotically optimal anytime search. Int. J. Robot. Res. **39**(5), 543–567 (2020)
9. Gasparetto, A., Boscariol, P., Lanzutti, A., Vidoni, R.: Path planning and trajectory planning algorithms: a general overview. In: Motion and Operation Planning of Robotic Systems: Background and Practical Approaches, pp. 3–27 (2015)
10. Hatch, K., Yu, T., Rafailov, R., Finn, C.: Example-based offline reinforcement learning without rewards. Proc. Mach. Learn. Res. **144**, 1–17 (2022)
11. Hüppi, M., Bartolomei, L., Mascaro, R., Chli, M.: T-PRM: temporal probabilistic roadmap for path planning in dynamic environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2022) (2022)
12. Kavraki, L., Latombe, J.C.: Randomized preprocessing of configuration for fast path planning. In: Proceedings of the 1994 IEEE International Conference on Robotics and Automation, pp. 2138–2145. IEEE (1994)
13. King, J.E., Cognetti, M., Srinivasa, S.S.: Rearrangement planning using object-centric and robot-centric action spaces. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 3940–3947. IEEE (2016)
14. Kostrikov, I., Agrawal, K.K., Dwibedi, D., Levine, S., Tompson, J.: Discriminator-actor-critic: addressing sample inefficiency and reward bias in adversarial imitation learning. arXiv preprint arXiv:1809.02925 (2018)
15. LaValle, S.M., et al.: Rapidly-exploring random trees: a new tool for path planning (1998)

16. Lee, M.C., Park, M.G.: Artificial potential field based path planning for mobile robots using a virtual obstacle concept. In: Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003), vol. 2, pp. 735–740. IEEE (2003)
17. Li, C., Ma, H., Xu, P., Wang, J., Meng, M.Q.H.: BiAIT*: symmetrical bidirectional optimal path planning with adaptive heuristic. arXiv preprint arXiv:2205.06940 (2022)
18. Liu, M., He, T., Xu, M., Zhang, W.: Energy-based imitation learning. arXiv preprint arXiv:2004.09395 (2020)
19. Patle, B., Pandey, A., Parhi, D., Jagadeesh, A., et al.: A review: on path planning strategies for navigation of mobile robot. Defence Technol. **15**(4), 582–606 (2019)
20. Pomerleau, D.A.: Efficient training of artificial neural networks for autonomous navigation. Neural Comput. **3**(1), 88–97 (1991)
21. Reddy, S., Dragan, A.D., Levine, S.: SQIL: imitation learning via reinforcement learning with sparse rewards. arXiv preprint arXiv:1905.11108 (2019)
22. Ross, S., Bagnell, D.: Efficient reductions for imitation learning. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. JMLR Workshop and Conference Proceedings, pp. 661–668 (2010)
23. Ross, S., Gordon, G., Bagnell, D.: A reduction of imitation learning and structured prediction to no-regret online learning. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pp. 627–635. JMLR Workshop and Conference Proceedings (2011)
24. Russell, S.J.: Artificial Intelligence a Modern Approach. Pearson Education Inc. (2010)
25. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)
26. Strub, M.P., Gammell, J.D.: Adaptively informed trees (AIT*): fast asymptotically optimal path planning through adaptive heuristics. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 3191–3198. IEEE (2020)
27. Strub, M.P., Gammell, J.D.: Adaptively informed trees (AIT*) and effort informed trees (EIT*): asymmetric bidirectional sampling-based path planning. Int. J. Robot. Res. **41**(4), 390–417 (2022)
28. Wu, M., Zhong, F., Xia, Y., Dong, H.: TarGF: learning target gradient field for object rearrangement. In: Thirty-Sixth Conference on Neural Information Processing Systems (2022)
29. You, H., Chen, G., Jia, Q., Huang, Z.: Path planning for robot in multi-dimensional environment based on dynamic PRM blended potential field. In: 2021 IEEE 5th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), vol. 5, pp. 1157–1162. IEEE (2021)
30. Zhong, F., Sun, P., Luo, W., Yan, T., Wang, Y.: AD-VAT+: an asymmetric dueling mechanism for learning and understanding visual active tracking. IEEE Trans. Pattern Anal. Mach. Intell. **43**(5), 1467–1482 (2019)
31. Ziebart, B.D., Maas, A.L., Bagnell, J.A., Dey, A.K., et al.: Maximum entropy inverse reinforcement learning. In: AAAI, vol. 8, pp. 1433–1438. Chicago, IL, USA (2008)

Improving Physics-Informed Neural Network Extrapolation via Transfer Learning and Adaptive Activation Functions

Athanasios Papastathopoulos-Katsaros[1,2(✉)], Alexandra Stavrianidi[3(✉)], and Zhandong Liu[1,2]

[1] Department of Pediatrics, Baylor College of Medicine, Houston, TX 77030, USA
athanasios.papastathopoulos-katsaros@bcm.edu
[2] Data Science Center, Jan and Dan Duncan Neurological Research Institute, Texas Children's Hospital, Houston, TX 77030, USA
[3] Department of Mathematics, Stanford University, Stanford, CA 94305, USA
alexst@stanford.edu

Abstract. Physics-Informed Neural Networks (PINNs) are deep learning models that incorporate the governing physical laws of a system into the learning process, making them well-suited for solving complex scientific and engineering problems. Recently, PINNs have gained widespread attention as a powerful framework for combining physical principles with data-driven modeling to improve prediction accuracy. Despite their successes, however, PINNs often exhibit poor extrapolation performance outside the training domain and are highly sensitive to the choice of activation functions (AFs). In this paper, we introduce a transfer learning (TL) method to improve the extrapolation capability of PINNs. Our approach applies transfer learning (TL) within an extended training domain, using only a small number of carefully selected collocation points. Additionally, we propose an adaptive AF that takes the form of a linear combination of standard AFs, which improves both the robustness and accuracy of the model. Through a series of experiments, we demonstrate that our method achieves an average of 40% reduction in relative L_2 error and an average of 50% reduction in mean absolute error in the extrapolation domain, all without a significant increase in computational cost. The code is available at https://github.com/LiuzLab/PINN-extrapolation

Keywords: Physics-informed neural networks · transfer learning · activation function optimization · extrapolation

A. Papastathopoulos-Katsaros and A. Stavrianidi—These authors contributed equally to this work.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-032-04555-3_24.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 289–301, 2026.
https://doi.org/10.1007/978-3-032-04555-3_24

1 Introduction

Physics-Informed Neural Networks (PINNs), introduced by Raissi et al. [20], are neural networks trained not only on data but also constrained by physical laws, expressed as partial differential equations (PDEs). By embedding physics as prior knowledge, PINNs restrict the space of admissible solutions and reduce overfitting. PINNs have been successful in various use cases, in fields such as fluid mechanics [4], medicine [13,26], and earth science [18]. In this work, we focus on using PINNs as solvers for nonlinear PDEs.

We consider a nonlinear PDE of the form

$$u_t + \mathcal{N}(u) = 0, x \in D, t \in [0, T]$$

where $u(t, x)$ denotes the latent solution, $\mathcal{N}$ denotes a nonlinear differential operator, D is the spatial domain and T is the final time. To ensure a well-posed problem, an initial condition is prescribed for u at $t = 0$, along with boundary conditions on ∂D. The mean squared error loss function of the neural network is given by

$$MSE = \frac{w_u}{N_u} \sum_{i=1}^{N_u} \left| u\left(t_u^i, x_u^i\right) - u^i \right|^2 + \frac{w_f}{N_f} \sum_{i=1}^{N_f} \left| u_t\left(t_f^i, x_f^i\right) + \mathcal{N}(u)\left(t_f^i, x_f^i\right) \right|^2$$

where $\left\{t_u^i, x_u^i, u^i\right\}_{i=1}^{N_u}$ denote the initial and boundary data prescribed to the problem and $\left\{t_f^i, x_f^i\right\}_{i=1}^{N_f}$ denote the collocation points in the domain where the PDE is enforced. The weights w_u and w_f reflect our trust in the data and the PDE, respectively.

Although PINNs require substantially less computational resources than finite elements methods [6,26], are mesh-free, and easily programmed, they perform poorly in extrapolation tasks [7,11] which assess their performance on test data $\left\{\left(t_{\text{test}}^i, x_{\text{test}}^i\right)\right\} \subset (T_{\text{train}}, T] \times D$, where $T > T_{\text{train}}$. In this paper, we propose a transfer-learning method that reduces the L_2 extrapolation error by 40% on average and the mean absolute extrapolation error by 50% on average across several benchmarks, including the Allen–Cahn (AC), Korteweg–de Vries (KdV), and Burgers equations.

2 Related Work

The original PINN architecture [20] uses the tanh activation function (AF) in all the layers. However, the performance of PINNs is strongly influenced by the choice of AF [1,8,14,22], and selecting AF that reflect the functional form of the solution has been shown to improve performance. In [1], the authors use components of the PDE problem at hand (e.g., the PDE coefficients, the initial condition, analytical solutions in lower dimensions), while in [14], the authors propose using $x + \sin^2(x)$ to learn periodic functions. In [8,22], the authors

introduce adaptive approaches. Following the ABU-PINN (Adaptive Blending Unit) method [22], one can take a linear combination of standard AFs (GELU, ELU, sigmoid, tanh, sin, exp, Softplus, and Swish) with adaptive weights and adaptive slopes as the AF f:

$$f(x) = \sum_{i=1}^{N} G(\alpha_i)\, \sigma_i(\beta_i x)$$

where $\sigma_i(\cdot)$, α_i , β_i denote the candidate AF, the learnable weight parameters, and the learnable scaling factors (adaptive slopes) respectively. The gate function $G(\cdot)$ is softmax, $G(\alpha_i) = \exp(\alpha_i) / \sum_{j=1}^{N} \exp(\alpha_j)$, so the search space is the convex hull of the set of candidate AF. In the rest of the paper, we will refer to these as ABU-PINN combinations.

The performance of the PINN also depends strongly on the selection of collocation points and the choice of the weights w_u and w_f. As demonstrated in prior works [24,25], adaptively sampling collocation points from regions that exhibit large PDE residuals during training can substantially improve accuracy.

Different PDEs have different generalization capabilities [23], as equations with a higher time derivative $|u_t|$ exhibit more variance in the time domain. For some PDEs, the extrapolation error increases exponentially fast as one moves away from the training domain [7]. In [11], the authors propose a method that adjusts the gradient descent vector update to always point in a direction that reduces the PDE loss error during training. However, this is ineffective when the initial and boundary conditions are imposed as hard constraints to accelerate training [2], as the loss function only includes the PDE residual.

Motivated by these considerations, we incorporate the initial and boundary conditions as hard constraints and propose a transfer-learning (TL) method to deal with the extrapolation issue, in which the AF selection plays a crucial role. By retraining only the final layer of the network on a small set of carefully selected collocation points in the validation region and learning an optimal linear combination of AF, we improve the extrapolation performance substantially for several benchmarks on the Allen-Cahn (AC), Korteweg–de Vries (KdV), and Burgers equations. We achieve an average reduction of 40% for the L_2 and of 50% for the mean absolute extrapolation error.

By fine-tuning only the parameters of the final layer and those of the adaptive AF, we achieve this error reduction with a minimal computational cost see table (S2). Fine-tuning only the final layer's parameters is a common practice in transfer learning [5], and this approach is further supported by a layer-wise feature importance analysis we conducted for the AC equation (see Figure S16).

Prior transfer learning approaches for PINNs have primarily focused on parameter tuning and data reuse to achieve generalization across different physical systems. In these works, transfer learning is applied across tasks, from one PDE to another, to improve performance in forward problems [5,9,10,16,19,21], and inverse problems [27]. In contrast, our method focuses on enhancing extrapolation performance for the same PDE in regions beyond the training domain. We also use a fundamentally different strategy that combines adaptive activation

function design with localized retraining of only the final layer based on newly sampled high-loss collocation points from an intermediate region.

3 Method Description

We split our temporal domain into three segments: the training range, the validation range, and the testing range, i.e. $[0, T_{\text{train}}]$, $(T_{\text{train}}, T_{\text{val}}]$, and $(T_{\text{val}}, T_{\text{test}}]$, where $T = T_{\text{test}} > T_{\text{val}} > T_{\text{train}} > 0$. We use $T_{\text{train}} = \frac{T}{2}, T_{\text{val}} = \frac{4T}{5}$, and $T_{\text{test}} = T$. The final 20% of the temporal domain is reserved for extrapolation, as in [11] and other time-series prediction studies.

To enforce the initial and boundary conditions of the PDE as hard constraints, we perform a change of variables to the governing equations. This ensures that the constraints are satisfied exactly and eliminates the need to balance the data loss against the PDE loss using fine-tuned weights w_u and w_f.

Initial Training Phase

1. We begin by training the neural network on the domain $[0, T_{train}]$. The training is monitored using the L_2 error on the validation interval $[T_{train}, T_{val}]$, and early stopping is applied to prevent overfitting.
2. After early stopping, we assess the extrapolation performance by computing the L_2 error on the test interval $(T_{\text{val}}, T_{\text{test}}]$.
3. To investigate how different AF impact extrapolation, we fix the AF tanh in all the layers except the final one, where we experiment with different alternatives.

Transfer Learning (TL) Phase

1. After the initial training phase is complete, we freeze the weights of all the layers except the final one, which is a typical practice in TL.
2. We uniformly sample a relatively large number of collocation points from the interval $[0, T_{\text{val}})$ and calculate their respective loss in a forward pass of the network. We then select a very small number of those collocation points: the ones that led to the highest loss.
3. We retrain the network on those few collocation points, using the Adam optimizer, updating only the final layer's weights and the parameters of its AF. This helps refine the model's predictive ability in the extrapolation domain without disrupting earlier learned dynamics. We use $L2$ regularization, which penalizes large deviations in the retrained weights by adding a term proportional to the square of their magnitude. Among several techniques to avoid catastrophic forgetting (including basic fine-tuning and Elastic Weight Consolidation [12]), $L2$ regularization yielded the best performance, so we adopted it for all our experiments (see supplementary material section A).

4 Experimental Setup

4.1 Hyperparameter Selection

We applied our method to the Allen–Cahn (AC), Korteweg–de Vries (KdV), and Burgers equations using neural networks composed of six layers with 32 neurons each. We use the L-BFGS optimizer [3], which ensures rapid convergence, paired with early-stopping. The network parameters are initialized using the Xavier scheme. For all equations, we use $8k$ uniformly sampled collocation points. For the KdV equation specifically, we also include 200 boundary points.

For the TL phase, we switch to the Adam optimizer for an additional 120 to 150 epochs, using a learning rate of `1e-3` to `1e-2`. A total of 80 collocation points are selected for training in the transfer phase out of $4k$ uniformly sampled collocation points.

4.2 Sampling Strategy

We experimented with varying the proportion of collocation points sampled in the training interval $[0, 0.5]$ and in the validation interval $(0.5, 0.8]$. Ultimately, we found that sampling exclusively from the validation interval produced the best results. Additionally, the existing mechanisms in our approach, including $L2$ regularization, the chosen number of training epochs, and the step size adjustments during TL, were sufficient to prevent catastrophic forgetting. The corresponding results are presented in Figures S1, S6, S11.

4.3 Evaluation Metrics

We consider the L_2 relative error and the mean absolute error (MAE) to evaluate the accuracy of the PINN solutions, defined as follows:

$$L_2 \text{ error} = \frac{\sqrt{\sum_{i=1}^{N} \left|u_\theta\left(x_i, t_i\right) - u\left(x_i, t_i\right)\right|^2}}{\sqrt{\sum_{i=1}^{N} \left|u\left(x_i, t_i\right)\right|^2}}$$

$$MAE = \frac{\sum_{i=1}^{N} \left|u_\theta\left(x_i, t_i\right) - u\left(x_i, t_i\right)\right|}{\sum_{i=1}^{N} \left|u\left(x_i, t_i\right)\right|}$$

where N is the number of test points, $u\left(x_i, t_i\right)$ is the reference solution at (x_i, t_i), and $u_\theta\left(x_i, t_i\right)$ is the PINN solution at (x_i, t_i). To measure the extrapolation error we take $\left\{\left(x_{\text{test}}^i, t_{\text{test}}^i\right)\right\}$, where x_{test}^i is sampled at a uniform spatial mesh grid in the spatial domain and t_{test}^i is on a uniform temporal grid in $(T_{\text{val}}, T_{\text{test}}]$. We use a temporal step size of 0.01 for Burgers, 0.005 for AC and KdV, and sample 400 collocation points at each time instant for AC, 600 for Burgers, and 500 for KdV.

To obtain the reference solution to the PDE, we used a finite difference method for spatial discretization combined with the implicit ODE solver ode15s

in MATLAB. The second spatial derivative was approximated using a centered difference scheme, and the other terms were evaluated pointwise. To enforce the boundary conditions, the solution at the domain boundaries was explicitly set at every time step. The solver's relative and absolute tolerances were set to `1e-6` and `1e-8`, respectively, ensuring numerical stability and accuracy.

5 Experimental Results

We benchmarked our method on three different equations: Allen-Cahn (AC), Korteweg–de Vries (KdV), and the viscous Burgers equation. Each experiment was repeated ten times, and we report the average results. In all examples, the training domain was set to $[0, 0.5]$, the validation domain to $[0.5, 0.8]$ and the extrapolation domain to $[0.8, 1]$. As mentioned previously, we evaluated the extrapolation performance of different AFs in the final layer while all other layers used tanh. The pool of candidate AFs for the final layer consists of:

tanh: The most commonly used AF for PINNs.

ABU-PINN from [22]: A linear combination of different AF including GELU, ELU, sigmoid, tanh, sin, exp, Softplus, and Swish. We take adaptive linear combinations of at least three and up to six AF at a time. The use of softmax for weight normalization is maintained as in [22].

$\mathbf{x} + \sin^2(\mathbf{x})$ from [14]: It has been shown to improve performance, potentially due to its semi-periodic component and its monotonicity.

Linear combinations of tanh (lctanh): $\sum_{i=1}^{n} w_i \tanh(a_i x + b_i)$, where the coefficients a_i, b_i, w_i are optimized.

Linear combinations of sin (lcsin): $\sum_{i=1}^{n} w_i \sin(a_i x + b_i)$, where the coefficients a_i, b_i, w_i are optimized.

Linear combinations of $x + \sin^2(x)$: $\sum_{i=1}^{n}(a_i x + b_i sin^2(c_i x + d_i))$, where the coefficients a_i, b_i, c_i, d_i are optimized.

As noted in [22], each PDE tends to have a preferred AF; for example, tanh performs best for the Allen–Cahn and viscous Burgers equations, while $x + \sin^2(x)$ is preferred for KdV. Building on this insight, we consider adaptive linear combinations of the preferred AF for each PDE (see supplementary material section D).

5.1 Allen-Cahn (AC) Equation

The AC equation is given by

$$u_t - 0.0001u_{xx} + 5u^3 - 5u = 0,\ x \in [-1, 1],\ t \in [0, 1],$$
$$u(0, x) = x^2 \cos(\pi x), u(-1, t) = u(1, t) = -1.$$

We enforce the initial and boundary conditions as hard constraints by setting

$$u(t, x) = x^2 \cos(\pi x) + t\left(1 - x^2\right) v(t, x).$$

Then u automatically satisfies the initial and boundary conditions, and we only need to impose the PDE constraint on v (see supplementary material section B).

The best ABU-PINN was the combination of tanh, GELU, and sigmoid. For lctanh and lcsin, the combination of three AF was shown to give the best results. Our findings are presented in Table 1. Overall, a linear combination of tanh provided the best extrapolation accuracy while introducing a similar number of trainable parameters as other linear combinations.

Regarding the TL results, we plot the collocation points with the highest PDE loss (the ones chosen for training) in Figure S1. As expected, these points are predominantly concentrated towards the end of the temporal validation domain and around the two regions with the largest spatial discrepancies. We trained for 150 epochs and set the learning rate for the Adam optimizer to `5e-3`.

The performance of various AF with TL is summarized in Table 1. Despite using only a small number of collocation points and training for a limited number of epochs, our results demonstrate substantial improvements. Specifically, we reduced the L_2 error from 0.20 to 0.12 compared to the standard PINN model and the MAE from 0.11 to 0.06. A benchmark of our method against the state-of-the-art methods of References [11,15,17] is presented in Table 2. The results show that our TL approach outperforms all other methods.

Figures S2, S3, S4, and S5 allow for a visual comparison of the model performance with and without TL, as well as the performance of the tanh AF versus the best lctanh. The results are shown for two different timestamps, $t = 0.82$ and $t = 0.99$, which highlight the challenges of extrapolation due to the rapid spatial transitions that occur over time. As illustrated in the figures, it is evident that both TL and a well-chosen AF contribute significantly to the improvement of the extrapolation performance. Furthermore, errors tend to be larger at later timestamps (t), a finding that aligns with [7].

Table 1. Comparative analysis of L_2 and MAE for various AF applied to the AC equation, demonstrating the effectiveness of TL and of carefully selected AF. Results for lctanh, lcsin, and ABU-PINN correspond to the optimal linear combinations described in the main text.

	L_2		MAE	
AF	Without TL	With TL	Without TL	With TL
tanh	0.20	0.17	0.11	0.09
$x + sin^2(x)$	0.23	0.19	0.14	0.1
ABU-PINN	0.19	0.14	0.11	0.08
lctanh	0.16	0.12	0.09	0.06
lcsin	0.18	0.14	0.10	0.07

5.2 Korteweg–De Vries (KdV) Equation

The KdV equation is given by

$$u_t + uu_x + 0.0025u_{xxx} = 0,\ x \in [-1, 1],\ t \in [0, 1],$$
$$u(0, x) = \cos(\pi x), u(t, -1) = u(t, 1), u_x(t, -1) = u_x(t, 1).$$

Table 2. Comparison of our methods with state-of-the-art approaches applied to the AC equation.

Method	L_2 extrapolation error
lctanh w/o TL (ours)	0.16
lctanh w/ TL (ours)	0.12
SA-PINN [15]	0.18
w-s PINN [17]	0.14
DPM [11]	0.18

We enforce the initial condition as a hard constraint by setting

$$u(t, x) = \cos(\pi x) + tv(t, x)$$

Then u automatically satisfies the initial condition, and we only need to impose the PDE constraint and the boundary conditions on v (see supplementary material section B).

The best ABU-PINN was the combination of tanh, GELU, and sin. For lctanh and lcsin, the combination of three AFs yielded the best results, while the combination of two AFs achieved similar performance. Our findings are presented in Table 3. Overall, a linear combination of $x + \sin^2(x)$ with $n = 2$ provided the best extrapolation accuracy while introducing a similar number of trainable parameters as other linear combinations. As is the case for AC, we observe that the best adaptive linear combination is the linear combination of the single preferred AF, in this case, $x + \sin^2(x)$.

Regarding TL, we plot the collocation points with the highest PDE loss in Figure S6. We trained for 150 epochs and set the learning rate for the Adam optimizer to `5e-2`. The performance of the different AF with TL is summarized in Table 3. Our method reduced the L_2 error from 0.15 to 0.08 compared to the standard PINN model and the MAE from 0.12 to 0.06. A benchmark of our method against the state-of-the-art method of Reference [17] can be found in Table 4. The results show that our TL approach outperforms the other method. Figures S7, S8, S9, and S10 present example calculations that compare model performance with and without TL, as well as the performance of the tanh AF versus the best lc$(x + sin^2(x))$.

Table 3. Comparative analysis of L_2 and MAE for various AF applied to the KdV equation, demonstrating the effectiveness of TL and of carefully selected AF. Results for lctanh, lcsin, lc$(x + sin^2(x))$, and ABU-PINN correspond to the optimal linear combinations described in the main text.

	L_2		MAE	
AF	Without TL	With TL	Without TL	With TL
tanh	0.15	0.13	0.12	0.10
$x + sin^2(x)$	0.12	0.09	0.08	0.07
ABU-PINN	0.14	0.11	0.09	0.08
lctanh	0.14	0.12	0.11	0.09
lcsin	0.12	0.09	0.07	0.07
lc $(x + sin^2(x))$	0.10	0.08	0.07	0.06

Table 4. Comparison of our methods with state-of-the-art approaches applied to the KdV equation.

Method	L_2 extrapolation error
lc $(x + sin^2(x))$ w/o TL (ours)	0.10
lc $(x + sin^2(x))$ w/ TL (ours)	0.08
s-d PINN [17]	0.14

5.3 Viscous Burgers' Equation

The viscous Burgers' equation is given by

$$u_t + uu_x - \frac{0.01}{\pi} u_{xx} = 0, \ x \in [-1, 1], \ t \in [0, 1],$$
$$u(0, x) = -\sin(\pi x), u(t, -1) = u(t, 1) = 0.$$

We enforce the initial and boundary conditions as hard constraints by setting

$$u(t, x) = -\sin(\pi x) + t(1 - x^2)v(t, x)$$

Then u automatically satisfies the initial and boundary conditions, and we only need to impose the PDE constraint on v.
The best ABU-PINN was the combination of tanh, GELU, sigmoid, and sin. For lctanh and lcsin, the combination of three AFs yielded the best results. Our results are summarized in Table 5. The linear combination of tanh provides the best extrapolation accuracy, and we see again that the adaptive linear combination that performs best is the linear combination of the single preferred AF.

We plot the collocation points with the highest PDE loss in Figure S11. As before, these points are predominantly concentrated towards the end of the temporal validation domain and around the region with the largest spatial discrepancies. We trained for 150 epochs and set the learning rate for the Adam optimizer to `5e-2`.

The performance of the various AF with TL is summarized in Table 5. We reduced the L_2 error from 0.09 to 0.06 compared to the standard PINN model, and the MAE from 0.07 to 0.03. A benchmark of our method against the state-of-the-art methods of References [11,15] can be found in Table 6. The results show that our approaches outperform all other methods. Figures S12, S13, S14, S15 present example calculations that compare model performance with and without TL, as well as the performance of the tanh AF versus the best lctanh.

Table 5. Comparative analysis of L_2 and MAE for various AF applied to the Burgers' equation, demonstrating the effectiveness of TL and of carefully selected AF. Results for lctanh, lcsin, and ABU-PINN correspond to the optimal linear combinations described in the main text.

	L_2		MAE	
AF	Without TL	With TL	Without TL	With TL
tanh	0.09	0.08	0.07	0.06
$x + sin^2(x)$	0.08	0.08	0.04	0.03
ABU-PINN	0.11	0.09	0.04	0.03
lctanh	0.07	0.06	0.04	0.03
lcsin	0.10	0.07	0.04	0.04

Table 6. Comparison of our methods with state-of-the-art approaches applied to the Burgers' equation.

Method	L_2 extrapolation error
lctanh w/o TL (ours)	0.07
lctanh w/ TL (ours)	0.06
SA-PINN [15]	0.08
DPM [11]	0.09

6 Conclusion

In this work, we introduced a transfer learning (TL) method paired with adaptive activation functions (AF) to improve the extrapolation performance of physics-informed neural networks (PINNs). Through a series of numerical experiments, we demonstrated that our approach significantly outperforms the vanilla PINN while requiring minimal additional training cost and is also superior to three state-of-the-art methods. Our results highlight that both the choice of AF and the use of targeted TL within the validation region contribute to substantial gains. We evaluated our method on three partial differential equations (PDEs)

and achieved an average of 40% L_2 error reduction and an average of 50% mean absolute error reduction in the extrapolation domain.

We observed across all tested PDEs that the points with the highest PDE loss were found in the validation region, primarily near the domain's boundary. Selecting even a small number of such high-loss collocation points for the TL and implementing a regularization scheme to prevent catastrophic forgetting resulted in notable improvements in extrapolation performance.

Additionally, we found that an adaptive linear combination of the PDE-specific preferred AF outperformed combinations of other AF, including multiple diverse AFs. These results suggest that the key to improved extrapolation lies in the adaptability introduced by learnable AF weights and that increasing the diversity of AFs in the combination (as in ABU-PINN) is suboptimal.

However, our method was evaluated only on 1D PDEs over regular domains, and its scalability to more complex systems remains uncertain. In higher dimensions, collocation point selection becomes more critical, and enforcing boundary conditions as hard constraints on irregular domains may not be feasible. Applying our approach in such settings may require substantial adaptations and increased computational resources. Nonetheless, our method can still be applied in cases where real-world data is available, as our adaptive AF and TL approaches can still be implemented.

Acknowledgments. This work was supported by the National Institutes of Health (NIH) under grant number 5R01HG011795 and by the Cancer Prevention and Research Institute of Texas (CPRIT) under grant number RP240131. The authors also gratefully acknowledge the ICANN 2025 reviewers for their valuable feedback.

References

1. Abbasi, J., Andersen, P.: Physical activation functions: an approach for more efficient induction of physics into physics-informed neural networks. Neurocomputing **608** (2024)
2. Alkhalifah, T., Huang, X.: Direct imaging using physics informed neural networks. In: 2022 IEEE International Conference on Image Processing (ICIP), pp. 2781–2785 (2022)
3. Byrd, R.H., Lu, P., Nocedal, J., Zhu, C.: A limited memory algorithm for bound constrained optimization. SIAM J. Sci. Comput. **16**, 1190–1208 (1995)
4. Cai, S., Mao, Z., Wang, Z., Yin, M., Karniadakis, G.E.: Physics-informed neural networks for fluid mechanics: a review. Acta. Mech. Sin. **37**, 1727–1738 (2021)
5. Desai, S., Mattheakis, M., Joy, H., Protopapas, P., Roberts, S.: One-Shot Transfer Learning of Physics-Informed Neural Networks, arXiv e-prints 2110.11286v2 (2021)
6. Faroughi, S.A., Soltanmohammadi, R., Datta, P., Mahjour, S.K., Faroughi, S.: Physics-informed neural networks with periodic activation functions for solute transport in heterogeneous porous media. Mathematics **12**(1) (2024)
7. Fesser, L., D'Amico-Wong, L., Qiu, R.: Understanding and mitigating extrapolation failures in physics-informed neural networks, arXiv e-prints arxiv: 2306.09478 (2023)

8. Jagtap, A., Kawaguchi, K., Karniadakis, G.E.: Locally adaptive activation functions with slope recovery for deep and physics-informed neural networks. Proc. Royal Soc. A Math. Phys. Eng. Sci. (2020)
9. Jalili, D., Jadidi, M., Keshmiri, A., Chakraborty, B., Georgoulas, A., Mahmoudi, Y.: Transfer learning through physics-informed neural networks for bubble growth in superheated liquid domains. Int. J. Heat Mass Transfer. **232**, 125940 (2024)
10. Kapoor, T., Wang, H., Núñez, A., Dollevoet, R.: Transfer learning for improved generalizability in causal physics-informed neural networks for beam simulations. Eng. Appl. Artif. Intell. **133**, 108085 (2024)
11. Kim, J., Lee, K., Lee, D., Park, J.: DPM:a novel training method for physics-informed neural networks in extrapolation. In: 35th AAAI Conference on Artificial Intelligence, pp. 8146–8154 (2021)
12. Kirkpatrick, J., et al.: Overcoming catastrophic forgetting in neural networks. Proc. Natl. Acad. Sci. **114**, 3521–3526 (2016)
13. Kissas, G., Yang, Y., Hwuang, E., Witschey, W., Detre, J., Perdikaris, P.: Machine learning in cardiovascular flows modeling: predicting arterial blood pressure from non-invasive 4D flow MRI data using physics-informed neural networks. Comput. Methods Appl. Mech. Eng. **358**, 112623 (2020)
14. Liu, Z., Tilman, H., Masahito, U.: Neural networks fail to learn periodic functions and how to fix it. Adv. Neural Inf. Process. Syst. **33** (2020)
15. McClenny, L.D., Braga-Neto, U.M.: Self-adaptive physics-informed neural networks. J. Comput. Phys. **474**, 111722 (2023)
16. Mustajab, A.H., Lyu, H., Rizvi, Z., Wuttke, F.: Physics-informed neural networks for high-frequency and multi-scale problems using transfer learning. Appl. Sci. **14**(8), 3204 (2024)
17. Penwarden, M., Jagtap, A.D., Zhe, S., Karniadakis, G.E., Kirby, R.M.: A unified scalable framework for causal sweeping strategies for physics-informed neural networks (pinns) and their temporal decompositions. J. Comput. Phys. **493** (2023)
18. Prabhav, B., Jacques, R., Chris, M., Ankur, M., Daniel, K., Parisa, S.: Using a physics-informed neural network and fault zone acoustic monitoring to predict lab earthquakes. Nat. Commun. **14** (2023)
19. Prantikos, K., Chatzidakis, S., Tsoukalas, L.H., Heifetz, A.: Physics-informed neural network with transfer learning (TL-PINN) based on domain similarity measure for prediction of nuclear reactor transients. Sci. Rep. **13**(1), 16840 (2023)
20. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. **378**, 686–707 (2019)
21. Shima, A., Ishitsuka, K., Lin, W., Bjarkason, E.K., Suzuki, A.: Modeling unobserved geothermal structures using a physics-informed neural network with transfer learning of prior knowledge. Geotherm Energy **12**(1), 38 (2024)
22. Wang, H., Lu, L., Song, S., Huang, G.: Learning specialized activation functions for physics-informed neural networks, arXiv e-prints 2308.04073 (2023)
23. Wang, Y., Yao, Y., Gao, Z.: An extrapolation-driven network architecture for physics-informed deep learning. Neural Netw. **183**, 106998 (2025)
24. Wight, C., Zhao, J.: Solving allen-cahn and cahn-hilliard equations using the adaptive physics informed neural networks. Commun. Comput. Phys. **29**, 930–954 (2021)
25. Wu, C., Zhu, M., Tan, Q., Kartha, Y., Lu, L.: A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. Comput. Methods Appl. Mech. Eng. **403**, 115671 (2023)

26. Zapf, B., Haubner, J., Kuchta, M., Ringstad, G., Eide, P., Mardal, K.: Investigating molecular transport in the human brain from mri with physics-informed neural networks. Sci. Rep. **12** (2022)
27. Zhu, J., Xue, Y., Liu, Z.: A transfer learning enhanced physics-informed neural network for parameter identification in soft materials. Appl. Math. Mech.-Engl. Ed. **45**(10), 1685–1704 (2024)



A Spiking Central Pattern Generator Capable of Adaptive Gait Control in Quadruped Locomotion

Narumitsu Ikeda(✉)

Mitsubishi Electric Corp., 5-1-1, Ofuna, Kamakura, Kanagawa 247-8501, Japan
Ikeda.Narumitsu@dc.MitsubishiElectric.co.jp

Abstract. Periodic patterns such as locomotion and breathing are believed to be generated by neural circuits known as central pattern generators (CPGs). In this study, we construct a CPG using spiking neurons with sensory feedback. The proposed spiking CPG autonomously generates periodic rhythms and adjusts the cycle of its activity according to the intensity of the bias current, which corresponds to higher-level commands. Furthermore, it modifies its activity pattern in response to sensory input, enabling quadruped locomotion in a self-organized manner. Although the structure of the spiking CPG is simple, it can achieve adaptive gait generation by integrating feedback from the environment, command inputs from higher centers, and the spiking CPG's activity.

Keywords: spiking neural networks · neuromorphic computing · self-organization · interlimb coordination · physical interaction

1 Introduction

Locomotive behaviors such as walking patterns of legged animals are thought to be generated by neural circuits called central pattern generators (CPGs) [1]. Although CPGs can generate rhythmic patterns for locomotion in the absence of sensory input [1–4], they adaptively generate motor patterns by integrating sensory feedback from the environment [1,5,6]. CPGs have been modeled to explain the mechanisms of animal locomotion [1]. Also, as a biologically inspired control method, CPG models are expected to achieve robust motion generation with low computational cost in response to changes in the environment and robot [7].

The models of CPGs have been proposed at several levels of abstraction [5,8]. CPGs are functionally reproduced at a high-level abstraction using sine functions. At a lower level of abstraction, neural networks have been used with reciprocal inhibition [5]. Various types of neurons have been employed, including continuous-valued neurons, such as in Matsuoka's neural oscillators [9] and spiking neurons [10,11]. Spiking neural networks (SNNs) operate with ultra-low power consumption on neuromorphic hardware [12–15], making them practical particularly for applications with power constraints, such as motion generation in microrobots [16].

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 302–313, 2026.
https://doi.org/10.1007/978-3-032-04555-3_25

CPGs using spiking neurons are referred to as spiking CPGs (SCPGs). The researchers have created various SCPGs to generate locomotion for lamprey-like [8,17], quadruped [18], and six-legged robots [19–23]. However, these SCPGs do not incorporate sensory feedback from the environment. In the previous works, interlimb coordination in gait patterns was predefined by adjusting the parameters of the SCPGs, making it difficult to adaptively generate and transition gait patterns. On the other hand, it has been shown that gait patterns such as walking and running can be self-organized through the interaction between the physical body and CPGs with respect to a speed control input [24,25]. If sensory feedback can be appropriately integrated into SCPGs, they should be able to achieve adaptive gait patterns as well.

In this study, we present an SCPG architecture that incorporates sensory feedback and can adaptively generate gaits of a quadruped robot. We form an SCPG by mutually inhibiting two leaky integrate-and-fire neurons with adaptive thresholds and use four to build distributed decoupled SCPGs. Each SCPG receives feedback from the environment as ground reaction force (GRF) on the corresponding leg and a non-rhythmic bias current, which functions as a command from higher centers. Changing the intensity of the bias current alters the cycle duration of SCPGs' activity. Although there are no connections between the SCPGs, similar to the findings of Owaki et al. [24,25], the activity of the SCPGs is regulated through physical interaction via GRF feedback. The results of this study suggest that simple-structured SCPGs suffice adaptive gait generation through the integration of feedback from the environment, command input from higher centers, and SCPGs' activity.

2 Methods

2.1 Central Pattern Generator Model

We used two LIF neurons with adaptive threshold to construct an SCPG. The membrane potential v and the firing threshold v_{th} change as follows:

$$\mathrm{C_{mem}}\frac{\mathrm{d}v}{\mathrm{d}t} = \mathrm{g_{leak}}(\mathrm{E_{rest}} - v) + \mathrm{I_e} + \sigma_{\mathrm{GRF}} N_{GRF}, \tag{1}$$

$$\tau_{\mathrm{th}}\frac{\mathrm{d}v_{th}}{\mathrm{d}t} = \mathrm{E_{th}} - v_{th}. \tag{2}$$

The parameters are listed in Table 1. $\mathrm{I_e}$ is the non-rhythmic bias current, which works as the high level instruction. N_{GRF} is the GRF. When the membrane potential v exceeds the firing threshold v_{th}, the neuron fires and the membrane potential is reset to -58 mV. Also, the firing threshold is appended by 0.5 mV. The two neurons numbered 0 and 1 have reciprocal inhibition as shown in Fig. 1A.

When one of the two neurons fires, the firing neuron inhibits the other decreasing the membrane potential by 15 mV. Note that only the 0th neuron receives the sensory feedback of GRF, meaning σ_{GRF} for the 1st neuron is set to 0 pA/N.

Table 1. Parameters of the SCPG

Parameter name	Symbol	Value	Unit
Membrane capacitance	C_{mem}	200	pF
Leakage conductance	g_{leak}	12	nS
Resting potential	E_{rest}	−70	mV
Magnitude of sensory feedback	σ_{GRF}	0.4	pA/N
Time constant for adaptive threshold	τ_{th}	100	ms
Initial threshold	E_{th}	−50	mV

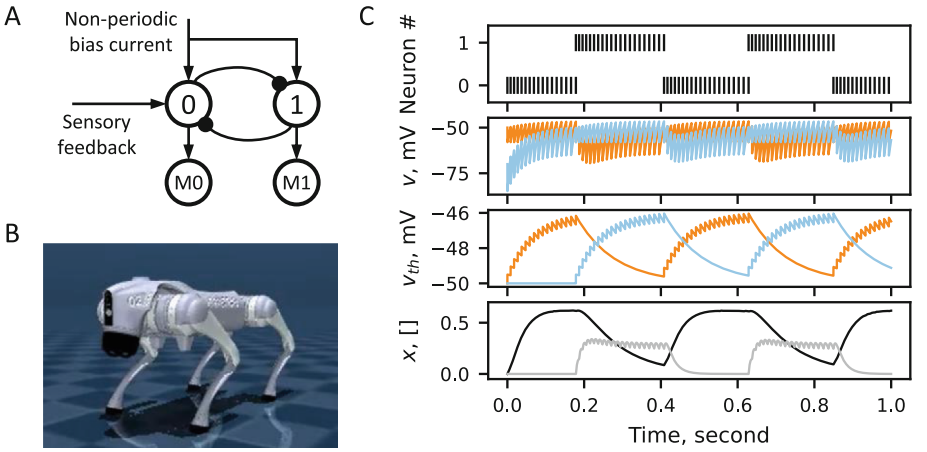


Fig. 1. A) The structure of the SCPG. The two neurons labeled 0 and 1 constitute an SCPG with reciprocal inhibition. The SCPG receives non-periodic bias current and sensory feedback. Each neuron in the SCPG excites a corresponding motor neuron. B) The quadruped robot used in this study. One SCPG controls the two joints of a limb: hip and knee joints. In total, four SCPGs generate quadruped locomotion. C) A representative activity of the isolated SCPG and motor neurons' outputs. The 1st, 2nd, and 3rd rows show the raster plot, membrane potentials, and firing thresholds of the SCPG neurons. The orange and light blue lines correspond to neurons 0 and 1, respectively. The last row shows the traces of the two motor neurons' outputs. The black and grey lines correspond to neurons M0 and M1, respectively. (Color figure online)

The two neurons in the SCPG send spikes to the corresponding motor neurons labeled as M0 and M1, respectively (Fig. 1A). Each motor neuron functions as a low-pass filter and converts the spike train to a continuous value x as follows:

$$\tau_a \frac{dx}{dt} = g_{motor}(1 - x), \tag{3}$$

$$\tau_b \frac{dg_{motor}}{dt} = -g_{motor}, \tag{4}$$

where τ_{a} and τ_{b} are time constants, and g_{motor} is a nondimensionalized conductance. Note that x is nondimensional. When the motor neuron receives a spike, the conductance g_{motor} is increased by 1. The time constants τ_{a} and τ_{b} were set to 100 ms and 20 ms for the neuron labeled with M0, 20 ms and 5 ms for M1. The two continuous values are used to calculate the target angles of the hip and knee joints, as in the next subsection.

2.2 Robot Control and Simulation

We used MuJoCo [26], MuJoCo Menagerie [27], and Cleo [28] for a quadruped robot simulation. From the collection of the quadruped robots in the MuJoCo Menagerie, we chose the model of the Unitree Go2 robot (Fig. 1B). We adhered to the definition of joint angles as specified in the model.

To achieve quadruped locomotion, we built four SCPGs using Brian2 [29]. The SCPGs are decoupled and distributed, as in the previous works [24,25]. Each SCPG receives GRF of a corresponding limb. The two motor neurons M0 and M1 give the target angles of hip and knee joints with the following equations:

$$\overline{\theta}_{hip} = \phi_{\mathrm{hip},0} + \Delta\phi_{\mathrm{hip}}\, x_{M0}, \tag{5}$$

$$\overline{\theta}_{knee} = \phi_{\mathrm{knee},0} + \Delta\phi_{\mathrm{knee}}\, x_{M1}. \tag{6}$$

The parameters $\phi_{\mathrm{hip},0}$, $\phi_{\mathrm{knee},0}$, $\Delta\phi_{\mathrm{hip}}$, and $\Delta\phi_{\mathrm{knee}}$ were set to 0.5, -1.75, 1.0, and -1.3 in radian, respectively. The corresponding hip and knee joints are controlled by proportional-derivative control with the values $\overline{\theta}_{hip}$ and $\overline{\theta}_{knee}$ as the target angles. Although controlling joint angles can also be achieved using SNNs, this study does not focus on it.

The neurons 0 and 1 correspond to the stance and swing phases, respectively. When the 0th neuron continues firing, the M0 neuron outputs a high value, while the M1 neuron outputs a low value close to 0. Such values correspond to the hip and knee angles where the limb extends to the ground, supporting the quadruped robot. When the 1st neuron continues firing, the M0 neuron outputs a low value, and the M1 neuron outputs a high value. Such values correspond to the angles where the limb folds and moves forward. Therefore, modulating the SCPGs' activity results in the adjustment of gait patterns.

3 Experiments and Results

3.1 Running the Isolated SCPG

We first conducted a fundamental analysis of the SCPG. Without the sensory input of GRF, the SCPG generated oscillating activity, as shown in Fig. 1C. Note that we initialized the membrane potentials of the neurons 0 and 1 to E_{th} and $\mathrm{E}_{\mathrm{rest}}$ to generate oscillatory behavior. The raster plot shows that spikes are denser at the beginning of each neuron's burst firing compared to the later part of the burst, indicating that the firing rate is higher at the beginning and lower towards the end of each burst. As the firing threshold increased, the firing rate

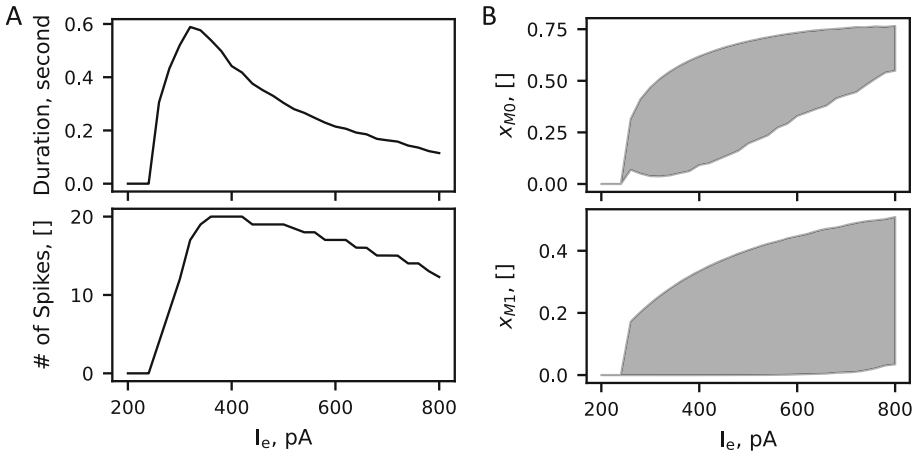


Fig. 2. A) Cycle duration and the number of spikes per burst as functions of the bias current intensity. B) Minimum and maximum values of the two motor neurons to the intensity of the bias current.

became low, weakening the inhibition against the other neuron. And the other neuron eventually started its bursting activity with the inhibition weakened. This oscillatory activity is typical of SNNs with spike-frequency adaptation [2,17,19].

As shown in Fig. 1C, the motor neurons converted the spike trains of the two neurons in the SCPG to two continuous values. M0's trace changed relatively slowly to M1's because of the difference between their time constants (Table. 1). M0's trace suites the forward and backward movement of the limb for the hip joint, while M1's trace flexion and extension for the knee joint.

Figure 2A shows that the duration of the oscillatory activity changes with the magnitude of the bias current to the two neurons in the SCPG.

This feature is preferable for generating locomotion with different gait cycles. The number of spikes in each neuron's bursting also differs with the bias current (Fig. 2A). Because the changes in the cycle duration and the number of spikes are not proportional, the firing rate of each neuron during bursting differs with the bias current, resulting in differences in the minimum and maximum values of x_{M0} and x_{M1} (Fig. 2B).

Next, we applied static pseudo-GRF as sensory inputs to the SCPG. Increasing the magnitude of the static pseudo-GRF trace prolonged the cycle duration (Fig. 3A) before the SCPG's state reached a non-oscillatory regime (Fig. 2B).

These two regimes, in which the SCPG exhibits periodic and non-periodic activities, correspond to the oscillatory and excitatory behaviors in the active rotator model [30]. With these two regimes, the SCPG can continue the stance phase if the GRF is sufficiently high when the limb should support the body and otherwise transition to the swing phase.

The threshold of the GRF that divides the two regimes differs with the intensity of the bias current (Fig. 3B). This feature is consistent with the varying

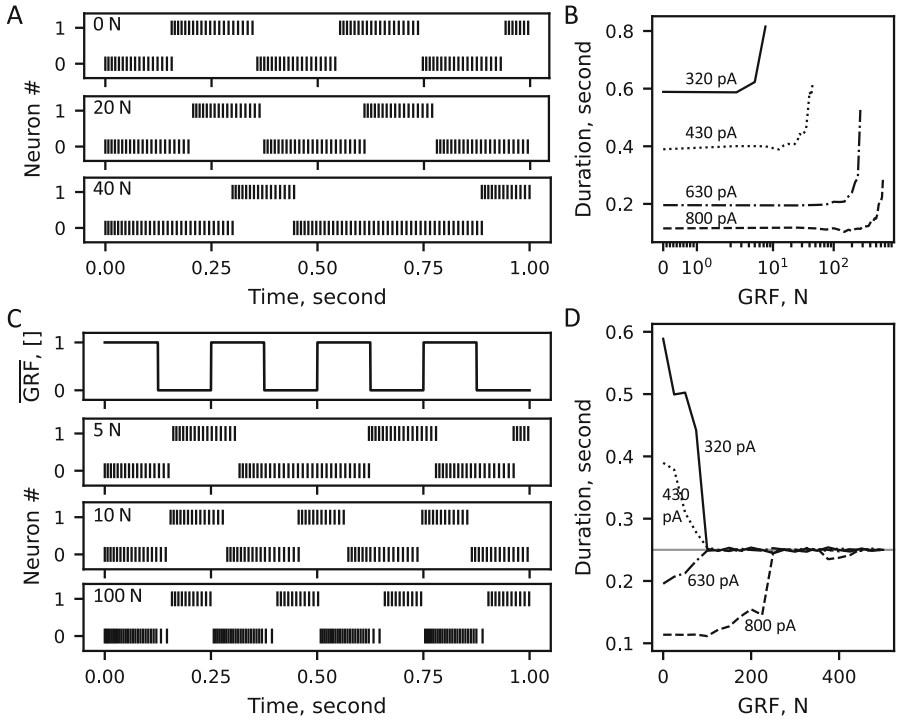


Fig. 3. A) Activities of the SCPG under static pseudo-GRFs of various magnitudes. B) Cycle durations of the SCPGs with various intensities of the bias current relative to the magnitude of static pseudo-GRF. The solid, dotted, dash-dotted, and dashed lines correspond to the intensity of the bias current of 320, 430, 630, and 800 pA, respectively. C) Activities of the SCPG under periodic pseudo-GRFs of various magnitudes. The top row shows the shape of the periodic pseudo-GRFs. The 2nd, 3rd, and 4th rows display the activities of the SCPGs. D) Cycle durations of the SCPGs relative to various magnitudes of the periodic pseudo-GRFs. Each line corresponds to the intensity of the bias current as in B.

threshold of the active rotator model with the constant driving force ω as in [24,25,30].

Figure 3C and D show that the SCPG can synchronize to the biphasic pseudo-GRF trace with a cycle duration of 0.25 s when the magnitude of the GRF is sufficiently high. This phenomenon implies the SCPG's potential to adapt its activity to, for example, the body movement of quadruped robots through sensory feedback.

3.2 Quadruped Gait Generation with the SCPGs

We conducted a quadruped gait generation simulation with the four decoupled SCPGs to control the four limbs: left fore (LF), left hind (LH), right fore (RF),

and right hind (RH) limbs. After initializing the environment, we ran the simulation for two seconds to stabilize the quadruped robot. We set the target angles of all the hip joints to 0.5 rad, the knee joints of the three legs (LF, RF, RH), and the LH leg to −1.75 and −1.9 rad, respectively. Note that we chose different target angles for the knee joints to distribute GRF to each leg unevenly. Then, we turned on the SCPGs to generate robot locomotion with the bias current of the SCPGs set to 430 pA for 6 s. Finally, we increased the bias current to 630 pA and simulated gait generation for 4 s. Figure 4A summarizes the timeline of the simulation.

Figure 4B and C show each limb's joint angles and GRF traces, respectively.

During the stabilization phase, each leg received a GRF of different magnitude due to the differences in knee angles. Immediately after the stabilization phase, the quadruped robot demonstrated a stable lateral-sequence (L-S) walk, where the stance phase of each leg transitions in the order of RH, RF, LH, and LF legs. This result indicates that the SCPGs instantly self-organized the interlimb coordination and generated stable locomotion. Furthermore, the robot exhibited a gait transition from L-S walk to bound right after 8 s when the bias current was increased. This type of gait transition corresponds to previous works [24,25].

We also conducted gait stability analysis with six initial knee angle conditions (Fig. 5A and B). The condition >LH is the same as in Fig. 4. Under the two

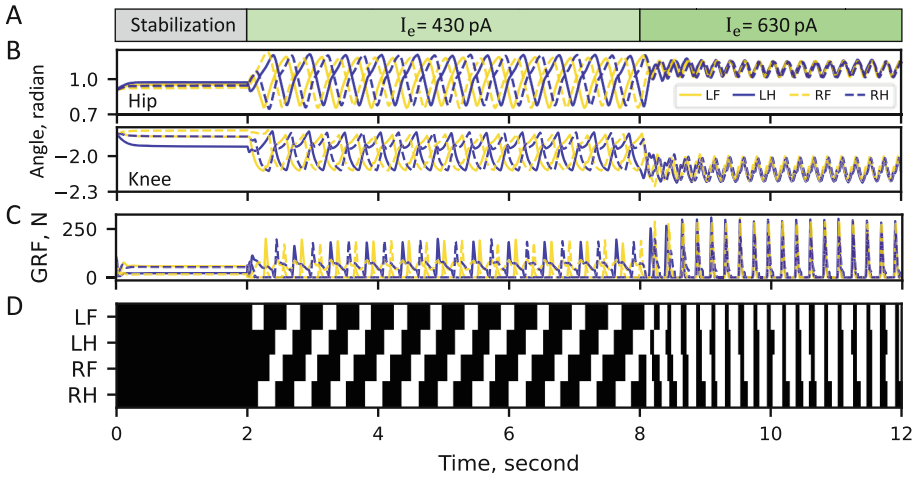


Fig. 4. A) The timeline of the simulation. The interval from 0 to 2 s is the stabilization phase. At 2 s, the SCPGs were activated with the non-periodic bias current I_e set to 430 pA. I_e was increased at 8 s. B) Angle traces of each limb's hip and knee joints. Each line corresponds to a limb (LF: solid yellow, LH: solid dark blue, RF: dashed yellow, and RH: dashed dark blue). C) Traces of GRFs to the limbs. The lines represent GRFs to the limbs as in A. D) Gait diagram of the quadruped robot. Black regions represent the stance phase where GRF on the corresponding limbs is greater than 0 N. (Color figure online)

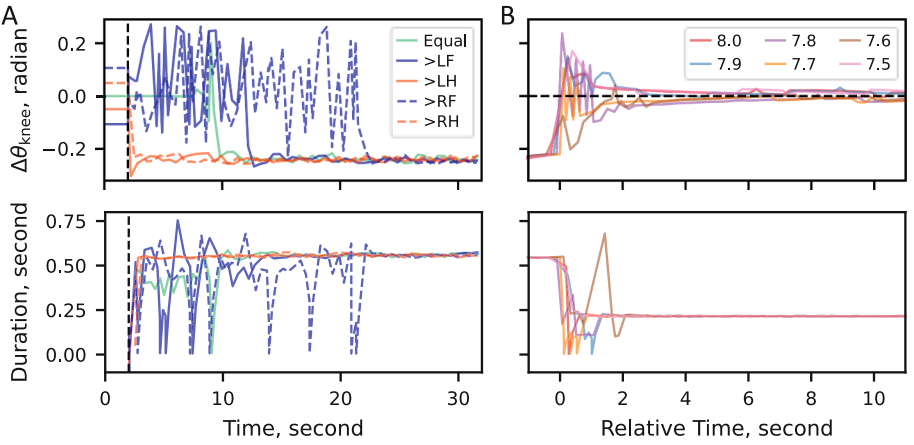


Fig. 5. A) Knee angle difference between the two forelimbs (LF - RF) at the touch-downs of the LF limb. Each line corresponds to a different initial knee angle condition: either all knee angles are −1.75 (Equal), or one knee angle is −1.90 while the others are −1.75 (>LF, >LH, >RF, >RH) in radians. B) Duration between the consecutive touch-downs of the LF limb. C) and D) Knee angle difference and duration as shown in A and B under different timings of the increase in the bias current, respectively. The condition 8.0 is the same as in Fig. 4. Each line corresponds to a different timing in seconds. The horizontal axis indicates the relative time from the timing of the increase.

conditions (>LH and >RH), the knee angle difference and the duration immediately converged. Although the robot exhibited transient unstable gaits under the other conditions, it reached a stable L-S walk, taking up to 22 s.

Furthermore, we analyzed the time it took to transition to a stable bound by varying the timing of the increase in the bias current. The differences in timing correspond to different states of the robot's joint angles. Regardless of the initial state, the gait transitioned from walking to a stable bound within 2 s at the latest. These results indicate that in the dynamics of SCPGs, the robot, and the environment, both the L-S walk and bound exist as stable states, emerging according to the magnitude of the bias current. Additionally, they demonstrate that immediate transitions to L-S walk or bound can be achieved from specific initial states.

4 Discussion

The isolated SCPG demonstrated rhythmic activities with different cycle durations, respective to the different magnitudes of the bias current (Fig. 1C). It also exhibited oscillatory and excitatory behaviors depending on the magnitude of sensory input (Fig. 2A). Furthermore, it was observed that the SCPG's activity synchronized to periodical sensory input with a sufficient magnitude (Fig. 3C and D). These features can be thought to enable self-organizing quadruped gait generation and transition (Fig. 4D and 5).

The proposed SCPG can be considered a spiking implementation of the active rotator model [24,25,30]. The active rotator model has two inputs as follows:

$$\frac{\mathrm{d}\phi}{\mathrm{d}t} = \omega - b\sin\phi, \tag{7}$$

where ϕ is the oscillator's phase, and the first and second terms are the constant driving force and the phase-dependent input, respectively. The model shows oscillatory behavior when $\omega > b$, whereas when $\omega < b$, it has stable and unstable fixed points showing excitatory behavior where the oscillator phase remains close to the stable fixed point. The bias current $\mathrm{I_e}$ of the proposed SCPG corresponds to the first term. The switching activity of the two neurons in the SCPG and the sensory feedback is compatible with the second term. Depending on the balance between $\mathrm{I_e}$ and N_{GRF}, the SCPG exhibits oscillatory and excitatory behaviors (Fig. 3B). This feature is consistent with that of the active rotator model. The key to exhibiting the two distinct behaviors is the structure of the sensory feedback, where the external input prolongs the 0th neuron's activity depending on its intensity.

The structure of the SCPG was based on the half-center model [2,31]. Although several works realized half-center models with spiking neurons, such works did not consider sensory feedback [17,19,32]. To our knowledge, this is the first SCPG that integrates sensory input from the environment as well as instructive input corresponding to a command from higher centers into its activity to generate adaptive quadruped locomotion. The phenomenological difference is that the models in the previous works only utilize the oscillatory behavior of the SCPGs. In contrast, the proposed SCPGs utilize oscillatory and excitatory ones to self-organize interlimb coordination, making it possible to instantly generate the stable L-S walk and transition to the bound (Fig. 4D).

Owaki et al. demonstrated that distributed decoupled CPGs could adaptively generate quadruped locomotion through physical communication via local sensory feedback to each CPG [24,25]. They designed the CPG model for a corresponding limb to remain in the stance phase while supporting the body; in other words, when its GRF feedback is sufficiently strong. The structure of the proposed SCPG is consistent with their design, where the sensory feedback prolongs the 0th neuron's activity which corresponds to the stance phase depending on the intensity of the feedback(Fig. 1A).

The gait transition occurred with the increase in the bias current, which corresponds to the increase in the oscillator frequency (Fig. 4D). Such gait transition was observed in the previous works [25,33,34]. Immediately after increasing the bias current, the robot is expected to lose balance temporarily due to the generation of leg movements different from the previous gait pattern. The increase in the GRF of the LH limb suggests this loss of balance. Here, although the stance phase of the LH limb was maintained due to the increase in GRF, the robot experienced a moment caused by the GRF and gravity, leading the LH limb to transition to the swing phase as the other three limbs entered the stance phase. Subsequently, the synchronous movement of the other three legs generates a significant upward force, causing the robot to leap. The GRF patterns generated

by the leap and landing were fed back to each SCPG, synchronizing each SCPG with the rhythm of the leap and landing. This mechanism is thought to have enabled the robot's gait to transition from walking to bounding. The characteristic of the SCPG synchronizing with external input (Fig. 3D) is considered to facilitate adaptive gait generation in response to external input.

Figure 5 indicates that initiating and transitioning gaits from specific postures is necessary to achieve immediate gait generation and transitions. Investigating which postures are appropriate for initiating and transitioning gaits is a subject for future work. In particular, experiments with actual robots are desirable to analyze the stability of gait generation and transitions across various terrains.

5 Conclusion

In this study, we designed an SCPG with sensory feedback from the environment and analyzed its features. The SCPG demonstrated rhythmic activities with different cycle durations according to the different magnitudes of the bias current, which corresponds to high-level commands. It also exhibited oscillatory and excitatory behaviors depending on the magnitude of sensory input, where the SCPG autonomously generates periodic activity and remains in its phase, respectively. Furthermore, it was observed that the SCPG's activity synchronized to periodical sensory input with a sufficient magnitude. These features of the SCPG enabled self-organizing quadruped gait generation and transition with the distributed decoupled SCPGs. The results of this study suggest that simple-structured SCPGs suffice adaptive gait generation through the integration of sensory feedback from the environment, command input from higher centers, and SCPGs' activity.

Acknowledgments. The author would like to thank Dr. Akio Ishiguro and Dr. Shura Suzuki at Tohoku University for their valuable insights and comments on the manuscript.

Disclosure of Interests. The author has no competing interests to declare that are relevant to the content of this article.

References

1. Grillner, S., El Manira, A.: Current principles of motor control, with special reference to vertebrate locomotion. Physiol. Rev. **100**(1), 271–320 (2019)
2. Marder, E., Bucher, D.: Central pattern generators and the control of rhythmic movements. Curr. Biol. **11**(23), R986–R996 (2001)
3. Selverston, A.I.: Invertebrate central pattern generator circuits. Philos. Trans. R. Soc. B Biol. Sci. **365**(1551), 2329–2345 (2010)
4. Kiehn, O.: Decoding the organization of spinal circuits that control locomotion. Nat. Rev. Neurosci. **17**(4), 224–238 (2016)

5. Yu, J., Tan, M., Chen, J., Zhang, J.: A survey on CPG-inspired control models and system implementation. IEEE Trans. Neural Netw. Learn. Syst. **25**(3), 441–456 (2013)
6. Bidaye, S.S., Bockemuhl, T., Buschges, A.: Six-legged walking in insects: how CPGs, peripheral feedback, and descending signals generate coordinated and adaptive motor rhythms. J. Neurophysiol. **119**(2), 459–475 (2018)
7. Sun, T., Dai, Z., Manoonpong, P.: Robust and reusable self-organized locomotion of legged robots under adaptive physical and neural communications. Front. Neural Circ. **17**, 1111285 (2023)
8. Angelidis, E., et al.: A spiking central pattern generator for the control of a simulated lamprey robot running on spinnaker and Loihi neuromorphic boards. Neuromorph. Computi. Eng. **1**(1), 014005 (2021)
9. Matsuoka, K.: Sustained oscillations generated by mutually inhibiting neurons with adaptation. Biol. Cybern. **52**(6), 367–376 (1985)
10. Gerstner, W., Kistler, W.M.: Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge University Press (2002)
11. Izhikevich, E.M.: Which model to use for cortical spiking neurons? IEEE Trans. Neural Netw. **15**(5), 1063–1070 (2004)
12. Davies, M., et al.: Advancing neuromorphic computing with Loihi: a survey of results and outlook. Proc. IEEE **109**(5), 911–934 (2021)
13. Akopyan, F., et al.: TrueNorth: design and tool flow of a 65 mW 1 Million neuron programmable neurosynaptic chip. IEEE Trans. Comput. Aided Des. Integr. Circ. Syst. **34**(10), 1537–1557 (2015)
14. Rajendran, B., Sebastian, A., Schmuker, M., Srinivasa, N., Eleftheriou, E.: Low-power neuromorphic hardware for signal processing applications: a review of architectural and system-level design approaches. IEEE Sig. Process. Mag. **36**(6), 97–110 (2019)
15. Young, A.R., Dean, M.E., Plank, J.S., Rose, G.S.: A review of spiking neuromorphic hardware communication systems. IEEE Access **7**, 135606–135620 (2019)
16. Clawson, T.S., Ferrari, S., Fuller, S.B., Wood, R.J.: Spiking neural network (SNN) control of a flapping insect-scale robot. In: 2016 IEEE 55th Conference on Decision and Control (CDC), pp. 3381–3388. IEEE (2016)
17. Donati, E., Corradi, F., Stefanini, C., Indiveri, G.: A spiking implementation of the lamprey's central pattern generator in neuromorphic VLSI. In: 2014 IEEE Biomedical Circuits and Systems Conference (BioCAS) Proceedings, pp. 512–515. IEEE (2014)
18. Espinal, A., et al.: Quadrupedal robot locomotion: a biologically inspired approach and its hardware implementation. Comput. Intell. Neurosci. **2016**(1), 5615618 (2016)
19. Strohmer, B., Manoonpong, P., Larsen, L.B.: Flexible spiking CPGs for online manipulation during hexapod walking. Front. Neurorobot. **14**, 41 (2020)
20. Rostro-Gonzalez, H., et al.: A CPG system based on spiking neurons for hexapod robot locomotion. Neurocomputing **170**, 47–54 (2015)
21. Cuevas-Arteaga, B., et al.: A spinnaker application: design, implementation and validation of SCPGS. In: Advances in Computational Intelligence: 14th International Work-Conference on Artificial Neural Networks, IWANN 2017, Cadiz, Spain, June 14–16, 2017, Proceedings, Part I 14, pp. 548–559. Springer (2017)
22. Espinal, A., Rostro-Gonzalez, H., Carpio, M., Guerra-Hernandez, E.I., Ornelas-Rodriguez, M., Sotelo-Figueroa, M.: Design of spiking central pattern generators for multiple locomotion gaits in hexapod robots by Christiansen grammar evolution. Front. Neurorobot. **10**, 6 (2016)

23. Gutierrez-Galan, D., Dominguez-Morales, J.P., Perez-Peña, F., Jimenez-Fernandez, A., Linares-Barranco, A.: Neuropod: a real-time neuromorphic spiking CPG applied to robotics. Neurocomputing **381**, 10–19 (2020)
24. Owaki, D., Kano, T., Nagasawa, K., Tero, A., Ishiguro, A.: Simple robot suggests physical interlimb communication is essential for quadruped walking. J. R. Soc. Interface **10**(78), 20120669 (2013)
25. Owaki, D., Ishiguro, A.: A quadruped robot exhibiting spontaneous gait transitions from walking to trotting to galloping. Sci. Rep. **7**(1), 277 (2017)
26. Todorov, E., Erez, T., Tassa, Y.: MuJoCo: a physics engine for model-based control. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5026–5033. IEEE (2012). https://doi.org/10.1109/IROS.2012.6386109
27. Zakka, K., Tassa, Y., Contributors, M.M.: MuJoCo menagerie: a collection of high-quality simulation models for MuJoCo (2022). http://github.com/google-deepmind/mujoco_menagerie
28. Johnsen, K.A., Cruzado, N.A., Willats, A.A., Rozell, C.J.: Cleo: a testbed for bridging model and experiment by simulating closed-loop stimulation, electrode recording, and optogenetics. bioRxiv (2023)
29. Stimberg, M., Brette, R., Goodman, D.F.: Brian 2, an intuitive and efficient neural simulator. elife **8**, e47314 (2019)
30. Shinomoto, S., Kuramoto, Y.: Phase transitions in active rotator systems. Progress Theoret. Phys. **75**(5), 1105–1110 (1986)
31. Brown, T.G.: On the nature of the fundamental activity of the nervous centres; together with an analysis of the conditioning of rhythmic activity in progression, and a theory of the evolution of function in the nervous system. J. Physiol. **48**(1), 18 (1914)
32. Russell, A., Orchard, G., Etienne-Cummings, R.: Configuring of spiking central pattern generator networks for bipedal walking using genetic algorithms. In: 2007 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1525–1528. IEEE (2007)
33. Aoi, S., Yamashita, T., Tsuchiya, K.: Hysteresis in the gait transition of a quadruped investigated using simple body mechanical and oscillator network models. Phys. Rev. E-Stat. Nonlinear Soft Matter Phys. **83**(6), 061909 (2011)
34. Aoi, S., et al.: A stability-based mechanism for hysteresis in the walk-trot transition in quadruped locomotion. J. R. Soc. Interface **10**(81), 20120908 (2013)



ViSMoE: Visual-Aware Sparse Mixture-of-Experts for Embodied Referring Expression Grounding

Shuo Feng[1,2,3] and Piji Li[1,2,3](✉)

[1] College of Artificial Intelligence, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
{fengshuo,pjli}@nuaa.edu.cn
[2] MIIT Key Laboratory of Pattern Analysis and Machine Intelligence, Nanjing 211106, China
[3] The Key Laboratory of Brain-Machine Intelligence Technology, Ministry of Education, Nanjing 211106, China

Abstract. Embodied Referring Expression Grounding is the task of enabling an agent to navigate in real environments and to localize a remote object based on natural language instructions. In this scenario, the agent needs to select one view for navigation at each step and identify a specific object among all candidate objects at the destination. However, most of the previous approaches fail to distinguish between views and objects, instead processing them using the vanilla vision encoder, which results in ambiguous representations of both views and objects. To address the above issues, we propose ViSMoE, which equips sparse Mixture-of-Experts with a visual-aware routing policy for the embodied agent. This framework processes different types of visual information specifically, resulting in discriminative visual representations for both views and objects. Experimental results on REVERIE and SOON datasets demonstrate that ViSMoE outperforms the previous state-of-the-art methods, showing the superiority of our proposed method.

Keywords: Embodied Referring Expression Grounding · Mixture-of-Experts · Views and objects

1 Introduction

Developing robots that can execute tasks based on natural language instructions has been a longstanding objective in AI and robotics. In this field, Vision-and-Language Navigation (VLN) [1] has recently attracted increasing attention from many researchers due to its broad real-world applications.

In VLN, navigation instructions are mainly divided into two types, *i.e.*, step-by-step instructions such as R2R [1] and goal-oriented instructions such as REVERIE [15]. Compared to detailed step-by-step instructions, high-level goal-oriented instructions are more prevalent in practical applications and present greater technological challenges. In this paper, we address the task of Embodied Referring Expression Grounding, specifically focusing on REVERIE [15] and

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 314–325, 2026.
https://doi.org/10.1007/978-3-032-04555-3_26

SOON [24]. In this task, the agent selects a discrete view to determine the next navigation step and identifies the target object among all candidate objects present at the destination, based on the provided instruction. Therefore, it is crucial to establish a comprehensive framework that effectively models both views and objects, addressing the dual challenges of language-guided navigation and referring expression grounding within Embodied Referring Expression Grounding.



Fig. 1. Illustration of navigation instruction, panorama, discrete views and candidate objects in SOON dataset. Discrete views provide a depiction of the scene layout, while candidate objects offer detailed information about specific items.

Previous approaches [4,21] treat discrete views and candidate objects indistinguishably by concatenating their respective embeddings to form a unified visual input. The visual input is then fed directly into the panorama encoder [3] and cross-model encoder to derive the contextual representations. Nevertheless, as shown in Fig. 1, though both discrete views and candidate objects belong to the visual input, there exist fundamental distinctions between them. Discrete views offer comprehensive overviews of the room layout, capturing the spatial relationships and broader environmental context, whereas candidate objects focus narrowly on detailed information about specific items within the scene. Moreover, discrete views are selected sequentially for the subtask of language-guided navigation, while candidate objects are identified upon reaching the destination for the subtask of referring expression grounding. In brief, discrete views

and candidate objects not only exhibit differences in their data distribution but are also tied to distinct tasks, reflecting a multi-input and multi-task challenge. Therefore, directly employing the vanilla encoder to process both types of inputs simultaneously introduces significant challenges.

To address the aforementioned issues, we propose a **Vi**sual-aware **S**parse **M**ixture-**o**f-**E**xperts (ViSMoE) framework, which integrates sparse Mixture-of-Experts (SMoE) [9] with the visual-aware routing policy. The ViSMoE incorporates multiple experts within its architecture, where each expert is specialized for a specific task or feature subspace, thus effectively addressing the challenges of handling multi-input and multi-task. In addition, we introduce the visual-aware routing policy to further enhance the specialization of experts for managing diverse visual inputs. Specifically, we employ learnable type embeddings for both discrete views and candidate objects. These type embeddings are added to the embeddings of views and objects to serve as the input for the gated network, which enables our proposed ViSMoE to consider the distributional difference between views and objects, capturing more distribution-specific information.

We summarize our key contributions as follows:

- We propose ViSMoE, which disentangles diverse task-specific visual features within the SMoE layer, tackling the challenges presented by heterogeneous inputs.
- We further introduce the visual-aware routing policy to enhance the specialization of experts in processing diverse visual inputs.
- Extensive results on REVERIE and SOON demonstrate the effectiveness of our framework, highlighting the benefits of using MoE for processing visual features in embodied AI.

2 Related Work

Embodied Referring Expression Grounding requires the agent to find an optimal path to the target by sequentially selecting the next viewpoint and to identify a specific object at the destination. It is critical to model the current observations as they are directly tied to navigation decisions and object grounding. One line of work [11,14] focuses on incorporating knowledge into view representation to enhance vision-language alignment. Additionally, [14] leverages corresponding object features to enrich the semantics of view representations. Previous methods focus on enhancing visual representations for views through new representation fusion techniques, with limited attention to object modeling. In contrast, our work introduces a novel Visual-Aware SMoE to better capture distribution-specific features for both views and objects, treating them with equal importance.

Mixture-of-Experts (MoE) has been widely explored across various domains [10,22], achieving significant advancements due to its divide-and-conquer strategy, which employs a series of specialization experts. Subsequent works [17] propose a sparsely activated method for utilizing experts, leading to the development of sparse MoE (SMoE), which reduces the cost at the training and inference

stage while maintaining effectiveness. Several recent studies have applied SMoE in multi-task scenarios. Specifically, [5,7] utilize task-specific router networks to activate relevant model components for each task, achieving excellent experimental results on multiple NLU tasks [20] and visual dense prediction tasks [12]. Besides, SMoE has demonstrated significant success in addressing the distributional difference in multi-input scenarios. For instance, EVE [2] utilize a shared transformer to encode the text and image input. Compared to them, the unified task of navigation and object grounding in Embodied Referring Expression Grounding satisfies both the multi-input and multi-task settings. We tackle this unified task using a method called ViSMoE, which equips the SMoE with the visual-aware router.

3 Method

3.1 Task Formulation

In Embodied Referring Expression Grounding [15,24], given a high-level natural language instruction $\mathcal{W} = \{w_1, w_2, \cdots, w_L\}$, where w_i denotes the i^{th} token and L is the length of the instruction, the agent needs to navigate through an indoor environment and localize a remote object at the destination. The indoor environment is represented as an undirected connectivity graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents the navigable nodes and $\mathcal{E}$ indicates the connectivity between them. At the time step t, the agent perceives a panoramic view $\mathcal{V}_t = \{v_{t,i}\}_{i=1}^{n}$. The agent decides an action a_t from a special "stop" token and the navigable viewpoints $\mathcal{N}(\mathcal{V}_t) = \{v_{t,i}\}_{i=1}^{K}$, where $\mathcal{N}(\mathcal{V}_t) \subseteq \mathcal{V}_t$. To enhance detailed visual perception and mitigate the challenges of direct object detection from the panorama [4], candidate object features $\mathcal{O}_t = \{o_{t,i}\}_{i=1}^{m}$ are extracted using annotated object bounding boxes [15] or object detectors. When the agent decides to stop at a location, it needs to select an object from the candidate objects.

3.2 Base Model

Inspired by [4], we design our model with the following key components, which employs dual-scale reasoning by leveraging both local and global observations.

Language Encoder. Each word embedding w_i is added with its corresponding position embedding in the sentence. The resulting embeddings are then fed into a multi-layer transformer [19] to get the contextual language representations $\hat{\mathcal{W}}$.

Vision Encoder. Discrete views $\mathcal{V}_t$ and candidate objects $\mathcal{O}_t$ are added their corresponding relative angle features a_t, bounding box features b_t and navigation type embeddings n_t, which can be described as:

$$\mathcal{V}'_t = \mathrm{LN}(\mathrm{W}_1^{\mathcal{V}}\mathcal{V}_t) + \mathrm{LN}(\mathrm{W}_2[a_t^{\mathcal{V}}; b_t^{\mathcal{V}}]) + n_t^{\mathcal{V}} \tag{1}$$

where the LN denotes layer normalization, $\mathrm{W}_1^{\mathcal{V}}$ and W_2 are learnable parameters, we can derive the expression for $\mathcal{O}'_t$ by replacing $\mathcal{V}$ with $\mathcal{O}$ in Eq. 1, $n_t^{\mathcal{V}}$ incudes

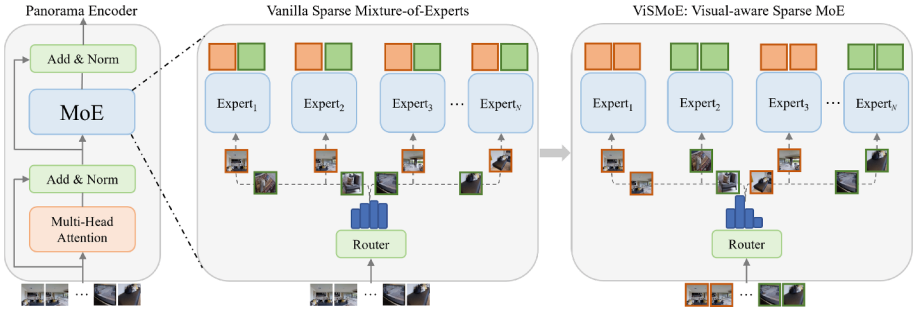


Fig. 2. Overview of ViSMoE: The dense FFN in the vision encoder is replaced by SMoE layer. We use brown boxes to denote scene images and green boxes to represent object images. Unlike vanilla SMoE, which ignores image categories, ViSMoE's routing input includes them, allowing the trained router to select the appropriate expert based on both the input token and its visual type. Therefore, it can be seen that while the vanilla MoE router balances the distribution of input tokens across experts, ViSMoE learns to route visual tokens to specific experts (e.g., expert-1 and expert-3 for view images, expert-2 and expert-N for object images). (Color figure online)

two type of embeddings: one for candidate views and another for non-candidate view. Similarly, $n_t^{\mathcal{O}}$ represents the type embedding for objects. A special "stop" token $\mathcal{V}_{t,0}$ is added to $\mathcal{V}_t$ for the stop action. They are then fed into a panorama encoder (*i.e.* a two-layer transformer) to obtain the contextual vision representations $\mathcal{V}_t^c$ and $\mathcal{O}_t^c$:

$$[\mathcal{V}_t^c; \mathcal{O}_t^c] = \text{Pano}([\mathcal{V}_t'; \mathcal{O}_t']) \tag{2}$$

Dual-Scale Cross-Modal Encoder. The topological graph $\mathcal{G}_t = \{\boldsymbol{v}, \boldsymbol{e} \mid \boldsymbol{v} \subseteq \mathcal{V}, \boldsymbol{e} \subseteq \mathcal{E}\}$ at time step t contains three types of nodes: current node, visited nodes, and navigable nodes. Current node representation $\mathcal{H}_t$ is updated by averaging the local features $\mathcal{V}_t^c$ and $\mathcal{O}_t^c$. The features of visited nodes remain unchanged. Navigable nodes, which may have been partially observed from different viewpoints in the past trajectory, are represented by averaging all the partially observed views. We employ the cross-modal transformer to model the global-scale and local-scale cross-modal encoders, ensuring effective multi-level observation representation. For local-scale reasoning, local visual features $\mathcal{V}_t$ and $\mathcal{O}_t$ perform cross-attention over language representations $\hat{\mathcal{W}}$, resulting in $\hat{\mathcal{V}}_t$ and $\hat{\mathcal{O}}_t$:

$$[\hat{\mathcal{V}}_t^c; \hat{\mathcal{O}}_t^c] = \text{Cross-Attn}([\mathcal{V}_t^c; \mathcal{O}_t^c], \hat{\mathcal{W}}) \tag{3}$$

For global-scale reasoning, node representations $\mathcal{H}$ perform cross-attention over language representations $\hat{\mathcal{W}}$, yielding $\hat{\mathcal{H}}$:

$$\hat{\mathcal{H}} = \text{Cross-Attn}(\mathcal{H}, \hat{\mathcal{W}}) \tag{4}$$

Global features $\hat{\mathcal{H}}$ and local features $\hat{\mathcal{V}}_t^c$ are employed for navigation. Object features $\hat{\mathcal{O}}_t^c$ are employed for object grounding.

3.3 Visual-Aware Sparse Mixture-of-Experts

In Embodied Referring Expression Grounding, navigation and object grounding rely on distinct visual input (*i.e.* views and objects) with different data distributions: views offer comprehensive room layouts, while objects focus on specific item details. Directly using the same Feed-Forward Networks (FFNs) for all visual tokens results in coarse visual representations, which negatively impact model performance. To solve this problem, we propose a Visual-Aware Sparse Mixture-of-Experts (ViSMoE) as shown in Fig. 2, which incorporates the visual-aware routing policy on the SMoE to capture distribution-specific information between views and objects.

Sparse Mixture-of-Experts (SMoE). An SMoE layer contains a gated network (router) and N experts which are FFNs. Given an input token $x \in \mathcal{R}^D$, which is replaced by visual representations $\mathcal{V}$ and $\mathcal{O}$ in practice, it is fed into the router $\mathcal{G}(\cdot)$ to get the assignment probability for N experts, which are used for further processing. This can be described as:

$$\mathcal{G}(x) = \text{top-k}(\text{softmax}(\text{W}_g \cdot x)) \tag{5}$$

where $\text{W}_g \in \mathcal{R}^{N \times D}$ is learnable and top-k$(\cdot)$ is a function that selects the largest k values.

The final output of the SMoE is the weighted sum of the k features derived from the expert selected by the router, which can be formulated as:

$$\hat{x} = \sum_{i=1}^{k} \mathcal{G}(x)_i \cdot \text{FFN}_i(x) \tag{6}$$

Visual-Aware Routing. The conditional sparse MoE imposes no specific constraints on the router network, resulting in insufficiently targeted routing for different token types. As shown in Fig. 2, the vanilla SMoE distributes visual tokens of various types across all experts, whereas ViSMoE directs them to specific experts.

Specifically, we propose a visual-aware routing policy that explicitly provides the router network with visual type information by incorporating visual type-specific embeddings. This visual-aware routing can be described as follows:

$$\mathcal{G}(x) = \text{top-k}(\text{softmax}(\text{W}_g \cdot (x + b_t))) \tag{7}$$

where $b_t \in \mathcal{R}^D$ represents the type embedding, with b_v for view representations $\mathcal{V}$ and b_o for object representations $\mathcal{O}$.

Auxiliary Loss. To prevent imbalanced token routing, we introduce a Load-Balancing Loss following [17] as an auxiliary loss for the router network. It is formulated as follows:

$$\mathcal{L}_{\text{Aux}} = \alpha \cdot N \sum_{i=1}^{N} p_i \times f_i \tag{8}$$

where α is set to 0.001, N is the number of experts, p_i is the average routing weight for the i^{th} expert, and f_i is the fraction of tokens allocated to the i^{th} expert. This loss function promotes an even distribution of tokens across experts.

3.4 Training and Inference

After incorporating Eq. 7, we employ Eq. 6 to replace the FFN layers in Eqs. 2 and 3, enabling a more specialized and efficient processing framework. ViSMoE's parameters are updated during pre-training and fine-tuning.

Pre-training. We pre-train the model on four tasks: masked language modeling (MLM), masked region classification (MRC), single-step action prediction (SAP) [3] and object grounding (OG) [13]. The SAP and OG losses are defined as follows:

$$\mathcal{L}_{\text{SAP}} = \sum_{t=1}^{T} -log\, p(a_t^* | \mathcal{W}, P_{<t}) \tag{9}$$

where $P_{<t}$ is a partial ground-truth demonstrations, a_t^* is the expert action of $P_{<t}$.

$$\mathcal{L}_{\text{OG}} = -log\, p(o^* | \mathcal{W}, P_T) \tag{10}$$

where o^* is the ground-truth object at the destination P_T. In summary, the total objective of pre-training is:

$$\mathcal{L}_{\text{P}} = \mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{MRC}} + \mathcal{L}_{\text{SAP}} + \mathcal{L}_{\text{OG}} + \mathcal{L}_{\text{Aux}} \tag{11}$$

Fine-Tuning. For fine-tuning, in addition to SAP, we employ the DAgger algorithm [16] to train the navigation policy, enabling the agent to navigate based on predicted probabilities during training and to correct bias through learning:

$$\mathcal{L}_{\text{DA}} = \sum_{t=1}^{T} -log\, p(a_t^* | \mathcal{W}, \hat{P}_{<t}) \tag{12}$$

where $\hat{P}_{<t}$ is the sampled demonstrations. The total objective of fine-tuning is:

$$\mathcal{L}_{\text{F}} = \mathcal{L}_{\text{SAP}} + \mathcal{L}_{\text{DA}} + \mathcal{L}_{\text{OG}} + \mathcal{L}_{\text{Aux}} \tag{13}$$

4 Experiments

4.1 Experimental Setup

Datasets. We conduct experiments and evaluate our model on benchmarks REVERIE [15] and SOON [24].

REVERIE contains high-level instructions that average 18 words in length, with navigation paths ranging from 4 to 7 steps. Each panorama is equipped with predefined object bounding boxes.

Table 1. Evaluation on REVERIE dataset.

Methods	Val Unseen						Test Unseen					
	Navigation				Grounding		Navigation				Grounding	
	TL↓	OSR↑	SR↑	SPL↑	RGS↑	RGSPL↑	TL↓	OSR↑	SR↑	SPL↑	RGS↑	RGSPL↑
CKR [6]	26.26	31.44	19.14	11.84	11.45	-	22.46	30.40	22.00	14.25	11.60	-
SIA [13]	41.53	44.67	31.53	16.28	22.41	11.56	48.61	44.56	30.80	14.85	19.02	9.20
HAMT [3]	**14.08**	36.84	32.95	30.20	18.92	17.28	**13.62**	33.41	30.40	26.67	14.88	13.08
TD-STP [23]	-	39.48	34.88	27.32	21.16	16.56	-	40.26	35.89	27.51	19.88	15.40
FDA [8]	19.04	51.41	47.57	**35.90**	32.06	**24.31**	17.30	53.54	49.62	36.45	30.34	22.08
DUET [4]	22.11	51.07	46.98	33.73	32.15	23.03	21.30	56.91	52.51	36.06	31.88	22.06
ViSMoE (Ours)	23.85	**54.98**	**50.18**	35.35	**33.71**	23.87	22.08	**58.63**	**54.16**	**37.27**	**33.38**	**22.49**

Table 2. Evaluation on SOON dataset.

Methods	Val Unseen					Test Unseen				
	TL↓	OSR↑	SR↑	SPL↑	RGSPL↑	TL↓	OSR↑	SR↑	SPL↑	RGSPL↑
GBE [24]	**28.96**	28.54	19.52	13.34	1.16	**27.88**	21.45	12.90	9.23	0.45
KERM [11]	35.83	51.62	38.05	23.16	4.04	-	-	-	-	-
GridMM [21]	38.92	**53.39**	37.46	24.81	3.91	46.20	48.02	36.27	21.25	4.15
DUET [4]	36.20	50.91	36.28	22.58	3.75	41.83	43.00	33.44	21.42	4.17
Ours	36.17	51.06	**38.23**	**26.13**	**4.31**	38.41	**48.56**	**36.69**	**21.53**	**5.20**

SOON offers detailed instructions that precisely describe the target scenes and objects. These instructions average 47 words in length, with navigation paths varying from 2 to 21 steps.

Evaluation Metrics. We employ standard and widely-used evaluation metrics, including trajectory length (TL), oracle success rate (OSR), success rate (SR), and success rate penalized by path length (SPL) to assess navigation performance. Additionally, remote grounding success rate (RGS) and remote grounding success rate weighted by path length (RGSPL) are used to evaluate the object grounding task.

Implementation Details. We employ 9, 2, 4 and 4 transformer layers in the text encoder, panorama encoder, coarse-scale cross-modal encoder and fine-scale cross-modal encoder. Except for the panorama encoder, the other encoders' parameters are initialized with the pretrained LXMERT [18].

4.2 Main Results

Results on REVERIE. Table 1 show the performance of our ViSMoE compared to previous methods on REVERIE. Our model outperforms the above model and improves almost all metrics which demonstrates the effectiveness and generalization ability of our ViSMoE. Specifically, compared to DUET [4], our model improves SR and RGS by 3.20% and 1.56% on validation split, and by 1.65% and 1.50% on test split.

Results on SOON. Table 2 presents the results on the SOON dataset. Our proposed ViSMoE outperforms DUET across all metrics. Especially, our ViS-

MoE achieves a 3.55% improvement in SPL and a 0.56% gain in RGSPL on the validation split, as well as a 0.11% increase in SPL and a 1.03% gain on the test split. Notably, our ViSMoE demonstrates a significantly larger improvement in RGSPL on the SOON dataset compared to REVERIE, highlighting its superior performance on tasks with greater visual and linguistic complexity. According to the statistics, each instruction in SOON contains an average of 11.3 nouns, significantly more than the 5.31 in REVERIE. Additionally, each navigation step in SOON includes an average of 45.8 candidate objects, compared to just 3.42 in REVERIE.

Table 3. Ablation study on the val unseen split of REVERIE.

Baseline	SMoE	Vis.	Aux.	SR↑	SPL↑	RGS↑	RGSPL↑
✓				46.98	33.73	32.15	23.03
✓	✓			48.37	34.09	32.57	22.94
✓	✓		✓	48.46	34.19	32.67	23.16
✓	✓	✓		49.92	35.01	33.42	23.74
✓	✓	✓	✓	**50.18**	**35.35**	**33.71**	**23.87**

Table 4. Effect of number of experts on ViSMoE.

#Experts	TL↓	OSR↑	SR↑	SPL↑	RGS↑	RGSPL↑
2	23.74	52.46	48.62	32.91	32.38	21.98
4	23.85	54.98	50.18	**35.35**	33.71	**23.87**
8	**23.54**	53.62	49.59	33.14	33.84	23.32
16	26.63	**55.78**	**50.38**	32.76	**33.91**	22.19

4.3 Ablation Studies

We conduct the ablation studies on the val unseen split of the REVERIE dataset.

Components of ViSMoE. As shown in Table 3, we begin by evaluating the necessity of each component in ViSMoE, with the number of experts set to 4 and the top-k strategy using $k = 2$. "Baseline" stands for the method of vanilla DUET [4]. "SMoE" denotes using the sparse Mixture-of-Experts. "Vis." denotes using visual-aware sparse Mixture-of-Experts. "Aux." denotes using load-balance loss. Row 1 in Table 3 presents the results of the original DUET model [4]. In comparison, Row 2 demonstrates improvements in all metrics except for RGSPL. Notably, when the auxiliary loss is introduced to balance the router load, all metrics show improvement over the baseline, which demonstrates the effectiveness of the vanilla SMoE and the auxiliary loss. Furthermore, Row 4 shows a substantial increase compared to Row 2, indicating that our proposed visual-aware routing policy enhances the router's ability to differentiate between various visual inputs, leading to better performance.

Effect of Number of Experts on ViSMoE. We investigate the impact of the number of experts on ViSMoE, using the top-k strategy with $k = 2$. Table 4

illustrates the variation in results across the metrics. It can be observed that as number of experts increases beyond 4, the model size grows, but the overall performance remains nearly unchanged. This suggests that, considering the size of the dataset and the complexity of the task, having the right number of experts is more effective than simply increasing the number.

Effect of Top-k on ViSMoE. As is shown in Table 5, We study the impact of the top-k routing strategy on ViSMoE, keeping the number of experts fixed at 4. When the top-k value is set to 2 or 4, the model maintains consistent performance across different configurations. However, as top-k increases, more experts of the model are activated, leading to higher computational overhead.

Table 5. Effect of top-k on ViSMoE.

top-k	TL↓	OSR↑	SR↑	SPL↑	RGS↑	RGSPL↑
1	25.53	51.31	47.96	31.12	32.11	20.84
2	23.85	**54.98**	**50.18**	**35.35**	**33.71**	23.87
4	**23.47**	54.76	49.64	34.74	33.68	**23.94**

Table 6. Effect of target layer selection.

Target layer	OSR↑	SR↑	SPL↑	RGS↑	RGSPL↑
Baseline	51.07	46.98	33.73	32.15	23.03
Pano.1	51.14	47.16	32.16	32.04	22.37
Pano.2	53.02	48.63	32.68	32.23	22.18
Pano.2, Cros.1	52.09	48.19	33.87	32.75	22.87
Pano.2, Cros.2	53.89	49.76	34.61	33.08	23.16
Pano.2, Cros.3	54.03	**50.37**	35.14	32.94	23.38
Pano.2, Cros.4	**54.98**	50.18	**35.35**	**33.71**	**23.87**

Effect of Target Layer Selection. As in shown in Table 6, we progressively insert ViSMoE layer, with 4 experts and the top-k strategy using k = 2, starting from the panorama encoder and moving to the local scale cross-modal encoder. "Pano" refers to the panorama encoder, while "Cros" indicates the local scale cross-modal encoder, where views and objects perform cross-attention over language. ".x" specifies the insertion of the MoE layer within the transformer blocks, from the 1^{st} to the x^{th} block. In "Cros", we initialize each layer's experts' parameters using the corresponding layer's FFN parameters from LXMERT. We observe that as more and more transformer blocks in the panorama encoder and the local scale cross-modal encoder are equipped with ViSMoE layers, the model's performance gradually increases.

5 Conclusion

We introduce ViSMoE, a framework that leverages the sparse Mixture-of-Experts with a visual-aware routing policy to process diverse visual inputs. Our contributions underline the advantages of using MoE for processing visual features in embodied AI, paving the way for more robust and precise navigation systems in real-world scenarios.

Acknowledgement. This research is supported by National Natural Science Foundation of China (No.62476127), Natural Science Foundation of Jiangsu (No. BK20242039), Basic Research Program of the Bureau of Science and Technology (ILF24001), Fundamental Research Funds for the Central Universities (No. NJ2023032), Scientific Research Starting Foundation of Nanjing University of Aeronautics and Astronautics (No. YQR21022), and the High Performance Computing Platform of Nanjing University of Aeronautics and Astronautics.

References

1. Anderson, P., et al.: Vision-and-language navigation: interpreting visually-grounded navigation instructions in real environments. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3674–3683 (2018)
2. Chen, J., et al.: EVE: efficient vision-language pre-training with masked prediction and modality-aware MoE. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 1110–1119 (2024)
3. Chen, S., Guhur, P.L., Schmid, C., Laptev, I.: History aware multimodal transformer for vision-and-language navigation. Adv. Neural. Inf. Process. Syst. **34**, 5834–5847 (2021)
4. Chen, S., Guhur, P.L., Tapaswi, M., Schmid, C., Laptev, I.: Think global, act local: dual-scale graph transformer for vision-and-language navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16537–16547 (2022)
5. Chen, T., et al.: AdaMV-MoE: adaptive multi-task vision mixture-of-experts. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 17346–17357 (2023)
6. Gao, C., Chen, J., Liu, S., Wang, L., Zhang, Q., Wu, Q.: Room-and-object aware knowledge reasoning for remote embodied referring expression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3064–3073 (2021)
7. Gupta, S., et al.: Sparsely activated mixture-of-experts are robust multi-task learners. arXiv preprint arXiv:2204.07689 (2022)
8. He, K., Si, C., Lu, Z., Huang, Y., Wang, L., Wang, X.: Frequency-enhanced data augmentation for vision-and-language navigation. In: Advances in Neural Information Processing Systems, vol. 36 (2024)
9. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Comput. **3**(1), 79–87 (1991)
10. Jordan, M.I., Jacobs, R.A.: Hierarchical mixtures of experts and the EM algorithm. Neural Comput. **6**(2), 181–214 (1994)

11. Li, X., Wang, Z., Yang, J., Wang, Y., Jiang, S.: KERM: knowledge enhanced reasoning for vision-and-language navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2583–2592 (2023)
12. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
13. Lin, X., Li, G., Yu, Y.: Scene-intuitive agent for remote embodied visual grounding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7036–7045 (2021)
14. Mohammadi, B., Hong, Y., Qi, Y., Wu, Q., Pan, S., Shi, J.Q.: Augmented commonsense knowledge for remote object grounding. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 4269–4277 (2024)
15. Qi, Y., et al.: Reverie: remote embodied visual referring expression in real indoor environments. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9982–9991 (2020)
16. Ross, S., Gordon, G., Bagnell, D.: A reduction of imitation learning and structured prediction to no-regret online learning. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pp. 627–635. JMLR Workshop and Conference Proceedings (2011)
17. Shazeer, N., et al.: Outrageously large neural networks: the sparsely-gated mixture-of-experts layer. arXiv preprint arXiv:1701.06538 (2017)
18. Tan, H., Bansal, M.: Lxmert: learning cross-modality encoder representations from transformers. arXiv preprint arXiv:1908.07490 (2019)
19. Vaswani, A.: Attention is all you need. In: Advances in Neural Information Processing Systems (2017)
20. Wang, A.: Glue: a multi-task benchmark and analysis platform for natural language understanding. arXiv preprint arXiv:1804.07461 (2018)
21. Wang, Z., Li, X., Yang, J., Liu, Y., Jiang, S.: Gridmm: grid memory map for vision-and-language navigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 15625–15636 (2023)
22. Yuksel, S.E., Wilson, J.N., Gader, P.D.: Twenty years of mixture of experts. IEEE Trans. Neural Netw. Learn. Syst. **23**(8), 1177–1193 (2012)
23. Zhao, Y., et al.: Target-driven structured transformer planner for vision-language navigation. In: Proceedings of the 30th ACM International Conference on Multimedia, pp. 4194–4203 (2022)
24. Zhu, F., Liang, X., Zhu, Y., Yu, Q., Chang, X., Liang, X.: Soon: scenario oriented object navigation with graph-based exploration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12689–12699 (2021)

FDFRL: Credit Card Fraud Detection Based on Federated Reinforcement Learning

Nana Zhang[1,2,3], Qin Li[1], Kun Zhu[2,3,4(✉)], and Dandan Zhu[5]

[1] School of Computer Science and Technology, Donghua University, No. 2999, Renmin North Road, Songjiang District, Shanghai 201620, China
[2] Key Laboratory of Embedded System and Service Computing (Tongji University), Ministry of Education, Shanghai 201804, China
kzhu00@tongji.edu.cn
[3] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, People's Republic of China
[4] Key Laboratory of Machine Intelligence and System Control, Ministry of Education, Shandong University, Jinan, People's Republic of China
[5] Institute of AI Education, Shanghai, East China Normal University, No. 3663, Zhongshan North Road, Putuo District, Shanghai 200333, China

Abstract. To tackle extreme class imbalance, non-independent and identically (non-IID) distributed transactions and dynamic fraud patterns in credit card fraud detection (CCFD), we propose FDFRL, credit card Fraud Detection based on Federated Reinforcement Learning, with three core modules: Kernel-guided adversarial representation learning for high-fidelity synthetic sample generation and robust feature embeddings via hierarchical adversarial refinement. PPO-based federated reinforcement learning employing gradient smoothing and dynamic weighting to harmonize updates from heterogeneous clients and ensure steady convergence. Label-driven federated fusion module that seamlessly integrates local representations into a unified global classifier. Extensive experiments on real-world fraud datasets show that FDFRL markedly outperforms eight state-of-the-art baselines.

Keywords: Transaction Fraud Detection · Federated Reinforcement Learning · Extreme Class Imbalance · Dynamic Fraud Patterns · Proximal Policy Optimization

1 Introduction

Nowadays, the use of credit cards in daily life is becoming increasingly frequent, and the transaction scale is growing. However, it is worrying that fraudulent transactions are also increasing, which has caused a significant impact on financial order and consumer rights [3]. According to the Nilson Report [17], in 2024, global payment card fraud (the sum of all types of cards such as credit cards and debit cards) lost approximately 33.26 billion dollars. The total global volume of payment card transactions will be estimated to reach 64.791 trillion dollars in

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 326–338, 2026.
https://doi.org/10.1007/978-3-032-04555-3_27

2028, of which the amount of card fraud will reach 39.37 billion dollars. Accordingly, banks and payment institutions need to devote substantial resources to tracking and remediation, which increases operational costs and can create credit risk and lower market confidence.

In real-world Credit Card Fraud Detection (CCFD), privacy and compliance restrictions prevent banks or departments from sharing sensitive transaction data, resulting in limited training data. Federated learning (FL) enables each party to train models locally and share only model parameters or gradients with a central server, thus facilitating cross-institution collaboration while safeguarding privacy and enhancing fraud detection capabilities.

Recently, several studies [12,13] have utilized federated learning techniques to detect credit card fraud. These methods effectively protect user privacy and achieve cross-bank collaboration by training models locally in each bank and sharing model parameters. Although federated learning provides an effective solution for CCFD, it still faces many challenges in a complex cross-bank collaboration environment. First, fraudulent transactions only account for a very small part of the total data. Under extremely unbalanced data distribution, traditional machine learning methods often have difficulty in accurately identifying minority class samples, resulting in high false positive rates or missed detections. In addition, credit card fraud is usually dynamic. With the maturity of underground fraud networks and the rapid update of fraud technology, fraud methods are constantly changing. Existing federated learning approaches [18,23] often suffer from insufficient update frequency and low coordination efficiency, making it difficult to promptly capture the characteristics of new fraud samples. Therefore, enhancing the adaptability of federated learning when dealing with rapidly evolving fraud patterns remains an urgent challenge.

Motivated by these limitations, we propose FDFRL, a credit card **F**raud **D**etection model based on **F**ederated **R**einforcement **L**earning. FDFRL comprises three modules: kernel-guided adversarial representation learning (KARL), PPO-based federated reinforcement learning, and label-driven federated fusion. KARL employs layered neural representation learning with kernel-guided adversarial training to correct class imbalance and generate robust feature vectors. PPO-based federated reinforcement learning enables each client to refine its local detection policy via clipped Monte Carlo advantages and transmit gradient updates to a central server, which consolidates them into a privacy-preserving global policy capable of swiftly adapting to new fraud patterns. Label-driven federated fusion clusters clients by their transaction label distributions and unifies their local embeddings into a shared global representation, mitigating heterogeneity and boosting detection accuracy.

Our contributions can be summarized as follows.

- We propose FDFRL, a credit card fraud detection model based on federated reinforcement learning that unifies kernel-guided adversarial representation learning(KARL), PPO-based federated reinforcement learning, and label-driven federated fusion(LFF). During aggregation, FDFRL applies a strategy-aware gradient clipping mechanism: client gradients are averaged, and parameter ratios between new and old policies are computed and clipped

within a PPO-inspired ϵ-bounded trust region. This constrains abrupt policy shifts and stabilizes updates under non-IID and label-skewed conditions.
- We design KARL, which uses a generator composed of two sequential stages: in the first stage it produces compact and information rich latent vectors, and in the second stage it refines these vectors through a critic network to synthesize realistic and diverse samples even under conditions of scarcity. Concurrently, the LFF module organizes clients into semantically consistent groups by label distribution and fuses their high-order features through hierarchical concatenation. The fused representation is used to train a global classifier with partial extractor freezing, enabling effective generalization under client heterogeneity.
- Experiments on three real-world fraud datasets show that FDFRL improves AUC by 3.09%, F1 by 5.75%, and AP by 12.70% on average over eight SOTA models.

2 Related Work

2.1 Deep Learning Methods in CCFD

In recent years, some typical deep learning techniques have been widely used to detect fraudulent transactions in CCFD. For example, some studies [20,31] use a variety of deep learning models (i.e., CNN, LSTM, etc.) to detect fraudulent transactions in credit cards. These studies [2,4,25] use deep auto-encoder technology to detect fraud in credit card transactions. In addition, these studies [6,7,28] use generative adversarial networks (GANs) to promote a deeper understanding and more accurate identification of credit card fraud.

2.2 Federated Learning in CCFD

HistFL [14] integrates weakly supervised attention multi-instance learning with differential privacy for cancer subtype and survival prediction. FedProx [11] employs a regularization function for robust convergence on non-independent and identically distributed (non-IID) data. Li et al. [10] propose a point transformer-based FL framework for improved HER2 prediction and privacy-preserving multi-site learning. These methods underscore FL's key trait of local training with gradient/weight sharing, enabling multi-party collaboration while protecting privacy [24].

CCFD involves highly sensitive data. Federated learning (FL) is vital for data security and privacy in decentralized settings. Yang et al. [27] tackle FL security in tilted datasets; Zheng et al. [29] study FL in cross-island non-IID environments with meta-learning; Singh et al. [19] use FL for anomaly detection via a neural autoencoder; Lv et al. [15] build a social financial fraud model with longitudinal FL leveraging financial and social traits; and Wang et al. [21] propose an Approx-SMOTE strategy for efficient parallel preprocessing. A weighting mechanism that factors sample size and accuracy, with salting and interference, further boosts detection efficiency, accuracy, and security.

FL in CCFD boosts privacy and pattern capture, but integrating generative models for augmentation and imbalance correction remains hard: sparse client-level data must be pooled without leaking privacy, while avoiding mode collapse, bias, and over-reliance on supervised methods.

3 The Proposed FDFRL Model

3.1 Problem Formulation

CCFD is tackled using federated learning by integrating self-supervised representation learning with reinforcement learning into a robust hybrid strategy. Figure 1 shows the model architecture. Let $T = \{T_i\}_{i=1}^{N}$ denote transaction records with features X_i and labels y_i. Given clusters $G = \{G_1, G_2\}$ grouping clients, the server maintains the global model M and manages gradient updates. The goal is to minimize detection error while preserving privacy.

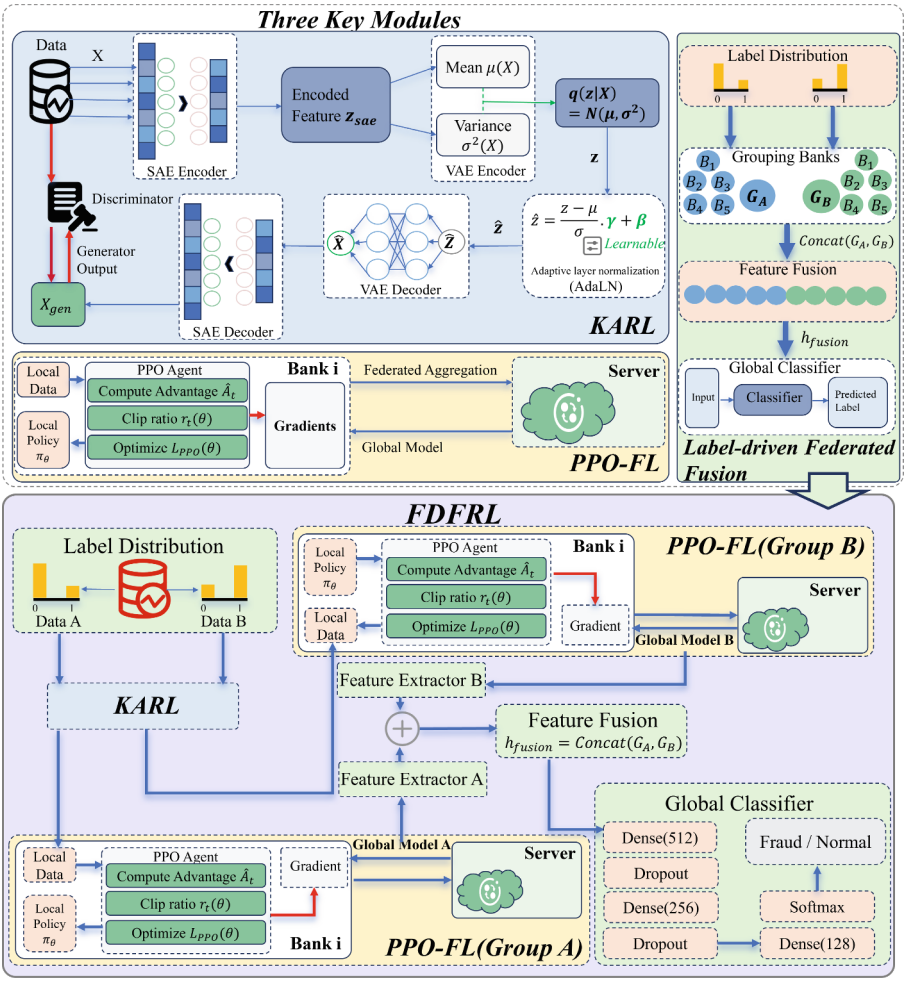


Fig. 1. The model architecture of FDFRL.

3.2 Kernel-Guided Adversarial Representation Learning

In fact, SVAE-GAN is the core module, which uses SVAE as a generator for high-quality adversarial samples. SVAE comprises an SAE encoder, VAE encoder, adaptive layer normalization (LN), VAE decoder, and SAE decoder.

First, the SAE encoder compresses input X into a latent code, as shown below:

$$z_{\text{sae}} = f_{\text{sae}}(X), \tag{1}$$

and the VAE encoder models it as a Gaussian distribution:

$$q(z \mid X) = \mathcal{N}\big(z \mid \mu(X), \sigma^2(X)\big). \tag{2}$$

Adaptive LN normalizes the latent features:

$$\hat{z} = \frac{z - \mu}{\sigma} \cdot \gamma + \beta. \tag{3}$$

Then, the VAE decoder reconstructs $\hat{z}$ to obtain $\hat{X}$,

$$p(\hat{X} \mid z) = \mathcal{N}\big(\hat{X} \mid g_{\text{vae}}(z), \sigma^2\big), \tag{4}$$

and the SAE decoder recovers the final sample:

$$X_{\text{gen}} = h_{\text{sae}}(\hat{X}). \tag{5}$$

Adversarial training refines both the generator and the discriminator. SVAE generates adversarial samples $G(z)$ evaluated by discriminator D, with losses defined as:

$$L_G = -\mathbb{E}_{z \sim p_z(z)}[\log D(G(z))], \tag{6}$$

$$L_D = -\mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] - \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]. \tag{7}$$

Additionally, farthest cosine sampling (FCS) selects the most representative features.

3.3 PPO-Based Federated Reinforcement Learning

PPO (Proximal Policy Optimization) is a reinforcement learning algorithm based on trust region optimization, which aims to maximize cumulative rewards by optimizing strategies. The core idea is to ensure the stability of training by limiting the amplitude of each strategy update and avoiding excessive strategy changes. The objective function of PPO can be expressed as follows:

$$L^{\text{PPO}}(\theta) = E_t\left[\min\left(r_t(\theta)\hat{A}_t, clip\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right], \tag{8}$$

in which $r_t(\theta)$ is the strategy ratio (that is, the ratio of the current strategy to the old strategy):

$$r_t(\theta) = \frac{\pi_\theta\left(a_t \mid s_t\right)}{\pi_{\theta_{old}}\left(a_t \mid s_t\right)}. \tag{9}$$

In Eq. (8): the core is the clipping operation, i.e., *clip*, which ensures that the strategy does not deviate too far at each update, thereby improving the stability of training. $\hat{A}_t$ is an advantage function, which represents the advantage of an action over the current strategy, and ϵ is the clipping parameter, which controls the magnitude of the strategy update.

One of the keys to PPO is the advantage function, which measures the quality of an action relative to the current strategy. In federated reinforcement learning, the calculation of the advantage function is usually implemented through the Monte Carlo method to evaluate the quality of each action. The advantage function $\hat{A}_t$ can be expressed as:

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + (\gamma\lambda)^2\delta_{t+2} + \ldots + (\gamma\lambda)^n\delta_{t+n}, \tag{10}$$

where δ_t is the temporal difference error, γ is the discount factor, and λ is the attenuation coefficient of the advantage function. The pseudocode of PPO-based Federated Reinforcement Learning is as shown in Algorithm 1.

Algorithm 1: Federated Policy Optimization for Fraud Detection

Input: Client set $C = \{C_1, \ldots, C_N\}$, global policy $\pi_{\theta_{\text{global}}}$, local data D_{local}.

Output: Final global policy $\hat{\pi}$.

Initialize $\pi_{\theta_{\text{global}}}$;
for *round* $r = 1$ *to num_ rounds* **do**
  Initialize gradient list $G \leftarrow []$;
  for *each client* $C_i \in C$ **do**
    Train local model $\pi_{\theta_{\text{local}}}$ on D_{local} via PPO;
    Compute advantage $\hat{A}_t$;
    Update local model using the PPO loss, calculated by Eq. (8);
    Compute and append gradients $\nabla\theta_{\text{local}}$;
  Aggregate gradients: $\nabla\theta_{\text{global}} = \frac{1}{N}\sum_{i=1}^{N} \nabla\theta_{\text{local},i}$;
  Update $\pi_{\theta_{\text{global}}}$ and send it to all clients;
  if r mod *evaluation_ interval* $= 0$ **then**
    Evaluate the global model.
return $\hat{\pi} = \pi_{\theta_{global}}$;

3.4 Label-Driven Federated Fusion

Clients are grouped by label distribution for consistency. Each group is processed by the kernel-driven adversarial module to obtain a local feature:

$$h^{(i)} \in \mathbb{R}^{d_i}, \quad i = 1, \ldots, K, \tag{11}$$

where K is the number of groups.

The server fuses these features by concatenation:

$$h_{\text{fusion}} = \text{Concat}\left(h^{(1)}, \ldots, h^{(K)}\right) \in \mathbb{R}^d, \quad d = \sum_{i=1}^{K} d_i. \tag{12}$$

This fused feature is passed through a fully connected layer, and prediction is obtained via softmax:

$$\hat{y} = \text{softmax}(z), \quad z = W h_{\text{fusion}} + b. \tag{13}$$

The model is then optimized using cross-entropy loss:

$$\mathcal{L}_{\text{cls}} = -\sum_{j=1}^{C} y_j \log\left(\hat{y}_j\right), \tag{14}$$

with C as the number of categories. Finally, the server updates the global model using gradient descent on the fused features.

4 Experiments

4.1 Experimental Setup

Dataset. To demonstrate the effectiveness of the FDFRL model, we select three publicly available real-world transaction datasets, including S-FFSD, Default of Credit Card Clients and Creditcard. For convenience, they are abbreviated as C_1, C_2 and C_3. The statistical details of the three datasets are shown in Table 1.

Table 1. Statistics of three datasets.

Dataset	Record	Fraud	Legitimate	Unlabeled
C_1	77881	5256	24387	48238
C_2	30000	6636	23364	0
C_3	284807	492	284315	0

The C_1 dataset from Xiang et al. [26] comprises partially labeled records contributed by collaborators, and its subset S-FFSD is publicly available on GitHub. The C_1 collection is a simulated, scaled-down version of the Financial Fraud Semi-supervised Dataset (FFSD). C_2 was introduced by Zhu et al. [30]. In C_2, there are 30000 samples in total, of which 6636 are positive samples. C_3, from European card transactions in September 2013 on Kaggle, comprises 284,807 transactions (492 fraudulent, 0.172%) with 30 PCA-derived features.

Each dataset is first stratified into an 80% train and 20% test split, preserving the original fraud-to-legitimate ratio. With no missing values present, imputation is skipped. The training portion is then balanced using SMOTEENN, its feature matrix is standardized and normalized, and finally the top k representative features are selected via kernel farthest cosine sampling.

Baselines. This paper uses eight baseline models, including FedAvg [16], FedProx [11], FedNova [22], SCAFFOLD [5], FedDyn [1], FedNTD [8], and two versions of FedSOL [9].

Evaluation Metrics. To evaluate the effectiveness of the FDFRL model and eight baseline models, we utilize area under curve (AUC), F1 score, average precision (AP) as evaluation metrics.

Parameter Settings. For each experiment, models are trained for 120 epochs using Adam with a batch size of 64, a learning rate of 0.001, and a label skew of 0.9. The *latent_dim*, which defines the SVAE's compressed space, is set to 100. The PPO clipping parameter is 0.2 to limit policy updates and stabilize gradients.

4.2 Fraud Detection Performance

FDFRL is compared with eight state-of-the-art anti-fraud models to validate its performance, and the results are presented in Table 2.

Table 2. Model performance on different datasets.

Models	C_1			C_2			C_3		
	AUC	F1	AP	AUC	F1	AP	AUC	F1	AP
FedAvg	0.9448	0.8573	0.8170	0.9149	0.8418	0.8159	0.9996	0.9421	0.9408
FedProx	0.9425	0.8649	0.8161	0.9147	0.8380	0.8113	**0.9999**	0.9477	0.9453
FedNova	0.9328	0.8363	0.7902	0.9139	0.8388	0.8111	**0.9999**	0.9464	0.9421
SCAFFOLD	0.8074	0.6704	0.7027	0.7973	0.6765	0.6797	0.9998	0.9411	0.9397
FedDyn	0.9446	0.8625	0.7818	0.8955	0.8232	0.7795	0.8325	0.7933	0.7993
FedNTD	0.9417	0.8419	0.7428	0.9080	0.8249	0.8038	0.9997	0.9312	0.9304
FedSOL-fixed	0.8813	0.7125	0.7373	0.8096	0.6784	0.6920	0.9996	0.9321	0.9280
FedSOL-adaptive	0.9424	0.8569	0.8117	0.9138	0.8358	0.8091	0.9997	0.9265	0.9242
FDFRL	**0.9698**	**0.8835**	**0.9682**	**0.9913**	**0.9783**	**0.9945**	0.9911	**0.9652**	**0.9965**
Training efficiency (ms)	38.33			26.85			6.32		
Inference speed (ms)	18.41			14.15			3.67		

As shown in Table 2, FDFRL outperforms 8 SOTA models on all metrics for C_1 and C_2, and leads in F1 and AP on C_3. On C_3, it achieves F1 = 0.9652 and AP = 0.9965 (AUC = 0.9911). In contrast, FedAvg and FedNova, using local SGD with simple averaging, yield high AUC under 0.172% positives by biasing toward negatives—resulting in lower F1 and AP. Likewise, FedProx, SCAFFOLD, FedNTD, and FedSOL favor negatives to boost AUC at the expense of sensitivity. Overall, FDFRL adapts well: it improves C_1 by 2.65% (AUC), 2.15% (F1), and 18.51% (AP); C_2 by 8.35%, 16.22%, and 21.89%; and on C_3, it surpasses the best baseline by 1.85% in F1 and 5.42% in AP.

4.3 Ablation Study

Three variants of FDFRL are further investigated to explore the effectiveness of each component:

– **w/o KARL**: The component of Kernel-guided Adversarial Representation Learning module is excised from FDFRL.

Table 3. Ablation analysis on three datasets.

Models	C_1			C_2			C_3		
	AUC	F1	AP	AUC	F1	AP	AUC	F1	AP
w/o KARL	0.9497	0.8694	0.9505	0.9770	0.9505	0.9800	0.9866	0.9367	0.9875
w/o PPO	0.9362	0.8543	0.9471	0.9570	0.9151	0.9640	0.9481	0.9332	0.9694
w/o LFF	0.9508	0.8656	0.9579	0.9788	0.9350	0.9833	0.9792	0.9411	0.9797
FDFRL	**0.9698**	**0.8835**	**0.9682**	**0.9913**	**0.9783**	**0.9945**	**0.9911**	**0.9652**	**0.9965**

- **w/o PPO**: In FDFRL, we remove PPO from the federated reinforcement learning framework.
- **w/o LFF**: In the training of base classifier, we remove Label-driven Federated Fusion module.

Table 3 compares the detection performance across three metrics. FDFRL outperforms all variants. The drop in w/o KARL confirms its role in feature learning and sample augmentation. The significant degradation in w/o PPO underscores PPO's importance for policy optimization and effective federated updates, while the decline in w/o LFF highlights the benefit of label fusion.

4.4 Sensitivity Analysis

In this section, the sensitivity of FDFRL to various parameters is investigated, and the results are shown in Fig. 2.

Figure 2 shows that a batch size of 64 gives the best AUC (0.9698 vs. 0.9438 at 32), F1 (0.8835), and AP (> 0.95); a learning rate of 10^{-4} maximizes AUC (0.9829), F1, and AP (where rates $\geq 10^{-3}$ drop AUC to $\sim$0.91); a label skew of 0.85 yields peak AUC (0.9738) and F1 (0.9043); and using the 50th latent dimension achieves the highest AUC (0.9703), with all dimensions delivering AUCs of 0.96âĂŞ0.97 (except a dip to 0.9036 at dim 150) and F1 > 0.88, AP $>$ 0.96.

4.5 Discussion

Simple averaging (FedAvg), proximal regularization (FedProx), update normalization (FedNova), and drift correction (SCAFFOLD) all work well under IID data but struggle when client data are non-IID and highly imbalanced—their updates either over-converge to majority classes or fail to sufficiently correct minority-class bias. Dynamic regularization (FedDyn) can pull local models toward the global but yields only modest minority-class gains and is hyperparameter-sensitive; self-distillation (FedNTD) preserves global knowledge yet cannot compensate for scarce local samples; and orthogonal constraints (FedSOL) reduce interference but lack a mechanism to generate additional positive examples. FDFRL overcomes these issues by using KARL to produce high-quality synthetic fraud samples locally and by constraining and dynamically

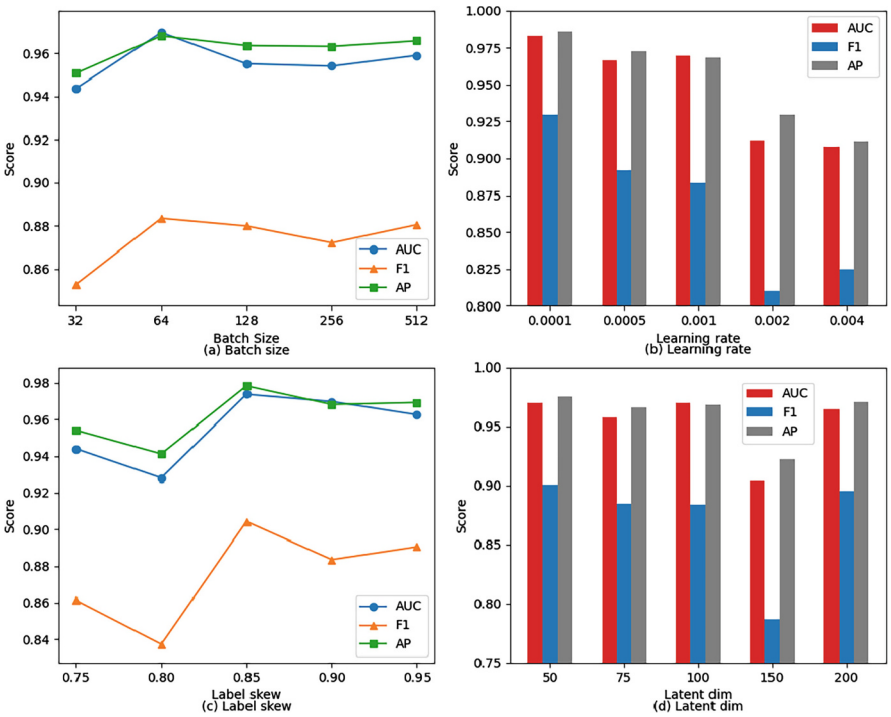


Fig. 2. Parameter sensitivity analysis.

weighting each client's gradient contributions during aggregation, thus balancing majorityâĂŞminority information conflicts effectively.

At its core, KARL performs hierarchical adversarial refinement in latent space, yielding both robust feature embeddings and realistic synthetic samples. Its PPO-based federated reinforcement learning component applies clipped advantage functions and dynamic weighting to reconcile conflicting updates, ensuring stable convergence and rapid adaptation to evolving fraud patterns in non-IID settings. The label-driven fusion (LFF) module then concatenates client features according to label distributions, harmonizing heterogeneous representations. Together, these mechanisms not only drive substantial performance gains under extreme imbalance and privacy constraints but also enhance interpretability: generated samples lie on a controlled manifold with measurable reconstruction quality, policy updates are traceable within a defined trust region, and fused features correspond to transaction-level label clusters—facilitating post-hoc analysis, auditability, and regulatory acceptance.

5 Conclusion

By orchestrating the kernel-guided adversarial learner, the federated policy stabilization engine, and the label-driven fusion scheme, FDFRL achieves a har-

monious interplay of modules that directly confronts extreme imbalance, distribution heterogeneity, and evolving fraud tactics. Extensive evaluations on real-world datasets reveal that FDFRL consistently surpasses eight leading methods in AUC, F1, and average precision, validating the power of its synergistic design. In future work, we will focus on automating hyperparameter tuning and embedding adaptive, low-latency inference capabilities to further elevate performance and deployment readiness.

Acknowledgments. This work is supported in part by the National Natural Science Foundation of China under Grant 62402098, 62302337, 62377011, in part by the Fundamental Research Funds for the Central Universities under Grant 2232024D-25, in part by the foundation of Key Laboratory of Embedded System and Service Computing (Tongji University), Ministry of Education, under Grant ESSCKF 2023-03, in part by the foundation of State Key Lab. for Novel Software Technology(Nanjing University), under Grant KFKT2025B26, KFKT2024B33, and in part by open research fund of Key Laboratory of Machine Intelligence and System Control, Ministry of Education (No. MISC-202401).

References

1. Acar, D.A.E., Zhao, Y., Navarro, R.M., Mattina, M., Whatmough, P.N., Saligrama, V.: Federated learning based on dynamic regularization. arXiv preprint arXiv:2111.04263 (2021)
2. Chen, J., Shen, Y., Ali, R.: Credit card fraud detection using sparse autoencoder and generative adversarial network. In: 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), pp. 1054–1059. IEEE (2018)
3. Cheng, D., Xiang, S., Shang, C., Zhang, Y., Yang, F., Zhang, L.: Spatio-temporal attention-based neural network for credit card fraud detection. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 362–369 (2020)
4. Fanai, H., Abbasimehr, H.: A novel combined approach based on deep autoencoder and deep classifiers for credit card fraud detection. Expert Syst. Appl. **217**, 119562 (2023)
5. Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A.T.: Scaffold: stochastic controlled averaging for federated learning. In: International Conference on Machine Learning, pp. 5132–5143. PMLR (2020)
6. Kaur, S., Singh, K.D., Singh, P., Kaur, R.: Ensemble model to predict credit card fraud detection using random forest and generative adversarial networks. In: Hassanien, A.E., Bhattacharyya, S., Chakrabati, S., Bhattacharya, A., Dutta, S. (eds.) Emerging Technologies in Data Mining and Information Security. AISC, vol. 1300, pp. 87–97. Springer, Singapore (2021). https://doi.org/10.1007/978-981-33-4367-2_10
7. Langevin, A., Cody, T., Adams, S., Beling, P.: Generative adversarial networks for data augmentation and transfer in credit card fraud detection. J. Oper. Res. Soc. **73**(1), 153–180 (2022)
8. Lee, G., Shin, Y., Jeong, M., Yun, S.: Preservation of the global knowledge by not-true self knowledge distillation in federated learning. arxiv 2021. arXiv preprint arXiv:2106.03097 (2021)

9. Lee, G., Jeong, M., Kim, S., Oh, J., Yun, S.Y.: Fedsol: stabilized orthogonal learning with proximal restrictions in federated learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12512–12522 (2024)
10. Li, B., Liu, Z., Shao, L., Qiu, B., Bu, H., Tian, J.: Point transformer with federated learning for predicting breast cancer HER2 status from hematoxylin and eosin-stained whole slide images. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 3000–3008 (2024)
11. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. Proc. Mach. Learn. Syst. **2**, 429–450 (2020)
12. Li, Y., Wen, G.: Research and practice of financial credit risk management based on federated learning. Eng. Lett. **31**(1) (2023)
13. Long, G., Tan, Y., Jiang, J., Zhang, C.: Federated learning for open banking. In: Yang, Q., Fan, L., Yu, H. (eds.) Federated Learning. LNCS (LNAI), vol. 12500, pp. 240–254. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-63076-8_17
14. Lu, M.Y., et al.: Federated learning for computational pathology on gigapixel whole slide images. Med. Image Anal. **76**, 102298 (2022)
15. Lv, B., Cheng, P., Zhang, C., Ye, H., Meng, X., Wang, X.: Research on modeling of e-banking fraud account identification based on federated learning. In: 2021 IEEE International Conference on Dependable, Autonomic and Secure Computing, International Conference on Pervasive Intelligence and Computing, International Conference on Cloud and Big Data Computing, International Conference on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), pp. 611–618. IEEE (2021)
16. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282. PMLR (2017)
17. Nilson Report 1276. https://nilsonreport.com/wp-content/uploads/NilsonReport_Issue_1276_0L24.pdf. Accessed 09 Jan 2025
18. Rathika, P., Pushparaj, A., et al.: Privacy preservation using federated learning for credit card transactions. In: 2023 International Conference on Intelligent Systems for Communication, ioT and Security (iCISCoIS), pp. 398–403. IEEE (2023)
19. Singh, S., Bhardwaj, S., Pandey, H., Beniwal, G.: Anomaly detection using federated learning. In: Bansal, P., Tushir, M., Balas, V.E., Srivastava, R. (eds.) Proceedings of International Conference on Artificial Intelligence and Applications. AISC, vol. 1164, pp. 141–148. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-4992-2_14
20. Voican, O.: Credit card fraud detection using deep learning techniques. Informatica Economica **25**(1) (2021)
21. Wang, J., Liu, W., Kou, Y., Xiao, D., Wang, X., Tang, X.: Approx-smote federated learning credit card fraud detection system. In: 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC), pp. 1370–1375. IEEE (2023)
22. Wang, J., Liu, Q., Liang, H., Joshi, G., Poor, H.V.: Tackling the objective inconsistency problem in heterogeneous federated optimization. Adv. Neural. Inf. Process. Syst. **33**, 7611–7623 (2020)
23. Wang, S., Gai, K., Yu, J., Zhang, Z., Zhu, L.: Pravfed: practical heterogeneous vertical federated learning via representation learning. IEEE Trans. Inf. Forensics Secur. (2025)

24. Wen, J., Zhang, Z., Lan, Y., Cui, Z., Cai, J., Zhang, W.: A survey on federated learning: challenges and applications. Int. J. Mach. Learn. Cybern. **14**(2), 513–535 (2023)
25. Wu, T.Y., Wang, Y.T.: Locally interpretable one-class anomaly detection for credit card fraud detection. In: 2021 International Conference on Technologies and Applications of Artificial Intelligence (TAAI), pp. 25–30. IEEE (2021)
26. Xiang, S., et al.: Semi-supervised credit card fraud detection via attribute-driven graph representation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 14557–14565 (2023)
27. Yang, W., Zhang, Y., Ye, K., Li, L., Xu, C.-Z.: FFD: a federated learning based method for credit card fraud detection. In: Chen, K., Seshadri, S., Zhang, L.-J. (eds.) BIGDATA 2019. LNCS, vol. 11514, pp. 18–32. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-23551-2_2
28. Zhao, C., Sun, X., Wu, M., Kang, L.: Advancing financial fraud detection: self-attention generative adversarial networks for precise and effective identification. Financ. Res. Lett. **60**, 104843 (2024)
29. Zheng, W., Yan, L., Gou, C., Wang, F.Y.: Federated meta-learning for fraudulent credit card detection. In: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, pp. 4654–4660 (2021)
30. Zhu, H., Liu, G., Zhou, M., Xie, Y., Abusorrah, A., Kang, Q.: Optimizing weighted extreme learning machines for imbalanced classification and application to credit card fraud detection. Neurocomputing **407**, 50–62 (2020)
31. Zioviris, G., Kolomvatsos, K., Stamoulis, G.: Credit card fraud detection using a deep learning multistage model. J. Supercomput. **78**(12), 14571–14596 (2022)

MENGLAN: Multiscale Enhanced Nonparametric Gas Analyzer with Lightweight Architecture and Networks

Zhenke Duan[1,2(✉)], Jiani Tu[1], and Jiqun Pan[1]

[1] Zhongnan University of Economics and Law, Wuhan, China
duanzhenke@sscapewh.com
[2] Wuhan SecureScape Technology Co., Ltd., Wuhan, China

Abstract. Accurate detection of ethylene concentrations in mixed gases is crucial in chemical production for safety and health purposes. Traditional methods are hindered by high cost and complexity, limiting their practical application. This study proposes MENGLAN, a Multiscale Enhanced Nonparametric Gas Analyzer that integrates a dual-stream structure, a Hybrid Multi-Head Attention mechanism, and a Feature Reactivation Module to enable real-time, lightweight, and high-precision ethylene concentration prediction. Results show that MENGLAN achieves superior performance, reduced computational demand, and enhanced deployability compared to existing methods.

Keywords: Gas Detection · Feature Reactivation · Lightweight · Edge-side model

1 Introduction

Ethylene is a highly flammable and toxic gas. Accurate monitoring is essential to prevent accidents and protect health in industrial settings. Traditional detection techniques such as gas chromatography and infrared absorption provide high accuracy but are impractical for real-time applications due to high costs and operational complexity. Recent advancements in electronic nose systems and deep learning models have improved gas detection. However, challenges such as real-time inference and deployment on edge devices remain. This work introduces MENGLAN, a lightweight and efficient analyzer that addresses these challenges.

The electronic nose mimics the human olfactory system. It utilizes highly sensitive sensors to detect and identify odors and volatile organic compounds. Through specific algorithms, it achieves odor classification and quantitative analysis. The electronic nose is widely applied in various fields, including environmental monitoring [1,2], food safety [3–5], and medical diagnostics [6–8], providing effective technical solutions for odor analysis.

In recent years, deep learning has been extensively used in electronic nose systems for gas concentration prediction in gas mixtures. Wu Jilong et al. [9] combined RESHA and ALW, integrating the Hybrid Attention (HA) model

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 339–350, 2026.
https://doi.org/10.1007/978-3-032-04555-3_28

to reweight features. Gan Wenchao et al. [10] integrated the Variational Mode Decomposition (VMD) method with the Long Short-Term Time-Series Network-Attention (LSTNet-Attention) model, enabling high-precision prediction of carbon monoxide and ethylene concentrations. Zeng Liwen et al. [11] developed a Dual-Channel Temporal Convolutional Network (Dual-Channel TCN) based on Temporal Convolutional Networks (TCN), accurately determining the concentrations of multiple gas mixtures. Zhuo Junwei et al. [12] designed the Parametric Rectified Linear Unit (PReLU)-based Multi-Head Attention Temporal Convolutional Network (PMH-TCN) model, effectively predicting gas concentrations. Xiong Lijian et al. [13] introduced the combination of Gramian Angular Field (GASF/GADF) and Convolutional Neural Networks (CNN) into electronic nose data processing, classifying odor intensities. Li Juan et al. [14] proposed a local dynamic neural network model, leveraging a pre-trained autoencoder network to achieve concentration prediction for multiple gases.

Existing gas concentration prediction models are accurate but suffer from slow inference and high computational demands, hindering real-time edge device deployment. Rapid gas concentration prediction is vital for chemical production safety, yet sensing device miniaturization limits computational power. Our proposed lightweight, high-sensitivity, multi-scale enhanced nonlinear gas analyzer addresses these issues. Its dual-stream architecture extracts local and global features, the multi-head attention mechanism processes key information in parallel, and the ReLU activation function improves the model's ability to capture nonlinear patterns in e-nose data. These improvements enable high-precision gas concentration prediction with reduced inference time, facilitating real-time ethylene monitoring and seamless deployment on compact sensing devices.

This study provides the following contributions:

- We adopted a specialized information processing mode, minimizing raw data preprocessing to closely mimic real - world applications. To address the low signal - to - noise ratio, we designed the following modules:
- A dual - stream structure is employed to focus on special local features like mutations and missing data. In the local receptive field convolution feature stream, a new local receptive field information stream is introduced to identify feature patterns. By integrating global and local feature convolutions into one module, we fuse features at different scales, enabling the model to perceive and recognize global characteristics and achieve high - precision measurements of gas mixtures.
- We propose the Hybrid Multi - Head Attention Mechanism (HAMH) to identify self - attention weights across different receptive fields, facilitating the fusion of feature information at various scales.
- We developed the Feature Reconstruction Model (FRM). This module reconstructs the features of local receptive field data and preserves gradient information during model iteration through skip connections, focusing on skipped and low - information segments.

2 Dataset Description and Processing

2.1 Data Description

To evaluate the performance of our proposed ethylene concentration prediction model, we conducted experiments using the public UCI dataset provided by Jordi Fonollosa *et al.* [15]. The dataset contains two types of gas mixtures: ethylene and carbon monoxide in air, as well as ethylene and methane in air. An array of 16 sensors was used to continuously collect signals at a frequency of 100 Hz for 12 h, yielding over 1 million samples.

The measurement system consists of a data acquisition platform, a power control module, and a gas delivery system. The data acquisition platform's measurement chamber is equipped with four types of sensors, namely TGS - 2600, TGS - 2602, TGS - 2610, and TGS - 2620. During the experiment, the power control module maintained a constant voltage of 5 V for the sensors. The gas delivery system injected gases into the measurement chamber at a flow rate of 300 ml/min to capture the time response of different gas concentrations.

Take the preparation of methane - ethylene mixtures as an example. Dry air, methane, and ethylene were fed into a pressurized cylinder controller (MFC) system through separate branches for mixing. The mixed gas was then injected into a 60 - ml measurement chamber. The sensors generated signals upon contact with the gas, which were transmitted to a PC and varied with gas concentration. Subsequently, the gas mixture was vented from the measurement chamber and collected by the exhaust system. The experiment recorded gas samples at 100 Hz for 12 h, resulting in approximately 1 million samples. The heatmap distribution of the dataset is shown in Fig. 1.

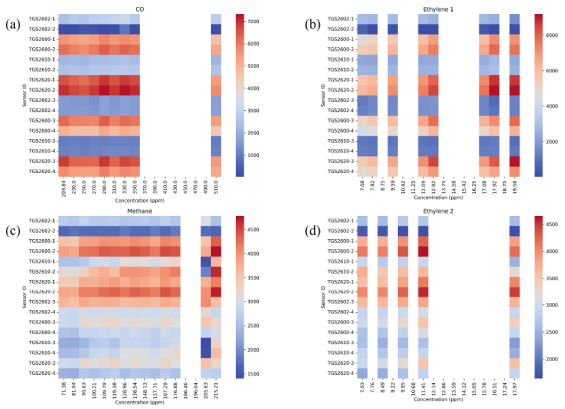


Fig. 1. (a) Thermodynamic distribution map of CO, (b) Thermodynamic distribution map of ethylene in mixed gas 1, (c) Thermodynamic distribution map of methane, (d) Thermodynamic distribution map of ethylene in mixed gas 2.

3 Methodology

3.1 Data Processing

The initial dataset includes 38 different concentration levels, comprising a total of 282 samples. To assess the predictive capability of the model, we retained the original data as much as possible, resulting in a dataset containing 282 samples. The dataset was then standardized using the following equation:

MENGLAN consists of three key components: Dual-Stream Structure for extracting local and global features, HMHA for combining spatial and temporal attention to optimize feature importance, and FRM to enhance feature representation and gradient flow.

3.2 Dual-Stream Structure

We adopt a dual-stream structure to process global and local features through multiple convolutions and max-pooling operations in Fig. 2. This design focuses on local feature irregularities while ensuring the model's global feature recognition and perception capabilities. Additionally, we employ the original ReLU as the activation function instead of other parametric ones (e.g., PReLU).

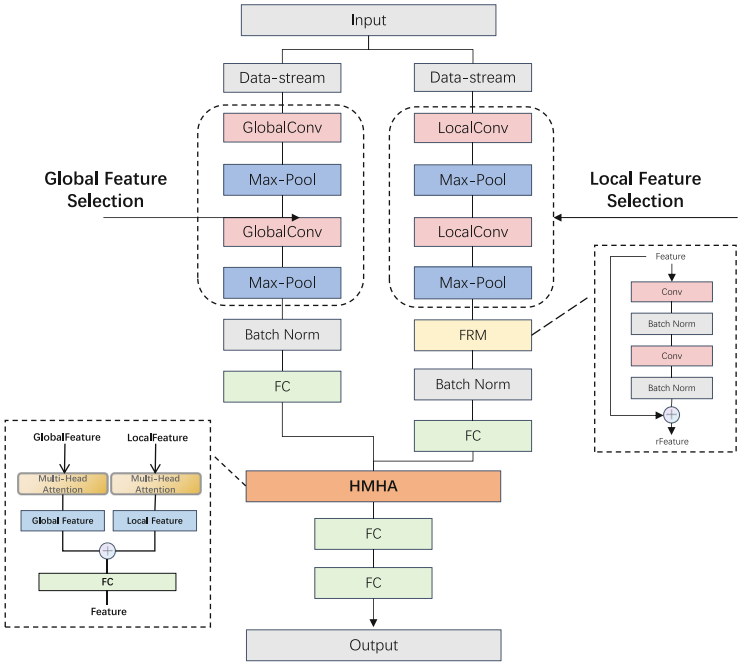


Fig. 2. MENGLAN integrates the FRM module for feature fusion activation and the HMHA mechanism for multi-scale convolutional feature fusion.

Convolution. In the dual-stream structure, we employ global and local convolutions for different feature processing requirements.

Global convolution applies the same kernel parameters across the entire feature dimension, reducing parameters through weight sharing while effectively extracting global patterns. This enables the network to detect similar structures across different spatial positions.

Local convolution (non-shared convolution) uses region-specific kernel parameters, allowing distinct feature representations for different areas. Despite increased parameter complexity, local convolution enhances the model's ability to capture special local features such as abrupt changes and missing data, improving generalization capability.

Max-Pooling. Max-pooling reduces feature map spatial dimensions by selecting the maximum value within sliding windows. This operation retains significant features while reducing computational complexity and improving network efficiency.

3.3 Hybrid Multi-head Attention (HMHA).

To identify self-attention weights across different channels, we propose a Hybrid Multi-Head Attention mechanism (HMHA) in Fig. 2, which integrates feature information from various receptive fields and scales. HMHA effectively enhances the model's representational power and feature integration capabilities by combining multi-head attention with local non-shared convolution operations. This approach allows for efficient extraction of both global contextual information and local position-specific patterns, thereby improving the model's ability to capture complex feature relationships.

Multi-head Attention. Multi-head attention enhances the model's representational and parallelism capabilities. It captures various relationships within the input sequence through multiple parallel attention heads. First, queries, keys, and values are projected into different subspaces via linear transformations. Then, each head computes attention outputs using scaled dot-product attention. The outputs from all heads are concatenated and projected into the final multi-head attention output.

$$\text{Output} = \text{Concat}(\text{Head}_1, \text{Head}_2, \ldots, \text{Head}_h)W^O \tag{1}$$

$$\text{Head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2}$$

where W_i^Q, W_i^K, and W_i^V are the projection matrices, and W^O is the output weight matrix.

Local Non-shared Convolution Operation. In our HMHA mechanism, we employ local non-shared convolutions to capture position-specific patterns. Unlike traditional shared convolutions that use the same kernel across all spatial locations, non-shared convolutions apply different kernels to different spatial regions, enabling location-specific feature extraction.

For an input feature map $X \in \mathbb{R}^{H \times W \times C}$, the non-shared convolution operation is defined as:

$$Y_{i,j} = \sum_{u=-k}^{k} \sum_{v=-k}^{k} X_{i+u,j+v} \odot K_{i,j,u,v} \tag{3}$$

where $Y_{i,j}$ is the output at spatial position (i, j), $K_{i,j,u,v}$ is the position-specific kernel for location (i, j) with offset (u, v), and $\odot$ denotes element-wise multiplication. This approach enables the model to learn spatially-adaptive filters that are particularly effective for capturing local variations in the input data.

3.4 Feature Reactivation Method

The Feature Reactivation Module (FRM) improves feature utilization and network efficiency, especially in lightweight designs, by using skip connections and local data reconstruction. The process is as follows:

1. Apply the first convolution with kernel K_1 to the input features F to obtain O_1:
$$O_1 = F * K_1 \tag{4}$$
2. Normalize O_1 using the first batch normalization layer to obtain BN_1. During training, we use batch statistics:
$$BN_1^{(train)} = \frac{O_1 - \mu_1^{(batch)}}{\sqrt{(\sigma_1^{(batch)})^2 + \epsilon}} \cdot \gamma_1 + \beta_1 \tag{5}$$
During inference, we use running statistics:
$$BN_1^{(test)} = \frac{O_1 - \mu_1^{(running)}}{\sqrt{(\sigma_1^{(running)})^2 + \epsilon}} \cdot \gamma_1 + \beta_1 \tag{6}$$
where $\mu_1^{(batch)}$ and $(\sigma_1^{(batch)})^2$ are the batch-wise mean and variance during training, $\mu_1^{(running)}$ and $(\sigma_1^{(running)})^2$ are the running statistics used during inference, ϵ is a small constant, and γ_1, β_1 are learnable parameters for the first BN layer.
3. Apply the second convolution to BN_1 to get O_2:
$$O_2 = BN_1 * K_2 \tag{7}$$

4. Normalize O_2 using the second batch normalization layer to obtain BN_2. During training:

$$BN_2^{(train)} = \frac{O_2 - \mu_2^{(batch)}}{\sqrt{(\sigma_2^{(batch)})^2 + \epsilon}} \cdot \gamma_2 + \beta_2 \tag{8}$$

During inference:

$$BN_2^{(test)} = \frac{O_2 - \mu_2^{(running)}}{\sqrt{(\sigma_2^{(running)})^2 + \epsilon}} \cdot \gamma_2 + \beta_2 \tag{9}$$

where $\mu_2^{(batch)}$, $(\sigma_2^{(batch)})^2$, $\mu_2^{(running)}$, $(\sigma_2^{(running)})^2$ are independent statistics for the second BN layer, and γ_2, β_2 are its corresponding learnable parameters.

5. Combine BN_2 with the input features F through a residual connection to produce the reactivated features $rFeature$:

$$rFeature = BN_2 + F \tag{10}$$

6. The reactivated features $rFeature$ are subsequently processed through the HMHA mechanism and fed into the final prediction layers. The relationship between $rFeature$ and the model prediction $\hat{y}$ is expressed as:

$$\hat{y} = \text{Decoder}(\text{HMHA}(rFeature)) \tag{11}$$

where HMHA$(\cdot)$ represents the hybrid multi-head attention operation that processes the reactivated features, and Decoder$(\cdot)$ denotes the final prediction layers that map the attention-enhanced features to the target output space.

4 Result and Discussion

In this study, to ensure applicability and real-time performance in industrial scenarios, raw data was not downsampled, and missing information was retained. This approach ensures feasibility in practical use, as it is difficult to distinguish between missing data and systematic errors from measurement instruments. The method's effectiveness and accuracy were validated across two datasets and four prediction targets. Additionally, comparisons with four lightweight algorithms and models of varying parameter sizes demonstrated the superiority of the proposed method. Ablation experiments further illustrated the role and effectiveness of FRM and HMHA. Moreover, the performance of the model under different activation functions was analyzed. Consistent with common practice, we used MSE as the loss function during model updates, defined as Eq. (12).

$$\text{MSE} = \frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2 \tag{12}$$

Here, y_t represents the true value of the target gas concentration, $\hat{y}_t$ denotes the predicted value, and $\bar{y}_t$ indicates the average value of the target gas.

Table 1. Model performance for ethylene in methane/ethylene and carbon monoxide/ethylene mixtures.

Model	Parameters	RMSE	MSE	MAE	R^2	Total Inference	Average Inference
Methane + Ethylene Mixture							
MENGLAN	8.93 MB	1.01	1.02	0.221	0.966	44.44 s	0.00005 s
MENGLAN	21.83 MB	0.988	0.996	0.212	0.967	44.56 s	0.00005 s
MENGLAN	71.63 MB	0.998	0.977	0.203	0.968	43.05 s	0.00005 s
2-ANN	17.02 MB	1.475	2.175	0.605	0.928	44.74 s	0.00005 s
ANN	13.55 MB	1.193	1.423	0.343	0.953	37.85 s	0.00005 s
CNN	4.31 MB	1.048	1.099	0.247	0.963	40.00 s	0.00005 s
CNN-RES	14.31 MB	1.030	1.062	0.240	0.965	47.65 s	0.00006 s
Carbon Monoxide + Ethylene Mixture							
MENGLAN	8.93 MB	0.790	0.625	0.161	0.980	53.67 s	0.00006 s
MENGLAN	21.83 MB	0.778	0.606	0.144	0.980	41.37 s	0.00005 s
MENGLAN	71.63 MB	0.772	0.596	0.143	0.981	40.86 s	0.00005 s
2-ANN	17.02 MB	1.548	2.398	0.710	0.924	43.71 s	0.00005 s
ANN	13.55 MB	0.815	0.664	0.172	0.978	39.12 s	0.00005 s
CNN	4.31 MB	1.097	1.20	0.318	0.961	37.32 s	0.00004 s
CNN-RES	14.31 MB	0.799	0.639	0.158	0.979	35.92 s	0.00004 s

4.1 Evaluation Indicators

To comprehensively evaluate model performance, several indicators were used. Root Mean Square Error (RMSE) measures the difference between predicted and true values; R^2 quantifies the model's fit to the dataset. Lower RMSE and SMAPE values indicate better fit, while R^2 values closer to 1 signify stronger fitting capability. Additionally, total and average inference time during the reasoning phase was considered to assess the model's deployment efficiency under lightweight conditions. These are defined as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2} \tag{13}$$

$$R^2 = 1 - \frac{\sum_{t=1}^{N}(y_t - \hat{y}_t)^2}{\sum_{t=1}^{N}(y_t - \bar{y}_t)^2} \tag{14}$$

4.2 Basic Information About the Experiment

All experiments were conducted using an NVIDIA Tesla V100 GPU. The datasets were divided into training, validation, and testing sets in a 6:2:2 ratio. An early stopping strategy with a patience of 15 epochs was employed to supervise the learning process. Python version 3.8.5 and PyTorch version 2.0.0 were used. The network was trained for a maximum of 200 epochs with a batch size

of 512. We used the Adam optimizer with a learning rate of 0.01 and decay rate of 0.08. The multi-head attention mechanism employed 8 attention heads, and ReLU was used as the activation function. A dropout rate of 0.2 was applied to prevent overfitting, with default values used for all other unspecified parameters.

Table 2. Prediction performance of different target gases in CO/ethylene and methane/ethylene mixtures

TARGET	Model List	RMSE	MSE	MAE	R^2	Average Inference
CO + Ethylene Mixture						
CO	w/o FRM	23.144	535.652	4.639	0.979	0.00006 s
CO	w/o HMHA	23.049	531.272	4.769	0.979	0.00004 s
CO	w/o FRM & HMHA	23.391	547.147	5.788	0.978	0.00004 s
C_2H_4	w/o FRM	0.876	0.769	0.221	0.975	0.00004 s
C_2H_4	w/o HMHA	0.931	0.867	0.237	0.972	0.00005 s
C_2H_4	w/o FRM & HMHA	0.972	0.945	0.244	0.970	0.00005 s
Methane + Ethylene Mixture						
CO	w/o FRM	16.874	284.751	4.498	0.951	0.00004 s
CO	w/o HMHA	16.770	281.264	3.998	0.952	0.00004 s
CO	w/o FRM & HMHA	16.935	286.827	4.095	0.951	0.00004 s
C_2H_4	w/o FRM	1.024	1.048	0.258	0.965	0.00006 s
C_2H_4	w/o HMHA	1.274	1.623	0.360	0.946	0.00006 s
C_2H_4	w/o FRM & HMHA	1.276	1.629	0.372	0.946	0.00005 s

4.3 Experimental Results

Comparison with Other Methods This work compares the prediction performance for ethylene concentration across two datasets with ANN [16,17], CNN [18], 2-ANN [19], CNN-RES, and MENGLAN models of different parameter sizes. Tables 1 present the results.

From Tables 1, MENGLAN outperforms all models in RMSE, MSE, MAE, and R^2, showcasing its effectiveness and scalability. Figures 3 illustrate training error trends and performance comparisons for models with varying parameter sizes, respectively.

Ablation Experiment. In this section, we further ablate different modules to validate their computational contributions, keeping all other experimental settings consistent.Table 2 illustrate the prediction performance of different target gases in two gas mixtures, respectively.

Table 2 indicate that the FRM and HMHA methods significantly enhance computational efficiency. Excluding the FRM module reduces the model's ability to extract local features, making it challenging to capture abrupt signals from raw data, thereby degrading performance. Similarly, excluding the HMHA module diminishes the model's capacity for weighted integration of global and local features, leading to performance loss.

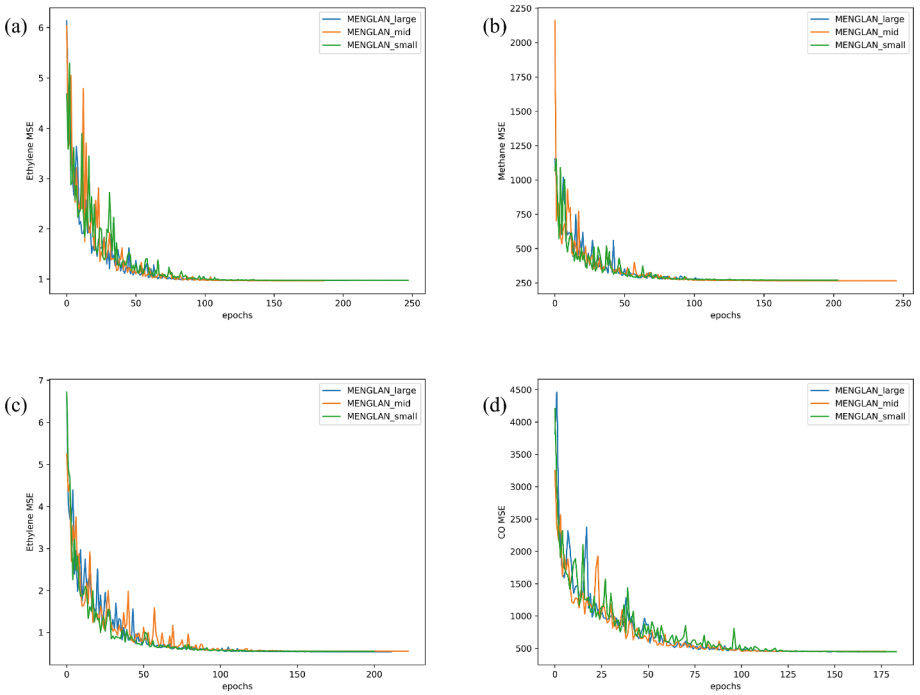


Fig. 3. Error comparison of models with different parameter amounts during the training process: (a) MSE for ethylene, (b) MSE for methane, (c) MSE for ethylene, (d) MSE for CO.

Table 3. Performance of predicting gas concentrations in the mixture using three different activation functions on the test set.

Activation	TARGET	RMSE	MSE	MAE	R^2	**Total inference (s)**
tanh	C2H4	1.683	2.834	0.657	0.907	40.16
	CH4	28.301	800.964	11.254	0.864	49.49
	C2H4	2.279	5.189	1.050	0.836	41.71
	CO	69.335	4807.373	30.625	0.814	41.58
sigmoid	C2H4	1.788	3.196	0.680	0.895	41.29
	CH4	40.040	1603.257	19.730	0.727	40.70
	C2H4	4.491	20.171	3.607	0.3627	41.42
	CO	160.736	25836.163	148.666	1.19×10^{-7}	41.50
softsign	C2H4	1.451	2.107	0.470	0.931	41.61
	CH4	24.000	576.009	8.559	0.902	41.11
	C2H4	1.261	1.589	0.388	0.950	41.79
	CO	43.771	1915.899	15.913	0.926	42.07

Comparison of Different Activation Functions. Considering the significant impact of activation functions on model performance, we explore the effects of replacing the ReLU function in the model with Sigmoid, Softsign, and Tanh functions. Experiments were conducted with consistent parameter settings, and the results are presented in Table 3. The experimental conclusions can be drawn from a comparison of Tables From 1 and 3. It can be observed that ReLU consistently outperforms other activation functions under various conditions. Therefore, when ReLU is chosen as the activation function, the MENGLAN model consistently achieves satisfactory results across different scenarios.

5 Conclusion

This study successfully developed MENGLAN, an instrument for real-time ethylene concentration prediction in chemical production gas mixtures. Its dual-stream structure, HMHA mechanism, and FRM model minimized preprocessing steps and achieved high accuracy and fast inference speeds. Experimental results showed reduced prediction errors and improved dataset fitting. MENGLAN offers a feasible solution for edge device deployment, enhancing production safety and efficiency. Future work could explore scalability and further optimization.

References

1. Guillem, D.G., Thanh, D.N., Jacob, W.J., et al.: Electronic nose for improved environmental methane monitoring. Environ. Sci. Technol. **57**(12), 4567–4576 (2023)
2. Zhichong, Y., Xu, T., Yichun, G., Xuehong, Y., Zhenming, X., Lingen, Z.: Monitoring the resources and environmental impacts from the precise disassembly of e-waste in China. Environ. Sci. Technol. **57**(22), 8900–8909 (2023)
3. Lin, L., Zhanqiang, H., Xianqiao, H., Dan, L., Shiyi, T.: Electronic tongue and electronic nose for food quality and safety. Food Res. Int. **154**(PB), 112–214 (2022)
4. Gui-Li, H., Tian-Tian, L., Xiao-Mei, M., et al.: Insights into the volatile flavor and quality profiles of loquat (Eriobotrya japonica Lindl.) during shelflife via HS-GC-IMS, E-nose, and E-tongue. Food Chem. X **100886**, 1–11 (2023)
5. Ali, M.M., Hashim, N., Aziz, S.A., Lasekan, O.: Principles and recent advances in electronic nose for quality inspection of agricultural and food products. Trends Food Sci. Technol. **99**(1), 1–10 (2020)
6. Teixeira, R.C., Rodríguez, M., de Romero, N.J., et al.: The potential of a portable, point-of-care electronic nose to diagnose tuberculosis. J. Infect. **5**(441), 447 (2017)
7. van Geffen, W.H., Huib, K.: Diagnosing viral and bacterial respiratory infections in acute COPD exacerbations by electronic nose. Euro. Respir. J. **s60** (2016)
8. Hao, S., Fengchun, T., Zhifang, L., et al.: Sensor array optimization of electronic nose for detection of bacteria in wound infection. IEEE Trans. Industr. Electron. **9**(7350), 1–9 (2017)
9. Wu, J., Zhao, W., Wu, F., et al.: A mixed gas concentration regression prediction method based on RESHA-ALW. Sens. Actuators B Chem. **136222**, 1–10 (2024)
10. Gan, W., Ma, R., Zhao, W., et al.: A VMD-LSTNet-attention model for concentration prediction of mixed gases. Sens. Actuators B Chem. **136641**, 1–12 (2025)

11. Liwen, Z., Yang, X., Sen, N., Min, X., Pengfei, J.: A mixed gas concentration regression prediction method for electronic nose based on two-channel TCN. Sens. Actuators B Chem. **382**, 133528 (2023)
12. Zhuo, J., Chen, X., Zhang, H., et al.: Rapid and high-accuracy concentration prediction of gas mixtures based on PMH-TCN. Measurement **242**, 116003 (2025)
13. Lijian, X., Meng, H., Can, H., Yuxin, H., Shaoyun, H., Xiuying, T.: Image presentation and effective classification of odor intensity levels using multichannel electronic nose technology combined with GASF and CNN. Sens. Actuators B. Chem. **395**, 134492 (2023)
14. Juan, L., Yilun, M., Zaihua, D., et al.: Local dynamic neural network for quantitative analysis of mixed gases. Sens. Actuators B Chem. **135230**, 1–9 (2024)
15. Fonollosa, J., Sheik, S., Huerta, R., Marco, S.: Reservoir computing compensates slow response of chemosensor arrays exposed to fast varying gas concentrations in continuous monitoring. Sens. Actuators B. Chem. **618**, 629 (2015)
16. Ren, W., Zhao, C., Niu, G., Zhuang, Y., Wang, F.: Gas sensor array with pattern recognition algorithms for highly sensitive and selective discrimination of trimethylamine. Adv. Intell. Syst. **4**, 2200169 (2022)
17. Liu, Y., Yang, Z., Huang, L., Zeng, W., Zhou, Q.: Anti-interference detection of mixed NOX via In2O3-based sensor array combining with neural network model at room temperature. J. Hazard. Mater. **463**, 132–857 (2024)
18. Chen, Z., He, Z., Lu, Z.-M.: DEA-Net: single image dehazing based on detail-enhanced convolution and content-guided attention. IEEE Trans. Image Process. **33**, 1002–1015 (2024)
19. Zeng, L., Xu, Y., Ni, S., Xu, M., Jia, P.: A mixed gas concentration regression prediction method for electronic nose based on two-channel TCN. Sens. Actuators B. Chem. **382**, 133–528 (2023)

Targeted Trust-Based Merging of Customers' Opinions

Jurij Ružejnikov[1,2(✉)] and Tatiana Valentine Guy[2]

[1] Czech University of Life Sciences Prague, Kamýcká 129, 165 00 Praha - Suchdol, Czech Republic
{ruzejnikov,guy}@utia.cas.cz
[2] Institute of Information Theory and Automation, Czech Academy of Sciences, Pod Vodárenskou věží 4, 182 08 Praha 8, Czech Republic

Abstract. In this article, we investigate how a rational agent forms their opinion based on prior knowledge, available information, and the opinions of other agents. We propose methodology of how to purposefully merge agent's opinion and expert opinions. We describe the agent's opinion and the opinions of experts in the form of distributions. Formulating opinion formation as a decision-making task and solve it using Fully Probabilistic Design (FPD). To demonstrate our approach, we apply the solution on simulated data describing features of mobile phone brands. Methodology is verified on a test bed example of choosing a mobile phone brand based on expert opinions while taking into account agent's trust in experts.

Keywords: Decision-making · Fully probabilistic design · Knowledge representation · Opinion dynamics · Opinion merging · Trust

1 Introduction

Opinion dynamics is a compelling field that explores how individual beliefs and preferences evolve through interactions and information exchange. Among the various methods employed to model these dynamics, linear pooling of opinions [1] stands out as a widely used technique. This method aggregates expert opinions into a single, coherent viewpoint by calculating a weighted sum of their inputs. Similarly, logarithmic opinion pooling [3] employs the weighted geometric mean to synthesize expert insights, providing a robust framework for consensus modelling.

Beyond these foundational approaches, innovative strategies such as consensual modelling [5] have emerged, leveraging social trust networks to enhance group decision-making processes. For example, in [14], experts collaborate to form a consensus, akin to an individual seeking diverse opinions from friends before making a significant purchase, such as a new computer.

Another compelling method, merging opinions by social sampling of posteriors [13], introduces a dynamic way to update opinions based on external information sources, paralleling the challenges we address in this paper. Additionally,

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 351–365, 2026.
https://doi.org/10.1007/978-3-032-04555-3_29

Bayesian updates can refine a modeller's opinion by integrating expert data [9], a concept we will explore further. A notable example is the Bayesian hierarchical model [11], which uses Kullback-Leibler divergence to objectively determine the influence of each expert and produce a calibrated consensus from multiple sources. Our approach, however, addresses the challenge from a distinct, single-agent perspective. Instead of seeking an objective consensus, we focus on how a rational agent personally forms an opinion by merging their own prior knowledge with the subjective trust they place in external experts. We explore how this agent can purposefully integrate diverse opinions, such as a consumer leveraging friend recommendations or online reviews before purchasing a new mobile phone. This paper develops a framework based on divergence minimization [2] and Fully Probabilistic Design [8] to model this subjective process and enhance the merging of opinion-defining distributions, ultimately aimed at improving decision-making outcomes.

The variety of proposed solutions highlights that the problem is far from resolved. In our approach, we emphasize that the data represent *subjective* opinions rather than objective truths. This distinction is essential, as it enables a nuanced update process that integrates expert insights while also reflecting the modeller's trust in these opinions and their own level of certainty. We quantify opinions as parametrised distributions of a random variable that encapsulates the subject at hand.

To showcase the effectiveness of our method, we will use the selection of a mobile phone brand as a practical case study, drawing on data collected from real participants. This scenario not only demonstrates the application of our approach but also underscores the complexities involved in merging diverse opinions in a real-world context. By addressing these challenges, we aim to provide valuable insights into the decision-making process and contribute to the ongoing conversation in opinion dynamics. The **contributions of the paper** are as follows:

- The paper introduces the innovative framework for dynamical merging of opinions that effectively differentiates between subjective insights and objective data. It allows for a nuanced integration of expert opinions while respecting the targeted scenario that involves multi-attribute preferences and expert prior opinions. This proposed approach not only enhances decision-making but also empowers users to make *informed choices* based on understanding of expert insights.
- The paper explores the role of the modeller's trust in expert opinions as well as their own level of certainty, emphasizing how these factors significantly impact the opinion merging process and shape decision outcomes. The research highlights the important relationship between trust and certainty, offering clear insights that can improve decision-making processes.
- Through a series of experiments conducted on real participant data, it validates the proposed method and demonstrates its effectiveness. This empirical validation not only demonstrates the effectiveness of the framework but also showcases its practical applicability in real-world scenarios.

Here are several potential use cases that showcase the relevance of the proposed framework for *dynamical* opinion merging to the themes of the ICANN conference, particularly in the context of AI and machine learning:

- *Medical diagnosis:* specialists like radiologists and oncologists often provide critical opinions for treatment decisions. The proposed framework can effectively merge this knowledge, considering each expert's trustworthiness and certainty. This leads to more accurate and reliable treatment plans, especially in complex cases with few data.
- *Risk assessment:* Financial analysts frequently depend on expert opinions to evaluate investment risks. The framework can merge these insights while incorporating trust scores based on past performance and domain expertise. This refined approach enhances risk assessment models, resulting in better investment decisions and improved financial outcomes.
- *Product development and marketing:* companies can merge expert opinions about product features and market trends with consumer insights to select product development and marketing strategies. Integrating consumer opinions with expert insights ensures that products meet market needs while aligning with consumer preferences.

The paper layout is as follows. Section 1 briefly outlines the contemporary state of the art in the field of opinion merging. Section 2 introduces an example of the target problem and outlines the essential theory behind opinion merging. Section 3 adapts this theory to the specific context of merging opinions about mobile phone brands, providing a tailored framework for our analysis. Section 4 discusses modeller's trust in experts' opinions and their own level of certainty, highlighting the importance of these factors in the merging process. Section 5 presents the experimental results, comparison with weighted linear pooling [1] and discusses advantages of proposed method. Conclusion summarises the key findings from these experiments and suggests potential avenues for future research.

Conventions and notation

$\mathbb{N}_0$ set of natural numbers including 0 is denoted; $\theta \in \boldsymbol{\Theta}$ is a model parameter and $\boldsymbol{\Theta}$ is the set of parameters; $X = (\mathcal{B}, F_1, \ldots, F_n)$ is a discrete multivariate random variable, where $\mathcal{B}$ represents the brand ($b = 1, \ldots, m$) and each F_j ($j = 1, \ldots, n$) represents a feature score, with scores $f_{j,b}$ taking values in $\{1, \ldots, k\}$; $P, \mathcal{A}, \mathcal{S}$ represent probability distributions; The Kullback-Leibler divergence between distributions P and Q is expressed as $\mathcal{D}(P||Q)$; The subscript I indicates that a function is related to an agent, referred to as the modeller, while the subscript E signifies a relation to an external source of information, such as an expert; Additionally, the superscript $\cdot^U$ denotes the updated version of the corresponding variable. A strict preference relation between two outcomes is written as $a \succ b$;

2 Opinion Merging Task

In today's market, consumers face the challenge of choosing the best product while balancing their personal preferences with expert evaluations. Our goal is to support informed decision-making that reflects both individual beliefs and external insights/knowledge/experience. In this contribution, we focus on the evaluation of mobile phone brands by integrating personal preferences with expert assessments.

Imagine you're in the market for a new phone. You have your own ideas about which brand might be the best based on your experiences and what you've heard. However, you also have a tech-savvy friend who shares their opinion on which phone they believe is superior. Your aim is to combine your own thoughts with your friend's advice to make a more informed choice.

In this scenario, you act as the modeller, holding your own initial preferences about phone brands, while your friend's opinion serves as the external opinion. The process of merging these two perspectives can be seen as blending personal preferences with expert advice, ultimately leading to a decision that considers both viewpoints. You might weigh your friend's advice differently based on how much you trust their expertise. This approach ensures that your final decision reflects both your personal preferences and your trust in external insights.

We examine a scenario in which two independent agents each has an opinion about a random variable, denoted by X. One of these agents, the *modeller*, seeks to update their opinion based on two sources: data they have observed directly and the external opinion provided by the other agent, referred to as the *expert*. The objective is to understand how the modeller's updated opinion is formed by integrating these two sources of information.

The modeller's preference is represented as a joint distribution $P_I(X, \theta)$, where $\theta \in \Theta$ is a parameter representing the modeller's uncertain opinion. The distribution can be decomposed using the chain rule as:

$$P_I(X, \theta) = P_I(X|\theta)A_I(\theta), \tag{1}$$

where $P_I(X|\theta)$ is the modeller's opinion about X, and $A_I(\theta)$ is the modeller's prior belief about θ. The result of merging the modeller's opinion with the external opinion is represented by the joint distribution $P_I(X, \theta, A|P_E)$, which can be decomposed as follows:

$$P_I(X, \theta, A|P_E) = P_I(X|\theta, A, P_E)P_I(\theta|A, P_E)P_I(A|P_E).$$

Incorporating expert opinion $P_E(X)$ into $P_I(X, \theta)$ implies updating the prior distribution $A_I(\theta)$ to $A_I^U(\theta|P_E)$. Paper [8] address this problem by using Fully probabilistic design of hierarchical Bayesian models. It constructs a so-called hyper-prior $S_I(A|P_E)$ on unknown A and provides a justified way how to find its optimal $A_I(\theta|P_E)$. Then the optimal updated opinion derived using [8] reads:

$$P^U(X,\theta,A|P_E) = P_I(X)A_I^U(\theta|P_E)S_I^U(A|P_E), \text{ where}$$
$$S_I^U(A|P_E) \propto S_I(A|P_E)\exp\left(-\mathcal{D}(A||A_I^U)\right) \tag{2}$$

describes how to select appropriate update of the modeller prior (1) based on the expert's opinion. The updated belief of the modeller about Θ then reads

$$A_I^U(\theta|P_E) \propto P_I(\theta)\exp\left(\sum_x \ln(P_I(x|\theta))P_E(x)\right) \tag{3}$$

Meaning the resulting updated belief is a modification of the original prior $P_I(\theta)$ by exponential mixture of the modeller's original opinion and the opinion of an expert. The exponential function here appears from KLD minimization of opinion $P_I(X,\theta,A)$ (original opinion) and $P_I(X,\theta,A|P_E)$ (opinion modified by expert). Exact derivation can be found in [8].
The result of merging the modeller's opinion about X with the expert's $P_E(X)$ can be found as:

$$P_I^U(X) = \int_{\Theta\times\mathcal{A}} P_I(X|\theta)A_I^U(\theta|P_E)S_I^U(A|P_E)d\theta dA.$$

3 Merging Opinions: Selecting a Mobile Phone Brand

To address the task of selecting mobile phone brands described in Sect. 2 we introduce a discrete multivariate random variable, $X = (\mathcal{B}, F_1, \ldots, F_n)$, where $\mathcal{B}$ represents the mobile phone brand and $F_j, j = \{1,\ldots,n\}$ denotes feature scores related to quality aspects such as camera performance, battery life, or display quality. The brands are indexed as b (ranging from 1 to m, with m being the total number of brands), and each feature score for brand b is represented as $f_{j,b}$ (with scores ranging from 1 to k, where k indicates the possible score range).

The modeller's opinion $P_I(X|\theta)$ reflects their belief about brands and corresponding features.

$$P_I(\mathcal{B},F_1,\ldots,F_n|\theta) = \underbrace{P_I(\mathcal{B}|\theta)}_{\theta_{\mathcal{B}}}\underbrace{P_I(F_1|\mathcal{B},\theta)}_{\theta_{F_1}}\cdots\underbrace{P_I(F_n|\mathcal{B},\theta)}_{\theta_{F_n}}, \tag{4}$$

where $P_I(\mathcal{B}|\theta)$ indicates brand preference and $P_I(F_j|\mathcal{B},\theta)$ reflects opinion on specific features. Note the opinions are assumed conditionally independent. Let us consider $\theta = (\theta_{\mathcal{B}}, \theta_{F_1}, \ldots, \theta_{F_n})$ be an unknown parameter. Then prior belief $P_I(\theta)$, which captures the modeller's initial uncertainty, can be modelled using a Dirichlet distribution:

$$P_I(\theta) \propto \prod_b \theta_b^{n_b-1} \prod_{\{f_{j,b}\}_{j=1}^n} \theta_{f_{j,b}}^{n_{f_{j,b}}-1}. \tag{5}$$

In (5) weights $n_\bullet$ express the modeller's preference for brand n_b and particular feature $f_{j,b}$. The expert provides opinion $P_{E_i}(X), i \in \{1, \ldots, l\}$ about X

$$P_{E_i}(X) = P_{E_i}(\mathcal{B}) \prod_{j=1}^{n} P_{E_i}(F_j|\mathcal{B}). \tag{6}$$

In practice, each expert's opinion is often specific, assigning a particular brand and its associated feature scores directly. Therefore, we assume that the distributions representing experts' opinions about phone brands and features are degenerate.

Model $S_I(A|P_{E_i})$ (2) remains consistent regardless of the external opinion, reflecting a stable approach to incorporating new insights:

$$S_I(A|P_{E_i}) \propto \prod_{b} \prod_{\{f_{j,b}\}_{j=1}^{n}} \theta_{f_{j,b}}^{s_{f_{j,b}}-1}. \tag{7}$$

In (7) parameters $s_{f_{j,b}} \in \langle 0, 2\rangle$ quantify the modeller's trust in the expert's opinion and confidence in their own prior beliefs, specification of quantification is presented in Sect. 4 below. The updated prior, $A_I^U(\theta|P_{E_i})$, after merging with an expert's opinion (6) can be derived as:

$$A_I^U(\theta|P_{E_i}) \propto \frac{1}{\mathcal{N}_{A_I^U(\theta|P_{E_i})}} \theta_b^{n_b-1} \prod_{\{f_{j,b}\}_{j=1}^{n}} \theta_{f_{j,b}}^{V_{f_{j,b}}-1},$$

where updated weights $V_{f_{j,b}}$ incorporate both prior weights and the expert's influence:

$$V_{f_{j,b}} = n_{f_{j,b}} + s_{f_{j,b}} + \delta(b, b_{E_i})\delta(f_{j,b}, f_{E_i,j,b_{E_i}}) - 1.$$

The detailed steps of this derivation can be found in Appendix A. The final updated opinion of the modeller is represented as:

$$P_I^U(X = x) = \frac{n_{b_I}}{\sum_b n_b} \prod_{j=1}^{n} \frac{V_{f_{j,b_I}}}{\sum_{f_{j,b_I}} V_{f_{j,b_I}}}.$$

This formula encapsulates how the modeller balances their own prior beliefs with the influence of expert opinions.

After merging opinions, the updated brand preference is given by the Bayes rule:

$$P_I^U(\mathcal{B}|F_1, \ldots, F_n) = \frac{P_I^U(F_1|\mathcal{B}) \cdots P_I^U(F_n|\mathcal{B}) P_I^U(\mathcal{B})}{\sum_b P_I^U(F_{1,\mathcal{B}=b}|\mathcal{B} = b) \cdots P_I^U(F_{n,\mathcal{B}=b}|\mathcal{B} = b) P_I^U(\mathcal{B} = b)}. \tag{8}$$

Distribution (8) depends on the specific realizations of feature scores $F_1, \ldots, F_n$. These feature scores need to be selected appropriately to extract the modeller's preference for a particular brand $\mathcal{B}$. We assume that the modeller selects the feature score $f_{j,b}$ with the highest probability $P_I^U(F_{j,b}|\mathcal{B})$, i.e.

$$f_{j,b}^{opt} = \arg\max_{f_{j,b}}(P_I^U(F_{j,\mathcal{B}=b} = f_{j,b}|\mathcal{B} = b)), \tag{9}$$

for all brands $b \in \{1, \ldots, m\}$.

4 Trust and Certainty

Trust in the opinion provided by the expert P_{E_i} is $t_{E_i} \in \langle 0, 1\rangle$. The lower value of t_{E_i} indicates a low trust in the opinion of experts E_i. Higher indicates otherwise. The trust value t_{E_i} can encapsulate various aspects of the relationship between the modeller and the expert E_i, such as friendship, prior knowledge of the expert's performance, or other subjective criteria set by the modeller. [1] For the purpose of experimentation, trust values t_{E_i} are predefined to specific values to illustrate the potential integration of trust t_{E_i} into the opinion merging process. The certainty of the modeller in their opinion $P_I(X)$, is represented by $c_{I,j,b} \in \langle 0, 1\rangle$. A lower value of $c_{I,j,b}$ indicates diminished certainty in their opinion $P_I(X)$, while a higher value signifies the opposite. Inclusion of both trust t_{E_i} and opinion certainty $c_{I,j,b}$ into s_{f_j} is carried out through a linear combination: $s_{f_j} = t_{E_i} + c_{I,j,b}$. The preference sub-setting process dictates how the modeller chooses f_j that form the modeller's preference on mobile brands during the opinion merging. The process of preference sub-setting is accomplished using the modeller's score preference defined as $r_{I,j} \in \{1, 2, \ldots, k\}$, representing the preference for the feature score F_j. This can be intuitively described as the modeller's inclination toward a specific range of scores f_j for a feature j across all mobile phone brands $b \in \mathcal{B}$. The score preference $r_{I,j}$ restricts the range for selecting scores $f_{j,b}^{\star}$ in (9) to $f_{j,b}^{\star} \geq r_{I,j}$.

Example (mobile phone selection task)

Let preference score $r_{I,j} = 4$ for quality of mobile phone display, the preferred feature score are highlighted in green:

Feature scores	1	2	3	4	5	6

. Consequently, this influences selection of the probabilities $P_I^U(F_j|\mathcal{B})$ entering the modeller's prior (8). These probabilities may assume a low value if the modeller's initial opinion P_I combined with expert's opinion P_{E_i} is outside the selection range defined by the score preference $r_{I,j}$.

5 Experiments

To showcase the functionality of the proposed opinion merging framework for the task of selecting a mobile phone based on the quality of its features, the following experimental setup on simulated data is considered. For comparison with the proposed method in this work, weighted linear pooling [1] of opinions

[1] It is important to note that the process of determining trust value t_{E_i} is not elaborated on in this work. Extraction of trust value can be achieved for example by similarity (dissimilarity measures) such solution can be found in [6,7,10].

was also implemented. The weights for the opinions of individual experts are given by the trust values t_{E_i}, while the weight for the opinion of the modeller is set to the certainty parameter $c_{I,j,b}$. Both parameters are defined as described in Sect. 4. Because the results obtained by applying both methods are similar, in the subsequent subsections we present the results of the opinion merging method proposed in this work, and discuss the comparison at the end of this section.

The simulated data were set up based on data collected from human participants in a survey about quality of features of several mobile brands. Showcase of proposed solution on simulated data was deemed to provide more insight into correctness of the proposed solution. The Python Jupyter notebook containing the experimental code and data is available at [12].

Let there be a person (the modeller) who is interested in buying a mobile phone. He chooses to be advised by their friends, colleagues and information on the internet. The modeller has a prior preference about mobile brands $\{B_1, B_2, B_3\}$ each having 3 features $\{F_1, F_2, F_3\}$. External information is in form of a score for each feature, the score scale is from 1 to 6, where 1 means the worst and 6 means the best. The experts E_i scores are set so that the solution to the selection process can be easily verified. The expert score tables applied in all Experiments are in Appendix C and are configured to the following ordering: $B_1 \succ B_2 \succ B_3$. The modeller's initial weights $n_{f_{j,b}}$ are set to 1 for each score. For simplicity the preference score $r_{I,j} = 4$ for all experiments, meaning the modeller prefers score of 4 or above for all the features. and brand preference $P_I(\mathcal{B})$ is configured to uniform distribution to ensure that the preference $P_I(\mathcal{B}|F_{1,\mathcal{B}}, \ldots, F_{n,\mathcal{B}})$ is not influenced by the modeller's overall preference of brand. That is, only the preference based on features is considered.

5.1 Opinion Merging and Preference Subsetting

This section illustrates the influence of expert opinion on the modeller's brand preference. In the following experiment, the opinions are merged without considering modeller's trust in expert opinions and without considering certainty of the modeller into his opinion. These factors will be investigated in the subsequent experiments.

Experiment 1: The aim of this experiment is to test if this ordering is shifted towards B_1 in the final updated brand preference (8). The modeller's prior preference on brands is given by the ordering: $B_2 \succ B_1 \succ B_3$ and can be found in Table: 6. Experts slightly prefer B_1 to B_2, which is a setup for the later demonstration of the trust value t_{E_i} (see Sect. 5.2). The result of final preference of mobile brands is in Fig. 1. The results suggest that the experts influenced the modeller to favour the B_1 brand. Which is the expected outcome.

5.2 Inclusion of Trust

In this section, we examine the integration of trust t_{E_i} for each expert E_i's opinion. In the following two experiments, the opinions are merged without con-

sidering modeller's certainty in his opinion. For simplicity, the setup of modeller's initial scores for the following two experiments is the same as in the first experiment.

Experiment 2: We start by configuring low trust values t_{E_i} for experts reacting to the B_1 brand and high trust values t_{E_i} for experts reacting to the B_2 and B_3, specific trust scores can be found in Table 4. The expected result is that the preferred brand should be the B_2, since it has similar score values provided by experts E_i as B_1, with the B_2 being slightly less favoured. The result of final preference of mobile brands is in Fig. 2. The result of this experiment complies with the described expectation.

Experiment 3: Setting the trust t_{E_i} for experts reacting to the B_1 and B_2 brands to low values and to high values for B_3, specific trust scores can be found in Table 5. The result of final preference of mobile brands is in Fig. 3. Both experiments complied with the described expected results. The trust process works as expected.

5.3 Inclusion of Certainty

To better demonstrate the proposed solution, trust values t_{E_i} are intentionally excluded. When the trust value is not set, it is equivalent to setting the trust t_{E_i} to the maximum value of 1.

Experiment 4: This experiment illustrates the impact of the certainty $c_{I,j,b}$ on the final ordering of brands. The modeller's opinion is configured to prefer the brands in the following order: $B_2 \succ B_3 \succ B_1$ and is presented in Table 7. Setting the maximum certainty of opinion $c_{I,j,b} = 1$ in the low scores of the modeller for the B_1 brand should lead to the preference for the B_2 as the main brand, with B_3 being the second most preferred brand. The modeller's certainty $c_{I,j,b}$ for each respective brand B_1, B_2, B_3 is:

Certainty	100%	40%	10%

. This can be intuitively interpreted as the modeller being less receptive to the advice of experts E_i about B_1. The result of final preference of mobile brands is in Fig. 4. The result of this experiment complies with the described expectation.

Experiment 5: In this experiment the modeller's opinion is configured to prefer the brands in the following order: $B_3 \succ B_2 \succ B_1$ and can be found in Table 8. Maximum certainty $c_{I,j,b}$=1 is applied to the modeller's high scores for B_3 brand and should lead to the highest preference for the B_3 brand. The modeller's certainty $c_{I,j,b}$ for each respective brand B_1, B_2, B_3 is:

Certainty	20%	20%	100%

. High values of certainty $c_{I,j,b}$ indicate that the modeller is very confident in brand B_3 and, therefore, unlikely to change his opinion, even if the experts overwhelmingly prefer other brands. The result of final preference of mobile brands is in Fig. 5. The result of this experiment complies with the described expectation.

5.4 Comparison of Linear Pooling and FPD

The resulting brand preference $P_I(\mathcal{B} \mid F_{1,\mathcal{B}}, \ldots, F_{n,\mathcal{B}})$ after applying linear opinion pooling was essentially the same as that obtained using the proposed method. The simple average difference in the preference for individual brands across all experiments are almost negligible:

Difference	0.041%	0.023%	0.034%

. This outcome is expected, as the implementation of both methods in this work is nearly identical. The difference arises from the initialization of opinions. In the FPD-based method, the modeller's opinion is initialized with non-zero weights using a Dirichlet prior, and the resulting probabilities are calculated as mean estimates of the parameters. In contrast, linear opinion pooling constructs the probability distribution directly, without parameter estimation. This result validates the proposed method. The main distinction is that the method presented here provides a theoretically sound Bayesian formulation, allowing for online updating of the modeller's opinion in a dynamic setting, which is subject of future research, see [4]. Achieving a similar effect with the linear pooling method would require recalculating the opinion mixture each time a new opinion needs to be incorporated to ensure theoretical correctness.

6 Conclusion

This work addresses the challenge of forming a new opinion by integrating insights from experts, the agent's prior beliefs, their trust in those experts, and their confidence in their own opinions. By adopting a *single-agent perspective* and modelling opinions probabilistically, we framed the opinion merging task as a *decision-making* problem. To validate the proposed approach, we employed the selection of a mobile phone brand with desired features as a practical test case and performed weighted opinion pooling [1] of the same opinions for baseline comparison. The resulting solution is represented as a posterior distribution of mobile brands, which captures the merged opinions of both the modeller and the experts. This distribution respects the agent's prior beliefs, their preferences for specific features, and the importance of those features. Data were collected from human participants. Given the lack of a definitive ground truth for merged opinions, we designed a series of simulated experiments based on real-world data that can be assessed through logic and intuition. The results obtained are robust and self-explanatory. The results from the proposed method are almost identical to results from weighted linear opinion pooling, which validates the method. Main advantage of the proposed solution is theoretically sound Bayesian formulation which allows on-line updating of opinions in dynamic setting. Several key questions remain for future research:

- How can we effectively learn an agent's trust in an expert's opinion based on previously observed data, external knowledge, and varying preferences?
- How can we model the dynamics of trust, particularly its evolution with growing experience and changing agent's preferences?

– In what ways does confidence in the provided information shape the formation of opinions?

Our long-term vision is to combine the learning of preferences with opinion merging, ultimately creating a reliable, analytically transparent human-in-the-loop support system for interactions within social or customer networks. This approach, in combination with decision-making systems, has the potential to enhance and transparetize decision-making processes and foster more effective communication across diverse contexts and domains such as merging of opinions of medical specialist, company managers, policy-makers or autonomous systems.

Acknowledgments. The results and knowledge included herein have been obtained owing to support from the EU-funded IAMI project, Grant agreement ID: 101168272 and the institutional grant no. 2025A1013, Internal Grant Agency of the Faculty of Economics and Management, Czech University of Life Sciences Prague.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

Appendix A

Let us substitute $P_I(X)$ (4) $P_I(\theta)$ (5) and $P_{E_i}(X)$ (6) into $A_I^U(\theta|P_{E_i})$ (3). Substituting $P_I(\theta)$ (5) and simple equivalent transformations gives:

$$A_I^U(\theta|P_{E_i}) = \prod_b \prod_{\{f_{j,b}\}_{j=1}^n} \theta_b^{n_b-1}\theta_{f_{j,b}}^{n_{f_{j,b}}-1} \exp\left[\sum_{x\in X} \ln(P_I(x|\theta))P_{E_i}(x)\right]. \quad (10)$$

Substituting (4) and (6) into (10), where expert provides, among other, $P_{E_i}(\mathcal{B})$. This distribution does not represent the expert's preference for brand $\mathcal{B}$. Therefore, can be omitted during the substitution of (6).

$$A_I^U(\theta|P_{E_i}) \propto \prod_b \prod_{\{f_{j,b}\}_{j=1}^n} \theta_b^{n_b-1}\theta_{f_{j,b}}^{n_{f_{j,b}}-1} \exp\Bigg[\sum_{x\in X} \ln(\theta_b\theta_{f_{1,b}}\cdots\theta_{f_{n,b}}) \times P_{E_i}(F_{1,\mathcal{B}=b} = f_{1,b}|\mathcal{B}=b)\cdots P_{E_i}(F_{n,\mathcal{B}=b} = f_{n,b}|\mathcal{B}=b)\Bigg]. \quad (11)$$

As noted, external opinion $P_{E_i}(X)$ degenerates, then (11) reads

$$A_I^U(\theta|P_{E_i}) \propto \prod_b \prod_{\{f_{j,b}\}_{j=1}^n} \theta_b^{n_b-1}\theta_{f_{j,b}}^{n_{f_{j,b}}-1}\theta_{f_{E_i,1,b_{E_i}}}\cdots\theta_{f_{E_i,n,b_{E_i}}}. \quad (12)$$

Next, we will make the following adjustment in (12):

$$A_I^U(\theta|P_{E_i}) \propto \prod_b \prod_{\{f_{j,b}\}_{j=1}^n} \theta_b^{n_b-1} \theta_{f_{j,b}}^{n_{f_{j,b}}+\delta(b,b_{E_i})\delta(f_{j,b},f_{E_i,j,b_{E_i}})-1}.$$

Next, we multiply by the modeller's strategy $S_I(A|P_{E_i})$ specified in (7), since it serves as a hyper-prior for the distribution $A_I^U(\theta|P_{E_i})$ and move the relevant exponential terms under one exponent:

$$A_I^U(\theta|P_{E_i}) \propto \prod_b \prod_{\{f_{j,b}\}_{j=1}^n} \theta_b^{n_b-1} \theta_{f_{j,b}}^{n_{f_{j,b}}+s_{f_{j,b}}-1+\delta(b,b_{E_i})\delta(f_{j,b},f_{E_i,j,b_{E_i}})-1} \tag{13}$$

We introduce the following definition of the updated weights.

$$V_{f_{j,b}} = n_{f_{j,b}} + s_{f_{j,b}} + \delta(b,b_{E_i})\delta(f_{j,b},f_{E_i,j,b_{E_i}}) - 1,$$

where b is a brand, b_{E_i} denotes brand selected by i-th expert, $f_{j,b}$ is score of j-th feature $f_{E_i,j,b_{E_i}}$ is a score selected by i-th expert for brand b_{E_i}. Then (13) reads

$$A_I^U(\theta|P_{E_i}) \propto \prod_b \theta_b^{n_b-1} \prod_{\{f_{j,b}\}_{j=1}^n} \theta_{f_{j,b}}^{V_{f_{j,b}}-1}, \tag{14}$$

To obtain the final form of updated prior (3), we need to calculate the normalisation constant. This is done by noticing that the final distribution is a product of the Dirichlet distributions. Therefore, the normalisation constant is a product of normalisation constants.

$$\mathcal{N}_{A_I^U(\theta|P_{E_i})} = \prod_b \frac{\Gamma(n_b)}{\Gamma(\sum_b n_b)} \prod_j \frac{\prod_{f_{j,b}} \Gamma(V_{f_{j,b}})}{\Gamma(\sum_{f_{j,b}} V_{f_{j,b}})}, \tag{15}$$

where $j \in \{1,\ldots,n\}$, $b \in \{1,\ldots,m\}$ and Γ is the Gamma function.

After merging with the expert's opinion $P_{E_i}(X)$ (6), the final form of the updated prior $A_I^U(\theta|P_{E_i})$ is :

$$A_I^U(\theta|P_{E_i}) = \frac{1}{\mathcal{N}_{A_I^U(\theta|P_{E_i})}} \prod_b \prod_{\{f_{j,b}\}_{j=1}^n} \theta_b^{n_b-1} \theta_{f_{j,b}}^{V_{f_{j,b}}-1}. \tag{16}$$

Appendix B

The modeller's updated opinion can be written as:

$$P_I^U(X) = \int_{\boldsymbol{\Theta}} P_I^U(X,\theta)d\theta = \int_{\boldsymbol{\Theta}} P_I(X|\theta) A_I^U(\theta|P_{E_i})d\theta. \tag{17}$$

In the second equality, $A_I^U(\theta|P_{E_i})$ is the updated prior specified in (16). Next, we substitute the modeller's opinion (4) and updated prior $A_I^U(\theta|P_{E_i})$ (16) into

(17). Substituting $P_I(X|\theta)$ (4) into (17), using assumption of conditional independence as in (4) and equivalent transformations gives:

$$P_I^U(X=x) = \frac{1}{\mathcal{N}_{A_I^U(\theta|P_{E_i})}} \int_{\boldsymbol{\Theta}} \prod_b \theta_b^{n_b+\delta(b,b_I)-1} \prod_{\{f_{j,b}\}_{j=1}^n} \theta_{f_{j,b},b}^{V_{f_{j,b}}+\delta(b,b_I)\delta(f_{j,b},f_{j,b_I})-1} d\theta. \tag{18}$$

Similarly to the calculation in (14), the integral is a product of Dirichlet distributions

$$P_I^U(X) = \frac{n_{b_I}}{\sum_b n_b} \prod_j \frac{V_{f_{j,b_I}}}{\sum_{f_{j,b_I}} V_{f_{j,b_I}}}. \tag{19}$$

Appendix C

Simulated expert opinion tables for each brand (Tables 1, 2 and 3).

Table 1. Brand 1.

Experts	Feature 1	Feature 2	Feature 3
1	4	3	4
2	5	3	5
3	5	6	5
4	6	5	3
5	6	6	6
6	5	6	5
7	6	6	5
8	6	3	4
9	4	5	4
10	6	4	3

Table 2. Brand 2.

Experts	Feature 1	Feature 2	Feature 3
1	5	5	5
2	5	6	5
3	3	4	4
4	3	4	5
5	4	5	5
6	5	6	4
7	6	6	6
8	5	6	6
9	4	3	4
10	4	6	3

Table 3. Brand 3.

Experts	Feature 1	Feature 2	Feature 3
1	3	4	3
2	3	3	4
3	4	3	4
4	3	3	3
5	5	4	3
6	3	5	5
7	4	5	6
8	4	3	2
9	3	4	3
10	4	3	4

Trust levels assigned to experts by the modeller for individual experiments.

Table 4. Experiment 2.

Expert	Brand 1	Brand 2	Brand 3
1	0.8	0.9	0.9
2	0.9	0.9	0.7
3	0.3	0.3	0.8
4	0.7	0.3	0.8
5	0.2	0.7	0.9
6	0.2	0.7	0.8
7	0.3	0.9	0.7
8	0.5	0.9	0.7
9	0.4	0.4	0.8
10	0.9	0.4	0.9

Table 5. Experiment 3.

Expert	Brand 1	Brand 2	Brand 3
1	0.8	0.3	0.5
2	0.9	0.3	0.7
3	0.5	0.9	0.9
4	0.9	0.9	0.2
5	0.2	0.5	0.9
6	0.2	0.5	0.8
7	0.2	0.2	0.9
8	0.5	0.2	0.7
9	0.4	0.8	0.8
10	0.9	0.8	0.9

The modeller's opinion setup for individual experiments.

Table 6. Experiments 1–3.

	Feature 1	Feature 2	Feature 3
Brand 1	5	4	5
Brand 2	6	5	6
Brand 3	4	4	3

Table 7. Experiment 4.

	Feature 1	Feature 2	Feature 3
Brand 1	3	4	1
Brand 2	6	5	6
Brand 3	4	4	3

Table 8. Experiment 5.

	Feature 1	Feature 2	Feature 3
Brand 1	3	1	1
Brand 2	2	3	1
Brand 3	6	5	5

Updated brand preferences of the modeller across five experiments.

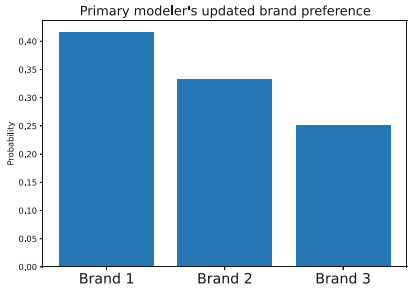


Fig. 1. Experiment 1.

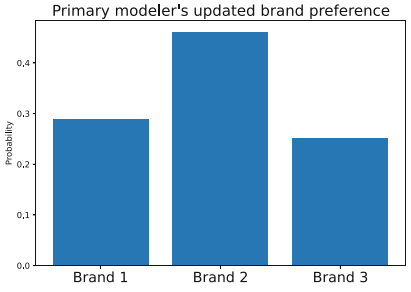


Fig. 2. Experiment 2.

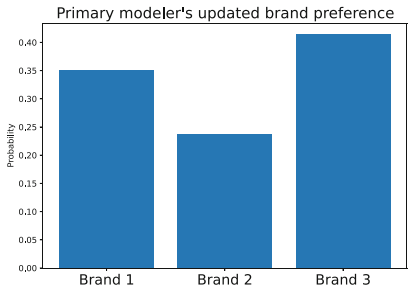


Fig. 3. Experiment 3.

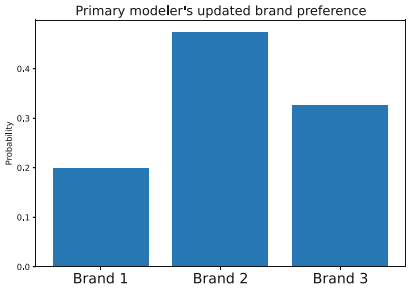


Fig. 4. Experiment 4.

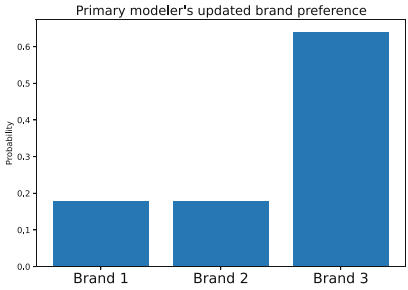


Fig. 5. Experiment 5.

References

1. DeGroot, M.H., et al.: Optimal linear opinion pools. Manag. Sci. (1991)
2. Garg, A., et al.: Generalized opinion pooling. In: AI&M (2004)
3. Genest, C., et al.: Allocating the weights in the linear opinion pool. J. Forecast. (1990)
4. Karny, M.: Optimized geometric pooling of probabilities for information fusion and forgetting. Automatica (2025)
5. Liu, P., et al.: Social trust-driven consensus reaching model for multi-attribute group decision making: exploring social trust network completeness. TFS (2023)
6. O'Donovan, J., et al.: Trust in recommender systems. In: Proceedings of the 10th International Conference on Intelligent User Interfaces (2005)
7. O'Doherty, D., et al.: Trust-based recommendation: an empirical analysis. In: Proceedings of the 6th ACM Workshop on Social Network Mining and Analysis (2012)
8. Quinn, A., et al.: Optimal design of priors constrained by external predictors. Int. J. Approximate Reasoning 153 (2017)
9. Romeijn, J.W.: An interpretation of weights in linear opinion pooling. Episteme (2020)
10. Roy, F., et al.: Comparative analysis of different trust metrics of user-user trust-based recommendation system. Comput. Sci. (2022)
11. Rufo, M.J., et al.: Merging experts' opinions: a bayesian hierarchical model with mixture of prior distributions. Eur. J. Oper. Res. (2010)
12. Ružejnikov, J.: Targeted trust based merging of customers opinions experiments: v1.0 (2025). https://doi.org/10.5281/zenodo.15738051
13. Sarwate, A.D.: Merging opinions by social sampling of posteriors. In: Proceedings of the 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton) (2012)
14. Ya, W., et al.: Consensus model of social network group decision-making based on trust relationship among experts and expert reliability. J. Syst. Eng. Electron. (2023)



DA-NeRF: High-Fidelity Talking Face Generation From Speech With Neural Radiance Fields

Yali Cai, Peng Qiao(✉), and Dongsheng Li

National Key Laboratory of Parallel and Distributed Computing, College of Computer Science and Technology, National University of Defense Technology, Changsha, China

Abstract. One-shot talking head generation aims to produce a video that is consistent with the driving audio from a single frame of the source character. Recently, NeRF-based methods have made significant progress in novel view synthesis. However, these methods often generate unnatural or degraded images when applied to dynamic objects, primarily due to the loss of geometric consistency during the generation process. To address this issue, we propose a depth-guided keypoint warp mechanism that provides geometric guidance for NeRF rendering. Additionally, by combining a multi-layer perceptron (MLP) with an attention mechanism, which can encode more neighborhood information while reducing the total number of parameters. Finally, we design an adaptive loss function to minimize floaters and enhance lip-sync consistency. By integrating these methods, we introduce the DA-NeRF, which generates coherent and realistic talking head videos by accurately depicting lip motion, facial expressions, and head poses. Experimental results from several benchmark tests demonstrate that our method produces more realistic talking head videos, surpassing the state-of-the-art methods in image quality.

Keywords: Talking head generation · Dynamic NeRF · Depth prediction · Audio driven animation · Video portrait

1 Introduction

Speech-driven talking head generation [5] refers to generating a video based on an image of the target character and a driving audio. Research in this field has increased significantly in recent years and is widely app lied in areas such as game production, virtual reality, and video conferencing [11]. However, generating realistic videos that accurately align speech content with natural facial expressions and maintain high image quality remains a significant challenge.

Warp-based methods [6] have been proposed to modify the mouth shape of actors by directly warping facial features during movie production. However, the generated results often appear rigid and unnatural. In earlier years, Generative Adversarial Networks (GANs) [10] were extensively used to synthesize high-resolution images, and subsequent works [4] incorporated various facial priors to stabilize GAN training. Nevertheless, these methods can lead to feature

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 366–377, 2026.
https://doi.org/10.1007/978-3-032-04555-3_30

loss during extraction, diminishing the model's ability to produce detailed facial features. Since the emergence of Neural Radiance Fields (NeRF) [17,25], numerous dynamic NeRF-based methods [13,16] have made significant progress in video generation. These methods utilize features from the input audio signal as conditions for an implicit function to synthesize high-fidelity images from novel perspectives. However, because these methods rely solely on dynamic NeRF, the quality of novel views degrades significantly when facial muscles change during pronunciation or when there is substantial head motion. Existing methods use facial template features as priors for NeRF, but their performance remains limited by the expressiveness and tracking accuracy of the template.

To address these limitations, we propose DA-NeRF, an attention-assisted parametric model driven by a depth-guided keypoint warp mechanism. The core idea is to extract a set of latent features from facial images to serve as priors for NeRF, which contain detailed facial geometric information for rendering and enable more controllable generation during significant head motion. Specifically, the depth information [9] recovered from the facial image is utilized to optimize the extraction of facial keypoints. The facial latent features are then extracted from the deformed keypoints, encoding the residuals of the deformation and facilitating the rendering of new poses not present in the input sequence. Moreover, we designed a novel architecture that integrates MLP with an attention mechanism, enhancing the focus on important lighting, key features, and details, thereby improving the quality of the generated images. Finally, the audio signal features, along with the latent facial features and camera parameters, are input into NeRF's conditional implicit function to generate high-fidelity speech frames of the target view through volume rendering. To better regularize the dynamic NeRF and the keypoint warp module, we propose an adaptive loss function that enhances image quality by constraining keypoint extraction and rendering results. In summary, our main contributions are as follows:

- We design an attention mechanism assisted NeRF model, which can capture the details of more important areas, and improve the quality of the generated image.
- We present a depth-guided keypoint warp module to provide geometric guidance for NeRF modeling, enabling the extraction of more accurate facial features.
- We propose an adaptive loss function that effectively reduces the appearance of artifacts.

2 Related Works

Audio Driven Talking Head Generation

Audio-driven talking head generation involves animating the head of a target character to match the speech. One possible approach is to use facial landmarks [7,27], which are highly correlated with expression, as priors for deformation. However, facial landmarks are sensitive to head pose, perspective and scale, presenting challenges for extreme head poses. Parametric models [14,26] which

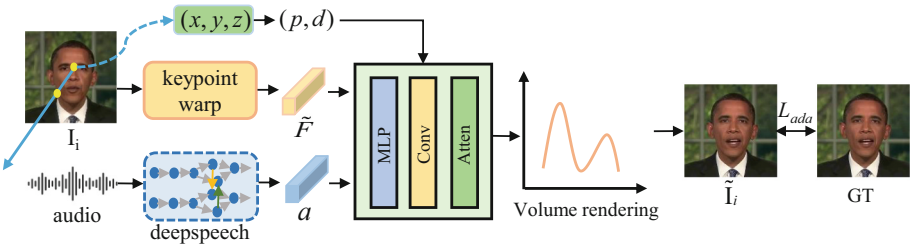


Fig. 1. Illustration of our proposed framework DA-NeRF.

retain almost all the expression and lip motion features, serve as more accurate priors. However, their ability to model non-rigid deformation is limited. In addition, they overlook the effect of vertex offsets on face shape, resulting in rough and unnatural facial expressions. To address these issues, NeRF, known for its performance in novel view synthesis, is adopted for talking head generation [1,13,16]. However, artifacts persist during significant head motion due to NeRF's limited performance in dynamic scenes. Recently, diffusion [3,23] have shown strong capabilities in generation tasks. However, they require much longer to synthesize views and perform poorly in terms of identity retention.

Dynamic Neural Scene Representation and Rendering

NeRF models the scene as a continuous 5D radiance field implicitly. D-NeRF [19] is the first work to extend NeRF to dynamic scene generation, capable of modeling dynamic scenes. However, modeling dynamic objects remains challenging for implicit representation. To further improve the control of facial details, some approaches [13,15,24] incorporated 3D parameters as a priori information into implicit representations. Other methods [2,16] learned latent codes related to facial expressions and motion from image frames in an end-to-end and self-supervised manner to drive NeRF, making the model 3D-aware while faithfully capturing high-frequency expressions. However, despite the efficiency of face templates, their expressive power is limited by the linear expression mixing shapes, resulting in coarse control of the head motion and a lack of detailed expressions specific to the individual.

3 Method

3.1 Model Representation

As shown in Fig. 1, the input of backbone comprises three components: the 3D coordinates of the source image $\mathbf{I_i}$, the head deformation features $\tilde{\mathbf{F}}$ extracted using the keypoint warp module, and the driving audio features $\mathbf{a}$ extracted by the deepspeech. The backbone of our DA-NeRF framework consists of MLP, convolutional layers, and self-attention mechanism modules. The model maps these inputs through implicit representation to generate the deformed head image frames $\tilde{\mathbf{I}}_i$. To avoid artifacts in the generated images, we propose an adaptive loss that controls discrete pixels in the synthesized images. During training, given

a talking video with audio of the target character, DA-NeRF learns the mapping between frames.

DA-NeRF is a parametric model based on NeRF, which implicitly stores the 3D scene in a neural network and outputs an image from a new perspective by inputting the camera poses. NeRF models the scene as a continuous 5D radiance field. Given the known positions of the point $\mathbf{p}$ in the scene and the observed direction $\mathbf{d}$, the neural network outputs $(\mathbf{c}, \sigma)$, where $\mathbf{c}$ represents the color and the σ indicates the density at that point. The implicit representation is as follows:

$$F_{\Theta}(\mathbf{p}, \mathbf{d}) = (\mathbf{c}, \sigma), \tag{1}$$

in which Θ is the model parameters.

After obtaining $\mathbf{c}$ and σ, the volume rendering is finished by the following formula:

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t), \mathbf{d})dt, \tag{2}$$

where, $T(t) = \exp(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds)$ denotes the accumulated transmittance along the ray from t_n to t. $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ indicates the light coming from a point. t_n and t_f represent near and far bounds of ray respectively.

The original NeRF aims at generating new views for static objects, while our focus is on dynamic scenes with morphological changes over time. To address this, both the temporal information and the features obtained from the deformation module are input simultaneously to guide the subsequent rendering process. Overall, DA-NeRF is formulated as follows:

$$F_{\Theta}(\mathbf{p}, \mathbf{d}, \mathbf{a}, \tilde{\mathbf{F}}) = (\mathbf{c}, \sigma), \tag{3}$$

where, $\mathbf{a}$ denotes speech embedding. $\tilde{\mathbf{F}}$ is the facial feature output by depth-guided keypoint warp module.

To focus ray sampling on semantically relevant areas, our DA-NeRF model replaces the final layer of the original NeRF model with a convolutional layer. This modification enhances the model's ability to capture local spatial structures. The MLP layer outputs the color and density values for each voxel point. The subsequent convolutional layers extract local spatial information from these color and density features, generating richer representations. The convolutional layer, through a sliding window approach, captures and represents local features within the volume, which is crucial for understanding complex structures and details in space. The features processed by the convolutional layers are used as the values for Q,K,V inputs in the self-attention module. Q,K,V represent the query, key, and value, respectively. The self-attention mechanism module generates a weight vector α_t for the ray as follows:

$$\alpha_t = soft\max(QK^T/\sqrt{d_k})V, \tag{4}$$

Here, Q,K,V represent the query, key, and value, respectively.

Finally, the color is calculated as follows:

$$\hat{C}(r) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t), \mathbf{d})dt \cdot \alpha_t \tag{5}$$

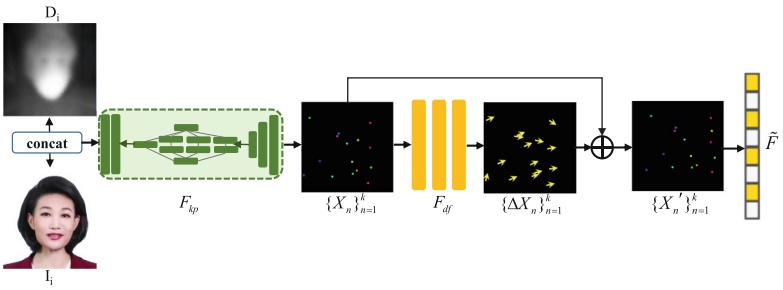


Fig. 2. Overview of the keypoint warp module.

3.2 Keypoint Warp Module

To avoid additional depth annotation, a self-supervised depth prediction module is utilized to extract depth features from human face images. This module consists of a depth network and a pose network. The depth network utilizes the UNet architecture, featuring an encoder-decoder structure. Specifically, ResNet18 serves as the depth encoder in this architecture. The decoder is constructed using four identical upconv structures. The pose network processes continuous frames to ascertain changes in angle and position relative to the scene. It shares a similar structural design to deep networks, incorporating encoder and decoder elements.

Two consecutive frames from the video, captured by a monocular camera, are used to determine changes in the camera's angle and position. Note that two adjacent frames are $\mathbf{I}_i$ and $\mathbf{I}_{i+1}$, the depth net is denoted as F_d, the pose net is F_p, the depth map output by the depth net is $\mathbf{D}_{I_i} = F_d(\mathbf{I}_i)$. The output relative camera poses matrix $[\mathbf{R}_{I_i \to I_{i+1}}, \mathbf{t}_{I_i \to I_{i+1}}]$ and the camera intrinsic matrix $\mathbf{K}_{I_i \to I_{i+1}}$ of the pose network are:

$$[\mathbf{R}_{\mathbf{I}_i \to \mathbf{I}_{i+1}}, \mathbf{t}_{\mathbf{I}_i \to \mathbf{I}_{i+1}}], \mathbf{K}_{\mathbf{I}_i \to \mathbf{I}_{i+1}} = F_p(\mathbf{I}_i || \mathbf{I}_{i+1}), \tag{6}$$

$$\mathbf{q}_k \sim \mathbf{K}_{\mathbf{I}_i \to \mathbf{I}_{i+1}} [\mathbf{R}_{\mathbf{I}_i \to \mathbf{I}_{i+1}} | \mathbf{t}_{\mathbf{I}_i \to \mathbf{I}_{i+1}}] \mathbf{D}_{\mathbf{I}_i}(\mathbf{p}_j) {\mathbf{K}_n}^{-1} \mathbf{p}_j, \tag{7}$$

$$\tilde{\mathbf{I}}_i = B_I(\mathbf{I}_{i+1}, \{\mathbf{q}_k\}_{k=1}^N), \tag{8}$$

where, $\mathbf{q}_k$ and $\mathbf{p}_j$ denote the warped pixels and the original pixels on the source image, respectively. $\tilde{\mathbf{I}}_i$ is the reconstructed image of $\mathbf{I}_i$. $\mathbf{B}_i$ denotes the differentiable bilinear interpolation function. N is the total number of pixels.

The training of this module is achieved by constructing the luminosity consistency error loss between $\tilde{\mathbf{I}}_i$ and $\mathbf{I}_i$. The loss is represented as:

$$P_e(\mathbf{I}_i, \tilde{\mathbf{I}}_i) = \alpha(1 - SSIM(\mathbf{I}_i, \tilde{\mathbf{I}}_i)) + (1 - \alpha)||\mathbf{I}_i - \tilde{\mathbf{I}}_i||, \tag{9}$$

here we take the value of α as 0.8 for the sake of image quality.

As shown in Fig. 2, after obtaining the depth map from F_d, it is concatenated with the RGB images I_i and input to the keypoint detection network F_{kp}. These fused features pass through two Conv-BN-ReLU-Average Pool blocks to extract advanced features, which are then input into a residual network module. Finally, two Upsample-Conv-BN-ReLU blocks restore the feature dimensions to their original size. The keypoint estimator outputs a set of sparse keypoint coordinates on the face:

$$\{\mathbf{X}_n\}_{n=1}^{k} = F_{kp}(\mathbf{D}_{\mathbf{I}_i}||\mathbf{I}_i), \tag{10}$$

where k is the number of the facial keypoints. After that, the deformation module calculates the deformation displacement of the keypoints:

$$\{\Delta\mathbf{X}_n\}_{n=1}^{k} = F_{df}(\{\mathbf{X}_n\}_{n=1}^{k}). \tag{11}$$

The deformed keypoints are:

$$\{\mathbf{X}_n{}'\}_{n=1}^{k} = \{\mathbf{X}_n\}_{n=1}^{k} + \{\Delta\mathbf{X}_n\}_{n=1}^{k}. \tag{12}$$

Finally, they are fed into a slot attention module to obtain the fusion features as follows:

$$\tilde{\mathbf{F}} = softmax(\{\mathbf{X}_n{}'\}_{n=1}^{k}). \tag{13}$$

3.3 Loss Function

During the training phase, we propose an adaptive loss function to train DA-NeRF. In deformable NeRF, $L_N^{\mathbf{r},i}$ is utilized to minimize the distance between the generated image and the ground truth. Additionally, we apply L_E, L_D and L_F to constrain the extraction and deformation process in the keypoint warp module. The overall loss function is expressed as follows:

$$\begin{aligned} L_{ada}^{\mathbf{r},i} = \lambda_1 \cdot L_N^{\mathbf{r},i} + \lambda_2 \cdot L_E(\{\mathbf{X}_n\}_{n=1}^{k}) + \lambda_3 \cdot L_D(\{\mathbf{X}_n\}_{n=1}^{k}) \\ +\lambda_4 \cdot L_F(\{\Delta\mathbf{X}_n\}_{n=1}^{k}), \end{aligned} \tag{14}$$

Robust Loss. Without regularization, floaters frequently appear in new views, significantly degrading the quality and authenticity of the generated video. To enhance the model's sensitivity to such floaters, we apply a robust loss [9,12], where floaters are defined as outliers rather than the traditional $L2$ loss. Given a robust kernel $\kappa(\cdot)$, we rewrite the loss function as follows:

$$L_N^{\mathbf{r},i} = \kappa(||\hat{C}(\mathbf{r};\theta) - \hat{C}_i(\mathbf{r})||_2), \tag{15}$$

where $\kappa(\cdot)$ is positively and monotonically increasing.

Equivariance Loss. L_E is defined to ensure consistency for each valid keypoint output by the keypoint warp module [22].

Keypoint Distance Loss. L_D is used to prevent the detected facial keypoints from clustering around a single point, with a predefined threshold set to penalize the model.

Keypoint Warp Loss. L_F is used to improve the training of the deformation module by introducing a regularization term that limits the prediction offset to a reasonable range. λ_1, λ_2, λ_3, λ_4 are the weights of the loss term.

4 Experiments

4.1 Implemention Details

We implemented our method with Pytorch. The DA-NeRF model was trained on a single NVIDIA 3090 GPU, with the Adam optimizer employed to update the learnable parameters. The learning rate is set at 1e-5. The training video, approximately 4 min long and with a resolution of 512×512, was sourced from [11,20,26]. Following the preprocessing method of AD-NeRF, we segmented the head and torso using a parsing method and extracted facial coordinate data using 3DMM. Additionally, deepspeech was utilized to extract speech features. In this work, several metrics were employed to evaluate the quality of the generated images. The Peak Signal-to-Noise Ratio (PSNR) and Learned Perceptual Image Patch Similarity (LPIPS) were used to measure the similarity between the generated results and the ground truth. Furthermore, the Landmark Distance around the mouth (LMD) and the confidence score (Sync_{conf}) proposed in SyncNet were utilized to evaluate lip-sync accuracy.

Table 1. Quantitative ablation experiment results.

Method	Testset 1				Testset 2			
	PSNR↑	LMD↓	LPIPS↓	Sync↑	PSNR↑	LMD↓	LPIPS↓	Sync↑
baseline	26.21	5.86	0.11	5.14	27.57	5.98	0.09	4.68
-a	28.97	4.76	0.05	6.37	28.34	4.88	0.05	6.24
-k	27.24	5.08	0.08	5.88	27.07	5.07	0.07	5.97
-l	28.64	4.96	0.06	6.41	28.46	4.97	0.05	6.18
ours	**30.54**	**4.51**	**0.02**	**6.94**	**31.97**	**4.37**	**0.02**	**6.81**

4.2 Ablation Study

In this section, we conduct ablation experiments to demonstrate the effectiveness of our proposed depth-guided keypoint warp module and adaptive loss in improving the model's performance. The results are reported in Table 1, where "ours" denotes the proposed model, "-a" represents the model that replaces the attention-assisted NeRF with the original NeRF, "-k" denotes the model without the keypoint warp module, and "-l" indicates the model obtained by replacing the adaptive loss in our model with the $L2$ loss. The baseline refers to the model



Fig. 3. Qualitative ablation studies on test set.

where a single feature vector is extracted from the source image before applying the original NeRF and $L2$ loss.

Effectiveness of the Attention-Assited NeRF
In DA-NeRF, we propose a novel backbone network that merges convolutional layers with attention mechanisms to enhance spatial understanding capabilities. We conducted ablation experiments to validate the positive impact of this design on overall generation quality. The quantitative evaluation results in Table 1 indicate that removing the attention-assisted NeRF proposed in this paper leads to a noticeable degradation in both image quality and lip-sync accuracy. This demonstrates that the proposed attention-assisted NeRF is crucial for obtaining keypoint information. Moreover, the qualitative results shown in Fig. 3 clearly illustrate that the proposed attention-assisted NeRF structure significantly enhances the quality of the facial representations.

Effectiveness of the Keypoint Warp Module
It is evident that the depth-guided keypoint warp module significantly improves the proposed model's performance across various metrics, as shown in Table 1. This indicates that it plays a crucial role in talking head generation. As illustrated in Fig. 3, the head predicted by "ours" appears more realistic than that predicted by "-k". The ablation of the keypoint warp module leads to blurry results, while the addition of the depth-guided keypoint warp module effectively focuses the pixels on the face, especially when the head is rotated at larger angles.

Effectiveness of the Adaptive Loss
To verify the effectiveness of adaptive loss in minimizing the presence of floaters and improving lip-sync consistency, we compare the generation results using

adaptive loss with those using the standard $L2$ loss. In Fig. 3, there are fewer floaters in the generated images when using adaptive loss compared to those generated by "-l". This demonstrates the effectiveness of the adaptive loss function in reducing floaters. As shown in Table 1, the PSNR value improved by approximately 2 dB, while the LPIPS value decreased by 0.04. Additionally, the corresponding lip-sync accuracy also improved.

Table 2. Quantitative comparison to MakeItTalk [28], FOMM [22], Wav2Lip [18], AD-NeRF [11], DFRF [21], NeRF-AD [2] and DEGSTalk [8].

Method	Testset 1				Testset 2			
	PSNR↑	LMD↓	LPIPS↓	Sync↑	PSNR↑	LMD↓	LPIPS↓	Sync↑
MakeItTalk	18.77	5.31	0.19	3.94	18.36	7.37	0.12	3.52
FOMM	11.36	8.65	0.44	2.13	13.34	9.61	0.35	2.88
Wav2Lip	25.50	5.65	0.14	**8.94**	27.38	7.24	0.13	**9.39**
AD-NeRF	27.89	5.44	0.03	5.64	28.02	6.32	0.04	4.33
DFRF	28.60	5.43	0.03	5.99	28.83	6.56	0.04	4.40
NeRF-AD	29.01	5.12	0.03	6.13	29.53	5.03	0.03	5.41
DEGSTalk	29.13	4.91	**0.02**	6.21	30.27	4.82	**0.02**	6.37
DA-NeRF	**30.54**	**4.51**	**0.02**	6.94	**31.97**	**4.37**	**0.02**	6.81

4.3 Comparative Study

In this section, we compare the proposed method with several widely used talking head generation techniques. As shown in Table 2, our DA-NeRF achieves the best results among all the compared methods. In terms of image quality, our method attains the highest scores in PSNR and LPIPS, indicating that using depth-guided keypoint deformation as a prior for rendering effectively enhances image quality.

The qualitative comparison results in Fig. 4 reveal that the FOMM method generates facial images with significant large-area distortions. The Wav2Lip method modifies only the mouth region, resulting in severe distortions of the teeth and lips. Although the images generated by MakeItTalk do not exhibit severe distortions, they lack many details, such as facial hair and skin textures. The results from AD-NeRF show ghosting artifacts, making it difficult to maintain the source identity details. The DFRF improves upon this, leading to a noticeable enhancement in image quality, but still suffers from blurriness in the teeth and contours. The NeRF-AD method represents a significant advancement over the previous two approaches, achieving substantial improvements in image quality. Nevertheless, certain limitations persist. Specific frames within the generated video sequences exhibit noticeable blurriness, and the generation of fine facial details, including teeth and beards, lacks realism. Although the overall visual quality of the outputs produced by DEGSTalk surpasses that of prior methods, blurred details can undermine the verisimilitude of the generated videos, thereby impacting their overall fidelity and practical utility. Our method produces more realistic results during significant head motion, suggesting that



Fig. 4. Comparison result between various methods on test set.

our model accurately captures head dynamics. In terms of lip-sync consistency, our method also outperforms most other methods in Sync, proving its effectiveness in promoting lip-sync accuracy.

5 Conclusion

In this paper, we propose DA-NeRF, an attention-assisted parametric model driven by a depth-guided keypoint warp mechanism for one-shot audio-driven

talking head synthesis, achieving higher image quality and more diverse head motions. To enhance lip-sync consistency and reduce floaters, we further introduce an adaptive loss that effectively improves the quality of the generated video. Numerous experiments demonstrate that our method has advantages in generating natural talking videos with limited training data and iterations.

References

1. Bai, Y., et al.: High-fidelity facial avatar reconstruction from monocular video with generative priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4541–4551 (2023)
2. Bi, C., Liu, X., Liu, Z.: NeRF-AD: neural radiance field with attention-based disentanglement for talking face synthesis. In: ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3490–3494. IEEE (2024). https://doi.org/10.1109/icassp48485.2024.10446195
3. Bigioi, D., Basak, S., Jordan, H., McDonnell, R., Corcoran, P.: Speech driven video editing via an audio-conditioned diffusion model. arXiv preprint arXiv:2301.04474 (2023)
4. Cai, C., et al.: Speak: speech-driven pose and emotion-adjustable talking head generation. In: ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5. IEEE (2025)
5. Chen, L., et al.: Talking-head generation with rhythmic head motion. In: European Conference on Computer Vision, pp. 35–51. Springer (2020)
6. Chen, L., Li, Z., Maddox, R.K., Duan, Z., Xu, C.: Lip movements generation at a glance. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 520–535 (2018)
7. Das, D., Biswas, S., Sinha, S., Bhowmick, B.: Speech-driven facial animation using cascaded GANs for learning of motion and texture. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12375, pp. 408–424. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58577-8_25
8. Deng, K., et al.: Degstalk: decomposed per-embedding gaussian fields for hair-preserving talking face synthesis. arXiv:2412.20148 (2024)
9. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3828–3838 (2019)
10. Gu, K., Zhou, Y., Huang, T.: FLNet: landmark driven fetching and learning network for faithful talking facial animation synthesis. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 10861–10868 (2020)
11. Guo, Y., Chen, K., Liang, S., Liu, Y.J., Bao, H., Zhang, J.: AD-NeRF: audio driven neural radiance fields for talking head synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5784–5794 (2021)
12. Hong, F.T., Zhang, L., Shen, L., Xu, D.: Depth-aware generative adversarial network for talking head video generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3397–3406 (2022)
13. Li, D., Zhao, K., Wang, W., Peng, B., Zhang, Y., Dong, J., Tan, T.: Ae-NeRF: audio enhanced neural radiance field for few shot talking head synthesis. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 3037–3045 (2024)
14. Li, J., et al.: Talkinggaussian: structure-persistent 3d talking head synthesis via gaussian splatting. In: European Conference on Computer Vision, pp. 127–145. Springer (2024)

15. Li, T., et al.: Lokitalk: learning fine-grained and generalizable correspondences to enhance nerf-based talking head synthesis. arXiv:2411.19525 (2024)
16. Liu, X., Liu, Z., Bi, C.: Nerf-3dtalker: neural radiance field with 3D prior aided audio disentanglement for talking head synthesis. In: ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5. IEEE (2025)
17. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: representing scenes as neural radiance fields for view synthesis. Commun. ACM **65**(1), 99–106 (2021)
18. Prajwal, K., Mukhopadhyay, R., Namboodiri, V.P., Jawahar, C.: A lip sync expert is all you need for speech to lip generation in the wild. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 484–492 (2020)
19. Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-NeRF: neural radiance fields for dynamic scenes (2020)
20. Sanderson, C.: The vidtimit database. IDIAP Commun. (2004)
21. Shen, S., Li, W., Zhu, Z., Duan, Y., Zhou, J., Lu, J.: Learning dynamic facial radiance fields for few-shot talking head synthesis (2022)
22. Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E., Sebe, N.: First order motion model for image animation. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
23. Stypułkowski, M., Vougioukas, K., He, S., Zięba, M., Petridis, S., Pantic, M.: Diffused heads: diffusion models beat GANs on talking-face generation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 5091–5100 (2024)
24. Xu, Y., et al.: Latentavatar: learning latent expression code for expressive neural head avatar. arXiv preprint arXiv:2305.01190 (2023)
25. Ye, Y., Yi, R., Gao, Z., Zhu, C., Cai, Z., Xu, K.: NeRF: neural edge fields for 3D parametric curve reconstruction from multi-view images. arXiv:2303.07653 (2023)
26. Zhang, Z., Li, L., Ding, Y., Fan, C.: Flow-guided one-shot talking face generation with a high-resolution audio-visual dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3661–3670 (2021)
27. Zhong, W., et al.: Identity-preserving talking face generation with landmark and appearance priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9729–9738 (2023)
28. Zhou, Y., Han, X., Shechtman, E., Echevarria, J., Kalogerakis, E., Li, D.: Makelttalk: speaker-aware talking-head animation. ACM Trans. Graph. (TOG) **39**(6), 1–15 (2020)



Beyond Reconstruction: A Physics Based Neural Deferred Shader for Photo-Realistic Rendering

Zhuo He, Paul Henderson(✉), and Nicolas Pugeault(✉)

School of Computing Science, University of Glasgow, Glasgow G12 8RZ, UK

Abstract. Deep learning based rendering has achieved major improvements in photo-realistic image synthesis, with potential applications including visual effects in movies and photo-realistic scene building in video games. However, a significant limitation is the difficulty of decomposing the illumination and material parameters, which limits such methods to reconstructing an input scene, without any possibility to control these parameters. This paper introduces a novel physics based neural deferred shading pipeline to decompose the data-driven rendering process, learn a generalizable shading function to produce photo-realistic results for shading and relighting tasks; we also propose a shadow estimator to efficiently mimic shadowing effects. Our model achieves improved performance compared to classical models and a state-of-art neural shading model, and enables generalizable photo-realistic shading from arbitrary illumination input.

Keywords: Neural rendering · Photo-realistic rendering · Re-lighting

1 Introduction

Photo-realism is a key goal of computer graphics [17], and is essential for creating immersive multimedia experiences and imagery indistinguishable from the real world. Significant advancements have been achieved in illumination representation, material appearance description, and high-precision geometric modeling [3,4,6,22]. These techniques can be defined as *forward approaches*, that study and model physical interactions between light and surfaces in the real world. However, they require approximations of complex real-world physics and calculations that can reduce the rendering fidelity; there still remain perceptible differences between synthesized and real-world images.

In contrast, neural network-based approaches allow learning aspects of the rendering process from real-world data; these can be seen as *inverse approaches* or *data-driven approaches*, which reconstruct 3D information from 2D images. For example, Neural Radiance Fields [16] regress a scene radiance distribution with a neural network by estimating color and density of sample points, then using volume rendering to construct images from novel viewpoints. Gaussian Splatting [12] uses lightweight 3D Gaussians as the primitive representation, with a fast approximate rasterization method to render these to pixels efficiently.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 378–389, 2026.
https://doi.org/10.1007/978-3-032-04555-3_31

Although these approaches can produce highly realistic images, they fit models that are specific to the content of individual scenes. They must be retrained from scratch for new scenes, and do not learn information that transfers or generalizes across scenes.



Fig. 1. Example of rendering results from our physics based neural deferred shader.

Another approach to photorealistic rendering is to use an auxiliary enhancement network to learn a mapping between the domain of traditionally rendered images and that of photorealistic images [13,18]. As the initial rendering result lacks complete 3D scene information such as spatial distribution of illumination and 3D object positions, it is difficult for such post-hoc methods to learn a mapping to photorealistic images. In this work, we combine a deferred shading framework with a neural shading network, trained to directly regress pixel color from physically-based rendering (PBR) textures and light input, to achieve high fidelity rendering results. Our contributions are as follows:

1. We propose a physics-based neural deferred shading pipeline that renders scenes with PBR textures (albedo, roughness, specular) and illumination (HDRI light map) to photo-realistic images (Fig. 1).
2. We develop a neural shadow estimator to efficiently approximate realistic shadows, improving the final rendering result.
3. We propose a new FFHQ256-PBR dataset which contains RGB images, material textures (albedo, roughness, specular), depth, screen space ambient occlusion and HDRI environment map.
4. We conduct comprehensive experiments to compare our approach with existing shading models on multiple datasets, demonstrating its superior performance.

2 Related Work

Photo-Realistic Novel View Synthesis. Novel view synthesis is an application of the 3D reconstruction task, and recent learning-based techniques allow to reconstruct photo-realistic 3D scenes from posed images, and to synthesize

high quality novel views from learned 3D information. Specifically, a series of works on Neural Radiance Field [7,16,19,26] represent a single 3D scene using an implicit radiance field function, it samples points along camera rays, regressing color and density to reconstruct the view image. Other works on Gaussian splatting [12,19,25,27], on the other hand, draw inspiration from point-based rendering to use a splatting procedure to render pixel colors. They use lightweight 3D Gaussians as the primitive representation and achieve real-time novel view synthesis without sacrificing quality. Both of those approaches require overfitting to specific scenes and are difficult to generalize since they compose geometry, material and illumination together. In contrast, our work is based on the standard deferred rendering pipeline which learns the shading process that can render the arbitrary scene and producing photorealistic results.

Photorealism Enhancement. Other works reduce the photo-realism gap between the traditional rendering output and real-world imagery by designing an enhancement model. Richter et al. [18] capture all shader inputs of the video game Grand Theft Auto 5 (GTA5) and design an adversarial learning process to improve the vividness of shading result by discriminating it with the real-world photo in a similar scene, enhancing the photorealism of render outcome. Kim et al. [13] use a neural network to improve the effect of the rendering outcome from a physically-based Cook-Torrance model, it does not use adversarial learning as it has the paired input and targets. However training a model to improve forward approaches (see Sect. 1) requires either unstable adversarial learning or is restricted by the limited input, as the initial rendering result has discarded much information, like the illumination angle, which would be valuable for the enhancement process. In contrast, we propose a more general process of shading, where a neural network regressor provides an atomic model of the shading process, calculating the RGB color of each rasterized shading point from highly informative PBR texture and illumination input, achieving better performance and explainability.

Neural Deferred Rendering. In classical rendering, there are two ways to render a spatial point: forward shading and deferred shading. Forward shading directly renders each object in turn, and lights it according to all light sources in the scene. Deferred shading instead delays intensive rendering tasks, such as shading, to a later stage, only processing them for the visible points on camera screen, reducing the calculation complexity [3]. Neural rendering is typically based forward shading, albeit learning to represent a scene from sample imagery [2,8,20,24,28], however the deferred pipeline can also be used in neural rendering, which is advantageous as it avoids the non-differentiable rasterization process in classical rendering, and allows replacing classical shading formulae with a neural model learnt from data. Neural deferred rendering was first proposed in Thies et al. [21], which integrates the conventional real-time rendering pipeline with a learnable neural texture and a deferred neural renderer. Worchel et al. [23] extend the deferred neural renderer to tackle 3D reconstruction, using a differentiable rasterizer in the classical deferred rendering pipeline to reconstruct a triangle mesh from multi-view input. However, existing works on neural

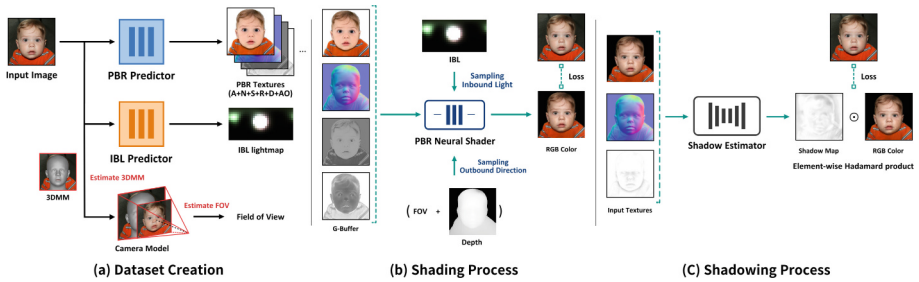


Fig. 2. The overall pipeline of our physics-based neural deferred shading. In data preprocessing (a), the input image is estimated for PBR textures (A: albedo, N: Normal, S: specular, R: roughness, D: depth, AO: Ambient Occlusion), IBL lightmap, field of view via pre-trained models. Then the estimated data are used to train the physics based neural shader (b). Subsequently a shadowing estimator (c) estimates the shadow map applied to the final shading result.

deferred shading do not consider the light-surface interaction during shading, and are not physics-based—thus unable to generalize to new illumination conditions. Our work redesigns the coordinate-based neural deferred shader [23] by incorporating the inbound light and bidirectional angle, regressing the shading result based on physically-meaningful information.

3 Method

Image rendering aims to model how light interacts with surfaces [13], as described by the rendering equation:

$$L_o(\mathbf{v}) = \int_{\Omega} F(\mathbf{v}, \mathbf{l}) L_i(\mathbf{l}) \langle \mathbf{n} \cdot \mathbf{l} \rangle d\mathbf{l} \tag{1}$$

where $L_o(\mathbf{v})$ is the outbound radiance leaving in direction $\mathbf{v}$; it is the integral of the incident light $L_i(\mathbf{l})$ from every possible direction $\mathbf{l}$ across the hemisphere Ω, centered around the surface normal $\mathbf{n}$. $F(\mathbf{v}, \mathbf{l})$ is the Bidirectional Reflectance Distribution Function (BRDF) describing how the surface reflects light.

In practice, approximations of this equation are used, limiting the photorealism of the rendering. We introduce *physics-based neural deferred shading* in this work, a novel framework for photorealistic rendering applied to human portraits. We use a neural network to regress the rendering equation with geometry, material and light inputs instead of using hand-coded approximations. An overview of our pipeline is presented in Fig. 2. We introduce the main approach in Sect. 3.1, then describe the loss functions in Sect. 3.2, and provide implementation details in Sect. 3.3.

3.1 Physics Based Neural Deferred Shading

To approximate the rendering equation more precisely, our work redesigns previous neural deferred shaders [23] to incorporate PBR inputs (PBR materials,

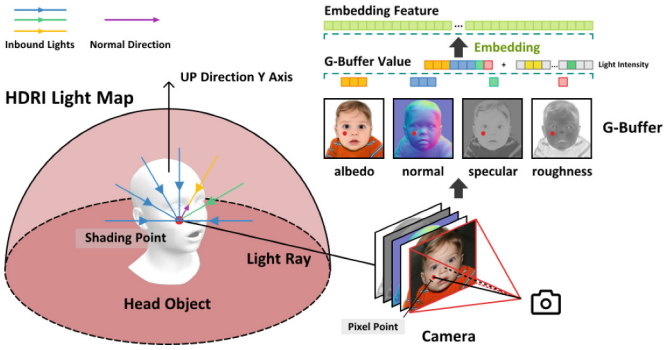


Fig. 3. Shading process of each point. Each spatial point would be acquired all PBR texture values to form the G-Buffer and sampling inbound light from HRDI map to get the high dimensional feature as the input of neural network.

normal map, depth map, HDRI light map). Due to the lack of paired samples for PBR inputs and real-world images, we first create a new dataset for training and evaluation; then train our neural deferred shader and shadow estimator by minimizing the difference between rendered and ground truth images. During inference, our model is capable of rendering an arbitrary G-buffer containing physically-based material and geometry information, under a given HDRI light-map, producing a photo-realistic image result.

Dataset Creation. We use the recent material and light predictor [13] to create our dataset from images. This includes physically-based rendering (PBR) material textures, normal map, depth map, screen-space ambient occlusion (SSAO) map and image-based lighting (IBL) environment map, where the PBR textures are 2D texture maps consisting of albedo, specular, roughness, combined with normal map and depth maps to form the G-Buffer (see Sect. 2). The light model is an equirectangular projection of a hemispherical HDR image that surrounds the object to provide the lighting. Moreover, calculating the outbound direction of each pixel requires the field-of-view (FOV) of the input image. We use a 3D Morphable Model (3DMM) to register input images and locate 3D landmarks on faces [5], which combined with projected 2D landmarks and acquired depth map, allows to estimate the FOV. Using this process, we extend the human face dataset FFHQ [11] into a new dataset called FFHQ256-PBR, consisting of 69,990 facial images with estimated PBR material textures, HDRI illumination and camera field-of-view. We manually remove failure cases (e.g. where background removal affects the head itself). Our main experiments focus on human faces due to their importance in many applications; however other object types are also supported in our rendering pipeline. We therefore collect a second dataset based on a subset (1,000 scenes) of the synthetic BlenderVault dataset [15]. For this, we estimate the relevant parameters using the inverse rendering approach proposed in [15].

Shading Process. We train the neural deferred shader to estimate the RGB color for each pixel, after which it can shade an arbitrary surface point during

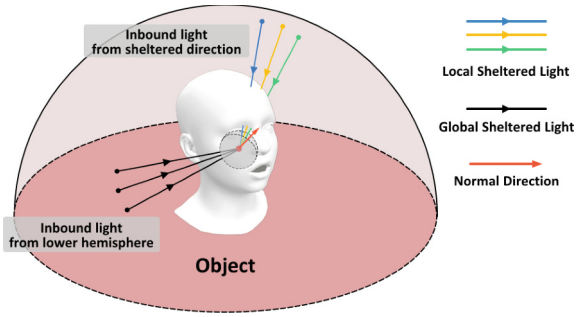


Fig. 4. Localized shadowing model: The light rays denoted by black arrows come from the lower hemisphere and would be filtered by the cosine term in the rendering equation, whereas the light rays denoted by colorful arrows are occluded by local geometry resulting in localized shadowing.

inference phase. Specifically, the shader processes each pixel of the G-Buffer separately (Fig. 3). Texture and normal information is represented in screen space and taken from the corresponding pixel, while illumination information is sampled from rays on the hemisphere around the 3D point found by casting a ray from the pixel into the scene.

The neural shader's input is inspired by the classical rendering Eq. (1), and consists of three material features corresponding to F term: albedo $\mathbf{a} \in [0,1]^3$, specular $\mathbf{s} \in [0,1]^1$ and roughness $\mathbf{r} \in [0,1]^1$; three geometric terms (as unit vectors): normal $\mathbf{n} \in [-1,1]^3$, inbound light direction $\mathbf{l} \in \mathbf{R}^3$ and outbound light direction $\mathbf{v} \in \mathbf{R}^3$; a single light feature: inbound light intensity $L_i \in (\mathbf{R}^+)^3$. The per-pixel values of $\mathbf{a}, \mathbf{n}, \mathbf{s}, \mathbf{r}$ form a geometry buffer (G-Buffer) that is concatenated with the n sampled inbound light rays $\{L_i(\mathbf{l}_1), L_i(\mathbf{l}_2), ..., L_i(\mathbf{l}_n)\}$ to form the input of a trainable neural shader with parameter θ. Accordingly, the neural shading function is:

$$\int_{\Omega} f_\theta(\mathbf{a}, \mathbf{n}, \mathbf{s}, \mathbf{r}, \mathbf{v}, L_i(\mathbf{l})\langle \mathbf{n} \cdot \mathbf{l} \rangle) d\mathbf{l} \in [0,1]^3, \mathbf{l} \in \Omega \tag{2}$$

where $L_i(\mathbf{l})\langle \mathbf{n} \cdot \mathbf{l} \rangle$ is the inbound light from upper hemisphere Ω^+ similar to Eq. 1. Thus the neural shader first regresses outbound light due to each inbound light direction for each pixel, then calculates the mean of these over the sampled inbound rays to get the final result. Thus neural shader learns to approximate the integral of the shading calculation from the reconstruction loss.

Shadowing Process. The architecture proposed above does not account for how inbound light rays may be affected by self-occlusions—for example, the top of the head may shelter light rays incident to a point below the eye, casting a shadow. Figure 4 illustrates this effect: light rays coming from the top side of the environment map are occluded by the top of the head. In practice, shadowing varies with differences in environment lighting: if the environment light map is even and smooth, screen space ambient occlusion (SSAO) can approximate the shadow effect well; conversely, if the environment light is highly anisotropic then

the resulting hard shadows must be calculated by costly tracing of individual light paths. To calculate the shadow effect with a unified approach suitable for diverse light maps, we design a neural shadow estimator that learns to improve the shadowing result from our neural deferred shader. Specifically, a UNet-like neural network f_η is designed to estimate the shadow map from the input of an unshadowed image $\hat{I}_{unshadow}$ and corresponding geometry information (normal map I_{normal} and SSAO map I_{ssao}). The final shading result is the element-wise (Hadamard) product of the neural shader's output and the estimated shadow map. The shadowing process is thus described by

$$\begin{aligned} \hat{I}_{shadow} &= f_\eta(\hat{I}_{unshadow}, I_{normal}, I_{ssao}) \\ \hat{I}_{rgb} &= \hat{I}_{unshadow} \odot \hat{I}_{shadow} \end{aligned} \tag{3}$$

where $\hat{I}_{\text{shadow}}$ is the pixel-wise shadow map, η is parameters of the neural shadow estimator, $\hat{I}_{rgb}$ is the shadowed image.

3.2 Loss Function

The proposed model is trained by solving the following minimisation for the neural network parameters θ:

$$\underset{\theta}{\operatorname{argmin}}\, L_{\text{appearance}}(\mathbf{a}, \mathbf{n}, \mathbf{s}, \mathbf{r}, \mathbf{v}, L_i) \tag{4}$$

where $L_{\text{appearance}}$ compares the rendered pixel to the ground truth pixel at the corresponding position. $\mathbf{a}, \mathbf{n}, \mathbf{s}, \mathbf{r}, \mathbf{v}, L_i$ are albedo, normal, specular, roughness, outbound direction, inbound light respectively.

More specifically, we train our model in two stages, with $L_{\text{appearance}}$ defined differently in each phase. In phase one, the model does not consider shadowing, thus $L^{(1)}_{\text{appearance}}$ is the shading loss defined by Eq. 5, which is an L_1 loss calculating the average distance between prediction and ground truth for all foreground pixels, N_S is the number of sampled pixels, and C is the RGB value of pixel.

$$L^{(1)}_{\text{appearance}} = \frac{1}{N_S} \sum_{i=1}^{N_S} \|C_i - \hat{C}_i\|_1 \tag{5}$$

$$L^{(2)}_{\text{appearance}} = \|I_{rgb} - \hat{I}_{rgb}\|_1 \tag{6}$$

In phase two, the neural deferred shader is frozen to produce an unshadowed image, which is fed to the shadow estimator through concatenating with the normal map (see Sect. 3.1), yielding the shadowed image $\hat{I}_{rgb}$. An L1 loss $L^{(2)}_{\text{appearance}}$ (Eq. 6) calculates the residual between the shadowed image $\hat{I}_{rgb}$ and ground truth image I_{rgb} for all pixels.

3.3 Implementation Details

We train each stage for 40 epochs, which balances efficiency and performance. We use the Adam optimizer [14], with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a

learning rate of 5×10^{-5}. We randomly sample 8,192 foreground pixels of each image to form each training batch; this provides enough diversity to avoid overfitting as the neural deferred shading works on a per-pixel basis and is agnostic to the overall scene structure. We uniformly sample 128 inbound light rays from the HDRI light map as input to the shader in each training step. Training converges within one day on a single NVIDIA GeForce RTX 3090 GPU. During inference, the physically based neural deferred shader can render an image in 1 s per frame.

4 Experiments

We conducted comprehensive experiments to evaluate the performance of our neural deferred shading approach, comparing qualitatively and quantitatively against three baselines. The baseline models include classical shading models (the empirical Blinn-Phong model and the physics-based GGX model) as well as a recent learning-based model, neural deferred shader [23].

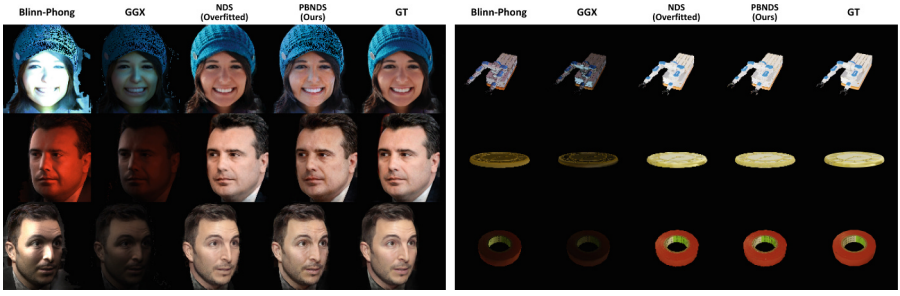


Fig. 5. Quality comparison between different shading models. **Blinn-Phong**: Blinn-Phong shading model; **GGX**: Trowbridge-Reitz GGX model; **NDS**: Neural deferred shader; **PBNDS**: Physics based neural deferred shader; **GT**: Ground truth image. Our model can reconstruct the ground truth scene with generalization and keep the photo-realism maximally through capturing the correct light-surface interaction.

4.1 Qualitative Evaluation

We visually compare shading results produced by different shading models from similar inputs. We evaluate examples from two different datasets, a separate test set drawn from the FFHQ256-PBR for facial data and a subset of BlenderVault [15] for other object types. Figure 5 illustrates the results for both datasets, comparing our neural shader with the classical Blinn-Phong [1] and GGX [22] models. Our physical based neural deferred shader (PBNDS) reconstructs the image photo-realistically from the estimated materials and illumination, and the overall hue and light reflection are more realistic than classical models. In contrast, the Blinn-Phong model is an empirical model where the light reflection cannot produce a photorealistic result. The GGX model is based on micro-facet theory which requires the additional alignment for the incident light intensity to

actualize the energy conservation, and is not suited for the estimated illumination application. We also compare our model to the state-of-art neural deferred shading (NDS) [23], and show that our approach can achieve similar quality to it. This is notable since ours is a generalized (scene-agnostic) model where NDS overfits a neural network separately to each scene, since it takes the scene-dependent point coordinates as an input.

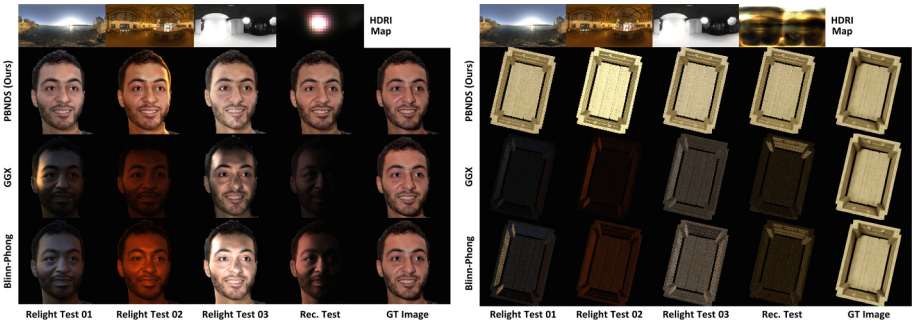


Fig. 6. Rendering results with different HDRI maps. The first row shows HDRI maps used to relight the original head; the other rows show the resulting rendered images after relighting from different shading methods. The results show our neural shader allows the environment map to realistically influence the shading of the head.

We next show how our model enables relighting images, by using recorded HDRI maps to render the image with different illumination. Specifically, we use our model trained from the previous section for both types of data but replace the HDRI map during rendering. All the new HDRI maps are captured in real-world scenes. We compare the relighting rendering results to classical rendering methods (Blinn-Phong, GGX) in Fig. 6. Our model correctly recovers the light-surface interaction within the scene with high fidelity, whereas the classical models do not work well in this inverse setting. Compared to rendering the scene with estimated illumination, the relighting test requires our model to adapt to real-world illumination inputs, which may lie outside the distribution of estimated illumination seen during training.

4.2 Quantitative Evaluation

We also compare the rendered images to the ground-truth quantitatively. We conduct 500 tests in each experiment to measure the performance of each shading model, using data from the test set of FFHQ256-PBR and BlenderVault [15]. To measure the realism of the shading result, we use Learned Perceptual Image Patch Similarity (LPIPS) [29] and Fréchet inception distance (FID) [9] as metrics; lower LPIPS and FID scores indicate better realism comparing ground truth and real world data. We also measure the Peak signal-to-noise ratio (PSNR) and Structural similarity index measure (SSIM), where higher PSNR or SSIM score indicates better reconstruction performance. As there is no paired ground-truth data for the relighting experiment, we only use FID score for evaluation.

Table 1. Quantitative evaluation for shading/relighting experiments

		MSE ↓		PSNR ↑		SSIM ↑		LPIPS ↓		FID ↓	
		FP	BV	FP	BV	FP	BV	FP	BV	FP	BV
SD	Blinn-Phong	0.078	0.044	11.90	14.94	0.623	0.784	0.181	0.080	0.244	0.239
	GGX	0.127	0.055	9.11	13.35	0.489	0.753	0.273	0.114	0.368	0.227
	NDS (Overfitted)	**0.005**	**0.001**	**29.68**	**38.53**	0.885	**0.988**	0.062	0.027	0.179	**0.037**
	PBNDS (Ours)	0.007	0.002	24.61	29.47	**0.919**	0.936	**0.032**	**0.025**	**0.056**	0.113
RE	Blinn-Phong	–	–	–	–	–	–	–	–	0.331	0.163
	GGX	–	–	–	–	–	–	–	–	0.556	0.162
	PBNDS (Ours)	–	–	–	–	–	–	–	–	**0.090**	**0.117**

SD: Shading experiment; **RE:** Relighting experiment; **FP:** FFHQPBR dataset; **BV:** BlenderVault dataset

Table 2. Comparison of training with different model components

	MSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
Default Config.	0.0046	24.4107	0.9167	0.0337	0.0607
+ Perc. Loss	0.0052	24.0763	**0.9298**	0.0273	0.0441
+ SDO (w/o AO)	0.0041	24.6948	0.9194	0.0319	0.0559
+ SDO (AO)	**0.0039**	**25.0053**	0.9247	**0.0262**	**0.0240**

Table 3. Comparison of different training batch size

	PSNR ↑	GPU Mem.	FLOPs(G)
2048	21.74	2.98	0.64
4096	22.81	5.96	1.28
8192	24.83	11.91	2.56
Full Res.	25.81	23.82	5.12

As shown in Table 1, our model exceeds classical shading models significantly in all metrics for both shading and relighting experiments, as the training process can adapt our model for the input with different data domain, especially for rendering the real world scene whose materials and illumination are estimated and not calibrated for classical models. Our model also outperforms the overfitted neural deferred shader model [23] in SSIM, LPIPS and FID for FFHQPBR dataset, and achieves competitive results in the less perceptually-relevant MSE and PSNR metrics for all datasets. We suggest this is caused by pixel-wise overfitting causing high-frequency noise especially for real-world images, that can be alleviated by the global shadowing process.

4.3 Ablation Study

We conducted an ablation study to evaluate the importance of the different components of our neural shading pipeline, following the same approach as in Sect. 3.1. We test our shading model with different configurations, each test follows a design similar to Sect. 4.2. The default configuration of our neural deferred shader lacks a shadow estimator, and uses a single reconstruction loss (as in Sect. 3.2), ReLU activation function, positional encoding [16]; other tests add specific components to this default setting. We add a perceptual loss [10] in the second test and our shadowing process in the third. We see in Table 2 that the perceptual loss improves results of perceptual metrics (SSIM, LPIPS, FID) but downgrades the reconstructive metrics (MSE, PSNR); the shadowing process without ambient occlusion (AO) can balance on both aspects, with AO the performance can be further improved to achieve the best result on most

metrics with only slight degradation on SSIM metric, thus we use the default setting with shadowing process (AO) as the final design. We also compare the training result of different batch size and find 8,192 is the best choice to balance the model performance and training efficiency (see Table 3).

5 Conclusion

We have presented a new method for photo-realistic rendering by using physics-based neural deferred shading, demonstrating improved performance compared to existing methods. We showed our method can be used for effective relighting and rendering. We also proposed a novel human facial dataset with PBR material textures, FFHQ256-PBR, which can be used for future research on real-world rendering.

Our method does not explicitly model the domain shift between synthetic, estimated and real-world data; although the relighting experiment proves it can handle real-world HDRI input, manually created 3D models still might be unsuitable for our shading model due to the, which could be addressed in future work by incorporating a domain adaptation process.

References

1. Blinn, J.F.: Models of light reflection for computer synthesized pictures. In: Proceedings of the 4th Annual Conference on Computer Graphics and Interactive Techniques, pp. 192–198. SIGGRAPH '77 (Jul 1977)
2. Boss, M., et al.: Neural-PIL: Neural pre-integrated lighting for reflectance decomposition. In: Beygelzimer, A., et al. (eds.) Advances in Neural Information Processing Systems (2021)
3. Deering, M., et al.: The triangle processor and normal vector shader: a VLSI system for high performance graphics. SIGGRAPH Comput. Graph. 21–30 (1988)
4. Dick, C., Schneider, J., Westermann, R.: Efficient geometry compression for GPU-based decoding in realtime terrain rendering. In: Computer Graphics Forum, vol. 28, pp. 67–83. Wiley Online Library (2009)
5. Feng, Y., et al.: Learning an animatable detailed 3D face model from in-the-wild images. ACM Trans. Graph. 1–13 (2021)
6. Guarini, G., Rossi, M., et al.: PBR material: a comparison between rendering for accurate material color reproduction. A case study. Color Colorimetry Multidisc. Contrib. **38** (2024)
7. Guo, Y.C., et al.: NeRFReN: neural radiance fields with reflections. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 18388–18397 (2022)
8. Hadadan, S., et al.: Neural radiosity. ACM Trans. Graph. 1–11 (2021)
9. Heusel, M., et al.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in Neural Information Processing Systems, pp. 6629–6640 (2017)
10. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 694–711. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_43

11. Karras, T., et al.: A style-based generator architecture for generative adversarial networks. In: Conference on Computer Vision and Pattern Recognition (2019)
12. Kerbl, B., et al.: 3D gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. 1–14 (2023)
13. Kim, H., et al.: SwitchLight: co-design of physics-driven architecture and pre-training framework for human portrait relighting. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 25096–25106. IEEE (2024)
14. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (2015)
15. Litman, Y., et al.: Materialfusion: enhancing inverse rendering with material diffusion priors. In: 3DV (2025)
16. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: representing scenes as neural radiance fields for view synthesis. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 405–421. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_24
17. Newell, M.E., Blinn, J.F.: The progression of realism in computer generated images. In: Proceedings of the 1977 Annual Conference on - ACM 1977, pp. 444–448 (1977)
18. Richter, S.R., et al.: Enhancing photorealism enhancement. IEEE Trans. Pattern Anal. Mach. Intell. 1700–1715 (2023)
19. Szymanowicz, S., et al.: Splatter image: ultra-fast single-view 3D reconstruction. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10208–10217 (2024)
20. Tewari, A., et al.: Advances in neural rendering. Comput. Graph. Forum 703–735 (2022)
21. Thies, J., et al.: Deferred neural rendering: image synthesis using neural textures. ACM Trans. Graph. (2019)
22. Walter, B., et al.: Microfacet models for refraction through rough surfaces. In: Proceedings of the 18th Eurographics conference on Rendering Techniques, EGSR 2007 (2007)
23. Worchel, M., et al.: Multi-view mesh reconstruction with neural deferred shading. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6177–6187 (2022)
24. Yao, Y., et al.: Neilf: neural incident light field for physically-based material estimation. In: Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXI, pp. 700–716 (2022)
25. Ye, K., et al.: 3D gaussian splatting with deferred reflection. In: Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers 2024, pp. 1–10 (2024)
26. Yu, A., et al.: pixelNeRF: neural radiance fields from one or few images. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4578–4587 (2021)
27. Yu, Z., et al.: Mip-splatting: alias-free 3D gaussian splatting. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 19447–19456 (2024)
28. Zeltner*, T., et al.: Real-time neural appearance models. ACM Trans. Graph. 33:1–33:17 (2024)
29. Zhang, R., et al.: The unreasonable effectiveness of deep features as a perceptual metric. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 586–595. Salt Lake City, UT (2018)



Accurate SDF Reconstruction with Geometric-Differential Regularization and Categorized Sampling Strategy

Kaiheng Li[1], Jiahui Chen[1], Chuanfeng Yang[1], Ziheng Zhang[1], Xuan Wei[1], Mingyu Shao[1], and Qingqi Hong[1,2,3(✉)]

[1] Department of Digital Media Technology, School of Film, Xiamen University, Xiamen, China
hongqq@xmu.edu.cn
[2] National Institute for Data Science in Health and Medicine, Xiamen University, Xiamen, China
[3] Institute of Artificial Intelligence, Xiamen University, Xiamen, China

Abstract. Implicit Neural Representation (INR) has emerged as a powerful approach for learning Signed Distance Function (SDF) in unsupervised 3D reconstruction from point clouds. However, current methods frequently suffer from over-permissiveness toward global viscosity solutions, resulting in spurious local minima that degrade both SDF optimization and final reconstruction quality. To overcome these limitations, we propose a novel loss function named Geometric-Differential Regularization (GDR) with two complementary components: Gradient-Projection regularization to suppress undesirable extrema formation, and Hessian-Degeneration regularization to enhance surface-proximal learning through higher-order geometric constraints. Furthermore, we develop a sampling strategy that dynamically adjusts constraints based on local geometric features of surface points. We validate our method through both 2D SDF learning visualizations and 3D reconstruction experiments using noisy point clouds, demonstrating significant improvements over existing approaches.

Keywords: Implicit Neural Representation · Signed Distance Function · 3D Reconstruction

1 Introduction

Recent advances in neural networks have significantly advanced 3D mesh reconstruction from point clouds [1,2,9,15]. The techniques involving SDF [5,14,16] have also provided significant insights for applications such as autonomous driving [21] and robot perception [13]. Building on this background, this study focuses on 3D reconstruction by leveraging INR [17–19] to learn the true SDF of an underlying watertight surface from unoriented point clouds, with the target surface extracted via zero-level set methods.

Previous methods either focus on near-surface sampling to learn the SDF [10,12,20], or fail to enforce constraints on the global viscosity solution [3,7,22]. Consequently, the learned function often exhibits uncontrolled inductive bias,

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 390–401, 2026.
https://doi.org/10.1007/978-3-032-04555-3_32

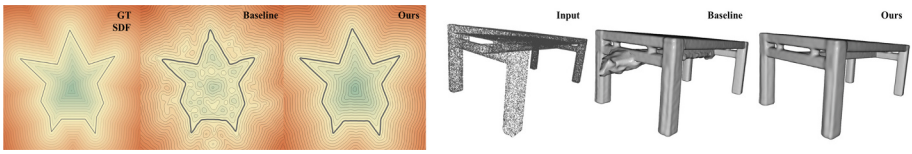


Fig. 1. 2D and 3D examples of SDF and surface reconstruction. In the 2D example, for the true SDF (left), previous methods are prone to undesirable local minima at the global scale(middle), whereas our method provides a better approximation of the global SDF(right). In the 3D example, for the input unoriented points (left), the aforementioned issue may cause local shape distortions (middle), while our method effectively mitigates this problem (right).

especially near singularities, leading to undesirable local minima and shape distortions (see Fig. 1, middle column). Moreover, conventional approaches still struggle with noise-induced surface roughness in low-curvature regions.

To overcome these limitations, this paper proposes Geometric-Differential Regularization (GDR) and a novel sampling strategy for network training. Specifically, we first use Gradient-Projection regularization L_{GP} to guide the sampling points toward correct projection estimation, ensuring proper gradient distribution and thereby preventing undesirable local minima. We complement L_{GP} with Hessian-Degeneration regularization L_{HD} based on Taylor expansion residuals, enhancing gradient linearity and singularity neighborhood smoothing through higher-order derivatives. To mitigate surface roughness, we adopt a distance-based categorized sampling strategy that enforces adaptive constraints across flat and high-curvature regions. In particular, we incorporate Voronoi vertices into non-manifold sampling to improve learning near non-differentiable surface features. We evaluate our method through 2D SDF visualizations (addressing 3D visualization challenges) and 3D reconstruction experiments on noisy unoriented point clouds, demonstrating effective local minima avoidance and improved surface smoothness both qualitatively and quantitatively (Fig. 2).

The main contributions of this paper are as follows:

- We propose a novel constraint term, Geometric-Differential Regularization (GDR), and demonstrate the effectiveness of the method and its components.
- We introduce a distance-based categorized sampling strategy that enables targeted constraints for sampling points with different features.
- We validate our method through 2D SDF visualization and 3D surface reconstruction experiments.

2 Preliminaries

Consider a continuous, compact geometric object $\Omega \subset \mathbb{R}^3$, where $\mathrm{int}(\Omega)$, $\partial\Omega$, and $\mathrm{ext}(\Omega)$ represent its interior, boundary surface, and exterior regions, respectively. We aim to reconstruct $\partial\Omega$ from a raw point cloud $\mathbf{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\} \subset \mathbb{R}^3$,

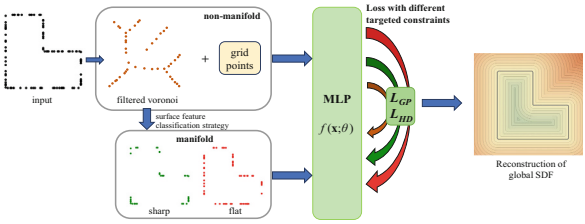


Fig. 2. Our pipeline begins with an input point cloud. We first compute its filtered Voronoi vertices, which are combined with grid points to form the non-manifold sampling set. These sampled points undergo surface feature classification to apply targeted constraints, visualized through color-coded sampling points and their corresponding loss function arrows. During function optimization, we introduce both L_{GP} and L_{HD} to achieve superior global SDF reconstruction.

sampled from the surface of Ω. Conventional methods train an implicit neural function $f(\mathbf{x};\theta)$ to accurately fit the true SDF f_t, then extract the zero-level set of the trained $f(\mathbf{x};\theta)$ as an approximation of $\partial\Omega$.

Specifically, for any point $\mathbf{x} \in \mathbb{R}^3$, the distance from $\mathbf{x}$ to $\partial\Omega$ is defined as $d(\mathbf{x}, \partial\Omega) := \min_{\mathbf{y}\in\partial\Omega} \|\mathbf{x}-\mathbf{y}\|$, where $\|\cdot\|$ denotes the ℓ_2-norm. Thus, the global true SDF $f_t : \mathbb{R}^3 \to \mathbb{R}$ assigns $d(\mathbf{x}, \partial\Omega)$ to interior points $\mathbf{x} \in \text{int}(\Omega)$, zero to boundary points $\mathbf{x} \in \partial\Omega$, and $-d(\mathbf{x}, \partial\Omega)$ to exterior points $\mathbf{x} \in \text{ext}(\Omega)$.

Subsequently, the neural network $f(\mathbf{x};\theta)$ with learnable parameters θ is trained unsupervised to fit $f_t(\mathbf{x})$, under specific constraints, as follows:

$$L_{\mathcal{M}} = \int_{\partial\Omega} |f(\mathbf{x};\theta)| d\mathbf{x}, \tag{1}$$

$$L_{\text{Eik}} = \int_{\mathbb{R}^3} |\|\nabla f(\mathbf{x};\theta)\| - 1| d\mathbf{x}, \tag{2}$$

$$L_{\mathcal{N}} = \int_{\mathbb{R}^3\setminus\partial\Omega} \exp(-\alpha|f(\mathbf{x};\theta)|) d\mathbf{x}, \quad \text{where } \alpha \gg 1. \tag{3}$$

These equations have been thoroughly discussed and detailed in [7,17]. As we only use coordinate inputs, the loss functions exclude ground-truth normal residuals. Finally, for the trained function $f(\mathbf{x};\theta)$, its zero-level set $\mathcal{S} = \{\mathbf{x}|f(\mathbf{x};\theta) = 0\}$ represents the desired reconstructed surface, typically extracted as an explicit triangular mesh via the marching cubes algorithm [11].

3 Method

Since our primary focus lies in global SDF approximation, we first introduce Gradient-Projection regularization (Sect. 3.1). Near the $\partial\Omega$, we approximate function residuals via Taylor expansion and introduce Hessian-Degeneration regularization (Sect. 3.2) to enhance training by incorporating higher-order information. Additionally, we introduce a categorized sampling strategy to apply distinct optimization strategies to points in different regions (Sect. 3.3).

3.1 Gradient-Projection Regularization

Traditional methods learning $f(\mathbf{x};\theta)$ through global energy minimization often fail to suppress undesirable local minima (Fig. 3(a)), where erroneous gradient estimations induce uncontrolled extrema (Fig. 3(b)). These extrema cause SDF deviations and gradient skew, degrading reconstruction accuracy. To address this, we introduce the Gradient-Projection regularization , which corrects gradient directions and promotes global viscosity approximation.

At its core, this method uses gradients from the true SDF $f_t(\mathbf{x})$ to compute surface projection points. Specifically, any query point $\mathbf{x}$ has a true projection $\mathbf{x}^*$ on $\partial\Omega$, such that $\|\mathbf{x}-\mathbf{x}^*\| = d(\mathbf{x},\partial\Omega)$, meaning $\mathbf{x}^*$ is the point closest to $\mathbf{x}$. On the other hand, $f_t(\mathbf{x})$ provides a specific method for computing the projection points, given by: $\mathbf{x}^* = \mathbf{x} - f_t(\mathbf{x}) \cdot \nabla f_t(\mathbf{x})/\|f_t(\mathbf{x})\|$. Let $\hat{\mathbf{x}}^*$ denote the projection estimate of the query point during training, defined as:

$$\hat{\mathbf{x}}^* = \mathbf{x} - f(\mathbf{x};\theta) \cdot \frac{\nabla f(\mathbf{x};\theta)}{\|\nabla f(\mathbf{x};\theta)\|}. \tag{4}$$

Figure 3(b) shows the projection estimate $\hat{\mathbf{x}}^*$, which aligns with the surface for well-trained points but deviates otherwise. Ideally, we expect $|f(\hat{\mathbf{x}}^*;\theta)| = 0$. While this helps align gradients with f_t, the regularization error grows as $f(\mathbf{x};\theta)$ deviates from $\partial\Omega$. To address this, we design an exponential weight function $\omega(\mathbf{x})$ based on the function value, defined as $\omega(\mathbf{x}) = \exp(-\beta|f(\mathbf{x};\theta)|)$, where β is a positive constant that controls the sensitivity of the weight function. Thus, we present the Gradient-Projection regularization:

$$L_{\mathrm{GP}} = \int_{\mathbb{R}^3} \omega(\mathbf{x}) \cdot |f(\hat{\mathbf{x}}^*;\theta)|\, d\mathbf{x} \tag{5}$$

L_{GP} guides function training from a geometric perspective, while the following L_{HD} complements it by incorporating higher-order differential information.

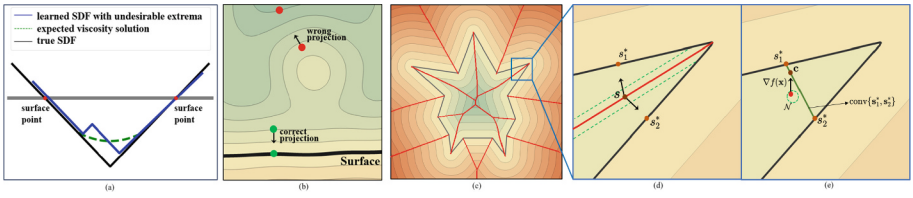


Fig. 3. The primary application scenarios of L_{GP} and L_{HD}. (a) 1D case. The green curve represents the expected smooth treatment near singularities. (b) Proper projections (green) yield accurate SDFs, while undesirable extrema distort projection directions (red). (c) True 2D SDF, with red regions marking non-differentiable singularities. (d) The singularity $\mathbf{s}$ has two projections. The green dashed line represents the small neighborhood of the singularity. (e) Under the guidance of L_{HD}, points in the neighborhood N exhibit stable degenerate gradients in the solution space of $\nabla^2 f(\mathbf{x};\theta)$. (Color figure online)

3.2 Hessian Degeneration Regularization

After regularizing the global SDF learning, we focus on the near-surface region, where function residuals can be expressed via Taylor expansion:

$$r(\mathbf{x}) = \underbrace{-f(\mathbf{x}) + f(\mathbf{x}^*) + \nabla f(\mathbf{x})^\top (\mathbf{x} - \mathbf{x}^*)}_{\text{Linear term}} + \underbrace{\frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^\top \nabla^2 f(\mathbf{x})(\mathbf{x} - \mathbf{x}^*) + o(\|\mathbf{x} - \mathbf{x}^*\|^2)}_{\text{Higher-order terms}}. \tag{6}$$

When $r(\mathbf{x}) = 0$, it indicates that $f(\mathbf{x};\theta) = f_t(\mathbf{x})$. In fact, enforcing $L_{\text{GP}} = 0$ leads to the relationships: $f(\mathbf{x}^*) = 0$ and $\mathbf{x} - \mathbf{x}^* = -f(\mathbf{x}) \cdot \nabla f(\mathbf{x}) / \|\nabla f(\mathbf{x})\|$, which constrain the linear term to be zero. For higher-order terms, a natural design choice is to set $\nabla^2 f(\mathbf{x})(\mathbf{x} - \mathbf{x}^*) = \mathbf{0}$. It is because the true SDF f_t satisfies $\|\nabla f_t\| = 1$ almost everywhere, and taking the gradient on both sides yield:

$$\nabla \|\nabla f_t\| = (\nabla^2 f_t)^\top \nabla f_t = \mathbf{0}, \tag{7}$$

where $\nabla^2 f_t$ is the Hessian matrix of f_t, and $\mathbf{0} = [0, 0, 0] \in \mathbb{R}^3$. To enforce this equality constraint, we formulate the L_{HD} as follows:

$$L_{\text{HD}} = \int_{\mathbb{R}^3} \left| \det\left(\nabla^2 f(\mathbf{x};\theta)\right) \right| \, d\mathbf{x}. \tag{8}$$

While a full-rank matrix drives $\nabla f(\mathbf{x};\theta)$ to vanish during optimization, a degenerate matrix retains gradients near singularities, thus smoothing non-differentiable regions. Consequently, while ensuring the residual $r = 0$ condition, our approach not only enhances surface reconstruction accuracy but also incorporates gradient linearity and non-differentiable smoothing properties through careful loss function design.

Gradient Linearity. Gradient linearity results from the geometric properties of the Hessian matrix, which characterizes the curvature $\kappa(\nabla f_t)$ in the direction of ∇f_t. Since the function is almost everywhere twice differentiable, combining this with (7) gives the curvature $\kappa(\nabla f_t) = \nabla f_t^\top \nabla^2 f_t \nabla f_t = 0$, which indicates the linearity of the function along the gradient direction.

Non-differentiable Smoothing. Singularities refer to the cusp positions of the SDF, as shown in the red regions of Fig. 3(c). With the underlying surface $\partial\Omega$, a point $\mathbf{s} \in \mathbb{R}^3$ can be defined as a singular point if it possesses two (or more) projection points $\mathbf{s}_1^*, \mathbf{s}_2^*$ on $\partial\Omega$, as shown in Fig. 3(d).

In the small neighborhood $\mathcal{N}(\mathbf{s}, \epsilon) = \{\mathbf{x} : \|\mathbf{x} - \mathbf{s}\| < \epsilon\}$ around $\mathbf{s}$, the function $f(\mathbf{x};\theta)$ may exhibit oscillating gradients. The proposed L_{HD} regularizes non-differentiable regions via degenerate solutions, constraining the gradients within the subdifferential range of $\mathbf{s}$, as shown in Fig. 3(e). Meanwhile, minimizing curvature energy smooths gradients within the neighborhood. In the next subsection, Voronoi vertices are introduced as approximations of singularities to further enhance the network's ability to handle non-differentiable regions.

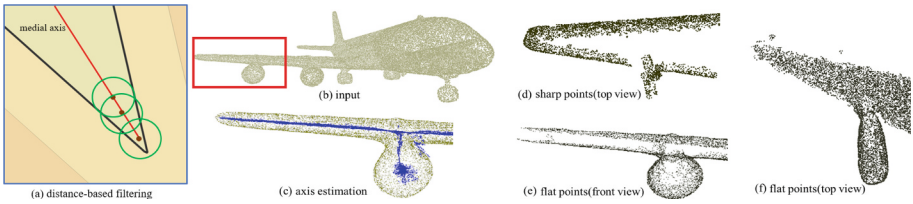


Fig. 4. Categorized sampling strategy. For the input point cloud (b), taking the wing as an example (red), we estimate the medial axis points (blue) using Voronoi points (c). Based on the distance between surface points and medial axis points, we dynamically classify regions into sharp (d) and flat (e)(f) areas. Different optimization strategies are then applied during training. (Color figure online)

3.3 Categorized Sampling Strategy

Standard methods apply uniform processing over the entire surface, causing roughness in planar regions and blurring sharp features. To address this, we identify surface points by their geometric characteristics and apply tailored optimization strategies.

Specifically, we first compute the Voronoi points $\mathbf{P}_{\mathcal{V}}$ of the input point cloud $\mathbf{P}$. To better approximate the medial axis, we refine $\mathbf{P}_{\mathcal{V}}$ via outlier removal into $\mathbf{P}'_{\mathcal{V}}$ (shown as brown points in Fig. 4(a) and blue points in Fig. 4(c)). $\mathbf{P}'_{\mathcal{V}}$ is also added to the non-manifold point sampling set. We then compute the nearest distances from surface points to $\mathbf{P}_{\mathcal{V}}$ and classify regions as flat ($\partial\Omega_F$) or sharp ($\partial\Omega_S$) using a dynamic threshold d_t derived from the global shape statistics (see green circles in Fig. 4(a)). Finally, we dynamically adapt the optimization strategy based on the classified surface samples. In addition to employing distinct training weights and annealing strategies, we introduce the divergence loss L_{div} from [3] for flat regions to suppress noise-induced surface undulations. Unlike the original, we reduce its truncation threshold to 0.01 to better match the geometry of our flat regions. Furthermore, $\mathbf{P}'_{\mathcal{V}}$ is incorporated into the non-manifold point sampling during the process, which enhances the network's learning of feature characteristics in the neighborhood of singularities.

In summary, we propose the overall Geometric-Differential Regularization:

$$L_{\text{GD}} = \lambda_1 L_{\text{GP}} + \lambda_2 L_{\text{HD}} + \lambda_3 \mathbb{I}_{\partial\Omega_F} \cdot L_{\text{div}}, \tag{9}$$

where $\mathbb{I}_{\partial\Omega_F}$ is an indicator function that takes the value of 1 in region $\partial\Omega_F$ and 0 elsewhere. Based on the sampling classification, we modify (1) and (2) as:

$$\begin{aligned} L'_{\mathcal{M}} &= u_1 \mathbb{I}_{\partial\Omega_F} \cdot L_{\mathcal{M}} + u_2 \mathbb{I}_{\partial\Omega_S} \cdot L_{\mathcal{M}}, \\ L'_{\text{Eik}} &= v_1 \mathbb{I}_{\partial\Omega_F} \cdot L_{\text{Eik}} + v_2 \mathbb{I}_{\partial\Omega_S} \cdot L_{\text{Eik}} + v_3 \mathbb{I}_{\mathbb{R}^3 \setminus \partial\Omega} \cdot L_{\text{Eik}}. \end{aligned} \tag{10}$$

where $u_i, v_i \in [0, 1]$. During training, we dynamically adjust the loss functions for different regions, yielding the total loss function:

$$L_{\text{total}} = L_{\text{GD}} + \lambda_4 L_{\mathcal{N}} + \lambda_5 L'_{\mathcal{M}} + \lambda_6 L'_{\text{Eik}}. \tag{11}$$

Table 1. Chamfer and Hausdorff Distances on the SRB Dataset

Method	$d_C \downarrow$	$d_H \downarrow$	$d_{\overrightarrow{C}} \downarrow$	$d_{\overrightarrow{H}} \downarrow$
IGR wo n	1.38	16.33	0.25	2.96
SIREN wo n	0.42	7.67	**0.08**	**1.42**
SAL	0.36	7.47	0.13	3.50
IGR+FF	0.96	11.06	0.32	4.75
DiGS	0.19	3.52	**0.08**	1.47
StEik	<u>0.18</u>	<u>2.80</u>	0.10	<u>1.45</u>
Ours	**0.169**	**2.280**	<u>0.095</u>	1.474

In the above formulation, the coefficients λ_i represent weight hyperparameters, whose optimal values will be determined in the experimental section.

4 Experiments

4.1 Experimental Datasets

All experiments are conducted on a RTX 3090 Ti GPU. We evaluate our method on the Surface Reconstruction Benchmark (SRB) [4] and ShapeNet [6] datasets to demonstrate its superiority over state-of-the-art (SOTA) methods in surface reconstruction tasks. To validate the global SDF learning capability of GDR, we additionally test it on 2D shapes, where SDF visualization is more intuitive.

The SRB dataset provides a standardized set of point clouds for testing and evaluating surface reconstruction algorithms. In our experiments, we use five complex shapes from SRB, which feature varying levels of geometric intricacy. To evaluate the alignment between the reconstructed meshes and the ground truth, we utilize Chamfer distance (d_C) and Hausdorff distance (d_H). Similarly, these metrics, denoted as $d_{\overrightarrow{C}}$ and $d_{\overrightarrow{H}}$, are used to assess the relationship between the reconstructed meshes and the input noisy point cloud.

ShapeNet is a large-scale, richly annotated dataset containing a diverse collection of 3D models across multiple object categories, representing a wide array of real-world objects. In this study, we use a preprocessed watertight subset of ShapeNet, consisting of 13 object categories, each containing 20 distinct shapes. For each shape, 100,000 points are uniformly sampled from the surface. We evaluate the performance of our method on the 3D reconstruction task using the Intersection over Union (IoU) metric and Squared Chamfer distance.

4.2 Comparison with Baselines

SRB. The network is trained for 10,000 iterations with a learning rate of 5×10^{-5}. The regularization weights λ_i are sequentially configured as: 20, 100, 10, 50, 3000, 100. Additionally, the model parameters are $\alpha = 100$ and $\beta = 75$. To accommodate diverse surface characteristics and mitigate noise interference, we

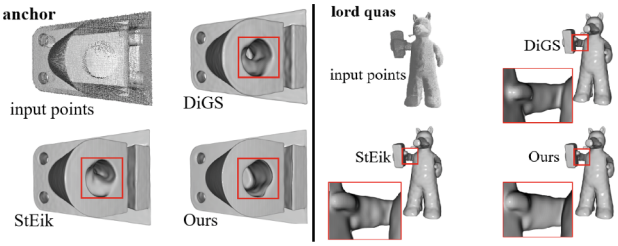


Fig. 5. Reconstruction results of SRB. GDR guides the function to learn the concave structures within the red box. In the flat regions, our method produces smoother surfaces with more distinct boundaries. In the lord quas shape, GDR provides a better approximation around the hinged surfaces, as indicated by the red box. (Color figure online)

Table 2. Squared Chamfer and IoU on the ShapeNet Dataset

Method	Squared Chamfer ↓			IoU ↑		
	Mean	Median	Std	Mean	Median	Std
IGR wo n	6.85e-3	8.12e-3	8.41e-3	0.3093	0.2690	0.1853
SIREN wo n	3.08e-4	2.58e-4	3.26e-4	0.3085	0.2952	0.2014
SAL	1.14e-3	2.11e-4	3.63e-3	0.4030	0.3944	0.2722
DiGS	1.32e-5	2.55e-5	4.73e-4	0.9390	0.9764	0.1262
StEik	6.86e-5	**6.33e-6**	3.34e-4	0.9671	0.9841	0.0878
PG-SDF	3.72e-5	2.40e-5	**5.39e-5**	0.9729	**0.9896**	0.0676
Ours	**3.35e-5**	2.27e-5	9.72e-5	**0.9733**	0.9893	**0.0636**

configure $u_1 = 0.8, u_2 = 1$ and $v_1 = 0.95, v_2 = 0.8, v_3 = 1$. We apply cosine annealing to sharp features to facilitate thin-structure learning in the network.

We evaluate GDR against SOTA methods, including IGR (without normals) [7], SIREN (without normals) [17], SAL [1], DiGS [3], and StEik [22]. Table 1 summarizes all the results for SRB complex shapes. From the table, it is evident that our method achieves the superior performance compared to SOTA methods. Additionally, our method achieves the suitable $d_{\overrightarrow{C}}$ and $d_{\overrightarrow{H}}$, demonstrating its robustness to noise. Qualitative results are shown in Fig. 5. Notably, for the anchor shape, our method produces smoother surfaces, more distinct boundaries, and better captures high-frequency concave structures.

ShapeNet. The experimental setup for ShapeNet remains nearly identical to SRB, except with $\lambda_3 = 5000$. Table 2 presents a quantitative comparison between our method and SOTA approaches, with the addition of the PG-SDF [8]. The results in Table 2 demonstrate that our method achieves the best or second-best performance across all metrics. While PG-SDF reduces variance by introducing discretized optimization, it introduces jagged artifacts (Fig. 6). Among purely neural network-based approaches, our method's lowest variance indicates

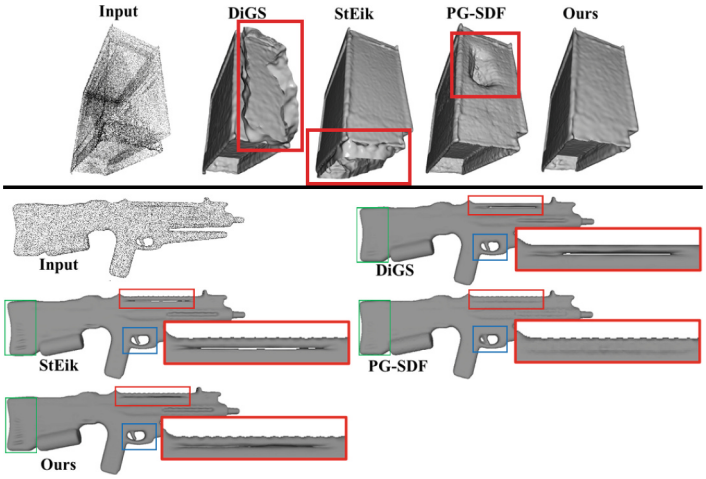


Fig. 6. Reconstruction results of ShapeNet. For the rifle shape, comparative results show that baseline methods either incorrectly classify these regions as through-holes (DiGS; Steik) or fail to capture the depression details entirely (PG-SDF). (Color figure online)

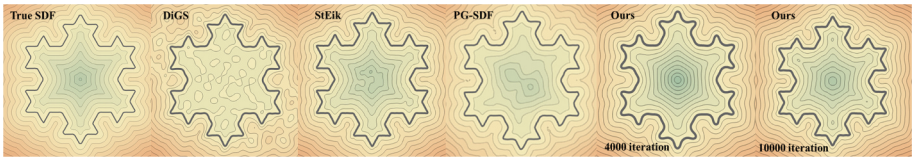


Fig. 7. Function learning in 2D. The result after 4,000 iterations of training with GDR, where the global SDF has roughly converged to the true values.

stronger robustness to diverse shapes. Figure 6 shows qualitative results on the ShapeNet dataset. For the cabinet shape, conventional methods fail to accurately reconstruct high-frequency thin structures within the interior geometry. In contrast, our approach demonstrates superior capability in preserving these challenging features. For the rifle shape, quantitative and visual comparisons reveal three key advantages of our method: (1) Precise reconstruction of the buttstock depression (green highlight), (2) Accurate separation of trigger components (blue highlight) and (3) Faithful recovery of both the undulating surface profile and correct indentation features (red highlight).

2D Shape. To demonstrate the effectiveness of our method in globally fitting the SDF, we visualize the SDF field for a 2D shape, as shown in Fig. 7. Our method converges to an approximate shape and a stable SDF approximation without undesirable extrema within 4,000 iterations. After 10,000 iterations, the method sacrifices some SDF accuracy to improve surface reconstruction. These visual results demonstrate that GDR performs effectively in learning globally approximated viscosity solutions.

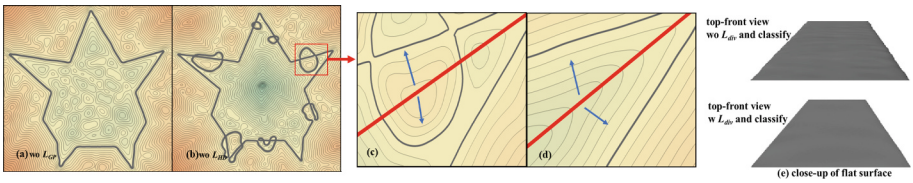


Fig. 8. Ablation Study. (c) An enlargement of the red region (b), where we incorrectly sampled two points near a singularity, resulting in erroneous projection directions. (d) The learned implicit SDF under GDR. (e) Close-up of flat planar regions. (Color figure online)

Table 3. Ablation Experiments on Shapenet

Method	Squared Chamfer	IoU	Time(min)
L_{total} wo L_{GP}	8.65e−5	0.9602	13.57
L_{total} w L_{GP} wo $\omega(\mathbf{x})$	5.14e−4	0.7412	13.35
L_{total} wo L_{HD}	8.43e−4	0.7036	11.46
L_{total} wo L_{div} and classify	4.24e−5	0.9701	14.08
L_{total}	3.35e-5	0.9733	15.22

4.3 Ablation Study

We evaluate each regularization term's impact through ShapeNet ablation studies, including cases without weighting and classification. The experimental results are shown in Table 3. To test the effectiveness of the proposed regularization terms in improving global SDF accuracy, we also conducted experiments on a 2D shape, as shown in Fig. 8. In (a), training without L_{GP} results in behavior similar to most SOTA methods, where the global viscosity solution only approximates the true SDF near the surface. In (b), although L_{GP} guides the sampling points toward the zero-level set, the absence of L_{HD}, which provides higher-order information and enforces gradient linearity, leads to errors near the surface and at the central singularity. The SDF learned under GDR, as shown in (d), demonstrates accurate function approximation. (e) illustrates a top-front view of planar regions, where the application of our surface point classification strategy yields significantly improved planarity in the learned surfaces. Additionally, the 'time' column in Table 3 represents the duration required for 10,000 iterations. Our method trades off some fitting speed to achieve higher accuracy. Moreover, multiple training strategies can be flexibly combined and embedded according to specific application scenarios.

5 Conclusion

In this paper, we propose a novel regularization method called Geometric- Differential Regularization (GDR) for accurate SDF reconstruction. This new regularization consists of two components, Gradient-Projection regularization and

Hessian-Degeneration regularization. GDR not only enables an accurate global approximation of the true SDF but also improves local fitting near the surface, ultimately achieving higher accuracy in surface reconstruction. We further introduce a medial-axis-distance-based surface feature sampling strategy to enhance surface reconstruction quality. We demonstrate the superior performance of our method in both SDF and surfaces reconstruction through various experiments.

Acknowledgment. This work was supported in part by the National Natural Science Foundation of China under Grant 62471418, in part by Fujian Provincial Natural Science Foundation of China under Grant 2024J01058, and in part by the Major Science and Technology Special Project of Fujian Province under Grant 2024HZ022003.

References

1. Atzmon, M., Lipman, Y.: Sal: sign agnostic learning of shapes from raw data. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2562–2571 (2020). https://doi.org/10.1109/CVPR42600.2020.00264
2. Belyaev, A.G., Fayolle, P.A.: On variational and PDE-based distance function approximations. Comput. Graphics Forum **34**(8), 104–118 (2015). https://doi.org/10.1111/cgf.12611
3. Ben-Shabat, Y., Koneputugodage, C.H., Gould, S.: Digs: divergence guided shape implicit neural representation for unoriented point clouds. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 19301–19310 (2022). https://doi.org/10.1109/CVPR52688.2022.01872
4. Berger, M., Levine, J.A., Nonato, L.G., Taubin, G., Silva, C.T.: A benchmark for surface reconstruction. ACM Trans. Graphics (TOG) **32**(2), 1–17 (2013)
5. Calakli, F., Taubin, G.: SSD: smooth signed distance surface reconstruction. Comput. Graphics Forum **30**(7), 1993–2002 (2011). https://doi.org/10.1111/j.1467-8659.2011.02058.x
6. Chang, A., et al.: ShapeNet: an information-rich 3D model repository (2015). https://doi.org/10.48550/arXiv.1512.03012
7. Gropp, A., Yariv, L., Haim, N., Atzmon, M., Lipman, Y.: Implicit geometric regularization for learning shapes. In: Proceedings of the 37th International Conference on Machine Learning. ICML'20, JMLR.org (2020)
8. Koneputugodage, C.H., Ben-Shabat, Y., Campbell, D., Gould, S.: Small steps and level sets: fitting neural surface models with point guidance. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 21456–21465 (2024). https://doi.org/10.1109/CVPR52733.2024.02027
9. Lei, H.: Offsetopt: Explicit surface reconstruction without normals (2025). https://arxiv.org/abs/2503.15763
10. Li, S., Gao, G., Liu, Y., Gu, M., Liu, Y.S.: Implicit filtering for learning neural signed distance functions from 3D point clouds. In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G. (eds.) Computer Vision – ECCV 2024, Part VI. LNCS, p. 234–251. Springer, Cham(2024). https://doi.org/10.1007/978-3-031-72658-3_14
11. Lorensen, W.E., Cline, H.E.: Marching cubes: a high resolution 3D surface construction algorithm. SIGGRAPH Comput. Graph. **21**(4), 163–169 (1987). https://doi.org/10.1145/37402.37422

12. Ma, B., Han, Z., Liu, Y.S., Zwicker, M.: Neural-pull: learning signed distance function from point clouds by learning to pull space onto surface. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 7246–7257. PMLR (2021)
13. Ortiz, J., Clegg, A., Dong, J., Sucar, E., Novotny, D., Zollhoefer, M., Mukadam, M.: iSDF: real-time neural signed distance fields for robot perception. In: Proceedings of Robotics: Science and Systems, New York City, NY, USA (2022). https://doi.org/10.15607/RSS.2022.XVIII.012
14. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: DeepsDF: learning continuous signed distance functions for shape representation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 165–174 (2019). https://doi.org/10.1109/CVPR.2019.00025
15. Ran, H., Liu, J., Wang, C.: Surface representation for point clouds. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 18920–18930 (2022). https://doi.org/10.1109/CVPR52688.2022.01837
16. Sitzmann, V., Chan, E.R., Tucker, R., Snavely, N., Wetzstein, G.: MetasDF: meta-learning signed distance functions. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20, Red Hook, NY, USA. Curran Associates Inc. (2020)
17. Sitzmann, V., Martel, J.N.P., Bergman, A.W., Lindell, D.B., Wetzstein, G.: Implicit neural representations with periodic activation functions. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20, Curran Associates Inc., Red Hook, NY, USA (2020)
18. Tancik, M., et al.: Fourier features let networks learn high frequency functions in low dimensional domains. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20, Red Hook, NY, USA. Curran Associates Inc. (2020)
19. Wang, S., et al.: Metricgrids: arbitrary nonlinear approximation with elementary metric grids based implicit neural representation (2025). https://arxiv.org/abs/2503.10000
20. Wang, Z., et al.: Neural-IMLS: self-supervised implicit moving least-squares network for surface reconstruction. IEEE Trans. Visual Comput. Graphics **30**(8), 5018–5033 (2024). https://doi.org/10.1109/TVCG.2023.3284233
21. Yang, H., et al.: Unipad: a universal pre-training paradigm for autonomous driving. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 15238–15250 (2024). https://doi.org/10.1109/CVPR52733.2024.01443
22. Yang, H., Sun, Y., Sundaramoorthi, G., Yezzi, A.: Steik: stabilizing the optimization of neural signed distance functions and finer shape representation. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. NIPS '23, Red Hook, NY, USA. Curran Associates Inc. (2023)



Optimized Supervised Control of Stochastic Timed Discrete Event Systems Using Supervisory Control Theory and Reinforcement Learning

Yingjun Liu, Fuchun Liu(✉), and Hongzhen Zhu

School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, China
fliu2011@163.com

Abstract. This paper proposes an optimization approach for supervised control of stochastic timed discrete event systems (STDES) that integrates reinforcement learning (RL) with supervisory control theory (SCT). Unlike crisp discrete event systems (CDES) with disregard for time, STDES analyzes more complex dynamic systems, which consider general distributions for sojourn time between states. So STDES is closer to complex physical systems and its optimal control problem is more challenging. In this paper, the language of the uncontrolled system and its deterministic specification are firstly generated by automata according to the logic features of STDES system; Then, a supervisor is derived using SCT to ensure safety control by disabling specific controllable event sequences; Next, the controlled automata structure from SCT is converted into a Semi-Markov Decision Process(SMDP) framework by adding the temporal and probabilistic variable of the uncontrollable events. Finally, a modified SCT-Timed RL adaptive control algorithm is developed to optimize system performance by designing a new Bellman equation and reward function. The results show that the proposed controller enhances system safety and flexibility, with an 8.9% performance improvement over non-intelligent method.

Keywords: Supervisory control theory · semi-markov decision process · reinforcement learning · stochastic timed discrete event systems

1 Introduction

As a mathematical modeling framework, discrete event system (DES) [1] has been widely used to depict and analyze intricate dynamic physical systems in various domains, including production line [2], supervised control [3]. Until now, many scholars have made in-depth studies on the control of DES. As a common denominator, the supervisory control theory (SCT) proposed by [4] is the

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 402–414, 2026.
https://doi.org/10.1007/978-3-032-04555-3_33

fundamental methodology for DES. The core of SCT is to derive a supervisor that ensures safe control through the synchronous operation of the uncontrolled system and deterministic specifications. However, with implicit specifications such as "producing the product with highest profit", the maximally permissive non-blocking supervisor generated by SCT cannot satisfy the requirements of optimal control. In general, this problem is often involved in stochastic discrete event system (SDES), which is based on probabilistic automata and attaches a probability value to each state transition on the basis of classical DES. Optimal control strategies widely used for SDES encompass adaptive control [5], reinforcement learning(RL) [6], etc. Among these methods, RL has emerged as a promising approach to optimal control, particularly with advances in deep learning.

The combination structures with SCT and machine learning for SDES have been addressed in some literature before, the selection of the regimen DES system was learned by [7]. The RL-based supervisor was proposed in [8]. And the optimal path of the cat and mouse problem was solved in [9]. The above work solved the optimal control problem in the SDES systems, in which the sojourn times between states were ignored. In some real physical systems, the transition function depends not only on states and events but also on the sojourn time between states. These kinds of system are called stochastic timed discrete event systems (STDES) [11], which are closer to complex physical systems. The safety analysis for STDES was proposed in [10], the parallel control for STDES is proposed in [12]. However, few studies have been performed on optimal control of STDES in the past literature. The literature [13] lists the planning, learning, and control process of the RL-based STDES system. Unlike minor adjustments, this paper focuses on explicitly modifying the system structure. Meanwhile, the temporal attribute is added to the supervised system for modeling as SMDP. And experiments based on the environment in [14] are carried out to corroborate the proposed approach with maximizing the production profit in the shortest time. The result shows 8.9% performance improvement compared with non-intelligent framework.

2 Modeling and Methodology

Two model representations will be carried out through the control processes based on the proposed framework. Firstly, model the uncontrolled system as a formulation representation for SCT operation. Before modeling STDES, consider to model CDES as the classical finite state automata $G = (Q, \Sigma, \delta, q_0, Q_m)$ with Q as the state space including initial state q_0 and goal state set $Q_m \subseteq Q$; Σ represents the event space with $\Sigma = \Sigma_{uc} \dot{\cup} \Sigma_c$ where Σ_{uc} is the uncontrollable events and Σ_c is the controllable events; the transition function is $\delta : Q \times \Sigma \rightarrow Q$; Σ^* denotes the Kleene closure, comprising all finite strings Str that can be constructed from Σ; the language from G is $L(G) = \{Str \in \Sigma^* : \delta(q_0, Str)!\}$, with "!" meaning "defined". Meanwhile, $L_m(G) = \{Str \in \Sigma^* : \delta(q_0, Str) \in Q_m\}$ denotes the marked language of G. Then based on the classical automata G,

the STDES can be modeled as stochastic automata $\mathcal{A} = (Q, \Sigma, \delta, q_0, Q_m, \mathcal{T})$ by importing the stochastic transition mapping function $\delta : Q \times \Sigma \times Q \to [0, 1]$ with $P\left[Q_{k+1} = q' \mid Q_k = q, \Sigma_k = \varrho\right]$. The function may be partial for events $\varrho \notin \Gamma(q)$. k corresponds to the time in which events occur. Then on this basis, turn to a stochastic timed automata (STA) $\mathcal{A} = (Q, \Sigma, \delta, q_0, Q_m, \mathcal{T})$ by adding the sojourn time $\mathcal{T} = \{\mathcal{T}_\varrho : \varrho \in \Sigma\}$ as a stochastic clock structure for the lifetimes of the event. Essentially, it assigns a stochastic clock sequence $\{\mathcal{T}_{\varrho,k}\} = \{\mathcal{T}_{\varrho,1}, \mathcal{T}_{\varrho,2}, \ldots\}$ to each event ϱ, which is independent and identically distributed by allowing general distributions for each timed event.

Secondly, define the control model for STDES as SMDP for the optimization procedure. SMDP is an extension of the classical MDP that relaxes the Markov property by permitting the transition probability to depend on the duration of the state stay. For the optimization part in the proposed framework, the control process is defined as $SMDP = (S, A, T, R, \gamma, Z)$ with $Z(s_{k+1}, \tau_k \mid (s_k, a_k))$ representing the probability distribution that a state changes in less than time τ_k, given the system in condition (s_k, a_k). Therefore, the proposed STA modeling method is suitable for the security control of STDES with the adoption of SCT. For the second form as SMDP for STDES, the goal is to seek a policy which can achieve the maximal accumulated reward.

$$Z(s_{k+1}, \tau_k \mid (s_k, a_k)) = P\left(S_{k+1} = s_{k+1}, \mathcal{T}_k \leq \tau_k \mid S_k = s_k, A_k = a_k\right) \tag{1}$$

The framework of the proposed method is depicted in Fig. 1. The STDES and the deterministic specification are first modeled as automata G and E, and then a supervisor is synthesized by the SCT operation. The supervised DES system is then translated into STA by adding the sojourn time and the transition probability matrix. Moreover, STA is translated into SMDP by adding a discount factor and a reward function. Finally, a timed-based optimal algorithm is performed in the SMDP environment to achieve an optimal policy. The details of each component are delineated as follows.

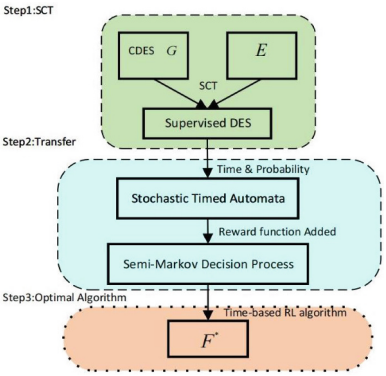


Fig. 1. Architecture of the proposed optimal control strategy.

2.1 Synthesizing Non-blocking Supremal Controllable System

For the logic part of STDES, generate an automata $G = (Q_g, \Sigma_g, \delta_g, q_{g0}, Q_{gm})$ and an automata $E = (Q_\iota, \Sigma_\iota, \delta_\iota, q_{\iota 0}, Q_{\iota m})$ respectively as the model of the original system and the deterministic specification; and then the parallel composition is used to generate the synthesized system U via the operation $U = G \parallel E$. The automata U is constructed as $U = (Q_g \times Q_\iota, \Sigma_g \cup \Sigma_\iota, \delta_u, \Gamma_g, (q_{g0}, q_{\iota 0}), Q_{gm} \times Q_{\iota m})$ with δ_u set as:

$$\begin{cases} (q_g, q_\iota) \xrightarrow{\varrho} (q_g^{'}, q_\iota^{'}), \text{ if } \varrho \in \Sigma_g \cap \Sigma_\iota,\ q_g \xrightarrow{\varrho} q_g^{'},\ q_\iota \xrightarrow{\varrho} q_\iota^{'}; \\ (q_g, q_\iota) \xrightarrow{\varrho} (q_g^{'}, q_\iota), \text{ if } \varrho \in \Sigma_g \backslash \Sigma_\iota,\ q_g \xrightarrow{\varrho} q_g^{'}; \\ (q_g, q_e) \xrightarrow{\varrho} (q_g, q_\iota^{'}), \text{ if } \varrho \in \Sigma_\iota \backslash \Sigma_g,\ q_\iota \xrightarrow{\varrho} q_\iota^{'}; \\ \text{undefined}, \qquad\qquad \text{otherwise.} \end{cases} \tag{2}$$

For any non-empty language $L(U) \subseteq L(G)$, $L(U)$ is prefix closed and controllable if and only if a supervisor Ψ is available such that $L(\Psi/G) = \underline{L(U)}$. The controllable condition is: $\forall s \in \overline{L(U)}, \forall \varrho \in \Sigma_{uc}, s\varrho \in L(G) \Rightarrow s\varrho \in \overline{L(U)}$. It should be emphasized that the language $L(U)$ may not always be controllable, the language $L(U)$ may not exist a supervisor Ψ due to its uncontrollable property. Thus, it is necessary to look for a maximal controllable language $L(U)^{\uparrow}$.

Theorem 1. *For any non-empty language $L(U) \subseteq L(G)$, there is a unique supervisor Ψ to make $L(\Psi/G) = L(U)^{\uparrow}$, where $L(U)^{\uparrow}$ is the maximal controllable, and prefix closed language of $L(U)$.*

Thus, undesirable states accessed through uncontrollable events and the associated paths that violate the specification need to be eliminated. Furthermore, it is generally expected that the sublanguage $L(U)^{\uparrow}$ is not blocked in the context of SCT. This ensures that the system can reach the goal states. Meanwhile, any state in $L(U)^{\uparrow}$ that is neither accessible nor co-accessible should be eliminated. This can be achieved using the "trim" operator. Denote $\mathrm{Sup}\, C(G, E)$ as the supremal controllable language for system G. In summary, to derive $\mathrm{Sup}\, C(G, E)$ for G, a supervisor Ψ must be designed by following the above steps to make:

$$\mathrm{Sup}\, C(G, E) = L_{\mathrm{m}}(\Psi/G). \tag{3}$$

Finally, the supervised system under supervised control is derived as an automata $\varsigma = (Q_\varsigma, \Sigma_\varsigma, \delta_\varsigma, q_{\varsigma 0}, Q_{\varsigma m})$. The complexity of this process can be assessed by the computational cost on $U = G \parallel E$. Specifically, if system G includes m states and E has n states, the computational complexity increases in $O\left(m \cdot n\right)^2$.

2.2 Translate Supervised Automata ς to SMDP Environment

Based on the supervised system ς, design the STA $\mathcal{A}$ automation's states, events and transition functions from ς based on $Q_\mathcal{A} \equiv Q_\varsigma, \Sigma_\mathcal{A} \equiv \Sigma_\varsigma, \delta_\mathcal{A} \equiv \delta_\varsigma$, equipped with sojourn time $\mathcal{T} = \{\mathcal{T}_\varrho : \varrho \in \Sigma_\mathcal{A}\}$ as a stochastic clock structure for every event lifetimes. The STA automata $\mathcal{A} = (Q_\mathcal{A}, \Sigma_\mathcal{A}, \delta_\mathcal{A}, q_{\mathcal{A}0}, Q_{\mathcal{A}m}, \mathcal{T})$ is then converted into an XML document to serve as the interface. Subsequently, according to $S \equiv Q_\mathcal{A}, A \equiv \Sigma_\mathcal{A}, T \equiv \delta_\mathcal{A}$, a parsing tool is developed to derive the state and event set, the partial transition function from the XML document for the

SMDP environment. Due to the time effect, the function $R(s_k, a_k)$ is designed for the reward value of SMDP as follows: $\kappa(s_k, a_k)$ is set as the value of immediate reward; $\mu(s_k, a_k)$ is set as the value of the reward rate. Add a discount factor for immediate reward. Unlike the classical RL algorithm, the parameter ϑ equivalent to the MDP discount factor γ is designed in the reward function according to the time delay. The expected reward for executing the action a_k as long as staying in state S_k, so the total reward $r(s_k, a_k)$ is the following:

$$r(s_k, a_k) = e^{-\vartheta \tau_k} \kappa(s_k, a_k) + \int_0^{\tau_k} e^{-\vartheta t} \mu(s_k, a_k) dt \tag{4}$$

Let $W_{s_k, s_{k+1}}(t)$ represents the probability function of the sojourn time with the probability density function as $w_{s_k, s_{k+1}}(t)$ by $W_{s_k, s_{k+1}}(t) = \int_0^{\tau_k} w_{s_k, s_{k+1}}(t) dt$:

$$W_{s_k, s_{k+1}}(\tau_k) = P\left(T_{k+1} - T_k \leq \tau_k \mid S_k = s_k, S_{k+1} = s_{k+1}, A_k = a_k\right) \tag{5}$$

Here T_k represents the jump moment. Given the system in (s_k, a_k), the probability from s_k changing to s_{k+1} in less than τ_k is calculated as follows:

$$\begin{aligned} Z(s_{k+1}, \tau_k \mid s_k, a_k) &= P\left(S_{k+1} = s_{k+1}, \mathcal{T}_k \leq \tau_k \mid S_k = s_k, A_k = a_k\right) \\ &= P\left(s_{k+1} \mid s_k, a_k\right) W_{s_k, s_{k+1}}(\tau_k) \end{aligned} \tag{6}$$

Obtaining the sojourn time $\mathcal{T}_k$ and the density associated with its STA distribution function, design the probability distribution function Z according to the Eq. (6). Lastly, the supervised system ς is converted to the SMDP model $SM = (S, A, T, R, \vartheta, Z)$, where the agent can maximize the expected total discount reward through the SCT-Timed RL algorithm for SMDP.

2.3 Design for SCT-Timed RL Algorithm

Two key algorithms are designed for the proposed method. The action selection policy is designed in Algorithm 1. Actions are selected according to the transition probability of uncontrollable events $p(a_k|s_k)$. Given that uncontrollable events $A^u_{s_k}$ in state s_k are inevitable, the inherent uncertainty of these events leads to an uncertain state transition within the system. The weight-based action policy operates as follows. When a controllable action a^k_c is chosen by the controller in state s_k, the system will prohibit other controllable actions in s_k. Consequently, lines 3–7 reflect the greedy strategy, where the actions with maximum $Q(s_k, a_k)$ are executed. The exploration mechanism for controllable events is represented in line 10. In contrast, when an uncontrollable action a^k_u is selected in s_k, the controllable actions are disabled, but other uncontrollable actions remain possible. Therefore, lines 13–15 utilize transition probabilities to select among uncontrollable actions. In line 17, the final action will be randomly selected from Δ.

Unlike traditional RL algorithms based on MDP, the control policy π for SMDP models will not only be based on states, but will also need to be taken into account with the impact of sojourn time and probabilistic features. Let $a_k = \pi(s_k)$. Like in MDP, the objective in SMDP is modified as the follows:

$$V_\pi(s_k) = E\left\{\sum_{k=0}^{\infty} e^{-\vartheta \tau_k} \left[r\left(s_k, a_k\right)\right] \mid S_k = s_0, \pi\right\} \tag{7}$$

Algorithm 1: Weight-based Action Policy

Input: $Q(s_k, a_k)$, A^k, A_u, A_c, $p(a_k|s_k)$, threshold ϵ ;
Output: Action a_k ;
$A_u^k = A^k \cap A_u$; $A_c^k = A^k \cap A_c$;
Initialize list Δ with empty element ;
if $A_c^i \neq \emptyset$ **then**
  Initialize number x randomly between $[0, 1]$;
  if $x > \varepsilon$ **then**
    Action a_c^k selected with highest $Q(s_k, a_k)$ in A_c^k;
    Store a_c^k in Δ;
  end
  else
    Action a_c^k selected randomly and stored in Δ;
  end
end
if $A_u^k \neq \emptyset$ **then**
  Select action a_u^k based on probabilities of $p(a_k|s_k)$;
  Store a_u^k in Δ ;
end
Select random action in Δ as a_k;
Return the action a_k;

where ϑ is the discount factor based on time. So, the modified Bellman equation of state value is derived as:

$$\sum_{a_k \in A} p(a_k \mid s_k) \left[r(s_k, a_k) + \sum_{s_{k+1} \in S} \left(P\left(s_{k+1} \mid s_k, a_k\right) \left(\int_0^{\tau_k} e^{-\vartheta t} w_{s_k, s_{k+1}}(t) \right) V^{\pi}(s_{k+1}) \right) \right] \tag{8}$$

Let $M(s_{k+1} \mid s_k, a_k)$ represent the discount coefficient that varies with the sojourn time τ_k and the probability distribution of state transition $P\left(s_{k+1} \mid s_k, a_k\right)$.

$$M(s_{k+1} \mid (s_k, a_k)) = P\left(s_{k+1} \mid (s_k, a_k)\right) \int_0^{\tau_k} e^{-\vartheta t} w_{s_k, s_{k+1}}(t) \tag{9}$$

Next, modify the classical Q-learning algorithm according to the value of the discount coefficient $M(s_{k+1} \mid s_k, a_k)$. So the updating process is as follows:

$$\begin{aligned} Q(s_k, a_k) \leftarrow & Q(s_k, a_k) + \alpha_k \left[r\left(s_k, a_k\right) \right. \\ & \left. + M(s_{k+1} \mid s_k, a_k) \left(\max_{a_{k+1} \in A_{k+1}} Q\left(s_{k+1}, a_{k+1}\right) \right) - Q(s_k, a_k) \right] \end{aligned} \tag{10}$$

where α_k denotes the learning rate. Finally, the optimal control policy π^* is obtained by selecting actions from $Q(s_k, a_k)$. In Algorithm 2, design the average reward value with the proposed method. The algorithm terminates the iteration with maxEpisode and returns an optimal controller that satisfies deterministic

specifications and implicit specifications. Firstly, SCT operation is carried out to achieve a superized automata ς in lines 1–3; then translate the above automata into SMDP environment in line 4; lastly, on lines 5–24, the learning algorithm is performed in which the main distinction from MDP is the calculation of the Bellman optimal function on lines 12–14 and the update of the optimal Q function $Q^*(s_k, a_k)$ on line 15 with the sojourn time considered. Finally, the agent obtains the optimal policy π^* in Line 26.

3 Case Study

The efficacy of the proposed framework is illustrated in the following section. The implementation is coded in Python and Supremica, and executed on a system featuring an Intel Core i9-11900 processor, operating under Ubuntu 20.04.1.

Algorithm 2: SCT-Timed RL Algorithm for STDES

Input: System G_i, Spec E_j, ϑ ;
Output: Optimal policy π;
1 $G_i = (Q_{gi}, \Sigma_{gi}, f_{gi}, \Gamma_{gi}, q_{gi0}, Q_{gim})$;
2 $E_j = (Q_{ej}, \Sigma_{ej}, f_{ej}, \Gamma_{ej}, q_{ej0}, Q_{ejm})$;
3 SCT on $\varsigma = G_{i_1} \parallel G_{i_2} ... \parallel E_{j_1} \parallel E_{j_n}$;
4 $SMDP = (S, A, T, R, \gamma, Z)$ from ς ;
5 EpiNum $\leftarrow$ 0; AveReward $\leftarrow$ 0; SumReward $\leftarrow$ 0;
6 Initialize Avelist as $L = \emptyset$;
7 **while** $EpiNum \leq maxEpisode$ **do**
8 s $\leftarrow s_0$;
9 StepNum $\leftarrow$ 0; EpiReward $\leftarrow$ 0;
10 **while** $StepNum \leq maxStep$ *or* $done$ **do**
11 Select a_k according to Algorithm 1 ;
12 Import $r_(s_k, a_k)$ as Eq. (4);
13 τ_k; $s_{k+1}, r(s_k, a_k), done \leftarrow Step()$;
14 $M(s_{k+1} \mid s_k, a_k) \leftarrow P(s_{k+1} \mid s_k, a_k) \int_0^{\tau_k} e^{-\vartheta t} dt$;
15 $Q(s_k, a_k) = r(s_k, a_k) + M(s_{k+1} \mid (s_k, a_k)) \max_{a_{k+1}} Q(s_{k+1}, a_{k+1})(1 - done)$;
16 Update $Q^*(s_k, a_k)$ as Eq. (10);
17 EpiReward $\leftarrow$ EpiReward $+ Q(s_k, a_k)$;
18 Set $S_k \leftarrow S_{k+1}$;
19 StepNum $\leftarrow$ StepNum $+ 1$
20 **end**
21 EpiNum $\leftarrow$ EpiNum $+ 1$;
22 Add EpiReward to Avelist;
23 SumReward $\leftarrow \sum_{v \in \text{Avelist}} v$;
24 AveReward $\leftarrow$ SumReward/len(Avelist);
25 **end**
26 Return optimal policy π^* from $Q^*(s_k, a_k)$;

Consider the flexible manufacturing line shown in Fig. 2, where six machines carry out operations to produce two types of cars with Type 1 as a customized car and Type 2 as a non-customized car. The production line is divided into three segments, denoted X, Y and Z, which correspond to the $Input, Customized, Qualified$ block. Suppose that vehicles are introduced into block X. In B_1, a car can be selected for customization to block Y or follow directly to block Z; Once the car does not pass the quality assessment performed by the machine M_Q, a rework or a discarded car can be performed statistically. Based on different sojourn times for customized events such as f_{Ri} and rework events such as f^1_{Qr}, selecting the car to block Y or block Z is a decision problem depending on the expected profit.

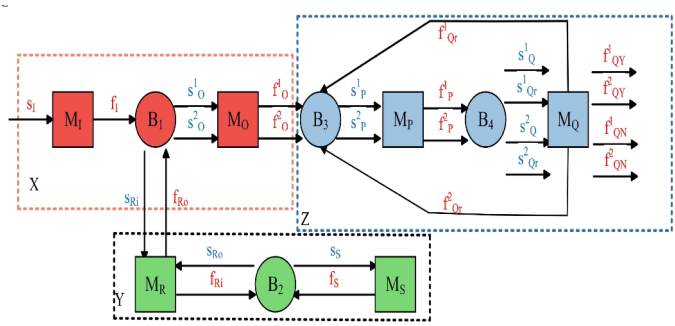


Fig. 2. Flow chart of a flexible automotive-assembly manufacturing system.

The manufacturing system is defined as a set of sub-automata which is shown in Fig. 3 with the software "Supremica", where MI and MS represent the machines featuring two states that transition through start and finish actions. MO, MP, and MR are designed as machines that operate on two different types of vehicles. MQ expresses both types can either leave or be reworked. The composition $G = MI \parallel MR \parallel Ms \parallel Mo \parallel Mp \parallel MQ$ creates 540 unrestricted states of the system.

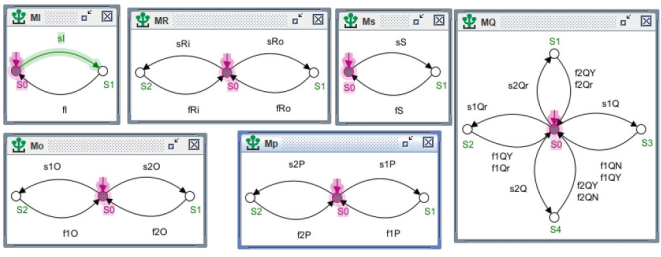


Fig. 3. Module FSM model of the automotive-assembly system.

And the deterministic specifications $B_{i=1,\ldots,4}$ are directly associated with restricted overflow and underflow. And deterministic specifications $C_{j=O,P,Q}$ are directly associated with restricted car types. The composition with all specifications is $E = B1 \parallel B2 \parallel B3 \parallel B4 \parallel CO \parallel CP \parallel CQ$ with 81 states in Table 1. By "pruned" composition on the above parallel automata, the synthesized system $\varsigma = (Q_\varsigma, \Sigma_\varsigma, \delta_\varsigma, q_{\varsigma 0}, Q_{\varsigma m})$ is shown in the last line of Table 1 with 1309 states and 4878 transitions. This supervisor eliminates all undesirable states and their associated paths, resulting in a supremal automata ς as permissive as possible while still satisfying the non-blocking requirement.

Table 1. States changed during the supervisory control process.

Operation	$\|Q\|$	$\|\rightarrow\|$	Type
$B1 \parallel B2 \parallel B3 \parallel B4 \parallel CO \parallel CP \parallel CQ$	81	567	Plant
$MI \parallel MR \parallel Ms \parallel Mo \parallel Mp \parallel MQ$	540	4536	Plant
$B1 \parallel B2 \parallel B3 \parallel B4 \parallel CO \parallel CP \parallel CQ \parallel$			
$MI \parallel MR \parallel Ms \parallel Mo \parallel Mp \parallel MQ$	18630	75438	Plant
$B1 \parallel B2 \parallel B3 \parallel B4 \parallel CO \parallel CP \parallel CQ$			
$\parallel MI \parallel MR \parallel Ms \parallel Mo \parallel Mp \parallel MQ$	1309	4878	Supervisor

Next, transfer the supervised system ς to SMDP environment by designing a STA automata $\mathcal{A} = (Q_\mathcal{A}, \Sigma_\mathcal{A}, \delta_\mathcal{A}, q_{\mathcal{A}0}, Q_{\mathcal{A}m}, \mathcal{T})$ firstly with states, events, and transition functions from the above automata ς. Secondly, add the sojourn time $\mathcal{T}$ for every transition. Controllable events are assumed to be untimed as they are commands with instantaneous time. Sojourn time is considered for uncontrollable events such as the MQ event, which may take lone time to work while the machine is in fault.

Table 2. Probability distribution function for timed events.

Action:	**Distribution class**	**CDF**	**Time**
f^1_{QY} ,f^1_{QN}	Rayleigh	$\int_0^{\tau_k} \frac{t}{\varrho^2} e^{-\frac{t^2}{2\varrho^2}} dt$	10 to 90
f^1_{Qr}	Rayleigh	$\int_0^{\tau_k} \frac{t}{\varrho^2} e^{-\frac{t^2}{2\varrho^2}} dt$	10 to 90
f^2_{Qr}, f^2_{QY}, f^2_{QN}	Rayleigh	$\int_0^{\tau_k} \frac{t}{\varrho^2} e^{-\frac{t^2}{2\varrho^2}} dt$	50
$f^{1,2}_P$, f_I, $f^{1,2}_O$	Exponential	$\int_0^{\tau_k} \lambda e^{-\lambda t} dt$	10
f^1_{Ri}	Rayleigh	$\int_0^{\tau_k} \frac{t}{\varrho^2} e^{-\frac{t^2}{2\varrho^2}} dt$	10 to 90
f_{Ro}, f_S	Exponential	$\int_0^{\tau_k} \lambda e^{-\lambda t} dt$	10

Specifically, conducte a variety of value tests from 10 to 90 time units on timed events of Type 1 products because of the uncertainty caused by customization.

By setting the probability density function (PDF) associated with each timed event in the Table 2, the probability function (CDF) of a state change to the next state occurring in less than time τ_k is calculated from its PDF. The car without customized has a stable sojourn time distribution, while the car type 1 has the expected value ranging from 10 to 90 time units since more equipment implies more uncertain values of sojourn time.

Therefore, the supervised automata ς is translated into the STA automata $\mathcal{A} = (Q_{\mathcal{A}}, \Sigma_{\mathcal{A}}, \delta_{\mathcal{A}}, q_{\mathcal{A}0}, Q_{\mathcal{A}m}, \mathcal{T})$. Then develop a tool that converts the STA automata $\mathcal{A}$ into a XML program, which serves as the interface of interaction. A parsing tool is then created to extract states, actions, and transition functions from the XML document into a SMDP tuple. Obtaining the sojourn time and PDF associated with the time-based events of STA, the probability Z of a state change to the next state occurring in less than time τ_k is calculated according to Eq. (6). Meanwhle, the reward value for every step is calculated according to Eq. (4) and the immediate and holding reward rate for each transition in the Table 3. Finally, a SMDP tuple $SM = (S, A, T, R, \vartheta, Z)$ is setup as the environment.

Table 3. Rewards and probabilistic parameters to be considered.

Action:	$\kappa(s_k, a_k)$	$\mu(s_k, a_k)$	$\mathcal{T}_k$	**Prob**(%)	**Type**
f^1_{QN}	-30000	-100	$\mathcal{T}_{f^1_{QN}}$	10 to 90	**Timed**
f^2_{QN}	-20000	-100	$\mathcal{T}_{f^2_{QN}}$	10	**Timed**
f^1_{QY}	30000	-100	$\mathcal{T}_{f^1_{QY}}$	10 to 90	**Timed**
f^2_{QY}	20000	-100	$\mathcal{T}_{f^2_{QY}}$	90	**Timed**
f^1_{Qr}	-1000	-100	$\mathcal{T}_{f^1_{Qr}}$	10 to 90	**Timed**
f^2_{Qr}	-1000	-100	$\mathcal{T}_{f^2_{Qr}}$	10	**Timed**
f_I, $f^{1,2}_O$	-10	-100	$\mathcal{T}_{f_I}, \mathcal{T}_{f^{1,2}_O}$	-1	**Timed**
$f^{1,2}_P$, f_{Ri}	-10	-100	$\mathcal{T}_{f^{1,2}_P}, \mathcal{T}_{f_{Ri}}$	-1	**Timed**
f_{Ro}, f_S	-10	-100	$\mathcal{T}_{f_{Ro}}, \mathcal{T}_{f_S}$	-1	**Timed**

According to the above Algorithm 2, perform experiments using the parameters in Table 2 with the discount factor $\vartheta = 0.1$. For training stage, the environment is trained over 50000 episodes, considering the stop critic with ending after producing 10 cars. For evaluation, experiments are carried out under two conditions: a purely supervisory method with a random action policy and maximizing utilization on the proposed method. The performance of the mean reward resulting from the SCT-Timed RL algorithm, compared to the purely supervisory method, is presented in Fig. 4. When the customization time of the Type 1 car is not too long, the average total reward generated by the proposed intelligent method is 10923000, while the non-intelligent SCT method is 1224830, which makes an enhancement of 8.9%.

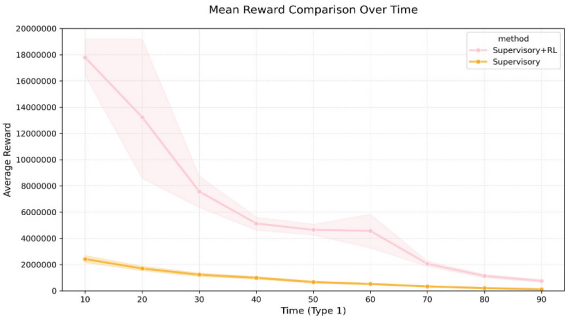


Fig. 4. Plot of average reward for SCT and proposed algorithm.

This is because, as the profits of customized cars are higher than that of non-customized cars, in the case of low customization time consumption, the system chooses to let the newly entered car move to the customized module, so that it can obtain higher profits. Meanwhile, for both methods, as the customized time increases, the total value decreases monotonically.

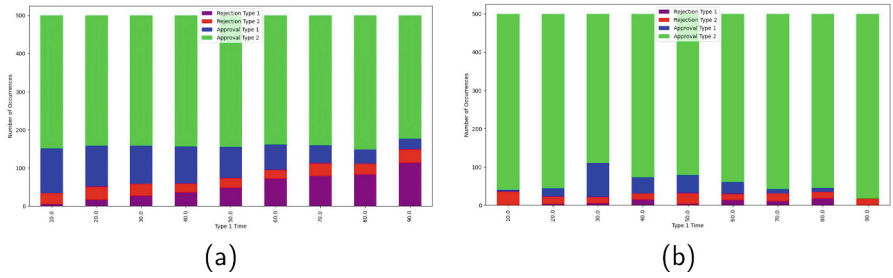


Fig. 5. (a) Plot of rejection/approval number with SCT. (b) Plot of rejection/approval number with proposed algorithm.

Next, carry out the experiments on the rejected - approved occurrences for each type of car. Figure 5(a) shows the number of rejected - approved occurrences of each type of car purely by the supervisory method. When under a longer sojourn time in the rework event f_{Qr}^1 with Type 1, the SCT decisions still allow the newly entered car to move to the customized module, and inevitably, there will be more reworks for Type 1 cars. In the proposed algorithm, the sojourn time in the rework event f_{Qr}^1 with Type 1 is incorporated into the Bellman equation and the reward function. When the rework time exceeds a certain threshold, the intelligent system will reduce the production quota for the Type 1 car, thus guaranteeing a higher profit. Figure 5 (b)shows the number of rejected - approved occurrences of each type of car purely by the SCT-Timed RL method. When under a longer sojourn time in the rework event f_{Qr}^1 with Type 1, the intelligent

method chose not to produce Type 1 car, due to the longer time consumption for Type 1 car.

Table 4. Comparison for different methods on higher rework time

Method	Type1 Ap.	Type2 Ap.	Type1 Num.	Type2 Num.	Blocked	Learn
SCT [4]	60	188	128	372	No	No
OURs	**10**	**19**	**22**	**478**	**No**	**Yes**

Table 4 shows the comparison for the different methods with the longest rework time (in 70–90 time units). Under the supervision of SCT, both methods can make the system run without blocking and continue to carry out the machining process. For higher rework time, the proposed method is more probable to adjust the ratio of customized cars to non-customized cars flexibly according to the real-time situation, thus achieving higher profits per unit time.

4 Conclusion

This research presents a novel optimal supervised control approach for STDES which integrates SCT and RL. Firstly, the language of STDES and the deterministic specifications are initially generated by automata G and E according to the logic features of STDES. Then a supervisor is synthesized using SCT to ensure system safety by prohibiting specific controllable event sequences. Subsequently, the controlled automata structure obtained via SCT is transformed into a SMDP framework by incorporating the temporal and probabilistic variable of the uncontrollable events. Lastly, a modified SCT-Timed RL algorithm is developed to optimize the control policy through the design of a new Bellman equation and reward function. The results demonstrate that the proposed controller significantly enhances system safety and flexibility, achieving a 8.9% performance boost compared to SCT method. This study advances optimal control research in diverse human-made industrial physical systems. Other DRL algorithms will be considered in the next work to realize the control of more complex systems.

Funding. This study was supported by the Guangdong Basic and Applied Basic Research Foundation (2023A1515012783) of China.

References

1. Lafortune, S., Cassandras, C.: Introduction to Discrete Event Systems, 2nd edn. Springer, New York (2008). https://doi.org/10.1007/978-0-387-68612-7
2. Billaut, J.-C., et al.: A survey on industrial applications of flexible job shop scheduling problems. Eur. J. Oper. Res. **205**(2), 241–251 (2010)

3. Wonham, W.M., Ramadge, P.J.: Supervisory control of timed discrete-event systems. IEEE Trans. Autom. Control **39**(2), 329–342 (1994)
4. Wonham, W.M.: Supervisory control of discrete-event systems: a brief history. Annu. Rev. Control. **45**, 250–256 (2018)
5. Zouari, F., et al.: Finite-time adaptive event-triggered output feedback intelligent control for noninteger order nonstrict feedback systems with asymmetric time-varying pseudo-state constraints and nonsmooth input nonlinearities. Commun. Nonlinear Sci. Numer. Simul. **136**, 108036 (2024)
6. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction, 2nd edn. MIT Press, Cambridge (2018)
7. Ying, Y., Lin, H.: Online self-learning fuzzy discrete event systems. IEEE Trans. Fuzzy Syst. **28**(9), 2185–2194 (2019)
8. Zhang, P., et al.: Control of black-box embedded systems by integrating automata learning and supervisory control theory of discrete-event systems. IEEE Trans. Autom. Sci. Eng. **17**(1), 361–374 (2019)
9. Yamasaki, T., Takai, S.: Decentralized supervisory control of discrete event systems based on reinforcement learning. IFAC Proc. Vol. **37**(11), 367–372 (2004)
10. Lefebvre, D., Kratz, F.: Stochastic timed discrete-event systems: modular modeling and performance evaluation through Markovian jumps. IEEE Access. **10**, 108332–108341 (2022)
11. Lefebvre, D.: Exposure and revelation times as a measure of opacity in timed stochastic discrete event systems. IEEE Trans. Autom. Control **66**(12), 5802–5815 (2020)
12. Wang, J., et al.: Event-triggered near-optimal control for unknown discrete-time nonlinear systems using parallel control. IEEE Trans. Cybernet. **53**(3), 1890–1904 (2023)
13. Neto, A.A.: Planning, learning and control under uncertainty based on discrete event systems and reinforcement learning. Ph.D. dissertation, Instituto Superior Técnico, Lisboa, Portugal (2010)
14. Zielinski, C., et al.: Flexible control of discrete event systems using environment simulation and reinforcement learning. Appl. Soft Comput. **111**, 107714 (2021)

A Classification Algorithm for Bronchiolitis Obliterans in Pediatric CT Images with Extreme Class Imbalance

Lu Liu[1,2], Xibin Feng[4], Ye Yuan[1,2], Wei Xu[4], and Wenjun Tan[1,2,3(✉)]

[1] School of Computer Science and Engineering, Northeastern University, Shenyang 110000, China
tanwenjun@cse.neu.edu.cn
[2] Key Laboratory of Intelligent Computing of Medical Images, Ministry of Education, Northeastern University, Shenyang 110000, China
[3] National Frontiers Science Center for Industrial Intelligence and Systems Optimization, Northeastern University, Shenyang 110000, China
[4] Shengjing Hospital Affiliated to China Medical University, Shenyang 110000, China

Abstract. Bronchiolitis obliterans (BO) is characterized by airflow obstruction resulting from inflammatory damage to the small airways in children. Current diagnostic methods heavily depend on clinical expertise, lacking efficient imaging-assisted tools. This paper introduces a novel deep learning framework based on ensemble learning to assist in the diagnosis of BO. Recognizing spatial feature similarities between BO and non-BO samples, an optimized network architecture has been devised to reveal high-order features. To addresss class imbalance, data redistribution strategies and grouped training methods have been proposed to ensure a balanced sample distribution. By integrating multiple independent models into an ensemble classifier, the classification accuracy has been significantly improved. Experimental results show that the single network model achieved an accuracy of 72.39% for BO and 82.81% for non-BO, which increase to 95.48% and 91.21% after model integration. These results underscore the efficacy of the proposed deep learning framework in pediatric BO classification, facilitating early diagnosis.

Keywords: Bronchiolitis Obliterans · Computer-aided Diagnosis · Data Redistribution · Ensemble Learning

1 Introduction

Bronchiolitis Obliterans (BO) is a chronic airflow limitation syndrome resulting from bronchiolar damage, posing a significant risk to pediatric patients [1]. Children are at increased risk due to their relatively immature immune systems, incomplete development of respiratory structures, and limited capacity to self-regulation of disease, making BO a significant threat to their health and quality of life. Most children with BO experience persistent or recurrent symptoms such as coughing, wheezing, dyspnea, and reduced exercise tolerance following airway injury, with severe cases potentially leading to respiratory failure [2]. If BO is

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 415–427, 2026.
https://doi.org/10.1007/978-3-032-04555-3_34

not diagnosed and treated promptly and accurately, it can result in irreversible airway fibrosis and obstruction [3]. Therefore, early diagnosis and effective therapeutic interventions for BO are crucial.

However, early diagnosis of BO presents significant challenges. Its initial symptoms are nonspecific, and radiographic findings in children may appear normal. Although high-resolution computed tomography (HRCT) is the main non-invasive diagnostic tool [4], it has high radiation sensitivity and limited use in young children. Although lung biopsy is considered the gold standard for diagnosing BO [5], the immature lung tissue in children and the heterogeneous distribution of lesions make it difficult to perform in pediatric patients. Therefore, physicians must have in-depth knowledge of the characteristics of BO, which requires a comprehensive evaluation integrating clinical manifestations, medical history, pulmonary function tests, and imaging for a timely diagnosis and treatment.

Deep learning technology automates feature extraction in medical image analysis to improve disease diagnosis. In breast cancer screening, it allows differentiation of malignant and benign tumors from mammography images [6]. In dermatopathology, it supports the automatic classification of skin lesions [7]. For ophthalmic diagnostics, it can detect Diabetic Retinopathy (DR) [8] and Age-related Macular Degeneration (AMD) [9] from retinal scans. In lung disease screening, it analyzes chest X-rays and CT images to identify and classify various lung conditions [10]. These applications optimize medical resource allocation and enable personalized treatment. Therefore, deep learning-based feature extraction from BO and non-BO CT images assists clinicians in early BO diagnosis, enhancing prognosis and long-term quality of life for pediatric patients.

Based on the remarkable achievements of convolutional neural network (CNN) models in image classification tasks, this study proposes the first direct application of innovative deep learning methods for BO disease diagnosis using CT images. Inspired by the Inception series network architecture [11–13], **1) this study designs an optimized network structure for BO disease CT image classification.** By integrating multiscale processing, parallel structures, and dense connections, the proposed model improves classification accuracy, offering clinicians more reliable diagnostic support. Addressing the serious class imbalance issue between BO and non-BO data, **2) this study employs data redistribution as a strategy to mitigate the adverse effects on the training results.** A class-ratio grouping strategy is adopted to balance subset distributions, alleviating the negative effects of imbalance on model training. By integrating multiple group-trained models and designing a classifier, **3) this study uses ensemble learning to enhance classification reliability and stability.** This approach not only fully utilizes data information but also enhances result reliability and stability through model complementarity. Finally, **4) this study the network's performance in BO disease classification using multiple metrics.** The results demonstrate that the proposed method excels in pediatric BO classification, offering a novel diagnostic aid for doctors and contributing significantly to improving treatment outcomes and the quality of life for affected children.

2 Method

This section describes the development of BO image classification methods. First, extracting discriminative spatial features between BO and non-BO data, along with handling severe class imbalance, were key challenges. Data redistribution was applied to reduce the imbalance's influence on training. Then, an optimized network was developed to improve BO classification accuracy, enabling more precise medical diagnosis. Finally, a multi - model ensemble strategy was implemented to further enhance classification performance, offering more reliable support for clinical diagnosis.

2.1 Data Redistribution

Class imbalance in machine learning significantly impacts predictive performance, particularly impairing minority class recognition [14]. This study tackles the severe class imbalance in BO classification: the sample size of BO children is much smaller than non-BO children, and their CT images show highly similar spatial features (Fig. 1), making it hard for classifiers to extract effective discriminative features. This exacerbates the negative impact of class imbalance, causing models to overfit the majority non-BO class and overlook critical BO pathological features.

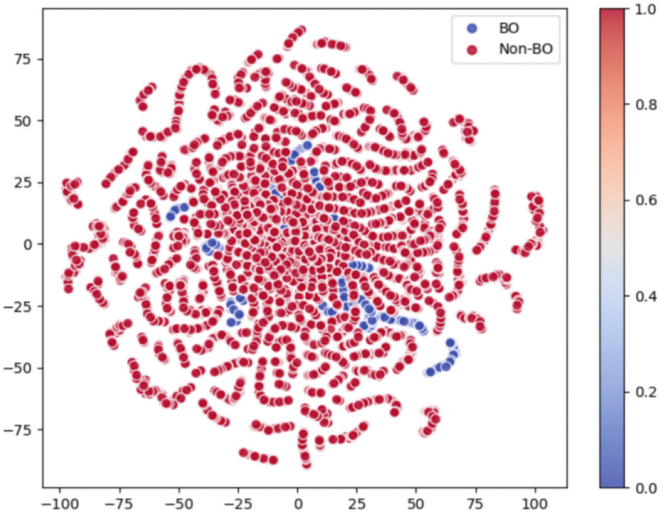


Fig. 1. T-SNE visualization of BO and non-BO medical image data.

To address this, a sample balancing strategy was employed to achieve a 1:1 ratio of BO and non-BO children in the training and validation sets. Given the small proportion of BO samples, a strategy of fixing BO samples and randomly sampling non-BO samples was adopted to expand negative data diversity and prevent sub-model overfitting to non-BO data. This approach aims to ensure

balanced cross-class learning, reduce prediction bias, and improve model generalization and accuracy, as described by formula (1).

$$Dataset\,(i) = \{BO, Non - BO\,(i)\} \tag{1}$$

2.2 BO Classification Network Architecture

With the development of deep learning, deep neural networks have become essential for image classification, where network architecture design and optimization significantly affect model performance. Inspired by the Inception series' design principles, this study proposes an improved network structure to enhance model performance and generalization while reducing overfitting risks, as shown in Fig. 2. The network design involves several key components:

1) Modular design and multi-scale feature extraction: Inspired by the modular design of the Inception architecture, the proposed network parallelizes convolution kernels and pooling operations of diverse sizes to extract hierarchical local image features, thereby improving its multi - scale information perception. Meanwhile, convolution kernel decomposition splits large kernels into smaller ones, reducing computational load and parameter count while maintaining model expressiveness.
2) Auxiliary classifiers and gradient back propagation: To accelerate the training process and alleviate gradient vanishing issues, the network introduces auxiliary classifiers in intermediate layers to provide early gradient feedback, thereby facilitating faster convergence during training. This approach allows the network to obtain effective forward signals earlier, expediting the model training process.
3) Global average pooling and parameter compression: To improve generalization ability and reduce model parameters, this study incorporates a global average pooling layer. By compressing spatial information into a single global statistic, this layer significantly reduces the number of parameters requiring training while preserving crucial feature information, thus enhancing model generalization performance.
4) Dense connections and ELU activation function: Towards the end of the network, three dense connection layers with 256, 128, and 64 neurons are included. Through the stacking of multiple dense connections, the network gradually extracts and combines more abstract feature representations, thereby enhancing classification performance. The dense layers employ the Exponential Linear Unit (ELU) activation function [15]. Compared to the traditional ReLU activation function, ELU has a non-zero gradient in the negative value range, alleviating gradient vanishing issues and speeding up network convergence.
5) Output layer and classification task: Finally, the model's output layer consists of two neurons to accommodate binary classification tasks. The output layer utilizes a softmax activation function to classify images into "BO" and "non-BO" classes, thereby achieving the binary classification task.

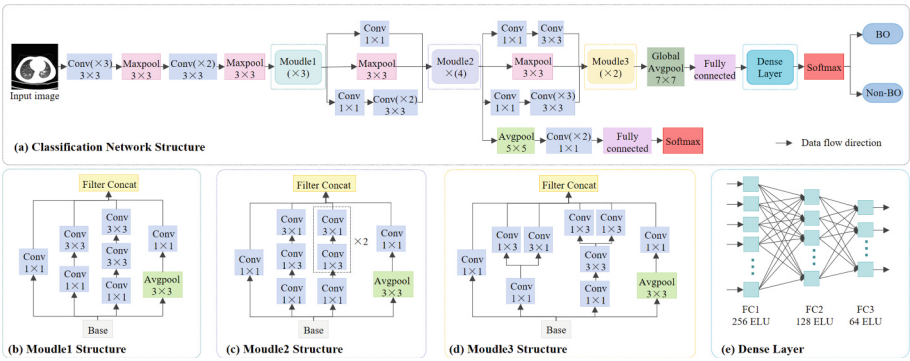


Fig. 2. Basic classification network structure and module structure diagram. (a) Classification network structure; (b) Moudle1 structure; (c) Moudle2 structure; (d) Moudle3 structure; (e) Dense layer structure.

2.3 Model Integration and Classification Strategy

Deep learning facilitates doctors in accurately analyzing lesion features from medical images. However, BO disease classification faces challenges like high similarity between BO and non-BO cases, as well as noise interference in pediatric CT images, which prevent single models from achieving optimal performance.

To address this, model ensemble, a technique combining multiple models to enhance classification accuracy and generalization, was adopted to improve the precision and stability of medical image classification tasks [16]. This approach integrates high-performing models trained on different datasets using the same deep learning architecture, capturing feature information under diverse data distributions by leveraging dataset characteristics. The schematic of the model ensemble and classifier module is illustrated in Fig. 3.

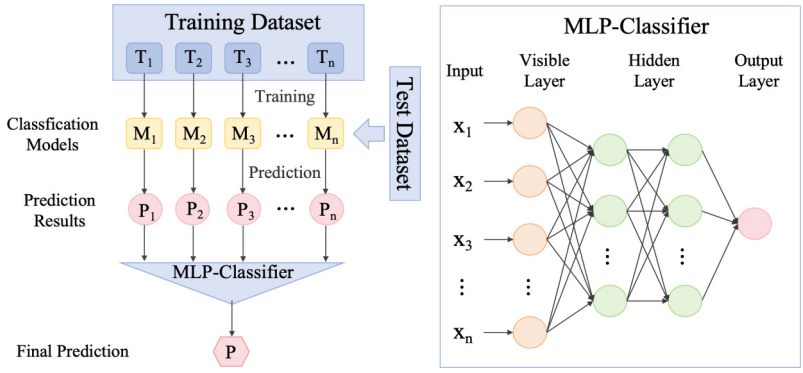


Fig. 3. Schematic diagram of model integration and classifier.

During the experiment, independently trained models generate classification predictions for the test set. To enhance classification stability and accuracy, a

Multi-Layer Perceptron (MLP) classifier module is used to integrate these predictions. The MLP combines individual model outputs via equal-weight voting to derive the final classification decision. This ensemble strategy alleviates overfitting or bias in single models, improves generalization across diverse samples, and yields more reliable and accurate classification results.

3 Experiments

3.1 Dataset

This study's dataset consists of 614 pediatric pneumonia cases with 512×512 - pixel chest CT images obtained at admission. After a one - year follow - up through parental communication, further clinical evaluations were conducted on pediatric patients with respiratory symptoms to confirm BO. Among these cases, 35 pediatric patients were ultimately diagnosed with BO, while the remaining 579 did not exhibit BO-related symptoms. To construct our model, the dataset was divided into training, validation, and testing sets at a ratio of 7:2:1. Rigorous data cleaning excluded invalid samples lacking lung information (Fig. 4), enhancing data quality and experiment accuracy.

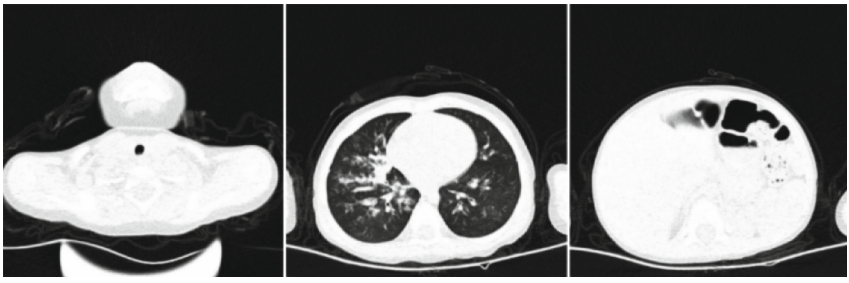


Fig. 4. CT images of lung. Images of lung apex, middle, bottom are displayed from left to right.

3.2 Implementation Details

All models were developed using the Keras framework, with training conducted 1000 epochs, a batch size of 32, and optimized by the Adaptive Moment Estimation (Adam) optimizer. The initial learning rate was set to 1.00E-03, with the ReduceLROnPlateau strategy adjusting the rate, to enhance training stability and convergence efficiency. CT image data preprocessing involved normalizing the training data to [0, 1] and adjusting input images to 299×299×3. Data augmentation techniques such as random flipping, rotation, and shifting were employed to enhance the model's generalization ability and prevent over-fitting.

The experimental setup included the following software and hardware environments: 1) CPU processor: Intel(R) Core(TM) i7-10700F CPU @ 2.90GHz; 2) GPU processor: NVIDIA GeForce RTX 2070 SUPER; 3) operating systems: Windows 10 64-bit edition, Ubuntu 20.04.2 LTS; 4) programming language and development environment: Python 3.9; PyCharm 2021.1 64-bit version.

3.3 Evaluation Metrics

To comprehensively assess the classification performance of the model, this study employed a multidimensional evaluation metric system. Given the significant class imbalance between BO and non-BO classes, this research calculated the classification accuracy for each class separately. Furthermore, to mitigate performance bias stemming from class imbalance, a weighted average method was introduced to determine the precision, recall, and f1 score of the model. As shown in formulas (2)–(4), where C indicates the total number of classes, and N represents the total number of samples. These evaluation metrics can holistically reflect the model's classification effectiveness from multiple dimensions, particularly in dealing with imbalanced datasets or when balancing precision and recall, providing a more detailed and comprehensive performance assessment.

$$Precision_{weighted} = \frac{1}{N}\sum_{i=1}^{C} N_i \times \frac{TP_i}{TP_i + FP_i} \tag{2}$$

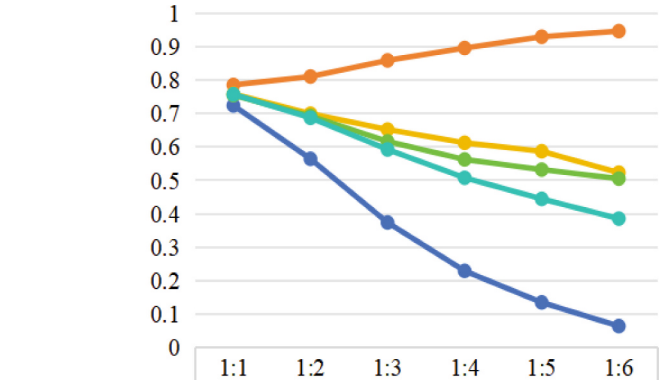
$$Recall_{weighted} = \frac{1}{N}\sum_{i=1}^{C} N_i \times \frac{TP_i}{TP_i + FN_i} \tag{3}$$

$$F1\ score_{weighted} = \frac{1}{N}\sum_{i=1}^{C} N_i \times 2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i} \tag{4}$$

3.4 Results

1) Data Redistribution Ablation Experimental Results: This study conducted a data resampling ablation experiment to investigate the impact of class imbalance on model performance. The ablation experiment employed a progressive approach to adjust the ratio of BO to non-BO samples. The training set included CT imaging data from 24 BO pediatric patients (1958 images), with the number of non-BO patient samples adjusted accordingly. The test set comprised CT data from 3 BO patients (221 images) and 3 non-BO patients (221 images). All experiments maintained a unified model architecture for consistency and comparability of results.

The experimental results are presented in the form of Fig. 5, detailing the BO acc, non-BO acc, precision, recall, f1 score. The results indicate that when the ratio of BO to non-BO samples is close to 1:1, the predictive performance of the model for both BO and non-BO children remains at a high level. As the ratio of BO to non-BO gradually increases, the accuracy of non-BO samples increases gradually until it approaches 1, while the accuracy of BO samples decreases gradually until it approaches 0. This experiment demonstrates that a sample balancing strategy can significantly enhance the model's generalization ability and accuracy on imbalanced class datasets, helping to avoid the model's over-fitting to the majority class during the training process.



	1:1	1:2	1:3	1:4	1:5	1:6
BO Acc	0.7239	0.5641	0.3737	0.229	0.1347	0.064
Non-BO Acc	0.7852	0.8104	0.8586	0.8956	0.9293	0.9461
Precision	0.7583	0.6994	0.6519	0.6121	0.5868	0.5228
Recall	0.7546	0.6921	0.6162	0.5623	0.532	0.5051
F1-Score	0.7573	0.6869	0.5922	0.5076	0.4443	0.3855

Fig. 5. Experimental results of data redistribution ablation.

2) Classification Network Comparison Results: This study compared the performance of nine popular deep learning models (ResNet50 [17], InceptionV3 [13], DenseNet121 [18], EfficientNetB0 [19], ViT [20], Swin Transformer [21], ConvNext [22], EfficientViT [23], MobileNetV4 [24]) on a dataset with adjusted data dimensions to match each model's input requirements. To assess the models' handling of imbalanced data classes, the study selected 3 samples of one class (221 images) and 51 samples of another class (4028 images) based on real diagnostic proportions. Given the higher proportion of non-BO samples in the dataset, we independently evaluated the classification accuracy for BO and non-BO classes. Key performance metrics such as precision, recall, and f1 score were calculated using weighted averages based on actual sample quantities to address class imbalances. The results are shown in Table 1.

Table 1. Comparative experimental results of classification network.

Network	BO Acc ↑	Non-BO Acc ↑	Precision ↑	Recall ↑	F1 Score ↑
ResNet50 (2016)	0.6599	0.7429	0.9312	0.7386	0.8104
InceptionV3 (2016)	0.6700	0.8016	0.9352	0.7948	0.8484
DenseNet121 (2017)	0.6364	0.8147	0.9336	0.8054	0.8551
EfficientNetB0 (2019)	0.7071	0.7566	0.9354	0.7540	0.8212
ViT (2020)	0.7168	0.7282	0.9348	0.7276	0.8030
Swin Transformer (2021)	0.6529	0.7675	0.9320	0.7615	0.8260
ConvNext (2022)	**0.7561**	0.6864	0.9359	0.6900	0.7762
EfficientViT(2023)	0.6910	0.7145	0.9321	0.7133	0.7928
MobileNetV4 (2024)	0.6356	0.7977	0.9325	0.7893	0.8445
Ours	0.7239	**0.8281**	**0.9407**	**0.8227**	**0.8673**

Performance comparisons show that the proposed network architecture outperforms existing models across key metrics. Specifically, it achieves 72.39% accuracy for BO cases and 82.81% for non-BO cases, with excellent performance in precision, recall, and F1-score, highlighting its effectiveness in balancing predictive performance for imbalanced classes. While the ConvNext model achieves 75.61% accuracy for BO cases in some indicators, the proposed architecture demonstrates significant overall advantages, especially in handling imbalanced data tasks, showcasing high robustness and effectiveness.

3) Model Ensemble Results: The study utilized a grouping training strategy to address class imbalance issues, integrating individual models to further enhance classification performance. Empirical results reveal a marked enhancement in model effectiveness across various key performance metrics through ensemble learning strategies. Despite a slightly lower accuracy rate in class BO (95.48%) compared to single model-1 (96.30%), the integrated model showed significant improvement in all other evaluation metric. This underscores the efficacy of ensemble strategies in improving overall model performance, particularly in enhancing comprehensive performance metrics. The results are shown in Table 2.

Further analysis reveals that the precision, recall, and f1 score all reached their peak values in the integrated model, affirming the efficacy of ensemble learning in addressing class-imbalanced datasets. Particularly in scenarios requiring comprehensive performance enhancement for more thorough evaluation and higher classification accuracy, ensemble learning strategies hold significant application prospects. The study results emphasize the role of ensemble methods in enhancing the stability and generalization capabilities of deep learning models for complex classification tasks, providing robust theoretical and practical foundations to further drive the resolution of data imbalance issues and advance research in related fields.

Table 2. Model ensemble experimental results.

Model	BO Acc ↑	Non-BO Acc ↑	Precision ↑	Recall ↑	F1 Score ↑
Model-1	**0.9630**	0.7566	0.9547	0.7673	0.8313
Model-2	0.7138	0.8795	0.9441	0.8709	0.8988
Model-3	0.7239	0.8281	0.9407	0.8227	0.8673
Model-4	0.7475	0.7944	0.9404	0.7920	0.8471
Model-5	0.8418	0.7560	0.9455	0.7605	0.8262
Model-Ensemble	0.9548	**0.9121**	**0.9648**	**0.9143**	**0.9312**

Based on the confusion matrix provided (Fig. 6), out of the 221 samples in the BO class, the model successfully identified 211 samples (95.48%), with only 10 BO samples (4.52%) misclassified as non-BO. For the 4028 samples in the non-BO class, the model accurately identified 3674 samples (91.21%), with 354 non-BO samples (8.79%) misclassified as BO. Despite the 354 misclassified non-BO samples exceeding the total number of samples in the BO class (221 images), this phenomenon does not indicate a severe misclassification by the model. In

fact, the 354 misclassified non-BO samples are from a relatively large non-BO class (4028 images), thus their misclassification rate should be evaluated as the proportion of misclassification for the non-BO class, rather than directly comparing sample numbers between BO and non-BO. Overall, this result demonstrates the model's outstanding performance in complex classification tasks. A detailed analysis of the confusion matrix reveals the model's high accuracy and robustness in distinguishing between BO and non-BO classes, particularly in handling imbalanced data, effectively mitigating the impact of misclassification and showcasing its strong discriminative capabilities in different class recognition tasks.

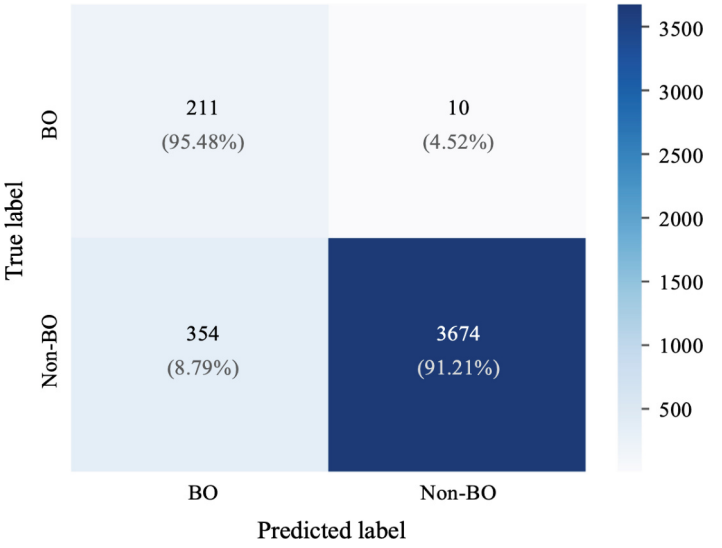


Fig. 6. Confusion matrix of classification results.

4) Diagnostic Patients Results: In this study, an in-depth analysis was conducted on diagnostic data of 3 BO children and 51 non-BO children from the test set. The box plot presented (Fig. 7) depicts disease classes on the horizontal axis and quantifies the proportion of corresponding patient data correctly classified on the vertical axis, a crucial indicator for evaluating the likelihood of BO disease in patients. It was observed that the classification confidence of all children exceeded the threshold of 50%, indicating the overall success of our classification model in distinguishing between BO and non-BO children. However, it is noteworthy that the classification accuracy of some children was only slightly above 50%, a phenomenon that may be closely related to the severity of the children's specific conditions. Therefore, this study recommends that healthcare professionals, when dealing with children with low classification accuracy, should not only rely on the current classification model but also take into account the clinical manifestations of the children and supplement with other diagnostic tools, to achieve a comprehensive assessment of the children's condition. This

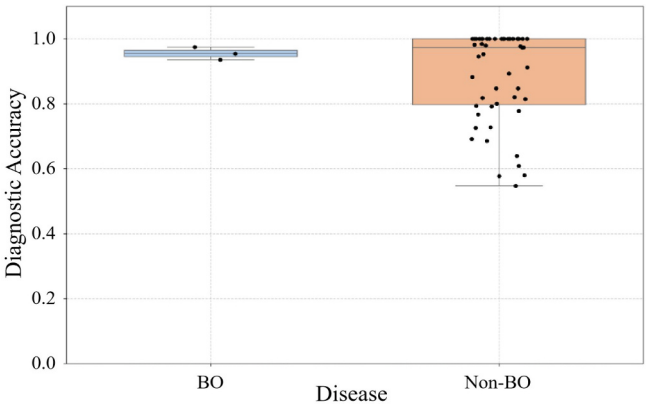


Fig. 7. Diagnostic accuracy of children with BO and non-BO.

multidimensional diagnostic approach not only enhances diagnostic accuracy but also provides more personalized and precise treatment plans for the children.

4 Conclusion

This study demonstrates a novel deep-learning-based CT image analysis technique for direct BO disease diagnosis. This approach not only offers a new method for early BO detection but also enables more accurate and timely clinical interventions for pediatric patients. Through comprehensive evaluation of experimental results, including the management of highly imbalanced BO disease classes, the proposed childhood BO classification framework achieved notable accuracies of 95.48% for BO cases and 91.21% for non-BO cases. These findings highlight the significant potential of this method in medical image analysis. Beyond clinical applications, the research provides insights into early BO diagnosis and treatment, as well as a paradigm for applying deep learning to rare diseases, contributing valuable academic and practical knowledge to the field.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (62471122) and Fundamental Research Funds for the Central Universities(N25GFZ017, N25BJD005).

References

1. Kavaliunaite, E., Aurora, P.: Diagnosing and managing bronchiolitis obliterans in children. Expert Rev. Respir. Med. **13**(5), 481–488 (2019)
2. Yu, J.: Postinfectious bronchiolitis obliterans in children: lessons from bronchiolitis obliterans after lung transplantation and hematopoietic stem cell transplantation. Korean J. Pediatr. **58**(12), 459 (2015)

3. Rosewich, M., Zissler, U.M., Kheiri, T., et al.: Airway inflammation in children and adolescents with bronchiolitis obliterans. Cytokine **73**(1), 156–162 (2015)
4. Chan, K.C., Yu, M.W., Cheung, T.W.Y., et al.: Childhood bronchiolitis obliterans in Hong Kong-case series over a 20-year period. Pediatr. Pulmonol. **56**(1), 153–161 (2021)
5. Jerkic, S.P., Brinkmann, F., Calder, A., et al.: Postinfectious bronchiolitis obliterans in children: diagnostic workup and therapeutic options: a workshop report. Can. Respir. J. **2020**(1), 5852827 (2020)
6. Wang, L.: Mammography with deep learning for breast cancer detection. Front. Oncol. **14**, 1281922 (2024)
7. Salma, W., Eltrass, A.S.: Automated deep learning approach for classification of malignant melanoma and benign skin lesions. Multimedia Tools Appl. **81**(22), 32643–32660 (2022)
8. Sangeetha, K., Valarmathi, K., Kalaichelvi, T., et al.: A broad study of machine learning and deep learning techniques for diabetic retinopathy based on feature extraction, detection and classification. Measur. Sens. **30**, 100951 (2023)
9. Gong, D., Kras, A., Miller, J.B.: Application of deep learning for diagnosing, classifying, and treating age-related macular degeneration. In: Seminars in Ophthalmology. Taylor & Francis vol. 36. no. 4, pp. 198–204 (2021)
10. Ibrahim, D.M., Elshennawy, N.M., Sarhan, A.M.: Deep-chest: multi-classification deep learning model for diagnosing COVID-19, pneumonia, and lung cancer chest diseases. Comput. Biol. Med. **132**, 104348 (2021)
11. Szegedy, C., Liu, W., Jia, Y., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
12. Szegedy, C., Vanhoucke, V., Ioffe, S., et al.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
13. Szegedy, C., Ioffe, S., Vanhoucke, V., et al.: Inception-v4, inception-ResNet and the impact of residual connections on learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31, no. 1 (2017)
14. Johnson, J.M., Khoshgoftaar, T.M.: Survey on deep learning with class imbalance. J. Big Data **6**(1), 1–54 (2019)
15. Clevert, D.A.: Fast and accurate deep network learning by exponential linear units (elus). arxiv preprint arxiv:1511.07289 (2015)
16. Ganaie, M.A., Hu, M., Malik, A.K., et al.: Ensemble deep learning: a review. Eng. Appl. Artif. Intell. **115**, 105151 (2022)
17. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
18. Huang, G., Liu, Z., Van Der Maaten, L., et al.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)
19. Tan, M., Le, Q.: EfficientNet: rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning. PMLR, pp. 6105–6114 (2019)
20. Dosovitskiy, A.: An image is worth 16×16 words: Transformers for image recognition at scale. arxiv preprint arxiv:2010.11929 (2020)
21. Liu, Z., Lin, Y., Cao, Y., et al.: Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012–10022 (2021)

22. Liu, Z., Mao, H., Wu, C.Y., et al.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11976–11986 (2022)
23. Cai, H., Li, J., Hu, M., et al.: EfficientViT: lightweight multi-scale attention for high-resolution dense prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 17302–17313 (2023)
24. Qin, D., Leichner, C., Delakis, M., et al.: MobileNetV4: universal models for the mobile ecosystem. In: European Conference on Computer Vision. Springer, Cham, pp. 78–96 (2025)



DISEncoder: A Dual-Branch Query Encoder Using Graph Models for Distributed Databases

Jianwen Yang[1,2], Qiuhong Zhang[1,2], Jin Yan[1,2(✉)], Shuo Zhang[1(✉)], Zhiming Ding[1], Meiling Zhu[1], and Xinrun Xu[1,2]

[1] Institute of Software, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
{yangjianwen18,zhangqiuhong22,xuxinrun20}@mails.ucas.ac.cn,
{zhangshuo,zhiming,meiling}@iscas.ac.cn, yvette.yan@mails.ucas.edu.cn

Abstract. Machine learning has emerged as a prominent approach to addressing various optimization problems in distributed databases, such as distribution-key recommendation, performance prediction, and parameter auto-configuration. However, the complex architecture and query execution processes of distributed databases pose significant challenges for feature selection and encoding in machine learning. Recently, several methods have been proposed that utilize physical plans or SQL queries as features. However, these methods often rely on using physical plans or SQL queries independently, neglecting the inherent characteristics of distributed databases, which limits their ability to fully represent the query features of such systems.

To overcome these limitations, we introduce DISEncoder, a query feature representation model specifically designed to address various optimization challenges in distributed databases. In this approach, we utilize three types of information-execution plans, distributed database architecture, and execution process logs-to construct representations of distributed queries. This information is integrated into two graph structures, and we develop a dual-branch graph encoding model to encode and fuse these structures, resulting in a feature representation vector for queries. We incorporated DISEncoder into two machine learning models for database optimization tasks, and experimental results demonstrate that DISEncoder significantly improves the performance of these models.

Keywords: GAT · Distributed databases · AI4DB

Jianwen Yang and Qiuhong Zhang are co-first authors who contributed equally to this work. Jianwen Yang is listed first.

Supported by the National Key R&D Program of China under Grant 2022YFF0503900, and the Research Project of Institute of Software, Chinese Academy of Sciences (ISCAS-ZD-202401, ISCAS-ZD-202403).

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 428–440, 2026.
https://doi.org/10.1007/978-3-032-04555-3_35

1 Introduction

The application of machine learning in distributed databases is growing, encompassing distribution-key recommendation [1,2], performance prediction [3], cost estimation [4], and automatic parameter configuration [5]. These tasks share a common challenge: effective and efficient distributed data query representation, focusing on feature extraction, model building, and feature encoding. Query representation forms the foundation of learning tasks. In current database machine learning applications, physical execution plans are predominantly used as query features [6–9]. These plans detail the sequence of operations and their associated costs during query execution, offering crucial performance insights for constructing machine learning models. Conversely, in single-node databases, SQL queries and database schemas are regarded as significant features for query representation [10,11]. SQL statements encapsulate semantic information about data operations, while database schemas convey structural information about data and table relationships.

Despite progress in distributed database query representation, limitations remain. First, existing methods often overlook the multi-node setup. Deployment configuration significantly impacts data distribution and query execution (e.g., node count affects data distribution; node location affects network latency). Second, current methods inadequately consider data transfer during query execution (e.g., data redistribution among nodes affects query performance). Finally, effective query feature encoding remains a challenge. TREE-CNN models [12,13] struggle to capture long-path information in physical query plans, while Tree-RNN models [7,8] present training challenges.

This paper introduces DISEncoder (DISTRIBUTED DATABASE QUERY ENCODER), a dual-branch joint query encoder designed for distributed databases. DISEncoder extracts features from execution plans, database architecture, and query execution logs to construct two graph structures. These are jointly trained in a dual-branch framework, resulting in a unified query representation for various learning tasks. DISEncoder's effectiveness is assessed by integration with several machine learning models. Our contributions are: 1) An execution plan encoder for distributed databases capturing operation types and resource utilization, encoded as a graph with a graph attention mechanism. 2) A method integrating distributed database architecture and query execution logs to construct a graph structure, capturing unique distributed database characteristics. 3) A dual-branch joint graph neural network that jointly trains execution plan and database architecture graph representations, adaptively fusing these features. 4) Experimental validation indicates DISEncoder's seamless integration into database tasks, potentially improving machine learning accuracy and query encoding efficiency compared to conventional database algorithms.

2 Related Work

Distributed Databases Query Representation. Existing methods primarily use resource usage and execution plans. Li et al. [3] used regression to predict

query latency with block I/O and network latency features. QIACO [4] models query cost based on I/O and communication overhead of JOIN operations. However, resource usage overlooks execution details. Wu et al. [14] used a GCN on execution plans in Amazon Redshift, combining the output with system features. Zero-shot [6] represents queries as graphs with transferable feature annotations. QTune [15] encodes operator costs and SQL statement types. A key limitation is the inadequate consideration of distributed database architecture.

Single-node Databases Query Representation. Single-node databases often use physical plans for tasks like cardinality estimation [8] and query optimization [12,13]. Tree-RNN models like RTOS [16] and TLSTM [7] process tree-structured plans, but are difficult to train on large plans. Tree-CNN models (NEO [13], BAO [12]) capture parent-child dependencies but have limited receptive fields. QueryFormer [9] uses a tree-based physical plan representation, capturing parent-child dependencies and long-path information. PreQR [10] uses SQL queries as features with an attention mechanism-based SQL encoder. SpiderSchemaGnn [17] represents the database schema as a graph and uses GNNs. QPSEncoder [11] extracts features from execution plans, SQL statements, and database schemas. However, these methods fail to capture the characteristics of distributed queries.

3 DISEncoder Framework

To learn a robust representation of distributed database queries, our objective is to select appropriate query representation features and subsequently encode them. In Fig. 1 illustrates the architecture of DISEncoder, which is primarily divided into three components.

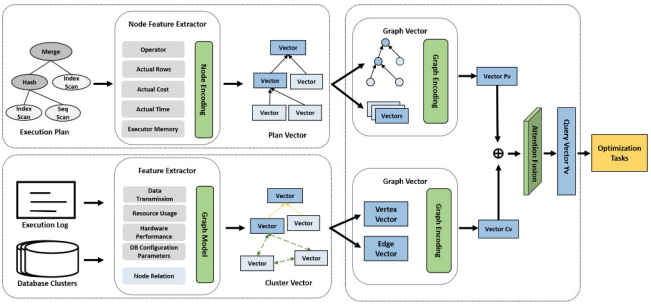


Fig. 1. Overview of the DISEncoder Framework: Dual-Branch Graph Encoding with Execution Plan Vector (P_v) and Cluster Vector (C_v).

First, key features from execution plan tree nodes are encoded into fixed-size node vectors. Nodes are vertices in a graph, with parent-child relationships

as adjacencies, constructing a plan vector graph. A Graph Attention Network (GAT) [18] encodes this graph into a fixed-size vector P_v (Sect. 3.1).

Second, static architecture and dynamic execution data combine to form a cluster vector graph, representing distributed database characteristics. Cluster nodes are vertices; deployment locations define vertex relationships. A GAT encodes this into a fixed-size vector C_v (Sect. 3.2).

Finally, a dual-branch model trains two GATs to obtain encodings P_v and C_v. An adaptive attention mechanism fuses these (P_v and C_v), resulting in query representation Y_v (Sect. 3.3).

$$Y_v = AttentionFuse(P_v, C_v) \tag{1}$$

3.1 Execution Plan Encoding

The database execution plan provides critical information on operations, data access paths, parallel execution, and resource utilization, which are vital for optimizing performance, managing resources, diagnosing issues, and enhancing query understanding [19]. The optimizer generates the execution plan based on the query, from which significant information is extracted from the nodes and encoded into feature vectors. By employing the GAT, we perform message passing on the dependencies between parent and child nodes within the execution plan tree, capturing its structural information. Finally, global pooling is applied to all nodes, resulting in a fixed-length vector that represents the execution plan.

Node Encoding. During query execution, execution plans are collected from the optimizer and parsed into a tree structure, as shown in Fig. 2. Each tree node represents a physical operator (e.g., Aggregation, Join, Scan). We identify seven features for each node: operator type, total cost, actual rows, plan width, actual startup time, actual total time, and execution memory. Operator types are encoded as a 21-dimensional one-hot vector, while the other six features are numerical. To address potential issues arising from differences in the scale of these features, we apply min-max normalization, standardizing values within the range of [0,1]. This results in a feature vector of length 27.

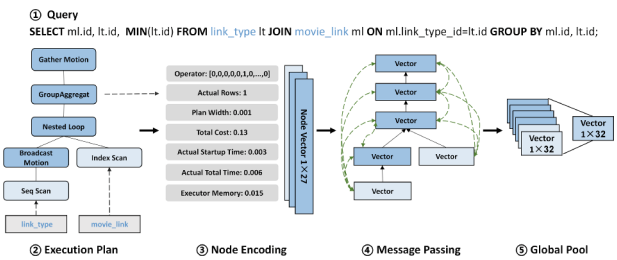


Fig. 2. Execution Plan Encoding Pipeline: Node Feature Extraction and GAT-Based Message Passing.

Message Passing. To capture the interactions among operators in the execution plan tree, we employ a GAT for message passing. In each layer, a node v_i first gathers the feature vectors h_j of its neighbors $v_j \in \mathcal{N}(i)$. We compute the unnormalized attention coefficient between v_i and v_j as:

$$e_{ij} = \text{LeakyReLU}\big(\mathbf{a}^\top [\mathbf{W}h_i \,\|\, \mathbf{W}h_j]\big), \tag{2}$$

where $\mathbf{W} \in \mathbb{R}^{F' \times F}$ is a learnable weight matrix projecting F-dimensional features into F' dimensions, $\mathbf{a} \in \mathbb{R}^{2F'}$ is the attention weight vector, and $\|$ denotes concatenation. We then normalize these coefficients across all neighbors of v_i using the softmax function:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik})}. \tag{3}$$

Finally, the updated feature of node v_i is obtained by a weighted sum of its neighbors' transformed features:

$$h'_i = \sigma\Big(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} \, \mathbf{W} \, h_j \Big), \tag{4}$$

where $\sigma(\cdot)$ is an activation function. By stacking multiple GAT layers, each node aggregates information from progressively larger neighborhoods, and a final global pooling step produces a fixed-size vector of the execution plan graph.

Global Pooling. Following the processing of plan vectors through multiple layers of attention mechanisms, the embedding of each node not only encapsulates its individual features but also integrates information from neighboring nodes. Ultimately, global pooling aggregates the embedding representations of all nodes into a fixed-dimensional vector that captures the semantic information of the entire graph, thereby representing the characteristics of the execution plan tree as a whole. This one-dimensional vector serves as input for subsequent classifiers or other machine learning models.

3.2 Distribute Databases Features Encoding

The architecture of distributed database clusters has a significant influence on data distribution, parallelism of query execution, and network transmission. Furthermore, the resource utilization of nodes and the data transmission between nodes during query execution are vital for overall performance. We develop a graph model that incorporates the cluster architecture and execution logs to capture the architectural features as well as the characteristics of distributed execution. These two components are integrated into a graph structure, which is subsequently encoded into a fixed-size embedding vector.

Cluster Graph Construction. Graph structure information comes from database cluster features and query execution process features (Fig. 3). Cluster features include node identifiers, major database parameter configurations, server hardware metrics, and node deployment relationships. Execution process

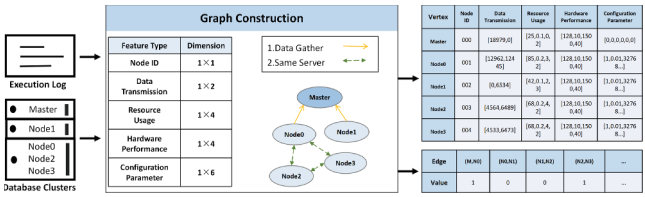


Fig. 3. Cluster Graph Construction: Integrating Static Architecture and Dynamic Execution Logs into Vertex and Edge Features.

features consist of node resource utilization and inter-node data transmission volumes. Combining these features constructs vertex and edge features, ultimately generating vertex and edge feature vectors.

Vertex Features. In distributed databases, nodes are typically Master and DataNode [20]. The Master generates the execution plan, while DataNodes execute it. We represent database nodes as vertices. The Master node, though not executing queries, receives and aggregates data from DataNodes and is included as a special vertex. Each vertex contains key DataNode features: node identifier (NodeID), major database parameter configurations (shared_buffers, work_mem), hosting server hardware metrics (memory size, disk size, network bandwidth, CPU cores), resource utilization during query execution (memory usage, CPU usage, I/O read/write time), and data transmission (DataNode input/output volumes). For vertex v_i, NodeID, Data Transmission, Resource Usage, Hardware Performance, and Configuration Parameter features are represented as N_i, D_i, U_i, H_i, C_i, respectively. The vertex feature vector is $V_i = [\hat{N}_i, \hat{D}_i, \hat{U}_i, \hat{H}_i, \hat{C}_i]$, where $\hat{N}_i$, $\hat{D}_i$, $\hat{U}_i$, $\hat{H}_i$, $\hat{C}_i$ are the normalized features of N_i, D_i, U_i, H_i, C_i, respectively.

Edge Features. Edge features reflect the topological structure and capture data aggregation during queries. As illustrated in Fig. 3, we categorize edge features into two types: "Same Node" indicates DataNodes deployed on the same server (bidirectional connection), and "Data Gather" denotes unidirectional data transmission from DataNodes to the coordinating Master node. For each vertex pair (v_i, v_j), a connection forms the edge matrix E if they satisfy either relationship. The cluster graph is defined as $G = (V, E)$, where V is the vertex matrix (each row is a node's feature vector), and E is the edge matrix (indicating node relationships).

3.3 Dual Branches Feature Encoding Model

To effectively process and integrate the features of the execution plan tree and distribute databases features, we propose a dual-branch joint deep learning model. Given that both types of features are represented as graph structures, the model employs GAT to encode the vertex and edge matrices, transforming the two graph structures into one-dimensional vectors. Finally, an attention

mechanism is utilized to merge these vectors, thereby enhancing the model's performance in downstream prediction tasks.

Encoding Phase. We use two GAT branches to encode graph features. GAT's self-attention mechanism assigns varying weights to each node, adaptively aggregating information from neighbors. This considers node features and captures complex relationships. Algorithm 1 details a single GAT branch. We design two graph attention layers. For each node pair (v_i, v_j), $\sigma^{(l)}$ and $W^{(l)}$ represent the activation function and learnable weights, respectively. The attention weight e_{ij} is calculated via attention. For each node v_i, a normalized attention distribution α_{ij} is derived based on its attention weights e_{ij} relative to other nodes. Using α_{ij}, information from each node's neighbors is aggregated to produce the aggregated feature representation $h_i^{'}$ for node v_i. Stacking two GAT layers progressively extracts local and global graph features, embedding structural information into vertex vectors. Global average pooling on weighted feature vectors yields the high-level representation feature H for the entire graph.

Algorithm 1. Single Branch Graph Encoding: SingleBranchGraphEncoding.

Require: V: a vertex matrix; E: an edge matrix; L: the graph encoding network
$h_1, h_2, \ldots, h_n \leftarrow V$
for $(\sigma^{(l)}, W^{(l)}) \in L$ **do**
  $e_{ij}^{(l)} \leftarrow \text{LeakyReLU}(\mathbf{W}^{(l)} h_i^{(l)} || \mathbf{W}^{(l)} h_j^{(l)})$
  $e_{ij}^{(l)} \leftarrow e_{ij}^{(l)} \cdot E_{ij}$
  $\alpha_{ij}^{(l)} \leftarrow \frac{\exp(e_{ij}^{(l)})}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik}^{(l)})}$
  $h_i^{'} \leftarrow \sigma^{(l)} \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(l)} \mathbf{W}^{(l)} h_j^{(l)} \right)$
end for
$H \leftarrow \text{GlobalMeanPooling}(h_1^{'}, h_2^{'}, \ldots, h_n^{'})$
return H

Algorithm 2. Dual Branches Feature Fusing.

Require: G_1: a graph matrix of plan; G_2: a graph matrix of cluster
$\mathbf{h_1} \leftarrow \text{SingleBranchGraphEncoding}(G_1)$
$\mathbf{h_2} \leftarrow \text{SingleBranchGraphEncoding}(G_2)$
$\mathbf{C} \leftarrow \mathbf{h_1} + \mathbf{h_2}$
$\mathbf{w} \leftarrow \text{sigmoid}(attentionLager(\mathbf{C}))$
$\mathbf{h}_{fused} \leftarrow \mathbf{w} \cdot \mathbf{h_1} + (1 - \mathbf{w}) \cdot \mathbf{h_2}$
Output $\leftarrow \text{FullyConnectedLayer}(\mathbf{h}_{fused})$
return Output

Feature Fusion Phase. We implement an attention-based fusion module that dynamically weights the vectors H from the dual-branch GAT. This assigns greater fusion weights to significant features to optimize information integration. This mechanism enhances sensitivity to key features and improves information fusion flexibility, providing more precise inputs for subsequent learning. As illustrated in Algorithm 2, features are encoded into vectors h_1 and h_2 via

their respective branches. These are concatenated, and attention layers generate attention weights w reflecting each vector's importance in the final fusion. Attention weights w are applied to compute the weighted sum of feature vectors h_1 and h_2, yielding the fused vector h_{fused}. A fully connected layer produces the final *Output*.

Joint Training. The training data consists of tuples $< Q, Pi, Ri, Ci, P >$, where Q represents queries, Pi execution plans, Ri execution logs, Ci database cluster architectures, and P actual query performance. Features are constructed from Pi, Ri, and C_i, which are input into a dual-branch model. Both branches perform forward propagation to produce feature vectors, which are then combined using an attention mechanism. This fusion creates a comprehensive feature representation for subsequent tasks, with the loss function computed and parameters updated via backpropagation. For cost estimation, the feature representations are concatenated with a multi-layer perceptron (MLP) to predict the cost value, enhancing the model's sensitivity and expressiveness.

4 Experiments

4.1 Experiment Setting

We deploy the Greenplum v6.21.0 distributed database across four servers (16 GB RAM, 256 GB disk, 4G Hz CPU), consisting of nine DataNodes and one Master. The encoding and predicting models are implemented using PyTorch and trained on an NVIDIA RTX 3080 GPU with 12GB of memory.

Datasets. We utilized three OLAP datasets. (1) *JOB* [21]: This OLAP benchmark utilizes a real-world IMDB dataset, consisting of 21 tables with primary and foreign key relationships, and 113 query templates. We generate 2,000 queries were used. These queries were generated by randomly combining different tables, columns, and predicates from the original set of 113 queries. We employ the default data volume of 7.5GB. (2) *TPC-H*: This dataset contains 8 tables and 22 query templates. Using these templates, we generate 2,000 queries that include complex operations such as joins, aggregations, and subqueries. We generate the data volume of 1GB. (3) *TPC-DS*: This dataset comprises 25 tables and 99 query templates. From these templates, we generate 2,000 queries. In addition to complex queries, TPC-DS includes more data loading and update operations, as well as a broader range of query patterns. Finally, to evaluate the performance of the trained model, we used the original 113 JOB queries, the 22 TPC-H templates, and the 99 TPC-DS templates as the validation dataset. We generate the data volume of 1.2GB.

Metrics. We use four metrics to assess DISEncoder: (1) *Error Rate*. We estimate prediction accuracy using the Q-Error [6,9], which is formally defined as:

$$Q\text{-}Error(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n} \frac{max(y_i, \hat{y_i})}{min(y_i, \hat{y_i})} \tag{5}$$

where y represents the true value and $\hat{y}$ represents the predicted value. (2) *Model Training Time*. This includes the time required to build input features and train the model. (3) *Model Predicting Time*. This includes the time required to build input features and predict. (4) *Peak GPU Memory Usage During Training*. This is the peak memory allocated during training.

Evaluation Techniques. We compare DISEncoder with four state-of-the-art methods: (1) *End-to-End (E2E) Model* [8]: A TLSTM-based cost model integrating data statistics into a tree-structured execution plan. LSTM units estimate operator costs in a tree structure to determine the total query cost. (2) *Zero-shot Model* [6]: Encodes table and column information within a graph-structured execution plan. MLP-extracted node features are used for GNN message passing, and an MLP-based model estimates execution time. (3) *Global-stage Model* [14]: Implements the Global-stage prediction model. Constructs execution plans as tree features, uses graph convolution, and concatenates encoded execution plan features with system features to predict execution time. (4) *QPSEncoder Model* [11]: Integrates physical plans, SQL statements, and database schemas. Features are extracted, encoded, and fused to capture workload characteristics, serving as a general database workload encoder. Uses BERT for SQL encoding, but BERT's 512-token limit prevents its evaluation on TPC-DS (queries often exceed this length).

Model Configuration. The DISEncoder consists of two GAT branches, each with 2 attention layers and 8 attention heads per layer. Each attention head has 32 hidden units, resulting in a total of approximately 0.48 million trainable parameters across both branches. We use the Adam optimizer with a learning rate of 0.001 and a batch size of 64.

4.2 Model Effectiveness Evaluation

We evaluate DISEncoder against four benchmarks on JOB, TPC-H, and TPC-DS datasets for execution time prediction and cost estimation. Predictions are made by concatenating the DISEncoder's one-dimensional feature vector with a two-layer MLP. Results are in Tables 1. DISEncoder consistently achieves the lowest Q-Error across all datasets, demonstrating strong adaptability and effective representation of distributed query characteristics. The GAT in DISEncoder adaptively allocates weights through its attention mechanism, enhancing its ability to handle diverse nodes and edges with high adaptability.

E2E shows the highest Q-Error due to its high-dimensional features (one-hot and bitmap encoding for operation, metaData, predicate, sample bitmap) that dilute critical information, making it unsuitable for distributed queries. Zero-shot and Global-stage use graph convolution, exhibiting some scalability but limited database schema dependency. Zero-shot's performance drops significantly on TPC-DS (20x compared to JOB/TPC-H) due to increased query complexity. QPSEncoder achieves the second-lowest error on JOB/TPC-H, effectively capturing information using physical plans, SQL statements, and database schema. However, BERT limits its effectiveness with complex queries.

Table 1. Comparison of Mean Q-Error of Execution Time(Time-Error) and Cost Estimation(Cost-Error) across Three Datasets.

Model	JOB		TPC-H		TPC-DS	
	Time-Error	Cost-Error	Time-Error	Cost-Error	Time-Error	Cost-Error
E2E	7448.99	8048.38	1835.27	2455.70	864.23	801.74
Zero-shot	12.84	22.85	10.72	18.19	200.33	173.46
Global-stage	86.31	2585.00	42.54	64.38	7.88	30.15
QPSEncoder	7.84	8.04	5.00	13.13	–	–
DISEncoder	**0.74**	**2.03**	**3.46**	**0.77**	**5.04**	**21.56**
Plan-Encoder	60.30	14.66	25.31	10.70	6.75	23.71
Cluster-Encoder	7154.22	13.77	14.66	-386.19	5.09	214.74

4.3 Model Component Evaluation

To assess branch impact, we compared DISEncoder with Plan-Encoder (no Cluster branch) and Cluster-Encoder (no Plan branch). Tables 1 shows that DISEncoder consistently achieves the lowest Q-Error. Plan-Encoder's Q-Error is lower than Cluster-Encoder's, indicating execution plans are more effective for representing queries. Plan-Encoder's performance on TPC-DS is comparable to DISEncoder, highlighting its suitability for complex queries. Cluster-Encoder, focusing on distributed architecture and execution, lacks detailed execution plan information, resulting in diminished performance. Therefore, only integrating execution plan features with distributed database features fully captures the characteristics of distributed queries and adapts to various workloads.

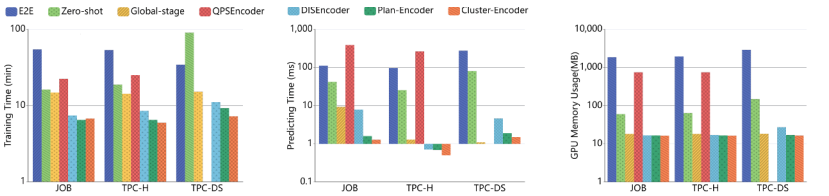
Ablation Study on Attention Mechanism. We introduce a new baseline: DISEncoder-GCN, which replaces the GAT in both branches with standard GCN that aggregate neighbor information uniformly. Table 2 shows that removing the GAT attention layers causes the mean execution time Error to increase from 5.04 to 10.72 and the mean cost estimation error to rise from 21.56 to 42.63. This significant degradation confirms that the attention mechanism is critical for capturing both execution plan semantics and distributed cluster characteristics.

Table 2. Ablation Study on Attention Mechanism (JOB dataset).

Model	DISEncoder	DISEncoder-GCN	Plan-Encoder	Cluster-Encoder
Execution Time Errors	5.04	10.72	6.75	5.09
Cost Estimation Errors	21.56	42.63	23.71	217.74

4.4 Model Efficiency Evaluation

In the execution time prediction task, we assessed seven models (including DISEncoder and four benchmarks) based on training time, peak GPU memory usage during training, and prediction time to evaluate DISEncoder's efficiency. Figure 4 illustrates the logarithmic values of these three metrics. Throughout training, DISEncoder required less time and peak GPU memory compared to benchmarks. This efficiency arises from concurrent training of two graph encoding branches, which accelerates model convergence, and its conversion of the execution plan tree into a graph structure that reduces data volume.



(a) Training Time (b) Predicting Time (c) GPU Memory Usage

Fig. 4. Model Efficiency Evaluation: (a) Training Time, (b) Prediction Time, (c) Peak GPU Memory Usage on the RTX 3080.

E2E and QPSEncoder used more time and memory. E2E encodes plan tree nodes into high-dimensional vectors and employs LSTM, demanding greater training resources. QPSEncoder encodes three structural features, increasing data volume and training complexity. Zero-shot's training time significantly increased under TPC-DS, indicating challenges with complex queries. Plan-Encoder and Cluster-Encoder require fewer resources than DISEncoder due to single-branch training. In prediction time, DISEncoder exhibited the lowest requirements among benchmarks, while E2E and QPSEncoder required more time due to complex feature encoding. Overall, DISEncoder is highly efficient, requiring less time and fewer resources for both training and prediction with its dual-branch encoding model.

5 Conclusion

This paper presents a method for representing queries in distributed databases based on graph models. We construct two graph structures using three features: the execution plan tree, the distributed database architecture, and the execution logs. We introduce the dual-branch graph encoding model, DISEncoder, which employs a graph attention mechanism to embed graph features and fuses these two types of features to create a comprehensive query representation. DISEncoder is integrated into two database learning tasks and is evaluated for its effectiveness and efficiency across three real-world datasets. The results indicate

that DISEncoder enhances the performance of existing methods by providing more effective query representations.

Performance Limitations on Specific Query Types. While DISEncoder demonstrates strong overall performance, it tends to underperform on queries with highly uniform or minimal structural complexity, such as single-table scans or simple projections without joins. In such cases, the benefit of dual-graph encoding diminishes, and traditional feature-based methods may suffice. Moreover, the encoding of inter-node communication becomes less informative when data locality or network cost is negligible.

References

1. Parchas, P., Naamad, Y., Van Bouwel, P., Faloutsos, C., Petropoulos, M.: Fast and effective distribution-key recommendation for amazon redshift. Proc. VLDB Endowment **13**(12), 2411–2423 (2020)
2. Zhou, X., Li, G., Feng, J., Liu, L., Guo, W.: Grep: a graph learning based database partitioning system. Proc. ACM on Manag. Data **1**(1), 1–24 (2023)
3. Li, H., Hou, X., Chen, M., Dai, Z., Zhu, M., Huang, M.: Performance prediction for concurrent workloads in distributed database systems. In: Algorithms and Architectures for Parallel Processing: 15th International Conference, ICA3PP: Zhangjiajie, China, 18–20 November 2015, Proceedings, Part IV 15, Springer, vol. 2015, pp. 626–639 (2015)
4. Mohsin, S.A., Darwish, S.M., Younes, A.: Qiaco: a quantum dynamic cost ant system for query optimization in distributed database. IEEE Access **9**, 15 833–15 846 (2021)
5. Saxena, G., et al.: Auto-wlm: machine learning enhanced workload management in amazon redshift. In: Companion of the International Conference on Management of Data, vol. 2023, pp. 225–237 (2023)
6. Hilprecht, B., Binnig, C.: Zero-shot cost models for out-of-the-box learned cost prediction. Proc. VLDB Endowment **15**(11), 2361–2374 (2022)
7. Marcus, R., Papaemmanouil, O.: Plan-structured deep neural network models for query performance prediction. In: Proceedings of the VLDB Endowment, vol. 12, no. 11 (2019)
8. Sun, J., Li, G.: An end-to-end learning-based cost estimator. In: Proceedings of the VLDB Endowment, vol. 13, no. 3 (2020)
9. Zhao, Y., Cong, G., Shi, J., Miao, C.: Queryformer: a tree transformer model for query plan representation. Proc. VLDB Endowment **15**(8), 1658–1670 (2022)
10. Tang, X., Wu, S., Song, M., Ying, S., Li, F., Chen, G.: Preqr: pre-training representation for sql understanding. In: Proceedings of the 2022 International Conference on Management of Data, pp. 204–216 (2022)
11. Yang, J., Zhang, Q., Yan, J., Ding, Z., Zhu, M., Lv, X.: Qpsencoder: a database workload encoder with deep learning. In: International Conference on Database and Expert Systems Applications, Springer, pp. 108–123 (2024)
12. Marcus, R., Negi, P., Mao, H., Tatbul, N., Alizadeh, M., Kraska, T.: Bao: making learned query optimization practical. In: Proceedings of the 2021 International Conference on Management of Data, pp. 1275–1288 (2021)
13. Marcus, R., et al.: Neo: a learned query optimizer. Proc. VLDB Endowment **12**(11), 1705–1718 (2019)

14. Wu, Z., et al.: Stage: query execution time prediction in amazon redshift. In: Companion of the International Conference on Management of Data, vol. 2024, pp. 280–294 (2024)
15. Li, G., Zhou, X., Li, S., Gao, B.: Qtune: a query-aware database tuning system with deep reinforcement learning. Proc. VLDB Endowment **12**(12), 2118–2130 (2019)
16. Yu, X., Li, G., Chai, C., Tang, N.: Reinforcement learning with tree-lstm for join order selection. In: 2020 IEEE 36th International Conference on Data Engineering (ICDE), pp. 1297–1308. IEEE (2020)
17. Bogin, B., Berant, J., Gardner, M.: Representing schema structure with graph neural networks for text-to-sql parsing. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4560–4565 (2019)
18. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., et al.: Graph attention networks. Stat **1050**(20), 10–48 550 (2017)
19. Fritchey, G.: Execution plan generation and the query optimizer. In: SQL Server, Query Performance Tuning: Troubleshoot and Optimize Query Performance, Springer, vol. 2022, pp. 23–40 (2022)
20. Lyu, Z., et al.: Greenplum: a hybrid database for transactional and analytical workloads. In: Proceedings of the 2021 International Conference on Management of Data, pp. 2530–2542 (2021)
21. Leis, V., Gubichev, A., Mirchev, A., Boncz, P., Kemper, A., Neumann, T.: How good are query optimizers, really? Proc. VLDB Endowment **9**(3), 204–215 (2015)

A Subject-Independent Stress Detection Model Based on Temporal Feature Disentanglement

Qingwei Zeng, Peibo Duan(✉), Yongsheng Huang, and ChangSheng Zhang

School of Software, Northeastern University, Shenyang, China
duanpeibo@swc.neu.edu.cn

Abstract. This study proposes a subject-independent stress monitoring model based on temporal feature disentanglement, aimed at improving the accuracy and generalizability of stress detection. Existing stress prediction models often perform poorly with new participants due to differences in individual physiological and psychological responses. Addressing this issue, we develop a method that effectively extracts and removes individual-specific features while retaining stress-sensitive features.

Using physiological signals and public datasets, we demonstrate that temporal disentanglement enhances model performance and generalization. Experimental results show significant accuracy improvements for detecting stress in new participants. Additionally, we introduce the use of ID labels for segment disentanglement, effectively reducing individual feature interference and increasing detection stability.

This research lays the groundwork for advanced stress monitoring systems, promoting employee health and sustainable corporate development through real-time, accurate detection.

Keywords: Occupational stress · Stress detection · Physiological signals · Temporal disentanglement · Generalization ability

1 Introduction

Occupational stress, often called the "invisible epidemic," poses significant threats to global health and economic efficiency. According to the World Health Organization (WHO), it results in an annual productivity loss of $1 trillion and is linked to chronic diseases like cardiovascular conditions (HR=1.38) and depression (OR = 2.17) [4,17]. Prolonged stress activates the autonomic nervous system (ANS), disrupting indicators such as heart rate variability (HRV) and cortisol secretion, which diminishes health and efficiency [11]. Thus, developing effective stress detection techniques is critical [14,17].

While traditional stress-monitoring methods such as surveys offer convenience, they suffer from bias due to small sample sizes, emphasizing the growing need for sensor-based physiological data collection. Physiological data, less prone to distortion, can reliably detect stress through changes in heart rate, sweating, and breathing patterns. Devices like Biopacs, Empatica E4, and RespiBAN have shown promise in this domain [13].

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 441–452, 2026.
https://doi.org/10.1007/978-3-032-04555-3_36

Despite advancements in machine learning-based stress prediction models [2], their **generalization ability**—particularly in subject-independent scenarios—remains limited [20,24]. Variations in individual physiology and coping mechanisms result in inconsistent stress detection across unseen individuals, restricting the use of generalized models [9].

To address this, we propose a method that uses **temporal sequence disentanglement** to isolate stress-related features from individual-specific traits, thereby improving generalization capabilities [21]. Through effective feature extraction and disentanglement, we hypothesize that it is possible to efficiently predict stress states across diverse individuals [22].

This study validates the approach using publicly available datasets [26] and introduces a novel disentanglement mechanism, enabling improved stress detection for unseen individuals. The contributions of this paper are summarized as follows:

- Leveraging temporal disentanglement to extract stress-related features [15].
- Utilizing ID labels to isolate individual-specific features and retain only stress-sensitive features [19].
- Providing a versatile algorithm applicable to various temporal physiological datasets and classification models [20].

The remainder of this paper is organized as follows: Sect. 3 details the DTS temporal disentanglement algorithm, Sect. 3.1 defines the problem, Sect. 3.2 elaborates on technical insights, and Sect. 4 presents experimental validations. Conclusions and future work are discussed in Sect. 5.

2 Related Work

Research on stress detection has increasingly focused on physiological signals as reliable stress indicators resistant to intentional distortion [9,11]. Among these signals, ECG is the most widely used [13,16]. Study examined HR, HRV, and temporal/frequency domain GSR features, reporting the highest detection accuracy [16]. Another study [2] utilized EDA and PPG signals with AdaBoost, SVM, and KNN classifiers, achieving 94

A personalized SVM-based stress detector using ECG+EDA and EDA+PPG features from the CLAS dataset [20] reached 88.9% accuracy under video stimuli, and 84.1% and 86.1% under image stimuli [2]. Another approach [2] using ECG and EDA features from the ASCERTAIN dataset achieved 68.7% accuracy with SVM classifiers [1,11]. Additionally, research on HRV and EDA features from the SWELL-KW and WESAD datasets indicates that general stress prediction models perform worse than individual-specific models in both subject-dependent and subject-independent settings [3,24,26].

Deep learning has furthered stress detection capabilities. For example, one-dimensional recursive CNNs analyzing ECG and EEG features achieved up to 90% accuracy [4,7], while ultra-short-term ECG with deep learning reached

87.4% [27]. A novel multi-core deep learning architecture based on cardiac activity attained an average accuracy of 89.8%with 6-fold cross-validation [7].

Existing studies mainly focus on personalized models [14,21]; subject-independent models still lag behind subject-dependent ones [25]. Although deep learning shows promise, its effectiveness in subject-independent scenarios is limited without disentangling individual-specific features [19,22].

This paper investigates deep learning models with feature disentanglement to enhance subject-independent stress detection. Prior research has not specifically explored the impact of disentangled learning on improving generalization in deep models [22]. The baseline models selected include classic recurrent models such as LSTM [6] and state-of-the-art time series classification models like MultiRocket [18] and Gated Transformer [23].

3 Methodology

In this section, we propose a multi-level disentanglement approach to extract stress-relevant features while removing individual difference features. We partition the latent space into segments, each representing distinct semantic information, ensuring effective learning and feature independence.

General models typically show reduced accuracy on unseen individuals because they cannot capture the varied physiological responses to stress. To verify this, we added subject IDs as control features. The consistently high average marginal impurity decrease (MDI) values across all datasets confirm that subject IDs effectively represent individual differences. Based on this insight, we developed a disentanglement framework for stress classification (Fig. 1).

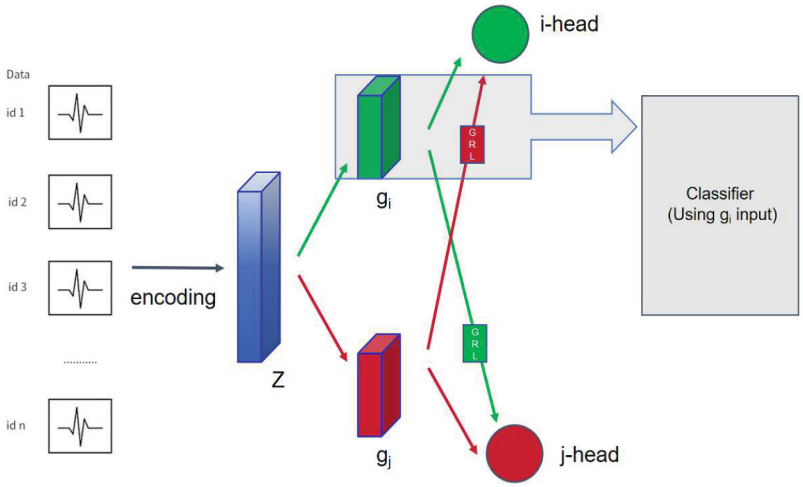


Fig. 1. The Overview of the Disentanglement Framework for Stress Classification

3.1 Fundamentals of Disentanglement Implementation

First, we elaborate on the concept of disentanglement. The goal of disentanglement is to maximize the marginal likelihood of observing samples by optimizing the model. This means that the model aims to accurately describe and generate input data. During this process, the data features learned by the model should be composed of different and independent latent factors. This independence is the core of disentanglement. Therefore, we can describe the generative process of observational data through latent variables:

$$p(x) = \int p(x|Z)p(Z)dZ \tag{1}$$

where θ represents the parameters of the neural network, and $p(x|Z)$ represents the true underlying distribution, which can be estimated through the training dataset. Thus, the marginal likelihood of observed data can be decomposed into an expectation of its conditional probabilities regarding latent variables. This approach is crucial in variational inference since we often need to optimize the distribution of latent variables to approximate the posterior distribution.

Based on this, we introduce the Evidence Lower Bound (ELBO):

$$ELBO = E_{q_\phi(Z|x)}[\log p_\theta(x|Z)] - D_{KL}(q_\phi(Z|x) \parallel p(Z)) \tag{2}$$

The motivation for disentanglement lies in our expectation that different latent variables will encode different information in the data. Thus, each latent variable should independently interpret different parts of the input data, avoiding information entanglement or over-reliance.

In many variational autoencoder frameworks, increasing the KL divergence penalty aims to encourage the model to produce disentangled latent representations, such as in β-VAE. However, increasing the intensity of the KL divergence penalty often leads to the vanishing KL divergence problem [8].

To address this, we need to balance the trade-off between KL divergence and mutual information. This leads to the optimized ELBO formula:

$$- L_{D_{KL}}(q_\phi(Z|x) \parallel p(Z)) + \alpha I_{q_\phi}(x; Z) + E_{q_\phi(Z|x)}[\log p_\theta(x|Z)] \tag{3}$$

The ELBO (Evidence Lower Bound) is an important metric used to maximize the posterior distribution of the latent variables Z given the observed data x [8].

The first term (reconstruction loss):

$$E_{q_\phi(Z|x)}\left[\log p_\theta(x \mid Z)\right] \tag{4}$$

reflects the model's ability to reconstruct the input data x, ensuring effective capture of the structural features of the data.

The second term (KL divergence):

$$D_{KL}(q_\phi(Z \mid x) \parallel p(Z)) \tag{5}$$

measures the difference between the posterior distribution and the prior distribution. By minimizing this difference, the training of the model promotes better disentanglement effects.

Motivation for disentanglement: When learning latent representations, we expect different latent variables to encode different aspects of the data. This way, each latent variable should provide an independent explanation for different parts of the input data, avoiding information overlap or excessive dependence. Extracting the desired features through disentanglement helps improve the model's generalization ability on new tasks.

In many variational autoencoder frameworks, increasing the KL divergence aims to encourage the model to produce disentangled latent representations, such as in β-VAE. It is a variant of the Variational Autoencoder (VAE) that attempts to learn disentangled representations by optimizing a heavily penalized objective, typically with $\beta > 1$:

$$L_{\beta\text{-ELBO}}(x) \quad = \quad \mathbb{E}_{q_\phi(Z|x)}\left[\log p_\theta(x \mid Z)\right] \quad - \quad \beta D_{\mathrm{KL}}\left(q_\phi(Z|x) \parallel p(Z)\right) \tag{6}$$

However, increasing the strength of the KL divergence penalty often leads to the problem of KL divergence vanishing, where the posterior distribution $q_\phi(Z \mid x)$ tends to approach the prior distribution $p(Z)$, making it impossible for the model to learn meaningful latent structures from the data. This phenomenon may lead to the following challenges:

1. **Information loss:** Excessive penalties may cause the posterior distribution of the latent variables to fail to accurately reflect the distribution of the input data, resulting in the loss of critical information.
2. **Performance degradation:** When the model relies too heavily on the prior distribution, its reconstruction ability may decline, making it unable to effectively capture the complexity of the input data.
3. **Disentanglement failure:** Although the model aims to achieve disentanglement, excessive KL penalties may blur the relationships between latent variables, leading to disentanglement failure.

Before addressing the issue of KL divergence vanishing, we need to understand its causes. Introducing a heavier penalty in the ELBO encourages independence among latent variables but simultaneously ignores the mutual information between the latent variables Z and the input X. According to references [8,24], we define

$$q_\phi(Z, x) = q_\phi(Z \mid x) p_\theta(x) \tag{7}$$

and represent

$$q_\phi(Z) = E_{p_\theta(x)}[q(Z \mid x)] \tag{8}$$

as the aggregated posterior, which captures the aggregated structure of the latent variables under the data distribution $p_\theta(x)$. Mathematically, the KL term in the ELBO can be decomposed using the factorized $p(Z)$ as follows:

$$
\begin{aligned}
D_{\mathrm{KL}}\left(q_\phi(Z \mid \mathbf{x}) \| p(Z)\right) = & \mathrm{KL}\left(q_\phi(Z, \mathbf{x}) \| q_\phi(Z) p_\theta(\mathbf{x})\right) \\
& + \mathrm{KL}\left(q_\phi(Z) \| \prod_j q_\phi(z_j)\right) \\
& + \sum_j \mathrm{KL}\left(q_\phi(z_j) \| p(z_j)\right)
\end{aligned} \tag{9}
$$

Here, z_j represents the j-th dimension of the latent variables. The first term can be interpreted as the mutual information (MI):

$$
I_{q_\phi}(Z; x) \tag{10}
$$

between the data variables and the latent variables. The second term is called the Total Correlation (TC) [5], which evaluates the dependencies between variables. Penalizing the TC encourages statistically independent factors in the data distribution. Therefore, imposing a heavier penalty on this term helps achieve disentangled representations. This term explains the success of β-VAE. The last term is the dimension-wise KL divergence, which prevents each latent dimension from deviating too far from its prior distribution.

From the decomposed KL terms, we can see that while using β to penalize the total correlation for better disentangled representations, it also penalizes mutual information. This, in turn, may lead to underfitting or neglect of latent variables. Approximate inference distributions often differ significantly from the true posterior. This is undesirable because a primary goal of unsupervised learning is to learn meaningful latent features that should depend on the observations. Thus, the ELBO objective tends to fit the data distribution rather than perform correct inference. When these two goals conflict, the ELBO objective tends to sacrifice correct inference to better fit (or overfit) the training data, a phenomenon we can refer to as the information preference problem [8].

To address the issue of KL divergence vanishing caused by information preference, proposed optimizing the ELBO by introducing the maximization of mutual information (MI) between the latent representation and the input, thereby guiding the model to retain attention to important information in the input data during disentanglement. This leads to the optimized ELBO formula [10]:

$$
-L_{D_{KL}}(q_\phi(Z \mid x) \parallel p(Z)) + \alpha I_{q_\phi}(x; Z) + E_{q_\phi(Z|x)}\left[\log p_\theta(x \mid Z)\right] \tag{11}
$$

Here, $I_{q_\phi}(x; Z)$ represents the mutual information between x and Z under the distribution $q_\phi(Z; x)$.

However, this objective cannot be directly optimized. The literature proposes an alternative equivalent form:

$$
-L_{D_{KL}}(q_\phi(Z \mid x) \parallel p(Z)) + \alpha D_{KL}(q_\phi(Z) \parallel p(Z)) + E_{q_\phi(Z|x)}\left[\log p_\theta(x \mid Z)\right] \tag{12}
$$

The mutual information maximization term functions similarly to the first term in the ELBO-TC decomposition but in the opposite optimization direction. This approach mitigates the KL divergence vanishing problem while enforcing the model to retain disentangled properties. By introducing the mutual information maximization term, we alleviate the KL divergence vanishing issue while maintaining penalties on total correlation and dimension-wise KL (with $\beta > 1$) to achieve disentanglement.

3.2 Application of Disentanglement in Stress Prediction Models

To achieve robust stress prediction, latent variables must be separated into two groups:

- g_i: Features relevant to stress states,
- g_j: Features encoding individual-specific characteristics.

The incorporation of subject IDs informs disentanglement by guiding the removal of individual-specific information, and the disentanglement framework employs loss functions fine-tuned for each feature group:

$$L_{\text{stress}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(p_{ij}), \tag{13}$$

$$L_{\text{id}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} z_{ik}\log(q_{ik}), \tag{14}$$

where y_{ij} and z_{ik} are true labels for stress and identity, and p_{ij}, q_{ik} represent their predicted probabilities.

To ensure independence between g_i and g_j, we introduce a Gradient Reversal Layer (GRL). During backpropagation, the GRL reverses gradients associated with individual-specific features (g_j), enforcing adversarial training that isolates stress-related features (g_i) [12].

After applying the above methods for disentangling latent variables, we can effectively isolate individual-specific features from the latent variables, preserving only stress-related features. This reduction in interference aids in improving the model's accuracy and generalizability of stress classification.

4 Experiments

This section evaluates the effectiveness of the proposed disentangling method through a detailed analysis of its performance on a selected dataset, validation of the unmixing effect, and a comparison of classification models. Key evaluation metrics such as F1-score, accuracy, and standard deviation are used, along with visualizations highlighting the improvements in classification performance.

4.1 Dataset Selection

The SWELL dataset [26](Stress and Workload in Office Life) was chosen for its comprehensive collection of data related to work-induced stress. Developed by researchers at Radboud University in the Netherlands, the dataset consists of data from 25 participants exposed to typical workplace stressors, such as sudden urgent emails and time pressure, while performing office tasks including writing reports, creating presentations, and searching for information. Various data modalities such as computer usage patterns, facial expressions, body posture, ECG signals, and EDA signals were recorded.

Participants experienced three experimental conditions:

- **No Stress Condition**: Participants completed tasks under normal conditions for approximately 45 min.
- **Time Pressure Condition**: Task completion time was reduced to two-thirds of the time required under the no-stress condition.
- **Interference Condition**: Participants received distracting emails during tasks, with some emails requiring specific actions while others were irrelevant to the tasks.

At the end of each condition, subjective stress levels were assessed using the NASA-TLX (Task Load Index), which evaluates mental demand, time demand, effort, and frustration. This dataset provides a robust foundation for analyzing work-related stress and its impact on physiological and psychological responses.

After data pre-processing, 16 participants with complete ECG signal recordings across all conditions were selected. From the no-stress condition, 30-minute ECG signal segments were extracted, combined with data from the stress conditions, and processed to extract R-R intervals (RRI). This balanced dataset of stressed and non-stressed samples formed the basis for subsequent comparative analysis.

4.2 Comparison of Classification Model Results

The DTS algorithm was used to extract stress-sensitive features, which were then fed into multiple classification models for stress prediction. Leave-one-out cross-validation (LOOCV) was employed as the evaluation method to avoid partitioning bias and ensure robust performance assessment across samples. Models were trained and tested with consistent hyperparameter settings and hardware environments.

Table 1 summarizes the binary classification metrics, including F1-score, accuracy, and standard deviation, demonstrating that the disentangling mechanism consistently improves model performance and stability. Figures 2, 3, and 4 provide visual comparisons of baseline and disentangled models for LSTM, MultiRocket, and GatedTabTransformer, respectively.

Table 1. Performance comparison of classification models with and without disentangling.

Technical Selection	F1-score (%)	Accuracy (%)	Standard Deviation (%)
LSTM	82.71	83.51	8.53
Disentangled-LSTM	85.64	86.01	6.04
MultiRocket	87.12	88.39	4.90
Disentangled-MultiRocket	87.99	89.18	4.17
GatedTabTransformer	85.26	86.21	4.62
Disentangled-GatedTabTransformer	87.02	87.75	3.95

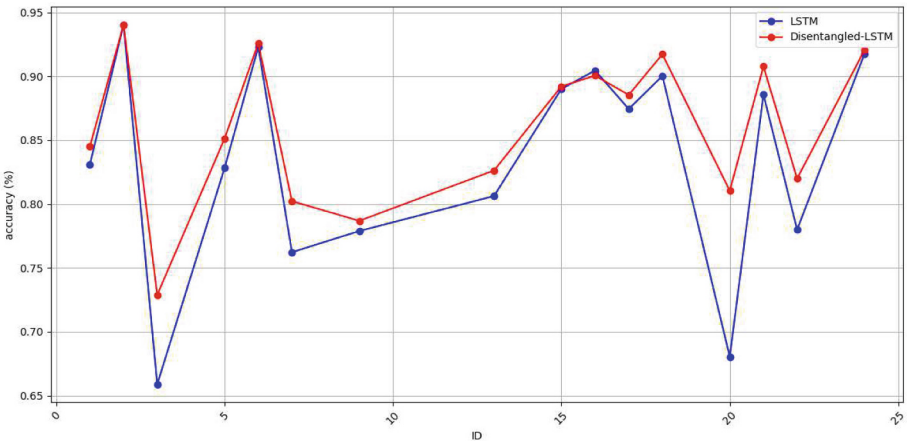


Fig. 2. Performance comparison between LSTM and Disentangled-LSTM on the SWELL dataset.

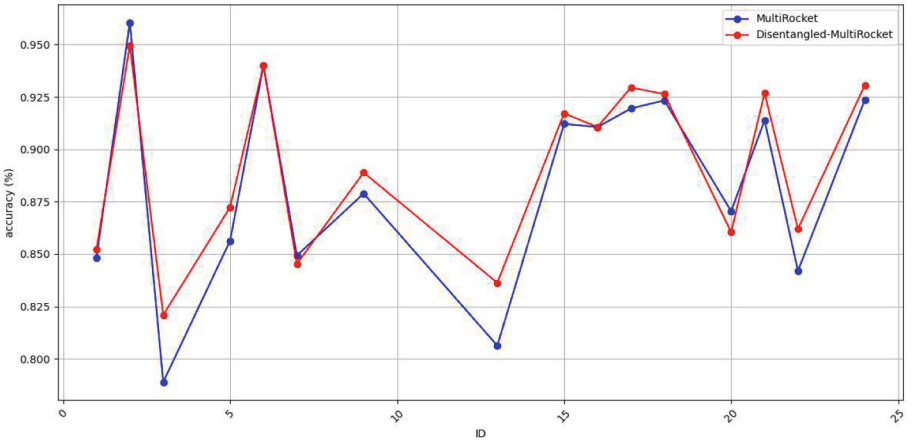


Fig. 3. Performance comparison between MultiRocket and Disentangled-MultiRocket on the SWELL dataset.

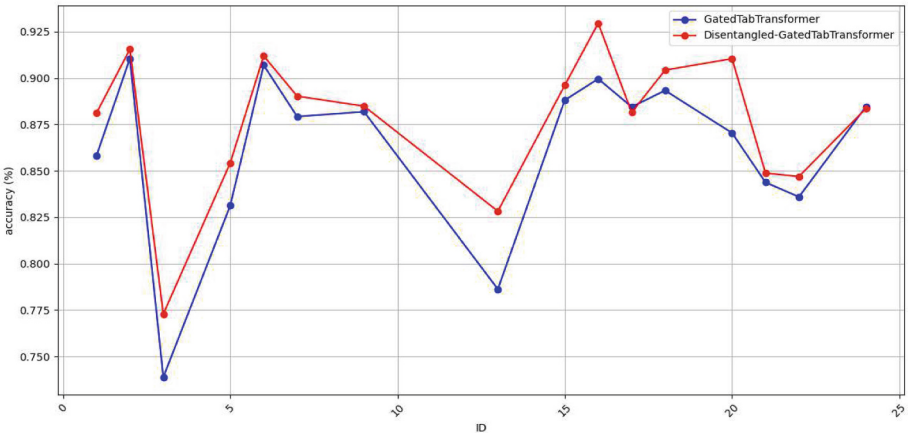


Fig. 4. Performance comparison between GatedTabTransformer and Disentangled-GatedTabTransformer on the SWELL dataset.

4.3 Analysis

- **LSTM-based model:** Accuracy improved from 83.51% to 86.01%, and standard deviation decreased from 8.53% to 6.04%. This indicates that disentanglement not only boosts accuracy but also enhances stability in sequential classification tasks.
- **MultiRocket model:** Already achieving high performance, it saw a slight accuracy gain (from 88.39% to 89.18%) and a reduction in standard deviation (from 4.90% to 4.17%).
- **GatedTabTransformer model:** Accuracy increased from 86.21% to 87.75%, with standard deviation decreasing from 4.62% to 3.95%.

Overall, these improvements validate that the disentangling mechanism effectively enhances model generalization and robustness by mitigating the impact of individual-specific variations.

5 Conclusion and Future Work

This study has proposed a subject-independent stress monitoring model utilizing temporal feature disentanglement techniques, aimed at improving the generalizability and accuracy of stress detection. Using publicly available datasets and deep learning models, we demonstrated the effectiveness of disentangling stress-relevant features from individual-specific characteristics. The experimental results confirmed the robustness of the disentanglement mechanism in reducing feature interference, improving classification accuracy, and enhancing model stability, particularly when addressing the stress states of previously unseen participants.

By leveraging R-R interval data for stress feature disentanglement, we achieved significant improvements in reducing the impact of individual differences, thus enhancing the reliability and practicality of the proposed model.

Importantly, the performance of the model under varying workloads provides a valuable basis for future stress management strategies and mental health interventions.

As part of future work:

- We aim to explore advanced cross-dataset transfer learning techniques to further enhance the generalization capabilities of subject-independent stress detection models.
- We will continue to optimize the disentanglement algorithm to ensure its real-time applicability and accuracy in practical environments.
- Finally, we plan to develop a scalable, fully-integrated stress monitoring system designed for long-term applications, promoting employee well-being and sustainable organizational growth.

This research serves as a crucial step toward building more efficient and robust stress detection methods, laying the groundwork for further studies and practical implementations in occupational health monitoring.

References

1. Alberdi, A., Aztiria, A., Basarab, A.: Towards an automatic early stress recognition system for office environments based on multimodal measurements: a review. J. Biomed. Inform. **59**, 49–75 (2016)
2. Marques, A.H., Silverman, M.N., Sternberg, E.M.: Evaluation of stress systems by applying noninvasive methodologies: Measurements of neuroimmune biomarkers in the sweat, heart rate variability and salivary cortisol. NeuroImmunoModulation **17**(3), 205–208 (2010)
3. Cinaz, B., Arnrich, B., La Marca, R., Tröster, G.: Monitoring of mental workload levels during an everyday life office-work scenario. Pers. Ubiquitous Comput. **17**(2), 229–239 (2013)
4. Fulcher, B.D., Jones, N.S.: HCTSA: a computational framework for automated time-series phenotyping using massive feature extraction. Cell Syst. **5**(5), 527–531 (2017)
5. Chen, R.T.Q., Li, X., Grosse, R., Duvenaud, D.: Isolating sources of disentanglement in VAEs. In: Proceedings of the NeurIPS (2019)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
7. Giannakakis, G., Trivizakis, E., Tsiknakis, M., Marias, K.: A novel multi-kernel 1D convolutional neural network for stress recognition from ECG. In: Proceedings o the 2019 8th International Conference o Affective Computing Intelligent Interaction Workshops and Demos (ACIIW), pp. 1–4. IEEE (2019)
8. Li, Y., et al.: Towards learning disentangled representations for time series. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 3270–3278 (2022)
9. Giannakakis, G., Grigoriadis, D., Giannakaki, K., Simantiraki, O., Roniotis, A., Tsiknakis, M.: Review on psychological stress detection using biosignals. IEEE Trans. Affect. Comput. (2019)
10. Hinton, G.E., Zemel, R.S.: Autoencoders, minimum description length, and Helmholtz free energy. In: Proceedings of the NeurIPS (1994)

11. Rahnuma, K.S., Wahab, A., Kamaruddin, N., Majid, H.: EEG analysis for understanding stress based on affective model basis function. In: Proceedings of the International Symposium on Consumer Electronics (ISCE), pp. 592–597 (2011)
12. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: Proceedings of the International Conference on Machine Learning (ICML) (2015)
13. Parameswaran, N.S., Venkataraman, D.: A computer vision based image processing system for depression detection among students for counseling. Indones. J. Electr. Eng. Comput. Sci. **14**(1), 503–512 (2019)
14. Adams, P., et al.: Towards personal stress informatics: comparing minimally invasive techniques for measuring daily stress in the wild. In: Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare, pp. 72–79 (2014)
15. Melillo, P., Bracale, M., Pecchia, L.: Nonlinear heart rate variability features for real-life stress detection: case study of students under stress due to university examination. Biomed. Eng. Online **10**(1), 96 (2011)
16. Subramanian, R., Wache, J., Abadi, M.K., Vieriu, R.L., Winkler, S., Sebe, N.: ASCERTAIN: emotion and personality recognition using commercial sensors. IEEE Trans. Affect. Comput. **9**(2), 147–160 (2016)
17. Cohen, S., Kamarck, T., Mermelstein, R.: A global measure of perceived stress. J. Health Soc. Behav. **24**(4), 385–396 (1983)
18. Tan, C.W., Dempster, A., Bergmeir, C., et al.: MultiRocket: multiple pooling operators and transformations for fast and effective time series classification. Data Min. Knowl. Discov. **36**(5), 1623–1646 (2022)
19. Markova, V., Ganchev, T.: Constrained attribute selection for stress detection based on physiological signals. In: Proceedings of the 2018 International Conference on Sensors, Signal Image Processing, pp. 41–45 (2018)
20. Markova, V., Ganchev, T., Kalinkov, K.: CLAS: a database for cognitive load, affect, and stress recognition. In: Proceedings of the 2019 International Conference on Biomedical Innovations Application (BIA), pp. 1–4. IEEE (2019)
21. Can, Y.S., Arnrich, B., Ersoy, C.: Stress detection in daily life scenarios using smartphones and wearable sensors: A survey. J. Biomed. Inform. **92**, 103139 (2019)
22. Uday, S., Jyotsna, C., Amudha, J.: Detection of stress using wearable sensors in IoT platform. In: Proceedings of the 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), pp. 492–498. IEEE (2018)
23. Cholakov, R., Kolev, T.: The GatedTabTransformer: an enhanced deep learning architecture for tabular modeling. arXiv preprint arXiv:2201.00199 (2022)
24. Shaw, A., Simsiri, N., Deznaby, I., Fiterau, M., Rahaman, T.: Personalized student stress prediction with deep multitask network. arXiv preprint arXiv:1906.11356 (2019)
25. Sriramprakash, S., Prasanna, V.D., Murthy, O.R.: Stress detection in working people. Procedia Comput. Sci. **115**, 359–366 (2017)
26. Koldijk, S., Sappelli, M., Verberne, S., Neerincx, M.A., Kraaij, W.: The SWELL knowledge work dataset for stress and user modeling research. In: Proceedings of the 16th International Conference on Multimodal Interaction, pp. 291–296 (2014)
27. Fan, T., Qiu, S., Wang, Z., et al.: A new deep convolutional neural network incorporating attentional mechanisms for ECG emotion recognition. Comput. Biol. Med. **159**, 106938 (2023)

Long Abstracts

Modulation of Brain Activity via Deep Brain Stimulation in Parkinson's Disease Patients: Individualized EEG Based Assessment

Gustavas Davidavicius[1,2(✉)], Vytautas Kucinskas[2], Pierpaolo Sorrentino[3], Andrius Radziunas[4], Augustinas Fedaravicius[2,4], Ieva Puleikyte[4], Karolina Reinyte[2], and Ausra Saudargiene[1,2]

[1] Department of Informatics, Vytautas Magnus University, Kaunas, Lithuania
gustavas.davidavicius@vdu.lt

[2] Neuroscience Institute, Lithuanian University of Health Sciences, Kaunas, Lithuania

[3] Institut de neurosciences des systemes, University Aix Marseille Université, Marseille, France

[4] Department of Neurosurgery, Medical Academy, Lithuanian University of Health Sciences, Kaunas, Lithuania

Parkinson's disease (PD) is a progressive neurodegenerative disorder characterised by motor and non-motor symptoms, and is associated with altered brain network dynamics. Understanding large-scale functional changes is crucial for improving diagnosis, treatment planning, and monitoring of disease progression or intervention effects. Electroencephalography (EEG) combined with source reconstruction and magnetic resonance imaging (MRI) data facilitates the analysis of cortical dynamics with greater anatomical precision. The present study investigates differences in preoperative and postoperative brain activity in a PD patient using source-constructed EEG. The objective of this study is to extract and compare the connectivity and dynamical features of brain networks to identify functional alterations associated with the deep brain stimulation (DBS) treatment.

The resting state EEG dataset of PD patients was collected before and after DBS electrode implantation. The patients were treated at the Department of Neurosurgery, Lithuanian University of Health Sciences Hospital in Kaunas, Lithuania, 2024–2025. Preoperatively, EEG recordings were obtained when the patients were off L-DOPA medication, and postoperatively with the DBS electrode activated and off L-DOPA medication. Post-surgery EEGs were recorded six months later after the surgical intervention. All EEG recordings were acquired using a 64-channel ANT Neuro device. The study protocol was approved by the Kaunas Regional Biomedical Research Ethics Committee (No. BE-2-115), informed consent was obtained from every study participant. EEG preprocessing was performed using Matlab EEGLab and Makoto's Miyakoshi pipeline. Brain sources were reconstructed utilising processed EEGs in Brainstorm, using patient MRI data to create Boundary Element Method (BEM) surfaces and an individualized head model via OpenMEEG. Electrode positions

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2026
W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 455–456, 2026.
https://doi.org/10.1007/978-3-032-04555-3_37

were manually adjusted, EEG data was concatenated, and downsampled to the Desikan-Killiany atlas. Source scouts were then extracted and resegmented to the 3-second epochs. The EEG features Amplitude Envelope Correlation (AEC) and Phase Lock Value (PLV) were calculated. A Wilcoxon permutation test (1000 permutations) was used to assess the statistical significance at the edge level, focusing on connections between brain regions. The resulting p-values were MaxT corrected to control the family-wise error rate, and the effect size was calculated. All statistically significant edges exhibited an effect size greater than 0.5. The most significant regions were selected by ranking PLV and AEC metrics. The top 15 of both features were compared to find regions with most statistically significant edges. A normalised sum of all statistically significant edges per EEG band was also calculated for comparison.

The most significant changes were found in the Beta band for both AEC and PLV features, as reported in previous studies [1]. Specifically, the normalised number of the significant AEC features in the following bands were as follows: Delta: 0.00, Theta: 1.18, Alpha: 2.71, Beta: 15.82, Gamma: 5.85; the number of the PLV features were: Delta: 0.00, Theta: 1.56, Alpha: 3.53, Beta: 13.62, Gamma: 9.68. The regions with the most significant edges were the Fusiform Gyrus on both left and right side, Pericalcarine Cortex, Parahippocampal Gyrus, Supramarginal Gyrus, Temporal Pole, Pars Opercularis, Precentral Gyrus, and Superior Frontal Gyrus on the left hemisphere, and Caudal Middle Frontal Gyrus and Superior Temporal Gyrus on the right hemisphere.

The findings of this study indicate that the most significant alterations in brain connectivity occurred in the Beta frequency band for both AEC and PLV metrics [1]. In contrast, other frequency bands showed comparatively minor alterations. These results suggest that DBS-induced modulation predominantly affects the Beta band connectivity within these critical brain regions, which may have implications for understanding the neural mechanisms underlying DBS efficacy in Parkinson's disease

Acknowledgments. The authors acknowledge support from the grant "Personalised brain models of Parkinson's disease patients. Modèle de cerveau personnalisé pour les pati ents att eints de la maladie de Parkinson". Lithuanian-French programme Gilibert, project No 2024-PRO-00148/ P-LZ-24-6, 2025–2026.

References

1. Conti, A., et al.: STN-DBS induces acute changes in β-band cortical functional connectivity in patients with Parkinson's disease. Brain Sci. **12**(12), 1606 (2022). https://doi.org/10.3390/brainsci12121606
2. Liu, H., et al.: STN-DBS rapidly suppresses cortical beta synchrony in Parkinson's disease. Cereb. Cortex **32**(20), 4480–4493 (2022). https://doi.org/10.1093/cercor/bhac252

Author Index

A
Abrego, Andre Luis Severino 102
Amakor, Augustina C. 251
Ang, Elisa Y. M. 228
Artelt, André 192

B
Baldy, Nina 78
Barry, Christian Laurence Almeida 102
Becker, Jürgen 66
Bee Kiat, Tay 228
Berkemeier, Manuel B. 251

C
Cai, Yali 366
Cao, Jian 1
Cao, Xixin 1
Cao, Yi 26
Chen, Guang 1
Chen, Jiahui 390
Chen, Yuehui 26
Cheng, Samuel 51
Cheng, Xiaobo 26
Colombo, Danilo 102

D
Davidavicius, Gustavas 455
Ding, Yi 26
Ding, Zhiming 428
Dong, Hao 276
Dong, Ruihai 204
Du, Bianxia 89
Duan, Peibo 441
Duan, Shengling 239
Duan, Zhenke 339

E
El Amm, Christian 51

F
Fedaravicius, Augustinas 455
Feng, Jiali 239
Feng, Shuo 314
Feng, Xibin 415
Feng, Zhijing 228

G
Gang, Xinhe 39
Gao, Chaoqi 166
Gu, Qinying 115
Gu, Yunlong 263
Guan, Lixin 263
Guy, Tatiana Valentine 351

H
Hammer, Barbara 192
Han, Yahong 166
Hartmann, Christoph 192
He, Liang 216
He, Zhuo 378
Henderson, Paul 378
Hong, Qingqi 390
Hu, Guoqiang 141
Hu, Qiao 89
Hu, Yupeng 89
Huang, Congshan 89
Huang, Yongsheng 441
Huang, Zhao 263

I
Ikeda, Narumitsu 302

J
Jaya, Evelyn Jessica 115
Jiao, Tianshuo 89
Jirsa, Viktor K. 78

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2026

W. Senn et al. (Eds.): ICANN 2025, LNCS 16071, pp. 457–459, 2026.
https://doi.org/10.1007/978-3-032-04555-3

Jodas, Danilo Samuel 102
Ju, Shengfeng 39

K
Kang, Wenxiong 128
Kim, Jonghoon 51
Koller, Lukas 192
Kong, Jie 12
Kucinskas, Vytautas 455

L
Lee, Sophia Yat Mei 39
Li, Dongsheng 366
Li, Jiawei 263
Li, Jiguang 263
Li, Kaiheng 390
Li, Mengshan 263
Li, Piji 314
Li, Qin 326
Li, Shoushan 39
Li, Xinjie 216
Li, Xintong 51
Li, Yahui 179
Liu, Fuchun 402
Liu, Jianhui 1
Liu, Lu 415
Liu, Yingjun 402
Liu, Yu 39
Liu, Zhandong 289

M
Martins, Guilherme Brandão 102
Mazur, Andreas 192

N
Niu, Yingjie 204

P
Pan, Jiqun 339
Papa, João Paulo 102
Papastathopoulos-Katsaros, Athanasios 289
Peitz, Sebastian 251
Peters, Henning 192
Poti, Valerio 204
Pugeault, Nicolas 378
Puleikyte, Ieva 455

Q
Qi, Yue 239
Qiao, Peng 366
Qu, Site 141

R
Radziunas, Andrius 455
Reinyt, Karolina 455
Ružejnikov, Jurij 351

S
Sandamirskaya, Yulia 66
Santana, Marcos Cleison 102
Saudargiene, Ausra 455
Scholz, Jona 66
Sextro, Walter 251
Shao, Mingyu 390
Shen, Shengjie 179
Shen, Tongfei 154
Shu, Lin 128
Sorrentino, Pierpaolo 455
Stavrianidi, Alexandra 289
Sui, Ziheng 1

T
Tan, Wenjun 415
Teng, Zixuan 154
Trächtler, Ansgar 192
Tu, Jiani 339

V
Vasilache, Alexandru 66

W
Wai Heng, Koh 228
Wang, Jincen 12
Wang, Liejun 179
Wang, Peng Cheng 228
Wang, Xinbing 115
Wang, Yizhou 276
Wang, Yuan 1
Wei, Xuan 390
Wohlleben, Meike 251
Woodman, Marmaduke M. 78
Wu, Mingdong 276

X
Xia, Yulong 276
Xiao, Shanlin 12

Xu, Hongbin 128
Xu, Meng 1, 263
Xu, Wei 415
Xu, Xinrun 428

Y
Yan, Jin 428
Yan, Zhixin 128
Yang, Chuanfeng 390
Yang, Jianwen 428
Yang, Yichen 39
Ye, Nanyang 115
Yi, Shangru 128
Yu, Yinfeng 179
Yuan, Ye 415

Z
Zeng, Qingwei 441
Zhang, Boyuan 166
Zhang, ChangSheng 441
Zhang, Dong 154
Zhang, Haoliang 51
Zhang, Nana 326
Zhang, Qiuhong 428
Zhang, Shuo 428
Zhang, Wenjun 128
Zhang, Xing 1
Zhang, Ziheng 390
Zhao, Mingchuan 204
Zhao, Yan 12
Zhong, Fangwei 276
Zhong, Yan 128
Zhou, Xian 39
Zhou, Yangyi 12
Zhu, Dandan 326
Zhu, Deli 216
Zhu, Hongzhen 402
Zhu, Jie 12
Zhu, Kun 326
Zhu, Meiling 428
Zong, Yuan 12
Zou, Chenglou 1