Towards an Efficient Synergistic Paradigm for Self-supervised Visual Representation Learning

Tarun Krishna, MSc

Supervised by Prof. Noel E. O'Connor, Dr. Kevin McGuinness

A Dissertation submitted in fulfilment of the requirements for the award of Doctor of Philosophy (Ph.D.)

School of Electronic Engineering
Dublin City University

September 2024

Declaration

I hereby certify that this material, which I now submit for assessment on the program of study leading to the award of PhD is entirely my own work, and I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: Tarun Krishna ID No.: 19215253 Date: 09/09/2024

Acknowledgements

"Research excellence emerges from collaborative synergy, not individual brilliance."

Firstly, I would like to extend my deepest appreciation to my supervisors, Dr. Kevin McGuinness and Prof. Noel O'Connor, for their invaluable guidance and unwavering support throughout this journey. Kevin, your patience and insight in navigating the complexities of machine learning have been truly invaluable. Your mentorship has been instrumental in shaping my growth and your wisdom will resonate with me for years to come. Noel, I am grateful for your exceptional contributions and stead-fast encouragement throughout my studies. Beginning my PhD against the backdrop of the COVID-19 pandemic presented numerous challenges, particularly as I found myself in a new country facing an impending lockdown. However, I am immensely grateful for your consistent accessibility, including our weekly Zoom calls and your round-the-clock availability.

I would like to thank Ayush Rai and Yasser Dahou for being my closest collaborators and friends. I am particularly grateful for our fruitful discussions, which have led to many interesting ideas. I would like to thank my lab mates Eric Arazo, Paul Albert, Enric Moreu and Luis Lebron for being available all the time for any kind of discussion, feedback, and guidance.

I would like to thank Prof. Alan Smeaton for always taking out time in our regular Friday meetings and being very insightful about the nitty-gritty details that one often never realizes. Thanks for being my "*other supervisor*". Further, I would like to thank Alex Drimbarean (Xperi now Tobii) for helping and guiding me throughout my stay at Xperi as a part of industrial collaboration. I am grateful for your invaluable support.

I would like to thank my parents for their constant support and encouragement. Finally, I would like to thank my wife for her love, unconditional support and her daily patience.

Last but not least, I would like to dedicate this thesis to the memory of Dr. Kevin McGuinness, whose contributions will always be remembered. This work stands as a testament to his enduring influence and our deep appreciation for his invaluable contributions.

Contents

List of Figures

List of Tables

Notations

AQE ..... Average Query Expansion

CE ..... Cross Entropy

CNNs ..... Convolutional Neural Networks

DBA ..... Database-side Augmentation

DC ..... Dynamic Computation

DNNs ..... Deep Neural Networks

FC ..... Fully Connected

FLOPs ..... Floating point operations per second

GANs ..... Generative Adversarial Networks

GD ..... Gradient Descent

k-NN ..... k-Nearest Neighbors

KD ..... Knowledge Distillation

MLP ..... Multi-Layer Perceptron

MSE ..... Mean Squared Error

NN ..... Neural Network

NCE ..... Noise Contrastive Estimation

ODI ..... Omnidirectional Images

PCA ..... Principal Component Analysis

SGD ..... Stochastic Gradient Descent

SSL ..... Self-Supervised Learning

SOTA ..... State-of-the-art

ReLU ..... Rectified Linear Unit

R-MAC ..... Regional Maximum Activation of Convolution

List of Publications

Publications arising directly from this thesis

- **Tarun Krishna**, Ayush Rai, Alexandru Drimbarean, Alan F. Smeaton, Kevin McGuinness and Noel E. O'Connor. "Unifying Synergies between Self-supervised Learning and Dynamic Computation." British Machine Vision Conference (2023).

- **Tarun Krishna**, *Ayush K. Rai, Yasser AD Djilali, Alan F. Smeaton, Kevin McGuinness, and Noel E. O'Connor. "Dynamic Channel Selection in Self-Supervised Learning." In 24th Irish Machine Vision and Image Processing Conference. 2022.

- *Djilali, Yasser Abdelaziz Dahou, ***Tarun Krishna**, Kevin McGuinness, and Noel E. O'Connor. "Rethinking 360° Image Visual Attention Modelling with Unsupervised Learning." In 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 15394-15404. IEEE, 2021.

- **Tarun Krishna**, Kevin McGuinness, and Noel O'Connor. "Evaluating contrastive models for instance-based image retrieval." In Proceedings of the 2021 International Conference on Multimedia Retrieval, pp. 471-475. 2021.

Other publications by the author

- Albert, Paul, Jack Valmadre, Eric Arazo, **Tarun Krishna**, Noel E. O'Connor, and Kevin McGuinness. "An accurate detection is not all you need to combat label noise in web-noisy datasets." arXiv preprint arXiv:2407.05528 (2024) (*Accepted in ECCV 2024*).

- Rai, Ayush K., ***Tarun Krishna**, *Feiyan Hu, Alexandru Drimbarean, Kevin McGuinness, Alan F. Smeaton, and Noel E. O'Connor. "Video Anomaly Detection via Spatio-Temporal Pseudo-Anomaly Generation: A Unified Approach."

*Equal contribution.

In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3887-3899. 2024.

- Rai, Ayush K., **Tarun Krishna**, Julia Dietlmeier, Kevin McGuinness, Alan F. Smeaton, and Noel E. O'Connor. "Motion Aware Self-Supervision for Generic Event Boundary Detection." In 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 2727-2738. IEEE, 2023.

- Albert, Paul, Eric Arazo, **Tarun Krishna**, Noel E. O'Connor, and Kevin McGuinness. "Is your noise correction noisy? PLS: Robustness to label noise with two stage detection." In 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 118-127. IEEE Computer Society, 2023.

Abstract

"Towards an Efficient Synergistic Paradigm for Self-supervised Visual Representation Learning"

Tarun Krishna

This thesis investigates the latest developments in self-supervised representation learning, which enables learning from a large un-labelled data corpus. The overarching objective of this work is to comprehensively assess, devise and harness self-supervised models with efficiency and effectiveness at the forefront. Taking an initial step in this direction, this research begins by evaluating the efficacy of contrastive models for instance-based image retrieval, demonstrating their capability to encode semantic similarity among instances induced through discriminative learning. Through extensive evaluation on Oxford5k/Oxford5k, Paris6k/rParis6k and INSTRE, it is shown that these models perform comparably with, and in some cases outperform pre-trained supervised baselines, highlighting their potential for building robust image retrieval engines without explicit supervision. Building upon this foundation, this work further delves into the realm of 360° image visual attention modeling, a domain largely unexplored in the context of self-supervised representation learning. More importantly, the solutions proposed for learning have been validated in realistic benchmarks (Salient 360 [Rai et al., 2017], VR-Eye Tracking, Sitzmann) built with datasets gathered from the Web. Further, contributions are made towards optimizing self-supervised learning strategies, particularly addressing challenges such as redundant channel features and computational complexity. Dynamic channel selection methods originally developed for supervised learning are adapted to self-supervised networks, resulting in significant reductions in computation without compromising performance. Additionally, a novel perspective is introduced on the synergy between self-supervised learning and dynamic computation paradigms. Through simultaneous learning of dense and gated sub-networks, a generic and efficient architecture is proposed, achieving comparable performance to vanilla self-supervised settings but with reduced computational costs. These approaches are rigorously benchmarked on the CIFAR-10/100, STL-10 and ImageNet-100 datasets. Finally, the conclusion of this thesis summarizes the contribution of this work and discusses some thoughts on directions for future research in this area.

Chapter 1

Introduction

1.1 Motivation

The introduction of Deep Neural Networks (DNNs) has led to significant improvements in performance for numerous visual analysis tasks in the field of computer vision. After the advent of AlexNet [Krizhevsky et al., 2012], which achieved outstanding results on the ImageNet Large Scale Visual Recognition Challenge [Russakovsky et al., 2015], DNNs became the building blocks for most visual analysis tasks. This was enabled by the availability of large computation resources and large corpora of labelled data. The growth in the availability of benchmark image datasets such as MNIST [LeCun et al., 1998], CIFAR [Krizhevsky and Hinton, 2009], SVHN [Netzer et al., 2011], COCO [Lin et al., 2014] and ImageNet [Russakovsky et al., 2015] facilitated significant advancement in supervised learning by the research community. Furthermore, such enormous labelled datasets led to innovations in the architectural design of DNNs, resulting in architectures such as VGG [Simonyan and Zisserman, 2015], ResNet [He et al., 2016], InceptionNet [Szegedy et al., 2015], ViTs [Dosovitskiy et al., 2021] etc. A consequence of all such research is consistent performance improvements for various visual analysis tasks. However, the rate of progress has slowed considerably in recent years, suggesting a saturation point may have been reached – for example, see Figure 1.1 for the annual trend in top-1 accuracy performance on ImageNet where this slowdown is visible. This performance saturation is because the labeled data is expensive and labour intensive. Furthermore, labelling data for one particular visual analysis task, often means that algorithms developed on this basis suffer from poor generalization.

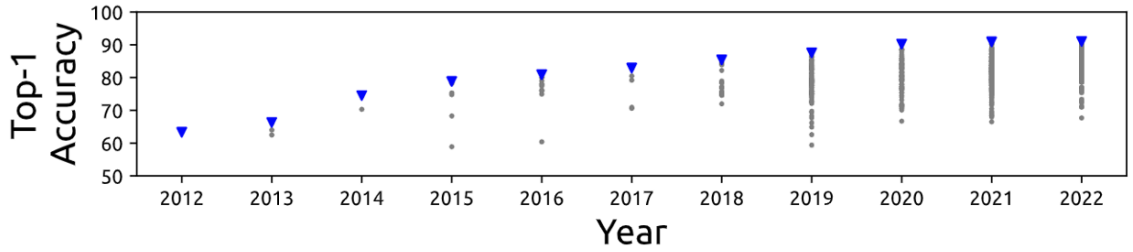
Of course, in practical scenarios, not all datasets have sufficient labelled training data to ensure the level of performance required. Transfer learning was thus introduced to address this challenge and facilitate the use of DNNs on smaller datasets, and it quickly became the primary method for transferring knowledge across image

datasets. Transfer learning [Zhuang et al., 2020] enables using DNNs on smaller datasets by leveraging features extracted from larger datasets. However, models trained this way tend to be brittle and sensitive to minor changes in the data [Jain et al., 2023] due to the reliance on supervised pre-training. Besides requiring large amounts of labelled data based on human (domain) expertise, supervised learning models can suffer from overfitting, especially when the training data sets are small or noisy. This can lead to poor *generalization* on unseen data. A similar issue arises when there is a data/class imbalance in the dataset [Menon et al., 2020].

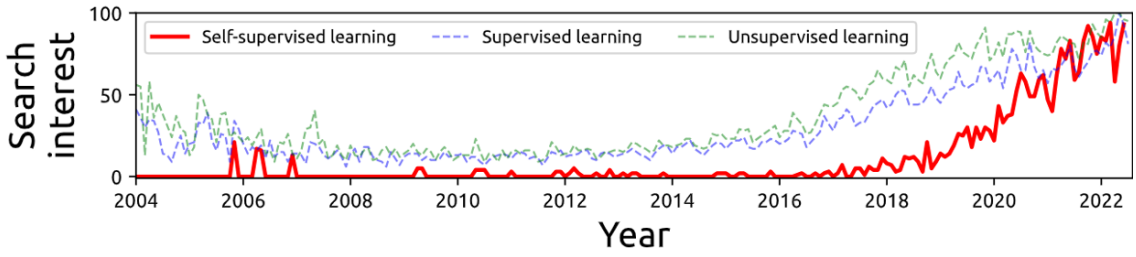
All these considerations indicated that new approaches were required to overcome the limitations of supervised learning. To this end, numerous studies explored unsupervised learning, which aims to enable robust feature extraction by training models without labeled data [Netzer et al., 2011]. Unfortunately, the results of these methods on visual analysis tasks have been underwhelming [Noroozi and Favaro, 2016, Pathak et al., 2016a]. Meanwhile, self-supervised learning (SSL) methods were being used in the field of Natural Language Processing (NLP) achieving state-of-the-art results and outperforming traditional supervised learning techniques [Devlin et al., 2018, Mikolov et al., 2013] in tasks like machine translation. Most recently, Large Language Models (LLMs) trained on web-scale large text corpora resulting in models like `BERT` [Devlin et al., 2019], `GPT`, and their variants [Zhao et al., 2023], have revolutionized both the performance and our current understanding of Artificial Intelligence (AI). The powerful performance of these models is in part due to the abundance of unlabelled text data, such as online websites and books, where SSL can be immediately exploited.

As a result, SSL started to be applied in computer vision tasks as a method for extracting robust features from unlabeled data using the properties of images, thereby avoiding the challenges associated with labeled data. SSL as the name suggests attempts to model unknown distributions (not making any assumptions about the data distribution) without relying on true annotations. As such, this learning paradigm potentially provides data efficiency and better generalization ability [Huang et al., 2022] compared to its supervised counterpart. The emergence of self-supervision has resulted in approaches that obtain state-of-the-art performance on numerous computer vision benchmarks [Chen et al., 2020b, Goyal et al., 2022].

SSL is a machine learning paradigm where a model learns from the data without explicit supervision in the form of labels/annotations. Instead, the learning algorithm generates its own supervisory signal from the input data. The key idea is to design *on the go* tasks, referred to as *objectives* or *pre-text* tasks, which can be derived from the data itself, eliminating the need for externally labeled datasets. In NLP, for example, the skip-gram algorithm [Mikolov et al., 2013] is based on a pretext task of predicting the context words (words surrounding a target word) given a



(a) ImageNet top-1 accuracy



(b) Interest over time for different learning paradigms

Figure 1.1: Performance trend (measured with Google trend) on ImageNet for supervised learning over the years 2012-2022. Figure source [Ozbulak et al., 2023].

target word. This objective of predicting the surrounding words from a given target word captures relationships among the words in large text corpora without the need for any explicit supervision. In computer vision, equivalent objectives exist for learning meaningful representations e.g., context prediction [Doersch et al., 2015], solving a jigsaw-puzzle [Noroozi and Favaro, 2016], colorization [Zhang et al., 2016a], image inpainting [Yang et al., 2017] to name but a few. However, most current research into SSL investigates joint-embedding architectures for mapping different views of the same image to a similar representation e.g., contrastive learning where the goal is to bring the image and its augmentation closer while contrasting it with other images (see Chapter 2 for a detailed description). As a result, a large family of generative models like auto-encoders [Kingma and Welling, 2014], generative adversarial networks (GANs) [Goodfellow et al., 2014], diffusion models [Song and Ermon, 2019] etc., have also come to prominence under the umbrella of generative SSL methods.

The success of SSL methods is based on the availability of large corpora of unlabeled data sets on the internet. This, coupled with the availability of large computational resources making it easy to scale the model size, has led to a surge in large deep networks [Oquab et al., 2023, Goyal et al., 2022]. However, there are some practical challenges hinder its widespread application and effectiveness:

- *Training efficiency*: Longer training times due to the need for generating

pseudo labels, which can require complex pretext tasks. Additionally larger model size leads to higher inference cost [Chen et al., 2021a].

- *Computational Resources*: Self-supervised learning can demand substantial computational resources, especially when processing large datasets. This poses accessibility challenges for researchers and practitioners who may not have access to high-performance computing environments. Efforts are ongoing to develop more efficient algorithms that can reduce resource requirements while maintaining performance

- *Model Collapse*: Self-supervised models can suffer from model collapse, where the model generates identical or nearly identical embeddings for different inputs. Without proper augmentation or loss term balancing, models might optimize for trivial solutions rather than meaningful distinctions.

- *Need for Large, Diverse Unlabeled Data*: SSL models thrive on diverse, high-quality datasets, which may not always be readily available. Datasets with biased or limited variability can lead to suboptimal feature learning.

Notwithstanding this, the approach is attractive as self-supervised large deep neural networks pre-trained on large-scale datasets serve as a general-purpose feature extractor and the common assumption is that this cost of pre-training can be amortized by transferring it to various downstream tasks. However, fine-tuning such large pre-trained models is also computationally expensive. Furthermore, downstream tasks are diverse and may vary widely and fine-tuning for each task makes the entire process tedious and cumbersome.

The study conducted in this thesis utilizes convolutional neural networks (CNNs) based deep networks as the base model instead of relying on vision transformers. There is no doubt transformers have revolutionized computer vision but due to computational complexity* it is difficult to train these models from scratch hence making it less accessible to small organization.

In this thesis, we address some of these considerations with a view to advancing the understanding of SSL as a paradigm for visual representation learning. Our objective is to comprehensively assess self-supervised models, devise new approaches and harness them in downstream applications, where efficiency and effectiveness are the overarching primary design considerations.

*The self-attention mechanism scales quadratically with the sequence length, leading to high memory and computational costs for long sequences.

1.2 Hypotheses and Research Questions

As explained above, self-supervised representation learning has emerged as a promising alternative to supervised representation learning due to its ability to learn underlying factors of variation from a large unlabelled data corpus. The large pre-trained self-supervised models obtained as a result, can serve as a starting point for different downstream tasks i.e., they can be exploited as a pre-trained (fixed) feature extractor, that can be used for transfer learning where fine-tuning is performed on the limited annotated dataset available for a given visual analysis task. The research reported in this thesis, focuses on self-supervised visual representation learning and proposes novel advances towards effectively exploiting limited resources to obtain maximal performance. Performance in this context means performance as measured by the generally accepted metrics for a given visual analysis task e.g. classification accuracy for recognition tasks, Intersection over Union (IOU) for object detection, etc..

In this thesis, we explore SSL through two important downstream visual analysis tasks. The first of these is instance retrieval, where the relevancy of image retrieval results is defined in terms of instances of the same object. This is a fundamental task in computer vision. It allows for searching and retrieving images based on their visual content, such as textures, colors, viewpoints, illumination, and shapes, rather than relying on manual annotations or metadata. The task itself serves as a proxy for several different tasks ranging from object recognition to scene understanding [Jing et al., 2015, Philbin et al., 2007], as it helps computers understand the context and relationships between objects in a scene, enabling applications like scene reconstruction, 3D modelling, and virtual reality.

We choose visual saliency prediction, i.e. predicting where people look in images and videos, as the second fundamental task. Saliency prediction is useful in a wide variety of applications across several domains (e.g. neuroscience, assistive systems, human-computer interaction). Some example applications where saliency prediction can be applied include gaze-aware video compression and summarization, activity recognition, object segmentation, object recognition and detection, image captioning, question answering, medical image processing, and surveillance [Borji, 2018]. Like instance retrieval, this task can also serve as a surrogate for other tasks [Le,]. Saliency prediction has been extensively investigated for 2D images and video, but its effectiveness for omnidirectional (360°) image data has received less attention. Applications leveraging 360° image data are growing in popularity and image saliency prediction techniques could substantially enhance this. For example, accurate and quick saliency detection for 360° image data could lead to advances in object detection, semantic segmentation, viewport prediction, etc in 3D applications.

Salience prediction for 360° data is an extremely challenging task compared to its 2D equivalent due to the use of head-mounted displays. However, it seems intuitive that the geometry of the image data captured in such a setting (i.e. different views of the same environment) would lend itself to SSL.

As outlined above, there is also a great need to explore more efficient inference methods on downstream tasks for SSL. In the supervised learning setting, as described in [Veit and Belongie, 2018], it is accepted that networks with dynamic data-dependent (conditional) channel computation architectures during inference can lead to enhanced representation power, adaptivity, and interpretability and can greatly reduce computation cost and memory resources without compromising on accuracy. This motivates us to investigate the behaviour of neural networks with a channel selection mechanism trained under self-supervision with some budget constraints in terms of FLOPs (Floating point operations per second).

To lessen the computational burden, it is common practice to extract (or learn) a lightweight network from an off-the-shelf pre-trained model. This has been successfully achieved through techniques such as knowledge distillation (KD) [Hinton et al., 2015], pruning [Frankle and Carbin, 2018] and the aforementioned dynamic computation (DC) [Veit and Belongie, 2018]. These approaches are effective but using fine-tuning to obtain a sub-network from large pre-trained models (such as Large Language Models) can be computationally expensive and cumbersome. Also, since downstream tasks are diverse and vary widely, any change in the task requires repeating the entire procedure multiple times, making it inefficient and less transferable. This motivates us to investigate whether both dense and sparse networks can be learned at the same time.

To this end, we follow a synergistic approach by designing a learning algorithm considering the downstream task or the objective (goal e.g. efficient inference) while amalgamating an SSL objective to achieve the desired objective. As a result, these four areas investigated in the context of SSL give rise to the following hypotheses and associated research questions that are explored in this thesis:

- **Hypothesis 1 (H1)**: Several *state-of-the-art* (SOTA) contrastive self-supervised models such as `SimCLR` [Chen et al., 2020a] and `MoCo` [He et al., 2020], have demonstrated that unlabelled data can be exploited in pre-training. These contrastive SSL models are learned by a joint embedding architecture i.e., mapping augmentation of the same image into a similar representation via instance discrimination. **We hypothesize that models that are trained to encode semantic similarity among instances via discriminative learning should perform well on the task of image instance retrieval.**
  - The above hypothesis leads us to our first research question (**RQ1**):

How effectively do contrastive SSL methods encode semantic identity in comparison to SL methods for the task of image instance retrieval?

- **Hypothesis 2 (H2)**: While self-supervised representation learning has shown success with 2D images, its application to 360° images remains under explored. **We hypothesize that omnidirectional images are particularly suited to an SSL approach due to the geometry of the data domain.**
  - This gives rise to our second research question (**RQ2**): *Can we efficiently design and effectively exploit contrastive self-supervised methods for a granular task like visual saliency prediction task for* 360° *omnidirectional images?*
- **Hypothesis 3 (H3)**: There is a key need to reduce the computational burden of SSL during training. **We hypothesize that self-supervised models are an ideal candidate for dynamic network structures as they capture highly redundant channel features during pre-training that can be removed to reduce computational load.**
  - This leads to our third research question (**RQ3**): *Do self-supervised models learn highly redundant channel features, so that important channels can be dynamically selected and the unnecessary ones removed?*
- **Hypothesis 4 (H4)**: Ideally some level of flexibility with respect to computational load is required when designing SSL networks. **We hypothesize that the Siamese setting can be utilized for simultaneously training an encoder that can serve the purpose of a dense encoder as well as a sparse lightweight encoder.**
  - This gives rise to our final research question (**RQ4**): *Is it possible to learn a single encoder (function) that could serve the dual purpose of being used as a dense and lightweight network with minimal additional overhead during pre-training?*

1.3 Thesis Structure

This thesis is structured as follows. Chapter 2 presents the the necessary technical background for understanding the research presented in the thesis and also provides a high-level overview of related work.

Chapter 3, investigates **RQ1**, measuring the semantic encoding capability of *contrastive* self-supervised models for the task of image instance retrieval for pre-trained models. We evaluate different contrastive models across different image

retrieval benchmark datasets across various settings to understand the feasibility of self-supervised (contrastive) pre-trained models for the task of instance retrieval. This chapter concludes by demonstrating that the learned representations are indeed well suited to the task and perform on par with supervised pre-trained models.

Chapter 4 explores **RQ2**. It builds upon the findings of Chapter 2 but here the goal is to utilize self-supervised models for the the tasks of saliency prediction. The goal is to design an effective way to utilize self-supervised (contrastive) learning to capture salient regions in 360° images. In this study, we extend recent advances in contrastive learning to learn latent representations that are sufficiently invariant so as to be highly effective for omnidirectional saliency prediction tasks.

Chapter 5 takes cues from Chapter 4 where it is observed that a relatively lightweight decoder could be used for saliency prediction, implying faster inference during test time. This finding motivates an investigation into the behavior of neural networks with a channel selection mechanism trained under self-supervision thereby addressing **RQ3**.

In Chapter 6, we investigate **RQ4** by designing efficient inference methodologies for self-supervised learning. We exploit the symmetric Siamese setting to explore the possibility of learning a light dynamic model along with a dense model. The end result of this training paradigm would be a dense SSL pre-trained model along with a dynamic network for efficient inference.

Finally, Chapter 7 provides a summary of the research conducted in this thesis. The results and findings are discussed by relating them to the hypotheses and research questions presented in this Chapter. We conclude with some suggestions for possible future work and some general remarks.

Chapter 2

Preliminaries and Background

This chapter provides the necessary theoretical and technical background for understanding the research reported in the remainder of the thesis. The chapter introduces technical intricacies essential for understanding self-supervised representation learning. In certain sections, a deeper exploration is provided into the underlying mathematical aspects, along with a comprehensive survey of relevant literature, all to enhance the reader's understanding of the work presented in subsequent chapters.

2.1 Introduction to Self-supervised Learning

Self-supervised learning exploits the underlying structure of the data, instead of relying on a manual supervisory signal (labeled training data) as used in supervised learning*. Such methods create an encoder f_θ by performing supervised learning on a *target* task specifically created for each given input $\mathbf{x}$. This *target* task is referred to as *pre-text* task. It is either a hand-designed task or a metric-based learning task used to understand the data. Let $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_{|D|}]$ be a set of un-labeled images (or a batch of images), where $\mathbf{x}_i \sim \mathbb{R}^{3HW}$, and $\mathcal{T}$ be set of transformations such that $t \sim \mathcal{T}$, $\mathbf{x}_i^t = t(\mathbf{x}_i)$. The set $\mathcal{T}$ contains standard image transformations, specifically small random crops, random jitter introduced into the color space, random conversion to gray-scale, and random horizontal flips. Let f_θ be the encoder function with learnable parameters $\boldsymbol{\theta}$. We denote the output from the encoder function as $\mathbf{Y} = [y_1, ..., y_n] = [f_\theta(\mathbf{x}_1), ..., f_\theta(\mathbf{x}_{|D|})]$. Self-supervised modules often comprise of projection and prediction networks g_ϕ and q_ψ respectively, which are used during pre-training*, after pre-training these networks are discarded and only f_θ is used for the

*Self-supervised labels are generated and designed on the go while supervised labels are provided externally.

*Pre-training is the process of training a model on a large dataset in order to learn general features before fine-tuning it for specific tasks. This pre-training stage could be supervised or self-supervised (unsupervised).

downstream tasks. Downstream tasks involve applying the outputs or representations from a pre-trained model to specific applications or problems.

Thus, SSL relies on generating pretext tasks for supervised learning, based on the belief that features learned while solving these pretext tasks prove beneficial in addressing various other downstream problems. In the following, we outline some of the most popular approaches to SSL under this learning paradigm.

2.1.1 Hand-designed SSL objectives

Images themselves provide several cues that can be used to design self-supervised objectives. *Image colorization* has been studied by [Larsson et al., 2017, Larsson et al., 2016, Zhang et al., 2016b] for self-supervised representation learning, however colorization as a pre-text task was proven to be too simple for DNNs. *In-painting*, ie. predicting the missing part of an image, has also been explored. For example, the authors of [Pathak et al., 2016b] proposed an approach leveraging context encoders that aim at in-painting large parts of images that are missing, forcing models to learn the image context. Similarly, geometric transformations of an image can also serve as an important cue for designing good learning objectives. Inspired by this, the authors of [Komodakis and Gidaris, 2018] proposed *RotNet* where the goal was to predict the rotation performed on an input image. The authors demonstrate that even a modest number of rotations proves adequate for acquiring a high-quality visual representation, where optimal outcomes are obtained when employing four rotations (0°, 90°, 180°, and 270°). Likewise, the authors of [Noroozi and Favaro, 2016] proposed a *Jigsaw* classification task. However, instead of rotating the image, the transformation consists of randomly permuting several patches of the image, like a jigsaw puzzle, where the model is then required to predict the class of the permutation that was used to shuffle the patches. Similarly, the authors of [Doersch et al., 2015] proposed the task of *context prediction*, where the goal is to correctly predict the nearby patch while given a context patch.

2.1.2 Discriminative SSL-methods

Discriminative self-supervised learning (SSL) frameworks can generally be categorized based on their use of the following techniques: clustering, contrastive learning, distillation, and information maximization. In the following, we explain each of these approaches.

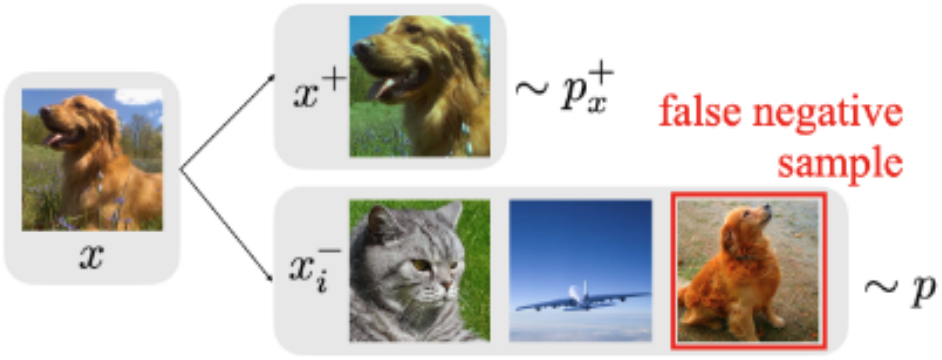


Figure 2.1: An example to demonstrate positive and negative samples for contrastive learning. Figure source [Chuang et al., 2020].

Clustering-based Methods

Clustering [Bishop, 2007] is one of the most popular methods for unsupervised learning. The objective of clustering-based representation learning is to group images with similar representations into a cluster. The objective is tractable, but it does not scale well with the dataset as it requires a pass over the entire dataset to form image "codes" (i.e., cluster assignments/pseudo labels) that are used as targets during training. Furthermore, designing end-to-end clustering-based objectives is not straightforward as it could lead to several issues such as empty clusters, learning trivial solutions (all images belonging to a single cluster), etc., prior knowledge of the number of clusters also plays a crucial role. Early works [Coates et al., 2011, Yang et al., 2016] paved the way for the adoption of the cluster-based objective for SSL, however, it was only when the authors of [Caron et al., 2018] proposed `Deep-Cluster` that clustering-based approaches came to the fore. Engineering tricks (e.g., automatically reassigning clusters during training to avoid empty clusters) were used to avoid the limitations mentioned above. However, the authors in [Asano et al., 2019], proposed `SeLa` as a more principled way of casting the problem of cluster assignment into an optimal transport problem. `SeLa` tackles the issue of model collapse by incorporating a more principled loss using the Sinkhorn-Knopp algorithm (Cuturi, 2013). A variant, `SwAV` was later proposed by [Caron et al., 2020a] that takes advantage of contrastive methods without requiring to compute pairwise comparisons. It is considered to be the most stable and accurate approach for clustering in SSL.

Contrastive Representation Learning

As the name suggests, contrastive approaches minimize the distance between positive samples while maximizing the distance between negative samples in the joint embedding space. This can be also understood as instance discrimination (i.e., learn

similarities between data by attracting positive samples and repelling negative ones), Figure 2.1. For vision tasks, the positives could, e.g., be random transformations of the same image (also referred as the anchor image), while the negatives are any other images, as depicted in Figure 2.1. The idea of contrastive learning is not new, it can be traced back to [Chopra et al., 2005], which presented one of the earliest training objectives for deep metric learning in a contrastive fashion. However, it has been popularised recently by [Wu et al., 2018b] and [Oord et al., 2018] through the introduction of `InstDic` and `CPC`. Consequently, based on the principle of maximizing mutual information (MI). The mutual information (MI) of two random variables is a measure of the mutual dependence between the two variables. More specifically, it quantifies the "amount of information" (in units such as shannons (bits), nats or hartleys). Maximization is a technique used in representation learning to train feature extractors by maximizing an estimate of the mutual information between different views of the data. [Hjelm et al., 2019] and [Bachman et al., 2019] proposed `DIM` and `AMDIM` respectively, and later `CMC` [Tian et al., 2019b] that build upon this notion of MI maximization and extends it to an arbitrary collection of views (as well as multiple sensory inputs). However, it was an approach named `MoCo` [He et al., 2020] which used delayed weight updates (momentum update) along with an efficient way of sampling negative samples that increased the popularity of contrastive SSL. Shortly after, [Chen et al., 2020a] introduced `SimCLR` with a much simpler framework that further improved the state-of-the-art (SOTA) performance by introducing projection layers and strong augmentation. Enhancements proposed in `SimCLR` became the foundations for many other SSL frameworks. Soon after, incorporating the finding in `SimCLR` authors [Chen et al., 2020c] introduced `MoCo-v2` that further improved the SOTA performance for image recognition and further, [Chen et al., 2021b] introduced a third version of `MoCo` exploring the usage of vision transformers as backbones. The simple design of `SimCLR` and `MoCo` became the foundations for several subsequent contrastive SSL frameworks. Learning under an SSL paradigm may suffer from dimension/representation collapse* [Jing et al.,], however in contrastive learning this is mitigated by utilizing negative samples. Contrastive SSL frameworks can suffer from another type of collapse, namely dimensional collapse, wherein representations collapse into a low-dimensional feature space (manifold) [Hua et al., 2021]. Avoiding network/representation collapse continues to be an active area of research in SSL – the reader is referred to [Le-Khac et al., 2020] for a more detailed review of contrastive learning.

The loss function used in contrastive learning (the contrastive loss) is derived from Noise Contrastive Estimation (NCE) [Gutmann and Hyvärinen, 2010] and its

*Network collapse is a phenomenon that occurs in self-supervised learning, where the network learns to map all input samples to a single point or a very small region in the representation space.

modifications. The idea is to us logistic regression to discriminate the target data from noise (as the negative samples). Let $\mathbf{x}$ be the target sample $\sim\ P(\mathbf{x}|C=1;\theta) = p_\theta(\mathbf{x})$ and $\tilde{\mathbf{x}} \sim\ P(\tilde{\mathbf{x}}|C=0) = q(\tilde{\mathbf{x}})$ be the noise sample. Note that the logistic regression models the logit (i.e. log-odds) and in this case the goal is to model the logit of a sample from the target data distribution instead of the noise distribution:

$$\ell_\theta(\mathbf{u}) = \log \frac{p_\theta(\mathbf{u})}{q(\mathbf{u})} = \log p_\theta(\mathbf{u}) - \log q(\mathbf{u}), \tag{2.1}$$

After converting logits ($\mathbf{u} = f_\theta(\mathbf{x})$) into probabilities with sigmoid $\sigma(.)$, the binary cross entropy loss can be applied:

$$\begin{aligned} \mathcal{L}_{\text{NCE}} &= -\frac{1}{N}\sum_{i=1}^{N}\big[\log\sigma(\ell_\theta(\mathbf{x}_i)) + \log(1-\sigma(\ell_\theta(\tilde{\mathbf{x}}_i)))\big] \\ \text{where } \sigma(\ell) &= \frac{1}{1+\exp(-\ell)} = \frac{p_\theta}{p_\theta + q} \end{aligned} \tag{2.2}$$

In the above, the loss is applied for a single negative sample, but it can be easily extended to multiple negative samples as well, e.g. see the generalization of this loss in Chapter 4 for use with multiple negative samples. Based on NCE, `InfoNCE` uses categorical cross-entropy loss to identify the positive sample among a set of unrelated noise samples [Chen et al., 2020a]. InfoNCE is defined for $2n$ instances of images from a given n instances in a batch $\mathcal{B} = [t(\mathbf{x_1}), t(\mathbf{x}_1), ..., t(\mathbf{x}_n), t(\mathbf{x}_n)]$, where $t \sim \mathcal{T}$ is a set of random transformation samples from the set of transformations $\mathcal{T}$:

$$\mathcal{L}_{\text{InfoNCE}}(\mathbf{x}_{i,j}) = -\log \frac{\exp(\text{sim}(\mathbf{r}_i, \mathbf{r}_j))}{\sum_{m=0}^{2n} \mathbb{1}_{m\neq i}\exp(\text{sim}(\mathbf{r}_i, \mathbf{r}_m))} \tag{2.3}$$

where $\text{sim}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i.\mathbf{x}_j}{||\mathbf{x}_i||\ ||\mathbf{x}_j||}$, which is also referred to as *cosine similarity* and $\mathbf{r}_i = g_\phi(f_\theta(\mathbf{x}_i))$.

Information Maximization Methods

To tackle dimension collapse, the authors of [Zbontar et al., 2021, Bardes et al., 2022, Ermolov et al., 2021] proposed different ways to maximize the information in the representation. These techniques prioritize the information richness of embeddings, aiming to prevent the loss of diverse representations. The central theme for these methods is to infer the relationship between two variables by analyzing their cross-covariance matrices. Figure 2.2 depicts a general pipeline for such approaches, however, not all the components as depicted in Figure 2.2 may be required. To be precise it is likely that both branches are not symmetric at all i.e., one branch processes input until predictor layer ($\mathbf{q}_\Psi$), while the other branch processes input until projector layer ($\mathbf{q}_\phi$), also both the branches may not be sharing weights at all

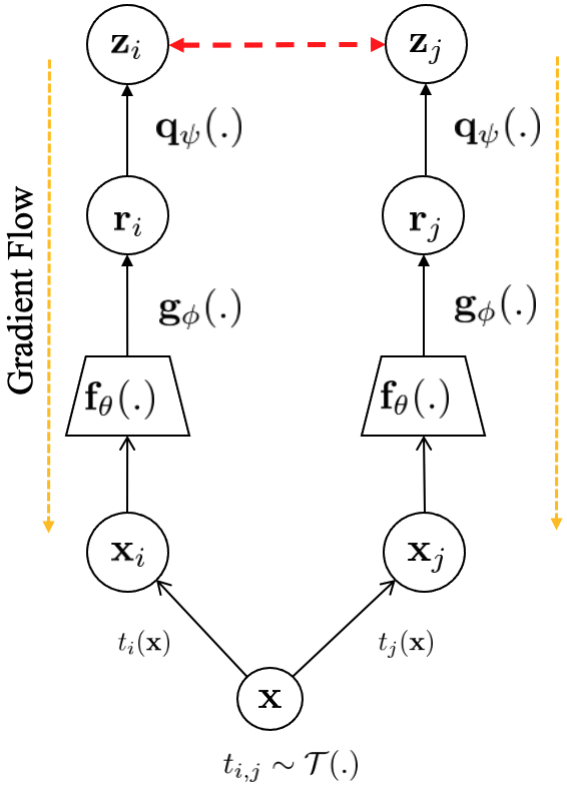


Figure 2.2: Typical pipeline for information maximization methods.

i.e., they are completely independent as in [Bardes et al., 2022]. The main takeaway is that the design choices vary as per the requirement and resources at one end.

To further illustrate this paradigm of SSL, [Zbontar et al., 2021] proposed `Barlow Twins (BT)`, where the idea is to regularize the cross-correlation matrix between the projections of both views. We denote a Siamese encoder as f_θ that computes representations $\mathbf{Y}^1 = f_\theta(\mathbf{X}^1)$ and $\mathbf{Y}^2 = f_\theta(\mathbf{X}^2)$, which are fed into a siamese projector g_ϕ to compute projections $\mathbf{R}^{(1)} = [\mathbf{r}_1^1, \ldots, \mathbf{r}_n^1] = g_\phi(\mathbf{Y}^1)$ and $\mathbf{R}^2 = [\mathbf{r}_1^2, \ldots, \mathbf{r}_n^2] = g_\phi(\mathbf{Y}^2)$ for both views.

$$\mathcal{L}_{\texttt{BT}}(\mathbf{R}^1, \mathbf{R}^2) = \underbrace{\sum_i^d (1 - C_{ii})^2}_{\textbf{Invariance}} + \underbrace{\lambda \sum_i^d \sum_{i \neq j}^d (C_{ij}^2)}_{\textbf{Redundancy Reduction}} \quad , \tag{2.4}$$

where λ is a positive constant trading off the importance of the first and second terms of the loss, and where C is the cross-correlation matrix computed between the outputs of the siamese networks along the batch dimension i.e., $C_{ij} \triangleq \frac{\sum_b \mathbf{r}_{b,i}^1 \mathbf{r}_{b,j}^2}{\sqrt{\sum_b (\mathbf{r}_{b,i}^1)^2} \sqrt{\sum_b (\mathbf{r}_{b,j}^2)^2}}$ b represents batch dimension while (i, j) are the indexes of the vector representation.

On a similar line `VICReg` [Bardes et al., 2022] is another popular method for information maximization that learns a joint embedding space governed by a loss

objective:

$$\mathcal{L}_{\texttt{VICReg}}(\mathbf{Z}^1, \mathbf{Z}^2) = \overbrace{\underbrace{\mu[v(\mathbf{Z}^1) + v(\mathbf{Z}^2)]}_{\textbf{V}\text{ariance}} + \underbrace{\nu[c(\mathbf{Z}^1) + c(\mathbf{Z}^2)]}_{\textbf{C}\text{o-Variance}}}^{\text{Regularisation Term}} + \underbrace{\eta s(\mathbf{Z}^1, \mathbf{Z}^2)}_{\textbf{I}\text{nvariance}}, \tag{2.5}$$

where $\mathbf{Z}^1 = \mathbf{q}_\psi(\mathbf{R}^1) = [\mathbf{z}_1^1, \cdots, \mathbf{z}_n^1], \mathbf{Z}^2 = \mathbf{q}_\psi(\mathbf{R}^2) = [\mathbf{z}_1^2, \cdots, \mathbf{z}_n^2]$, where μ, ν and η are hyperparameters, and `inavriance term` : $s(\mathbf{Z}^1, \mathbf{Z}^2) = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{z}_i^1 - \mathbf{z}_i^2\|_2^2$ aims to learn invariance to data transformations by making $\mathbf{z}_i^1$ and $\mathbf{z}_i^2$ similar, `variance term` : $v(\mathbf{z}_i^j)$ aims to prevent norm collapse by giving the components of $\mathbf{z}_i^1$ and $\mathbf{z}_i^2$ a standard deviation equal to γ (a fixed hyperparameter). It is defined as a hinge loss $v(\mathbf{z}_i^j) = \max(0, \gamma - S(\mathbf{z}_i^j), \epsilon))$, with $S(\mathbf{z}_i^j, \epsilon) = \sqrt{\text{Var}(\mathbf{z}_i^j) + \epsilon}$ the regularized standard deviation and the `covariance term` strives to remove correlations between the different components of $\mathbf{z}_i^1$, and is given by $c(\mathbf{Z}^1) = \frac{1}{d}\sum_{i \neq j}[C(\mathbf{z}_i^j)]_{mn}^2$ over the off-diagonal elements of the d-dimensional covariance matrix $C(\mathbf{Z}^1) = \frac{1}{n-1}(\mathbf{z}_i - \bar{\mathbf{z}})(\mathbf{z}_i - \bar{\mathbf{z}})^T$, where $\bar{\mathbf{z}} = \frac{1}{n}\sum_i^n \mathbf{z}_i^1$.

One thing to note in Equation 2.5 is that both branches are regularised independently and this is a very nice aspect of `VICReg` which we exploit in our work.

In the above approaches, the emphasis is on de-correlating the features in embedding space to avoid dimension collapse.

Distillation (Teacher-Student Paradigm)

Teacher-student [Hinton et al., 2015] learning methods (AKA knowledge distillation) are somewhat similar to information maximization techniques. In these methods, the student attempts to predict the teacher's representations across various transformations of images. This enables the student to learn invariant representations, resilient to diverse alterations (refers to augmentations) applied to the same image. The process typically involves two branches, one designated as the student and the other as the teacher. To circumvent representational collapse, where the student's learning stagnates, the teacher provides consistent target representations for the student to anticipate. Unlike conventional training where gradients update both teacher and student, the teacher remains unaltered, with fixed parameters during student updates. Occasionally, a momentum encoder intervenes between teacher and student, gradually transferring student weights to the teacher to enable more up-to-date targets. While the teacher usually mirrors the student's architecture, its parameters may differ. It can function as a moving average of the student's representations, updated via a momentum encoder with student weights – a typical schema is depicted in Figure 2.3. In some instances, the teacher shares weights with the student, necessitating an additional predictor network to forecast the teacher's representation. Interestingly, this is how network collapse is avoided without any

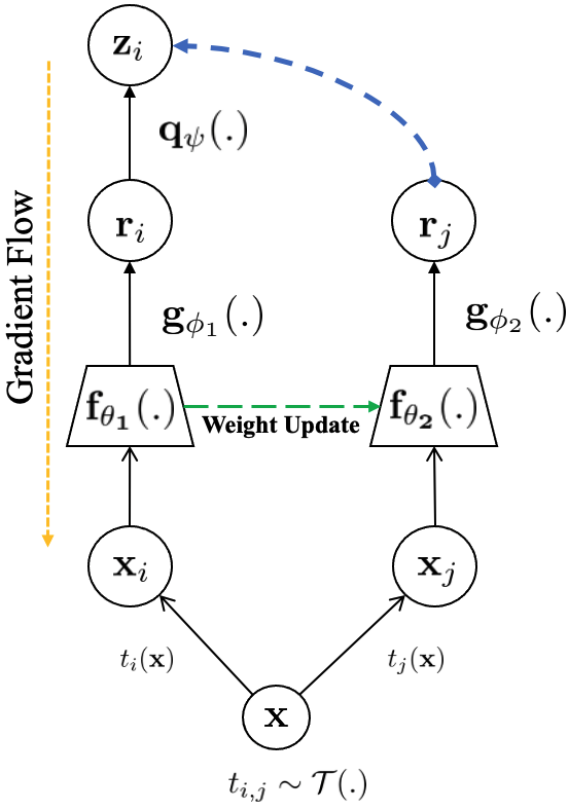


Figure 2.3: Typical pipeline for distillation-based methods. Note: gradient flow is only in one direction.

negative samples as used in contrastive learning. `BYOL` proposed in [Grill et al., 2020] follows a student-teacher non-symmetric architecture (similar to Figure 2.3). Soon after, a similar line of work by [Chen and He, 2021] proposed `SimSiam`, a similar architecture and loss function to `BYOL` [Grill et al., 2020]. However, the teacher and student had shared parameters, i.e., there was no momentum encoder in the setting, but instead leveraged stop-grad* in the other branch (to be precise in branch without predictor in Figure 2.3). This was followed by `OBOW` [Gidaris et al., 2021], in which the task is to reconstruct a bag-of-visual-words representation. However, it should be noted that the mechanisms to allow these methods avoid network collapse is an active area of research [Tian et al., 2020b, Li et al., 2022].

2.1.3 Generative SSL-methods

Generative SSL methods are one of the simplest examples of self-supervised learning. At their heart are autoencoders [Kingma and Welling, 2014] that learn two functions: an encoding function that transforms the input data, and a decoding function that recreates the input data from the encoded representation. The autoencoder learns an efficient representation (encoding) for a dataset, typically for dimensionality reduction.

*no backpropagation in branch as depicted in Figure 2.3

GAN frameworks based on this concept [Goodfellow et al., 2014] popularised generative machine learning and several GAN variants have been proposed. However, GANs have been primarily used for improving the fidelity of image reconstruction rather than learning good representation useful for other downstream tasks. Nevertheless, some research has investigated them for this task. BiGAN [Donahue et al., 2017] trains an additional encoder within a GAN framework and after training, this encoder can be used for different downstream tasks. Several other works in this direction have been proposed, e.g. [Brock et al., 2018, Donahue and Simonyan, 2019], however, the usage of GANs in SSL has not become a mainstream method due to a number of GAN related limitations, ranging from mode collapse to limitations related to scalability, as well as lack of flexibility in backbone networks. Recently, new paradigms in generative modelling have been gaining in popularity in the computer vision community such as masked auto-encoders [He et al., 2022], diffusion models [Ho et al., 2020] etc.

Overall this section covered different methods for performing self-supervised learning. Above mentioned methods presents different strategies to learn visual features, each following different school of thoughts. Each of these methods in itself presents interesting research directions. However, the goal of this thesis is not to dive deep into each of these aspect, instead this thesis follows a more practical approach towards self-supervised learning. However, as a researcher it is important to point out good research directions.

2.2 Dynamic (Conditional) Computation

The surge in popularity in large-scale neural networks in recent years can be attributed to their exceptional performance in tackling complex supervised/self-supervised learning tasks, such as achieving state-of-the-art results across various visual analysis tasks. However, despite their effectiveness, training such networks presents significant challenges. These include extended training durations, often lasting several weeks even on modern computers for certain tasks, and more importantly, the practical challenges associated with deployment of such large models in practical industrial settings. This highlights the need for approaches that mitigate against the challenges of very large networks, such as compression [Neill, 2020], knowledge distillation [Hinton et al., 2015], pruning [Frankle and Carbin, 2018] and dynamic computation [Veit and Belongie, 2018]. This work focuses on dynamic computation and a detailed explanation of the other techniques is beyond the scope of this thesis - the reader is referred to [Hoefler et al., 2021] for appropriate descriptions.

Dynamic computation (DC) is a resource-efficient mechanism that reduces model complexity by skipping unimportant parts of the network while preserving the networks topology. Several authors including [Figurnov et al., 2017, McGill and

Perona, 2017, Huang et al., 2018, Wu et al., 2020, Wang et al., 2018b] have proposed adding decision branches to different layers of convolutional neural networks (CNN) for learning early exiting strategies leading to faster inference. `BlockDrop` [Wu et al., 2018a] and `SpotTune` [Guo et al., 2019] learn a policy network to adaptively route the inference path through fine-tuned or pre-trained layers. `ConvNet-AIG`, introduce in [Veit and Belongie, 2018], [Herrmann et al., 2020] is a network that adaptively selects specific layers of importance to execute depending on the input image by specifying a target rate* for each layer. `GaterNet`, proposed by [Chen et al., 2019], introduced a network to generate input-dependent binary gates to select filters in the backbone network. The authors of [Li et al., 2021] proposed `DGNet`, a dual gating mechanism to induce sparsity along spatial and channel dimensions. Furthermore, dynamic channel pruning methods have also been devised such as feature boosting and suppression (FBS) [Gao et al., 2018], to dynamically amplify and suppress output channels computed by CNN layers. Other works learn sparsity through a three-stage pipeline, e.g. *pretrain-prune-finetune* as in [Tiwari et al., 2021], or use pre-trained models. The reader is referred to [Hoefler et al., 2021] for a more detailed explanation of sparsity, pruning and dynamic computing.

However, dynamic computing presents open challenges yet to be solved. A key challenge in deploying dynamic networks is their reduced parallelism due to input-conditioned computation graphs, which can lower efficiency on high-end GPUs. To address this, it is crucial to design dynamic models that are more hardware-friendly while also advancing hardware that supports dynamic computation. Combining dynamic models with techniques like quantization and pruning offers additional potential. There are still many research challenges that persist in the design, deployment and comprehension of dynamic networks. It is expected that these issues are likely to attract substantial research interest in the near future, given the remarkable advantages of dynamic models in terms of efficiency, efficacy and adaptiveness. Additionally, dynamic networks offer a promising solution to the low-power computation of large foundational models

2.2.1 Gated Computation

The core concept of dynamic computation is gating. Unlike pruning, DC maintains the network topology but dynamically determines for each input which subset of channels to compute, see Figure 2.4. A gating module (a small neural network) chooses between two discrete states, 0 for `off` and 1 for `on`, which can be seen as a hard attention mechanism. However, this has some repercussions, as choosing

*The target rate/budget refers to the budget constraints applied locally on each layer or globally to the overall architecture e.g., a target rate of 10% implies that only 10% of the channels will be retained for a particular layer or the entire network.

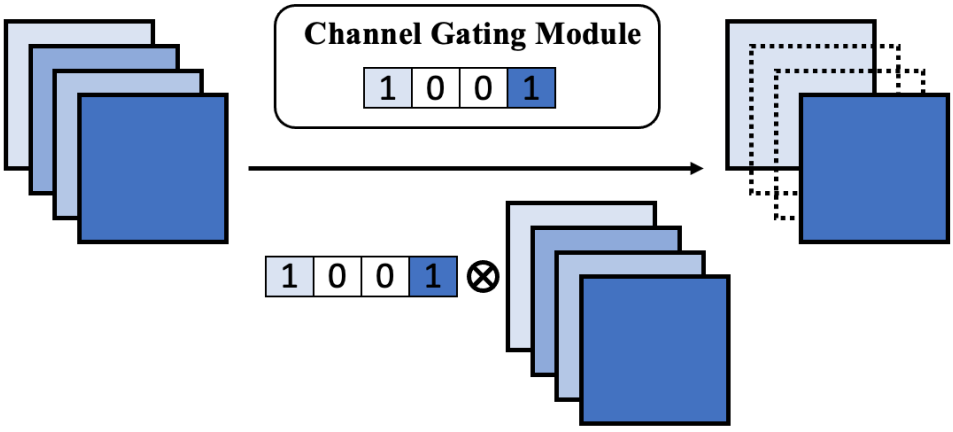


Figure 2.4: This figure illustrates the gating mechanism. Given an input feature map, the channel gating module decides which channels to accept or reject.

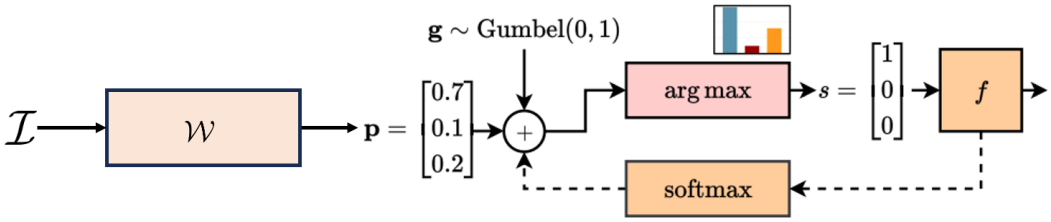


Figure 2.5: This figure depicts the Gumbel-Softmax re-parameterization trick. This figure is partially derived from [Scardapane et al., 2024].

"0/1" is a discrete selection that breaks the computational graph during training. As a result, often techniques like the `Gumbel-Softmax` reparameterization technique [Maddison et al., 2017] (referred to as the *gumbel-trick*) is used to relax the discrete binary masks to continuous variables. The `Gumbel-Softmax` trick provides a way to sample from discrete distributions in a differentiable manner, enabling the training of models with discrete outputs using standard gradient-based optimization algorithms. It is particularly useful in tasks involving structured outputs, such as sequence generation or discrete latent variable models.

The *gumbel-trick* can be understood by considering the sampling of a single element from the set of inputs. As depicted in Figure 2.5, first the inputs are processed with some trainable layer $\mathcal{W}$ to provide a relevance score $\mathbf{p}$ corresponding to each element in the set $\mathcal{I}$, which is proportional to its probability of being sampled. This is explored in further detail in Chapter 6. The implementation of $\mathcal{W}$ depends on the use case, which ensures that the approach used is computationally inexpensive.

Let us denote samples from the so-called Gumbel distribution [Jang et al., 2017] by g_i. The following operation provides an unbiased sample from the set $\mathcal{I}$:

$$s = \underset{i}{\text{argmax}}\{g_i + \mathbf{p}_i\} \tag{2.6}$$

Removing the Gumbel noise g_i, corresponds to taking the element with the highest score $\mathbf{p}_i$, while sampling provides a degree of freedom that is helpful in exploring possible alternatives. Since the probabilities $\mathbf{p}_i$ are implicitly trained via $\mathcal{W}$, the network can learn to select the element from $\mathcal{I}$ which is most useful for the specific task. In the case of channel selection, for example, this can lead to specializing single channels to specific types of inputs. It is important to note that sampling is not a differentiable operation and this often breaks the computational graph hence it is not easily trainable via gradient descent, but it can be relaxed with a softmax approximation:

$$s_i = \frac{\exp[(g_i + p_i)/\tau]}{\sum_j [(g_j + p_j)/\tau]}, \tag{2.7}$$

where τ denotes the temperature parameter, which is used to control the quality of the approximation. Note that s_i is not a binary variable but instead a positive real number. During the forward pass in training the argmax* operation is used whereas softmax relaxation is used during backpropagation as depicted in Equation 2.7. The complete process is shown in Figure 2.5.

Later in Chapters 5 & 6, we dive into more practical application of the above approach for designing an efficient approach for inference and additionally unifying the approach of dynamic computation in conjunction with self-supervised learning from a novel perspective.

2.3 Conclusion

This chapter presents the technical and theoretical background and basic terminology related to DNNs and SSL required to understand the research presented in the remainder of this thesis. The following chapters adapt these ideas to task-specific learning methodologies whilst investigating the various research hypotheses and questions introduced in Chapter 1.

*The argmax operation finds the argument that gives the maximum value from a target function

Chapter 3

Contrastive Self-supervised Learning and Instance-based Image Retrieval

In this chapter, we evaluate the effectiveness of contrastive self-supervised models for the task of image retrieval. This pertains to hypothesis **H1** defined in Chapter 1, where we hypothesize that models that learn to encode semantic similarity* among instances via discriminative learning should perform well on the task of image instance retrieval. Through our extensive evaluation, we find that visual representations from models trained using contrastive methods perform on par with (and in some cases outperform) a pre-trained supervised baseline trained on the Image-Net labels in retrieval tasks under various configurations. This is remarkable given that the contrastive models require no explicit supervision. Thus, we conclude that these models can be used to bootstrap base models to build more robust image retrieval engines. The research that emanated from this work was published at the ACM International Conference on Multimedia Retrieval 2021 (ICMR), Taipei, Taiwan.

3.1 Motivation

The task of large-scale image instance retrieval is to search a large image collection for the most relevant image/content for a given query – see Figure 3.1. The goal is to retrieve specific instances of the same concept from a large dataset of images e.g. pictures of the same object or location captured from different viewpoints and/or at different times. As explained in Chapter 1, this is a fundamental task in computer

*In the context of computer vision, semantic similarity refers to the measure of how similar two or more visual elements (such as images, objects within images, or visual features) are based on their content or meaning. This concept goes beyond simple pixel-by-pixel comparison and focuses on understanding the underlying features or objects depicted in the images to assess their similarity.

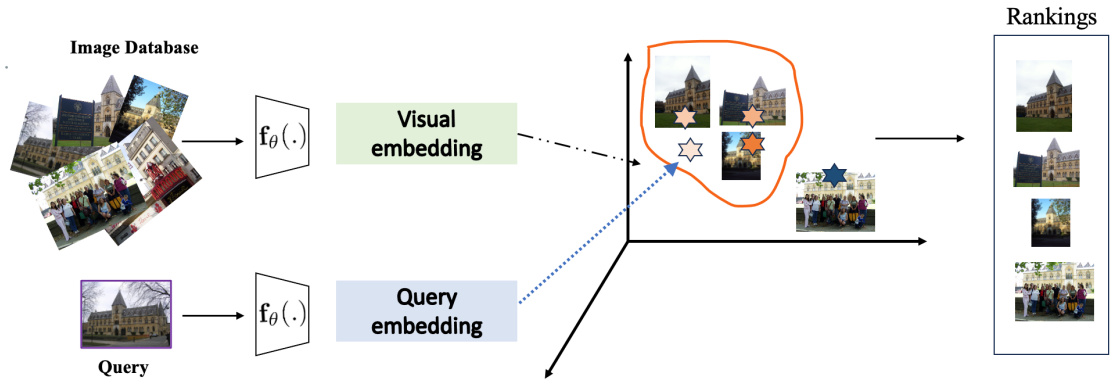


Figure 3.1: A typical pipeline for instance-based image retrieval. The goal is to retrieve specific instances of the same concept from a large dataset of images.

vision. Since their inception, Convolutional Neural Networks (CNNs) [Krizhevsky et al., 2012, Simonyan and Zisserman, 2015] have become the prominent approach for extracting descriptors for image retrieval. These descriptors perform very well in capturing the global semantics of an image and this has led to state-of-the-art results on many computer vision benchmarks [Ren et al., 2016, Chen et al., 2014, He et al., 2016].

The activations in the intermediate layers of CNNs can be used as a descriptor for an image. These descriptors are followed by some encoding techniques for a compact representation. These encoding techniques range from traditional approaches of VLAD [Jégou et al., 2010], BoW [Mohedano et al., 2016], and Fisher Vectors [Perronnin and Dance, 2007], to simple pooling methods like maximum activation of convolution [Azizpour et al., 2015a] (MAC), sum pooling convolution [Yandex and Lempitsky, 2015] (SPoC), Regional-MAC [Tolias et al., 2016] (RMAC) etc. The drawback of these methods is that these (off-the-shelf) networks are trained to reduce intra-class variance through supervision on ImageNet classes and this might affect the performance of instance retrieval (i.e. retrieving images that represent the same object or scene as in a query), which is a more fine-grained task. This drawback has been addressed in the literature by fine-tuning [Gordo et al., 2016, Radenović et al., 2016, Gordo et al., 2017, Revaud et al., 2019]. Fine-tuning is an effective approach for adapting neural networks to a different downstream task. However, in this work the goal is not to achieve SOTA performance on image retrieval benchmarks.

In this chapter, we explore an alternative approach, and specifically our hypothesis that models that are trained to encode semantic similarity among instances via discriminative learning perform well on the task of image instance retrieval (**H1**). We do this by performing a comprehensive suite of experiments to answer the question of how effectively SSL methods encode semantic identity in comparison to SL methods for the task of image instance retrieval (**RQ1**). To this end, we investigate contrastive SSL in the context of traditional off-the-shelf regimes using models that are trained

based on instance-wise supervision using a similarity-based metric. As explained in detail in Chapter 2, self-supervised models learn meaningful representations of visual data without the need for explicit labels. These learned representations can capture a high-level of semantic and structural information about the images, which could be valuable for tasks like image instance retrieval. To this end, in this chapter we investigate a class of self-supervised methods known as contrastive learning. that are trained in an unsupervised fashion using a contrastive loss [Wu et al., 2018b, Gutmann and Hyvärinen, 2010] as the sole objective (see Chapter 2 for a detailed description of contrastive learning). Our main contributions include:

- Extensively evaluate contrastive methods as a fixed feature extractor across different benchmarks;
- We provide experimental evidence showing that these models (trained without any explicit supervision) perform on par with a pre-trained supervised baseline (Table 3.2 and 3.3);
- Further investigate the role of the dimensionality of the feature embeddings for this task (Table 3.2).

3.2 Related Work

There has been tremendous progress recently in contrastive learning, with benefits demonstrated for a variety of tasks, ranging from image understanding [Khosla et al., 2020, Falcon and Cho, 2020, Tian et al., 2020a] to video understanding [Sermanet et al., 2018, Jabri et al., 2020]. This learning regime relies on learning a meaningful embedding that captures the inherent similarity between instances using discriminative approaches [Wu et al., 2018b]. This work investigates the effectiveness of contrastive methods that capture this very idea of instance similarity.

Conventional image retrieval methods [Sivic and Zisserman, 2003, Nister and Stewenius, 2006] relied on bag-of-words models that exploit local invariant features such as SIFT [Lowe, 2004] and large visual vocabularies (e.g. [Philbin et al., 2007]). To aggregate local patches and build a global summary, encoding methods such as Fisher vectors [Perronnin and Dance, 2007] or VLAD [Jégou et al., 2010], have also been proposed [Perronnin et al., 2010, Gordoa et al., 2012, Radenović et al., 2015].

Since the introduction of DNNs [Krizhevsky et al., 2012, Simonyan and Zisserman, 2015, Donahue et al., 2014, He et al., 2016] there has been a paradigm shift to exploit deep features instead of hand-crafted ones. Intermediate layers in convolutional nets can be used as global or local descriptors. As a result, so-called off-the-shelf [Azizpour et al., 2015b, Sharif Razavian et al., 2014] features can be used for retrieval. Based

on this, the authors of [Yandex and Lempitsky, 2015] used sum pooling with a centre prior for aggregating features across spatial dimensions. Other conventional encoding techniques like VLAD [Gong et al., 2014] or Fisher Kernels [Perronnin and Larlus, 2015] have also been used in combination with these local feature maps. For example, in [Mohedano et al., 2016] the authors proposed BoW-encodings of convolutional features for instance retrieval, whilst [Tolias et al., 2016] proposed R-MAC using max activations over a grid of windows of different scales to obtain compact representations.

Most of the off-the-shelf features are trained on ImageNet [Russakovsky et al., 2015] to reduce inter-class variance and deep models are used to extract feature representation. This approach is convenient and is able to produce reasonable performance for various retrieval tasks. However, the approach can experience some issues due to the reliance on pre-trained deep models. One of the most common issues is the domain gap, that could exist between the benchmark image dataset used to pre-train the deep model and the dataset of a given retrieval task. When this gap is not negligible, the efficacy of the pre-trained deep feature representation will be reduced. One way to address this is to fine-tune the model as shown in [Babenko et al., 2014, Gordo et al., 2016, Radenović et al., 2016, Gordo et al., 2017, Revaud et al., 2019] on an auxiliary dataset. However, the requirement to access an auxiliary dataset that is labelled and that is similar in nature to the dataset for retrieval can be difficult to realize in practice. In this case, fine-tuning a pre-trained deep model for retrieval is infeasible. In the context of image retrieval, most of the fine-tuning research has been performed on the *Landmark* dataset [Babenko et al., 2014], which requires cleaning of non-related images and potentially expensive and time-consuming post-processing. Another approach is to exploit methods that are trained to reduce intra-class variance, as is the case in contrastive learning on an unlabelled dataset. Unlike supervised learning, these approaches learn to discriminate among individual instances without any concept of categories. The work reported in [Wu et al., 2018b] discusses this notion of instance discrimination. Building on this, a simple formulation is presented in SimCLR [Chen et al., 2020a]. The intuition behind these approaches is to maximize the agreement among augmented views of the same instance using Noise Contrastive Estimation (NCE) [Gutmann and Hyvärinen, 2010]. Minimizing NCE is equivalent to maximizing mutual information (MI) as was formally shown in CPC [Oord et al., 2018] as InfoNCE. DIM [Hjelm et al., 2019] and AMDIM [Bachman et al., 2019] further extend the idea of InfoNCE across multiple views and scales. See Chapter 2 for the associated mathematical formulations for NCE and InfoNCE. One of the downsides of these approaches is that they require large batch sizes on large GPU clusters. To address this drawback, the authors in [He et al., 2020, Chen et al., 2020c] introduced MoCo, which uses an online and

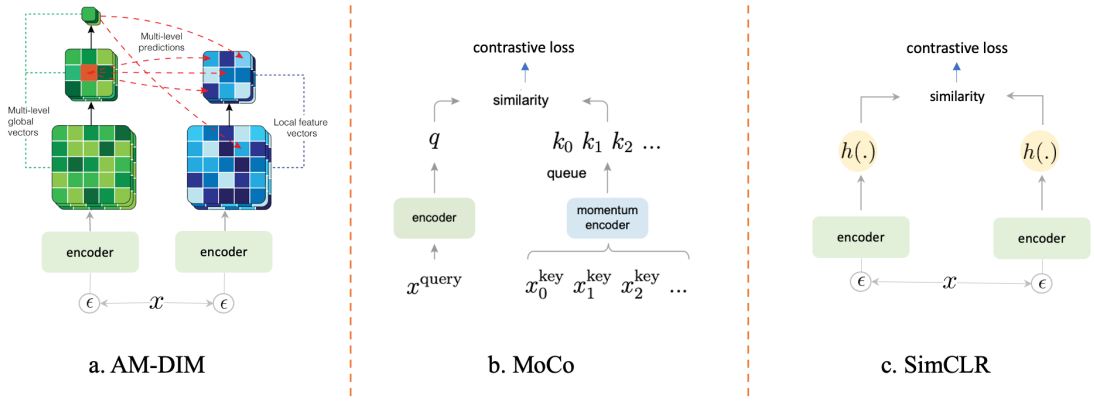


Figure 3.2: Self-supervised Contrastive Methods.

momentum-updated offline network that views contrastive learning as a dictionary lookup task. Our hypothesis is based on the intuition that the ability to discriminate among individual instances inherently encoded through the learning process makes contrastive learning a good candidate for the task of instance retrieval.

`Note`: This chapter mainly focuses on exhaustive evaluation of contrastive self-supervised learning for instance retrieval. However, several self-supervised models have been proposed during the completion of this thesis. For a more up-to-date comprehensive review please refer to [Ozbulak et al., 2023] as well as Chapter 2.

3.3 Contrastive Models

Contrastive learning refers to learning by comparison. This comparison is performed between positive pairs of "similar" and negative pairs of "dissimilar" inputs, which is achieved via a contrastive loss [Chen et al., 2020a, Le-Khac et al., 2020] derived from Noise Contrastive Estimation (NCE) [Gutmann and Hyvärinen, 2010] and is given by:

$$\mathcal{L}_{\text{Contrastive}} = -\sum_{i=1}^{N} \log \left(\frac{e^{\text{sim}(\mathbf{z_i}, \mathbf{z_i^+})}}{e^{\text{sim}(\mathbf{z_i}, \mathbf{z_i^+})} + \sum_{j=1}^{\mathcal{K}} e^{\text{sim}(\mathbf{z_i}, \mathbf{z_j^-})}} \right), \tag{3.1}$$

This work targets representative contrastive models for investigation, which are diverse in terms of the way they fuse information. Intuitively, this should lead to each model spanning different feature spaces from an image-understanding perspective. The following briefly describes the models considered in this study.

AMDIM: *Augmented Multiscale DIM* [Bachman et al., 2019] extends the Deep Info-Max (DIM) framework [Hjelm et al., 2019] by maximizing the mutual Information (MI) across features extracted independently from augmented views of each input along with features across multiple scales, as shown in Figure 3.2.

MoCo: *Momentum Contrast Representation Learning* [He et al., 2020, Chen et al.,

Table 3.1: Summary of different models.

Method	Epoch	Model	Params	Pre-trained	Fine-tuned
Baseline	90	ResNet-50	24M	✔	✗
SimCLR	100	ResNet-50	24M	✔	✗
SimCLR (2×)	100	ResNet-50 (2×)	94M	✔	✗
SimCLR (4×)	100	ResNet-50 (4×)	375M	✔	✗
MoCo_{v1}	200	ResNet-50	24M	✔	✗
MoCo_{v2}	200	ResNet-50	24M	✔	✗
AmDIM	150	-	-	✔	✗

2020c] alleviates the need for storing offline representations of the entire dataset in memory [Wu et al., 2018b] through the use of a *dynamic* memory *queue*. The samples in the dictionary are progressively replaced. This approach uses contrastive learning as a way of building a discrete dictionary (*queue*) of inputs (data samples) for a high-dimensional space. We consider MoCo_{V1} [He et al., 2020] and MoCo_{V2} [Chen et al., 2020c] in this evaluation.

SimCLR: *Simple framework for Contrastive Learning* [Chen et al., 2020a] is a simplified framework for contrastive learning compared to the previous ones. Here stochastic data augmentations are applied on an input $\mathbf{x}$ to get two views of $\mathbf{x}_i$ and $\mathbf{x}_j$. Sequentially these augmented inputs are passed through a *base encoder* $\mathbf{f}(.)$ followed by *projection head*, a small network that projects representations from $\mathbf{f}(.)$ to a space where a contrastive loss is applied. We investigate $\text{SimClr}_{\text{1x, 2x, 4x}}$.

It is also important to understand that the above methods differ only in the way in which they apply and scale the learning process while the objective remains the same as in Equation 3.1. In Table 3.1 we compare the different hyper-parameters in-order to take into consideration different aspects before making any comment over models performance.

3.3.1 Feature Extraction

We closely follow the implementation of R-MAC as in [Tolias et al., 2016] for obtaining compact representation. A typical setup for obtaining compact representation is depicted in Figure 3.3. Here the output feature representation of a CNN is depicted as a 3D tensor. Different regions are selected ($\mathcal{R}_i$) at different scales and for each of the ($\mathcal{R}_i$) regional max is calculated for each channel indices. Then we calculate the feature vector associated with each region and post-process it with $\mathbb{L}_2$ normalization, PCA-whitening and $\mathbb{L}_2$-normalization. PCA dimension d is set to 512. We combine the collection of regional feature vectors into a single image vector by summing them and $\mathbb{L}_2$-normalizing in the end.

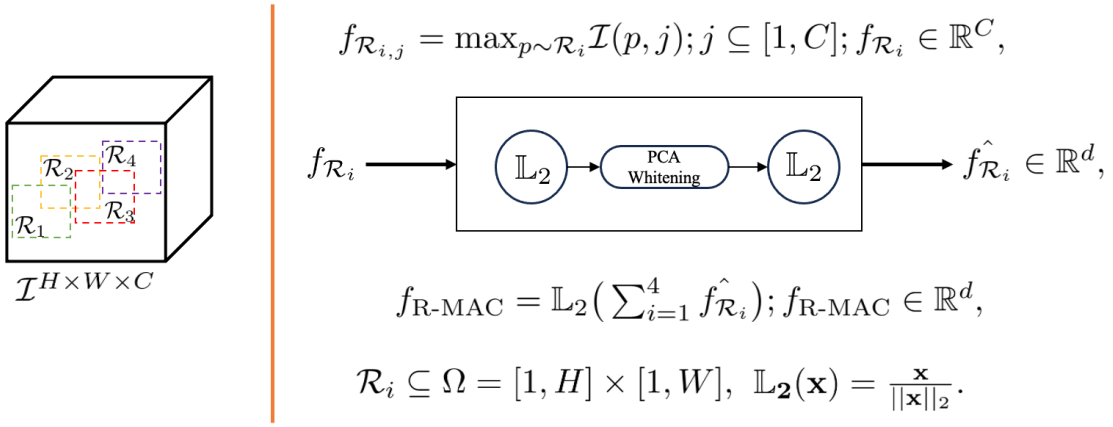


Figure 3.3: R-MAC feature extraction and post-processing. **Left**: depicts different selected regions $\mathcal{R}_i$ for calculating R-MAC representation. **Right**: for each $\mathcal{R}_i$ a corresponding representation is calculated by taking max across the spatial resolution which results in a feature vector ($f_{\mathcal{R}_i}$) of dimension $\mathbb{R}^C$, where C denotes channel dimension. Further these $f_{\mathcal{R}_i}$ representations are pre-processed via $\mathbb{L}_2$-PCA-$\mathbb{L}_2$ to a reduced textitd dimension which is fixed to 512 (across all evaluation). To accumulate the information across all R_i, elementwise sum is taken and again $\mathbb{L}_2$ normalization is applied to get a final representation.

3.4 Experimental Setup and Results

3.4.1 Setup

Implementation details: The goal in this work is not to fine-tune the models but instead evaluate them as a fixed feature extractor to obtain visual descriptors. The base encoder of each of the models is some flavor of ResNet50 with varying complexity. $\text{SimCLR}_{\text{2x,4x}}$ are 2x and 4x times in width compared to ResNet50 which serves as backbone for $\text{SimCLR}_{\text{1x}}$. Also, $\text{MoCo}_{\text{v1, v2}}$ have ResNet50 as the backbone, the only difference in v1 and v2 is that v2 further builds upon the findings in SimCLR [Chen et al., 2020a]. To this end, we consider the output of the last convolutional layer, i.e. just before the adaptive pooling layer, as our descriptor, which leads to feature maps of size $\mathbb{R}^{C\times H\times W}$. To obtain compact representations we use R-MAC ($L = 3a$) [Tolias et al., 2016] over spatial dimensions to get a fixed representation of size $\mathbb{R}^C$. We resize our input images to a fixed resolution of 724×724 giving a feature spatial dimension of 23×23 except in the case of AMDIM where the dimensions are 40×40. However, we downsample this to 23×23 to keep uniformity across the evaluation. Also, before running the final evaluation we first run each of the models in training mode (PyTorch `model.train()` just a forward pass) to tune the batch-normalization statistics to the current dataset and then finally test models in evaluation mode (`model.eval()`).

Dataset: We evaluate models on three standard benchmark datasets:

- Oxford5k [Philbin et al., 2007]: The Oxford Buildings Dataset comprises 5,062 images sourced from Flickr through searches for specific Oxford landmarks. Each of the 11 landmarks in the dataset has been manually annotated to create detailed ground truth data. For evaluation purposes, each landmark is associated with 5 query images, resulting in a total of 55 queries that can be used to assess an object retrieval system.

- Paris6k [Philbin et al., 2008]: contains 6,300 high resolution (1024×768) images obtained from Flickr by querying the associated text tags for famous Paris landmarks such as "Paris Eiffel Tower" or "Paris Triomphe." More details about the setup can be found here `https://www.robots.ox.ac.uk/~vgg/data/parisbuildings/`.

- INSTRE [Wang and Jiang, 2015]: Images are collected from multiple sources, which include various image search engines(e.g. Baidu, Bing, Google, Picsearch, Altavista), social networks (e.g. weibo, facebook) and photo sharing communities (e.g. Flickr, nipic). The whole dataset is split into three disjoint subsets INSTRE-S1 (for single object case 1), INSTRE-S2 (for single object case 2) and INSTRE-M (for multiple object case). INSTRE-S1 and INSTRE-S2 are collected for measuring single object case, both of which have 100 object classes. INSTRE-S1 contains 11011 images and INSTRE-S2 contains 12059 images. We group all the 100 objects from INSTRE-S1 into 50 two-tuples. INSTRE-M contains 5473 images distributed into such 50 two-tuple classes.

We further evaluate the performance on revised rOxford5k and rParis6k using the new evaluation protocol based on *easy*, *medium*, and *hard* ground truth labels [Radenović et al., 2018].

Evaluation metric: Retrieval performance is measured using mean Average Precision (mAP) following standard procedures for Oxford5k and Paris6k benchmarks and for INSTRE evaluating mAP over 1200 images as described in [Iscen et al., 2017]. For the revised benchmarks (rOxford5k and rParis6k) we report both mAP and mean precision@(10,5) (mp@10, mp@5).

Baseline. For comparing across all the contrastive models we use ResNet50 [He et al., 2016] trained on ImageNet [Russakovsky et al., 2015] as a fixed feature extractor as our pre-trained supervised baseline model. For completeness, we also evaluate a fine-tuned model [Revaud et al., 2019], which uses Generalized Mean Pooling (instead of R-MAC) trained with Average Precision loss (GeM (AP)) [*]. This method uses pre-trained ResNet101 [He et al., 2016] on ImageNet [Russakovsky et al., 2015] and finally fine-tuned on the public Landmarks-clean dataset [Gordo et al., 2016]. The purpose of this is to provide an indicative upper bound to the evaluation scores.

[*] `https://github.com/naver/deep-image-retrieval`

Ranking. We consider global search (G) in this evaluation, along with that we further integrate global search with the following techniques:

- Average Query Expansion [Chum et al., 2007] (`AQE`): A new query is constructed by averaging verified results of the original query (Q_0). First, the top N-verified results returned by the search engine are selected. A new query Q_{avg} is then formed by taking the average of the original query Q_0 and the N results; $Q_{avg} = \frac{1}{N+1}(Q_0 + \sum_{i=1}^{N} r_i)$, where r_i is retrieved result for the query.
- Database-side Augmentation [Arandjelović and Zisserman, 2012] (`DBA`): database-side augmentation (DBA) replaces every image signature in the database by a combination of itself and its neighbors. The objective is to improve the quality of the image representations by leveraging the features of their neighbors. We sum-aggregate the nearest k neighbors as in the query expansion case. Optionally, the sum can be weighted depending on the rank of the neighbors, and in our experiments, we use $weight(r) = \frac{kr}{k}$ as a weighting scheme, with r the rank of the neighbor, and k the total number of considered neighbors.

For `AQE` we consider nearest neighbour $N = 10$, for `DBA` we consider $N^{'} = 20$ while combining both of these we consider $N = 1$ and $N^{'} = 20$, based on the findings in [Gordo et al., 2017].

We use a `PCA` dimension of 512 and evaluate on a global search for R-MAC representations unless otherwise stated.

3.4.2 Results

Quantitative Assessment: Table 3.2 compares different models along with different expansion techniques. GeM (AP) serves as an upper-bound indicator rather than a benchmark.

In Table 3.2, if we consider SSL models with the same model complexity as compared to supervised baseline, for a *naive global* approach $MoCo_{v2}$ achieves an mAP of 58.36% which is 3.24% higher than supervised baseline which obtains mAP of 55.12% for Oxford 5k. However, $SimCLR_{4x}$ (375M) outperforms the baseline with the farthest margin i.e 4.28%. Also, there is a performance increment along all the methods including the baseline with the inclusion of different ranking techniques. However, it is difficult to comment on which ranking algorithm gives the overall best performance across all the methods and datasets. We also include results for ensembling different contrastive models. For the ensemble, the R-MAC representations are concatenated (feature vectors are vertically stacked i.e., one after the other) and dimensionality reduced via PCA, which seems to give a further performance boost consistently for the Oxford 5k dataset, Table 3.2. Concatenation ($SimCLR_{2x}$,

SimCLR_{4x}) seems to be the best performing combination for Oxford 5k dataset, although all the ensemble improves the performance. A similar trend can be observed for the Paris 6k dataset as well. For global search query, SimCLR_{1x} easily beats the baseline with a mAP of 42.28%, while MoCo_{v2} beats it by a large margin with a mAP of 49.72%. Ranking techniques improve search performance; however, no single ranking technique or combination consistently outperforms the others across all scenarios. Interestingly, an ensemble of different methods doesn't perform better than individual methods except in the case of AMDIM. In the case of the INSTRE dataset, the best-performing contrastive model is SimCLR_{4x} with the highest parameters.

To further consolidate our findings, we also conducted an evaluation on the revised rOxford 5k and rParis 6k datasets as depicted in Table 3.3. On rOxford 5k SimClr_{2x} gives the best performance on all labels. mP@10 is almost 70% for the *easy* category, with the drop in performance for *hard* label. Similarly, on rParis 6k, SimClr_{4x} gives the best performance with mP@10 over 90 for easy and medium, but again this drops off for *hard* labels. As in Table 3.2, here ensembling further boosts performance. Again, contrastive models surpass the baseline pre-trained supervised model.

Effect of descriptor dimension on performance. Table 3.4 reports our findings for global search*. Interestingly, true dimensions (L_2 normalized R-MAC representations) appear to perform worse for almost all the models. The best dimension varies across the dataset but it is never the true dimension. This could be attributed to dimensions with small principal components being noisy and redundant and adversely affecting performance.

Remarks: 1. GeM (AP) [Revaud et al., 2019] serves as a benchmark (upper bound) for evaluating the quality of representations produced by SSL methods relative to fine-tuned models. Notably, GeM was pre-trained on ImageNet using ResNet-101 (44.5M parameters) and fine-tuned on the Landmark-clean dataset [Gordo et al., 2016]. While it outperforms all models across all settings for Oxford 5k and Paris 6k datasets, its performance on the INSTRE dataset is significantly lower due to fine-tuning on a specific source domain. Interestingly, SSL models with larger parameter counts show only a minor drop in performance. This highlights a key advantage of self-supervised methods: their superior generalization to unseen target domains, as demonstrated quantitatively in Table 3.2. Furthermore, this analysis suggests

*Comparison is across the horizontal dimension (columns)

Table 3.2: Comparing mAP (%) score across different models. Bold (red) best-performing ensemble if it exists.

Dataset	Method	**G**	**G** +AQE	**G** +DBA	**G** +DBA +AQE
	Baseline	55.12	67.85	62.63	70.60
	SimClr_{1x}	51.47	62.96	61.00	66.32
	SimClr_{2x}	58.59	**70.49**	67.34	**72.62**
	SimClr_{4x}	**59.40**	69.80	**67.52**	71.03
	MoCo_{v1}	56.76	66.89	64.12	69.77
Oxford 5k	MoCo_{v2}	58.36	67.49	65.09	69.86
	AmDim	36.95	43.44	39.97	44.69
	SimClr_{2x}+SimClr_{4x}	**61.89**	**72.64**	**69.27**	**74.43**
	SimClr_{2x}+AmDim	57.06	67.37	64.69	69.29
	SimClr_{2x}+MoCo_{v2}	61.33	71.58	68.47	72.94
	SimClr_{4x}+MoCo_{v2}	61.87	71.97	68.20	72.94
	SimClr_{4x}+AmDim	58.13	68.76	64.81	69.78
	GeM (AP)	66.90	78.57	75.70	81.92
	Baseline	41.46	61.76	64.71	**75.56**
	SimClr_{1x}	42.28	61.15	64.36	74.45
	SimClr_{2x}	44.48	59.65	63.28	71.91
	SimClr_{4x}	44.96	56.3	60.40	68.46
	MoCo_{v1}	43.22	60.34	63.00	72.11
Paris 6k	MoCo_{v2}	**49.72**	**64.46**	**67.06**	75.01
	AmDim	34.87	49.2	47.72	56.06
	SimClr_{2x}+SimClr_{4x}	45.27	56.84	59.88	68.07
	SimClr_{2x}+AmDim	44.45	60.06	63.02	71.12
	SimClr_{2x}+MoCo_{v2}	47.93	61.46	64.96	72.37
	SimClr_{4x}+MoCo_{v2}	47.44	58.44	61.88	69.48
	SimClr_{4x}+AmDim	44.74	57.05	60.66	68.60
	GeM (AP)	49.87	63.46	67.99	76.78
	Baseline	36.64	**64.42**	**63.96**	**74.32**
	SimClr_{1x}	27.51	47.85	47.34	57.73
	SimClr_{2x}	36.68	57.04	56.89	65.27
	SimClr_{4x}	**44.76**	63.73	63.26	70.17
	MoCo_{v1}	33.44	54.88	54.84	64.65
INSTRE	MoCo_{v2}	33.36	51.79	50.47	58.83
	AmDim	24.34	37.92	36.69	42.74
	SimClr_{2x}+SimClr_{4x}	45.58	63.94	63.54	70.54
	SimClr_{2x}+AmDim	39.37	59.02	59.09	66.68
	SimClr_{2x}+MoCo_{v2}	41.99	60.99	60.64	68.41
	SimClr_{4x}+MoCo_{v2}	**47.07**	**65.23**	**64.93**	71.42
	SimClr_{4x}+AmDim	45.83	63.88	63.84	70.83
	GeM (AP)	20.78	31.37	30.29	35.93

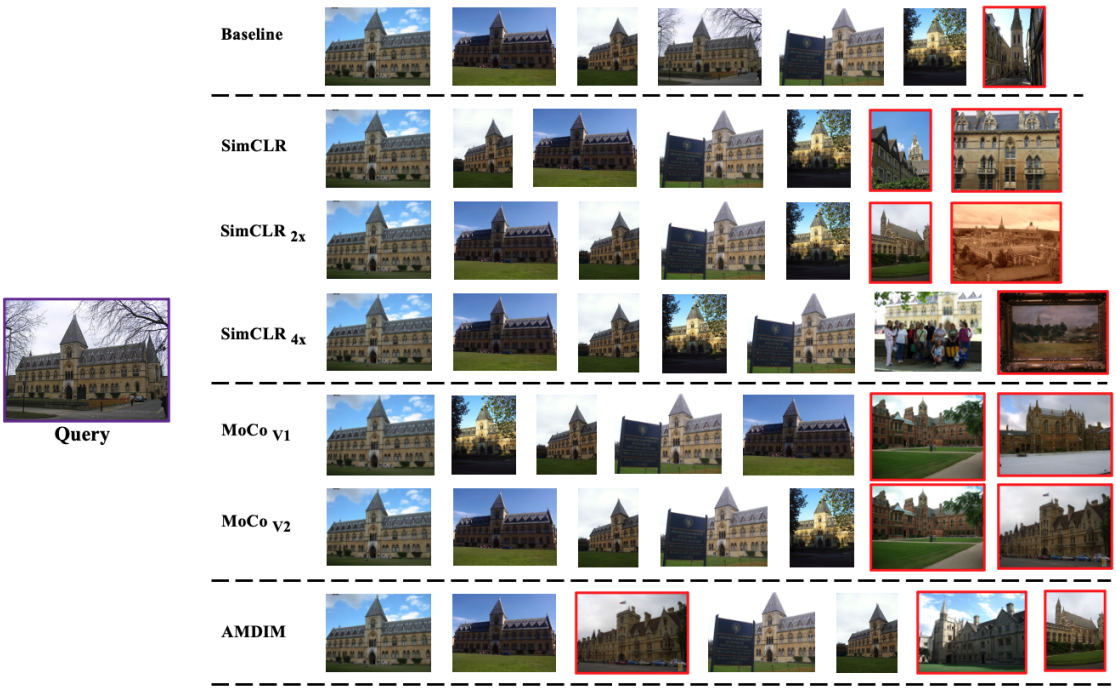


Figure 3.4: **Qualitative Analysis:** Comparison of retrieval results for a given query sample from Oxford 5k dataset.

Figure 3.5: **Qualitative Analysis:** Comparison of retrieval results for a given query sample from Paris 6k dataset.

Table 3.3: Comparison on global ranking across different models. Bold (red) best-performing ensemble if it exists.

		Easy			Medium			Hard		
Dataset	Method	mAP	mp@5	mp@10	mAP	mp@5	mp@10	mAP	mp@5	mp@10
	Baseline	45.65	67.55	61.81	32.89	64.00	58.57	12.10	23.71	19.43
	SimCLR_{1x}	47.17	69.78	63.90	31.98	65.24	57.52	9.52	20.57	15.86
	SimCLR_{2x}	**54.95**	**76.69**	**69.49**	38.54	73.14	**65.33**	**14.41**	**31.21**	**22.93**
	SimClr_{4x}	54.65	74.80	68.37	**38.57**	**73.71**	65.05	14.05	30.07	22.79
	MoCo_{v1}	48.29	68.48	64.36	33.57	64.43	58.58	9.27	20.05	15.62
rOxford 5k	MoCo_{v2}	52.69	72.33	65.86	36.49	67.24	59.24	10.72	23.38	18.67
	AmDim	21.24	38.31	31.35	17.54	36.38	32.57	4.11	5.92	6.33
	SimClr_{2x}+SimClr_{4x}	**55.23**	76.15	68.87	39.60	**74.29**	**65.64**	15.01	29.43	**23.97**
	SimCLR_{2x}+AmDim	52.08	72.28	65.51	36.19	66.86	58.57	10.20	21.64	17.64
	SimCLR_{2x}+MoCo_{v2}	55.11	76.47	69.35	38.46	70.52	62.57	12.51	27.12	21.83
	SimCLR_{4x}+MoCo_{v2}	53.98	72.40	67.90	38.31	70.10	64.69	13.38	28.43	23.13
	SimCLR_{4x}+AmDim	49.23	70.76	64.00	35.72	67.62	61.33	11.72	23.79	18.80
	GeM (AP)	64.07	84.93	80.56	51.03	89.43	83.86	30.30	54.86	44.00
	Baseline	47.20	91.14	87.00	31.63	93.71	91.71	10.44	57.14	43.43
	SimCLR_{1x}	49.35	93.14	89.52	31.61	95.71	92.00	8.93	46.57	38.43
	SimCLR_{2x}	54.28	**94.00**	91.14	34.47	96.00	92.86	10.87	62.00	49.71
	SimCLR_{4x}	54.97	93.71	**91.71**	35.39	**96.29**	**94.86**	12.34	**66.57**	**56.29**
	MoCo_{v1}	50.47	92.29	89.71	31.48	95.14	92.57	8.13	46.57	36.29
rParis 6k	MoCo_{v2}	**55.77**	92.86	89.24	**36.32**	94.86	92.14	**11.71**	60.00	50.43
	AmDim	38.62	80.29	73.71	25.92	82.57	75.71	6.36	25.14	19.57
	SimCLR_{2x}+SimCLR_{4x}	55.54	94.00	**91.86**	35.59	96.00	94.00	12.37	66.29	56.00
	SimCLR_{2x}+AmDim	53.71	93.71	90.43	34.57	95.14	93.14	10.46	55.71	46.00
	SimCLR_{2x}+MoCo_{v2}	**57.00**	93.43	91.00	**36.66**	95.14	93.29	11.88	64.00	52.43
	SimCLR_{4x}+MoCo_{v2}	56.86	**94.29**	91.71	36.61	96.29	94.86	**12.68**	**69.43**	55.57
	SimCLR_{4x}+AmDim	54.65	92.86	90.57	35.06	95.71	93.43	11.41	61.14	48.71
	GeM (AP)	54.90	94.38	91.95	37.36	98.86	97.00	14.65	76.86	63.71

that contrastive methods trained to minimize inter-class variance effectively capture instance similarity, which is reflected in these evaluations.

2. Large models generally perform better across most scenarios, as demonstrated by $\text{SimCLR}_{(4x>2x>1x)}$. However, performance improvements are not solely reliant on model size; better design can also lead to significant gains. For instance, MoCo_{v2}, despite having fewer parameters than $\text{SimCLR}_(4x, 2x)$, outperforms them due to its design choices. It incorporates improvements proposed in SimCLR [Chen et al., 2020a] for the projector head, resulting in enhanced performance. Additionally, the effectiveness of these SSL contrastive methods heavily depends on various training strategies, such as extended training epochs, large batch sizes, and effective augmentations.

3. Design of contrastive models and downstream tasks plays a crucial role. AmDIM [Bachman et al., 2019] performs worse across all the settings and this could be attributed to the way contrastive learning is applied. AmDIM minimizes the mutual information between global representation and a collection of local feature vectors pulled from an intermediate layer in the encoder, as illustrated in Figure 3.2. This might be useful for tasks like segmentation or correspondence learning but might not be good for instance retrieval. We investigate this aspect of mutual information minimization in the next chapter where we specifically propose similar design for the task of saliency prediction.

Table 3.4: Comparison of mAP (%) across different PCA dimension and the true dimension.

Dataset	Method	PCA-Whitening 32	64	128	256	512	1024	2048	True dim.
	Baseline	48.31	54.61	56.68	57.52	55.12	49.44	37.23	**58.47**
	$SimCLR_{1x}$	34.96	44.96	54.17	**54.79**	51.47	44.16	34.94	50.63
	$SimCLR_{2x}$	36.79	47.96	60.32	**61.68**	58.59	49.82	24.62	53.72
Oxford 5k	$SimCLR_{4x}$	35.32	45.86	57.77	**61.70**	59.40	53.83	44.71	41.94
	$MoCo_{v1}$	29.90	41.95	54.62	**57.38**	56.76	50.20	39.37	38.73
	$MoCo_{v2}$	39.92	47.88	57.92	**60.43**	58.36	52.14	40.05	51.76
	AmDim	11.52	15.30	21.71	28.71	36.95	**37.58**	30.73	16.10
	Baseline	**72.22**	71.83	63.65	53.40	41.46	28.99	17.89	68.36
	$SimCLR_{1x}$	69.90	**73.29**	66.62	54.58	42.28	30.7	20.06	66.60
	$SimCLR_{2x}$	75.21	**77.90**	69.25	57.83	44.48	33.05	23.27	72.20
Paris 6k	$SimCLR_{4x}$	77.16	**78.04**	69.19	57.29	44.96	35.02	25.79	72.89
	$MoCo_{v1}$	56.84	**63.63**	61.87	54.07	43.22	32.57	21.63	53.66
	$MoCo_{v2}$	70.08	**75.11**	71.24	61.66	49.72	35.73	22.98	69.99
	AmDim	21.49	33.96	**41.19**	40.85	34.87	26.82	17.85	25.50
	Baseline	26.44	34.92	**38.68**	38.55	36.64	29.25	20.20	33.03
	$SimCLR_{1x}$	16.47	23.27	29.01	**30.35**	27.51	21.85	15.97	21.85
	$SimCLR_{2x}$	18.42	26.83	35.88	**39.67**	36.68	29.64	21.57	25.94
INSTRE	$SimCLR_{4x}$	18.92	28.97	40.45	**46.99**	44.76	36.68	27.65	28.98
	$MoCo_{v1}$	20.08	27.85	33.77	**36.24**	33.44	26.66	18.46	23.01
	$MoCo_{v2}$	19.00	27.38	34.34	**36.22**	33.36	26.86	18.79	26.22
	AmDim	10.65	15.88	20.55	24.15	**24.34**	20.45	14.45	10.25

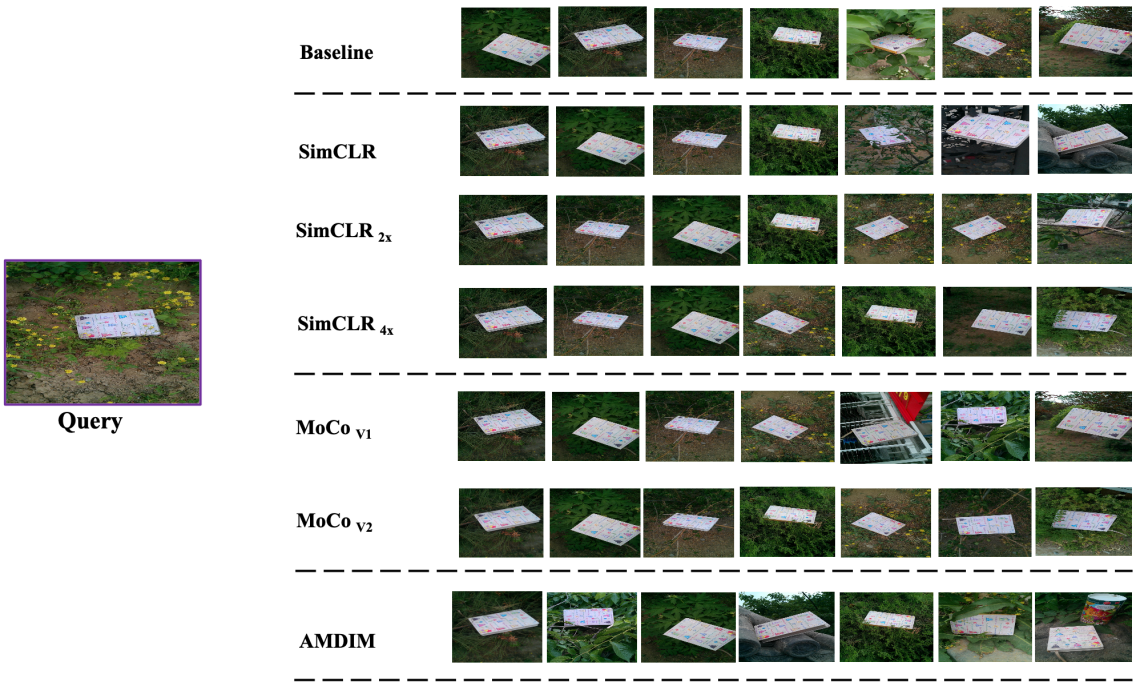


Figure 3.6: **Qualitative Analysis:** Comparison of retrieval results for a given query sample from INSTRE dataset.

Qualitative Assessment: We further qualitatively visualize some query results for Oxford 5k (Figure 3.4), Paris 6k (Figure 3.5) and INSTRE 3.6 dataset. Self-supervised models learn to encode semantic similarity without any sort of labels, instead, this learning paradigm itself learns to exploit underlying factors of variation within data just by minimizing the distance between identical samples while maximizing the margin between negative samples. This visualization further consolidates our finding for generalizability and encoding of semantic similarity for SSL contrastive models.

3.5 Discussion and Conclusion

This chapter evaluates contrastive self-supervised pre-trained models for the task of instance-based image retrieval. The goal of the experiments is to answer **RQ1**: *How effectively do SSL methods encode semantic identity in comparison to SL methods for the task of image instance retrieval?* Our evaluation found that these methods are on par with those trained on class labels. In fact, in many settings in Tables 3.2 and 3.3, contrastive approaches surpass the performance of the supervised model. The quantitative evaluation shows that these contrastive methods can easily surpass supervised models without any explicit supervision. This study proves that contrastive models learn features that can be used to bootstrap image retrieval engines. However, particularly impressive is the fact that the contrastive pre-training regime did not take the image retrieval task into consideration yet it learned features that can be used for such a task. This is interesting from the perspective of utilizing self-supervised models as a fixed feature extractor.

However, this evaluation leads to some important future research questions. One question is regarding the design of such contrastive models. As can be seen in our evaluation, global contrastive learning seems to work (i.e global representations) for such tasks while on the other hand optimizing for global-to-local (AMDIM) might not be useful for tasks like image retrieval or classification. Furthermore, it can also be seen how large models easily surpass the baseline. This is good from the perspective that annotations are not needed and models can be scaled accordingly but this scaling has its own implications such as larger training time, large batch size, etc. Furthermore, it is important to be careful about the application of such large models because then inference will become costly. In the following chapters (5, 6), we investigate this problem in the context of a SSL paradigm and provide perspectives on resolving some of these issues.

Chapter 4

Contrastive Self-supervised Learning and 360° Visual Attention Modelling

In Chapter 3 it can be observed that self-supervised (contrastive) pre-trained models serve as a good starting point to bootstrap an image retrieval pipeline for robust performance. In this chapter, we explore hypothesis **H2** defined in Chapter 1 that omnidirectional images are particularly suited to an SSL approach due to the geometry of the data domain. To this end, we extend recent advances in contrastive self-supervised learning to learn latent representations that are sufficiently invariant to be highly effective for 360° (spherical) saliency prediction as a downstream task. To verify this hypothesis, we design an unsupervised framework that effectively maximizes the mutual information between the different views from both the equator and the poles [*]. We show that the decoder is able to learn high-quality saliency distributions from the pre-trained frozen encoder embeddings. Our model compares favorably with fully-supervised learning methods on the Salient360! [Rai et al., 2017], VR-EyeTracking [Xu et al., 2018b] and Sitzman [Sitzmann et al., 2018] datasets. This performance is achieved using an encoder that is trained in a completely unsupervised manner and a relatively lightweight supervised decoder (3.8 × fewer parameters in the case of the ResNet50 encoder). The research that emanated from this work was published at CVF International Conference on Computer Vision (ICCV), Montreal, 2021.

[*]Equator represents the central part of the sphere while the poles are the top and bottom of the sphere.

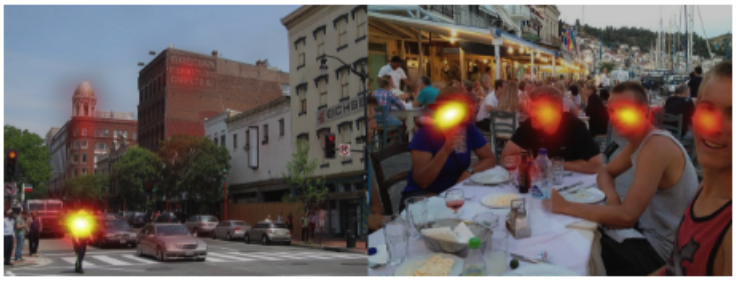


Figure 4.1: 2D Saliency heatmaps overlaid on the original images, where it can be seen that important parts of the image such as humans or faces are deemed to be particularly salient. Figure source: `https://www.cs.princeton.edu/courses/archive/spring18/cos598B/public/projects/LiteratureReview/COS598B_spr2018_Saliency.pdf`

4.1 Motivation

The task of visual saliency prediction is to predict the area in an image/video that captures human attention. In other words, saliency prediction aims to identify which parts of an image will attract the human gaze. The target output is a saliency map, showing which areas are most likely to draw attention, with higher values indicating more eye-catching spots (depicted in Figure 4.1). Visual saliency prediction focuses on identifying where attention is drawn in an image, often around objects, rather than focusing on what the objects are. Factors like local and global contrast, spatial distribution, focus, background relevance, and central bias all influence an object's saliency. These characteristics align with human visual attention and are commonly used as features in systems for predicting visual saliency.

Unlike traditional media (2D images/videos), omnidirectional images (ODIs) provide users with the ability to explore different regions of the viewing sphere. The average person's head movements (HM) are typically a good prediction of the most probable viewport localized within the sphere, while eye movements (EM) reflect regions of interest (ROIs) inside the predicted viewports. Thus, when predicting the most salient pixels for 360° images, it is necessary to predict both HM and EM [Xu et al., 2020b]. Despite remarkable advances in the field of visual attention [Itti and Koch, 2000, Borji, 2018, Xu et al., 2020b], existing approaches for 360° saliency prediction are still limited in scope/power for two main reasons.

First, all previous state-of-the-art 360° saliency static approaches are trained end-to-end in a supervised manner. This limits their capacity to leverage unlabelled data. Compared to the large-scale 2D video/image saliency datasets [Borji, 2018] (i.e., up to 10000 images,1000 video sequences), 360° video/image HM/EM datasets are rather small. This is due to the complex annotation process, which limits the capacity of the fully supervised approaches. Therefore, exploiting unlabelled data for learning better features is critical and intuitively a good design. Second,

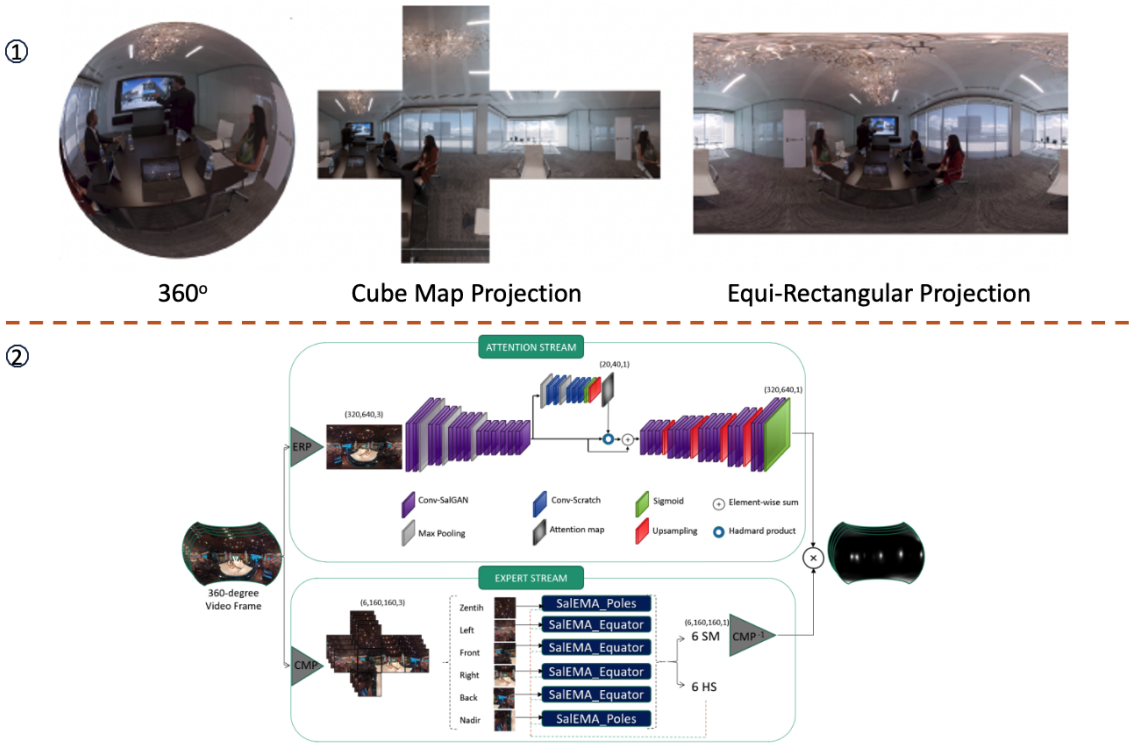


Figure 4.2: A typical pre-processing step for a 360° image often involves projecting it into cube map projection (CMP) and equi-rectangular projection (ERP) as shown above in **1**. Deep models can then be trained on these two post-processed images for the tasks of visual saliency prediction following two stream networks as shown below in **2**. Figure source for **1** [Rana et al., 2019] and **2** [Dahou et al., 2021] .

most of the previous approaches apply a CNN on each patch/cube resulting from the equi-rectangular (ERP) and cube map (CMP) projections (Figure 4.2). The former suffers from geometric distortions near the poles, whereas the latter stretches the salient regions into different cube faces, forcing the model to lose the global contextual information. These methods are also of high computational complexity, which may limit their applicability. A powerful recent paradigm for estimating mutual information (MI) is contrastive learning based on noise contrastive estimation (NCE) [Gutmann and Hyvärinen, 2010], where multiple views of the same scene are brought together in embedding space while pushing apart views from different scenes. Additionally, as the choice of views is important for contrastive learning, 360° data offers a new set of choices for more effective MI estimation. The projections used in the scope of this work are task-relevant, but also, make the optimization problem harder, since they are not as susceptible to optimization short-cuts [Doersch et al., 2015] as simple augmentations like color jitter and horizontal flips. This improves the expressive power of the encoder. This also motivates us to discard the use of CMP at training/inference, as we argue that the encoder is inherently sensitive to the signal coming from the Zenith and Nadir regions.

Our goal in this chapter is to learn representations that capture the signal shared

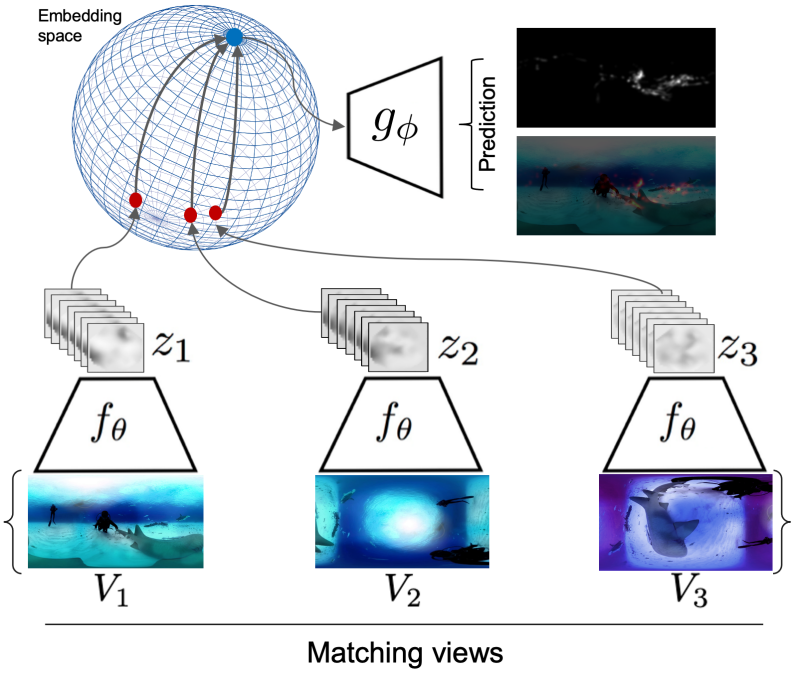


Figure 4.3: Given a set of 360° images and associated projections, a deep representation is learned by maximizing the mutual information between views of the same scene in the embedding space, while discarding views of different scenes.

between a support image $\text{View}_1(\text{V}_1)$ and its corresponding projections $\text{View}_2(\text{V}_2)$ and $\text{View}_3(\text{V}_3)$ as shown in Figure 4.3. This is achieved by maximizing the agreement between global and local representations of support images and their projections respectively. The approach is inspired by the notion of mutual information (MI) maximization as proposed in Deep Info Max (DIM) [Hjelm et al., 2019] and Augmented Multi-scale DIM (AMDIM) [Bachman et al., 2019]; at this point, it is important to address why we opted for an approach like DIM/AMDIM which relies on local-to-global contrastive learning. The global approach of contrastive learning focuses on extracting global representations while not explicitly learning local distinctive features. As a result, the former approach will work for tasks like image recognition or retrievals as described in Chapter 3 while the latter may be useful for tasks like segmentation or saliency prediction where local features play a crucial role. However, we introduce some important differences. First, we add self-attention to induce a soft feature selection mechanism over local representations (intermediate activation maps). Second, we formulate the total loss (Section 4.3.2) in a way to induce invariance to projections as in [Misra and Maaten, 2020] and maximize the MI across different augmented (projected) views. Finally, unlike (AMDIM), instead of relying on batch sizes for negative samples, a memory bank is adopted for computational efficiency. Our contributions are as follows:

- We propose a contrastive framework to extend the idea of self-supervised learning to a new data domain, specifically 360° images, and show how it

can be effectively used for a regression downstream task rather than a simple recognition task.

- Through extensive evaluation as shown in Table 4.1, we show that contrastive learning can be exploited for saliency prediction, and furthermore that it performs on par with fully supervised methods.

- Our approach addresses one of the key challenges encountered when predicting 360° saliency by excluding any use of CMP (Dividing the panorama into smaller tiles and applying local perspective projection introduces significant computational overhead, as saliency detection must be performed on each individual tile). The design implicitly embeds the geometric specifications in the model weights.

- A single subsequent stream of learning on the equi-rectangular projection (ERM) images significantly reduces the computational cost (8× faster than the most efficient model among other 360° saliency approaches).

4.2 Related Work

This section reviews important works related to attention modelling for 360° images. We focus on works related to the prediction of the HM/EM saliency maps in 360° images, which can be grouped into heuristic and data-driven approaches.

Visual attention modelling for ODIs. The authors of [De Abreu et al., 2017] introduced the fused saliency map (FSM) approach for HM saliency prediction to ODIs, where the input 360° image is rotated by several angles and then projected as a set of 2D patches using the ERP. SALICON [Jiang et al., 2015] (a SOTA 2D image saliency prediction model) is applied to each patch separately and the FSM approach fuses local saliency maps to generate the final prediction. The motivation of the approach introduced in [Maugey et al., 2017] is to reduce the border artifact after sphere-to-plane projection.

Unlike previous approaches, [Startsev and Dorr, 2018] combined both the ERP and CMP, to better reduce the negative impact of border artifacts. The former swaps the left and right halves of the image to reduce the distortions on the vertical sides. 2D saliency prediction approaches are applied to obtain two saliency maps, corresponding to the top and bottom faces of the cube, after incorporating CMP. The final saliency map is obtained by pixel-wise maximum multiplication as a method for fusing the two ERP and CMP generated maps.

Other approaches [Lebreton and Raake, 2018, Luz et al., 2017, Cornia et al., 2016], adjusted predictions on the extracted view-ports rather than ERP/CMP

projections, assuming that view-ports feature fewer geometric distortions. The main challenge is how to project several view-ports back into the final spherical saliency map. Rather than adapting 2D saliency prediction approaches on ODIs, some works [Achanta et al., 2012, Thakur et al., 2011, Zhu et al., 2018b, Fang et al., 2018, Battisti et al., 2018] proposed the extraction of handcrafted low-level features such as hue, saturation, luminance, texture, color channels, boundary connectivity, but also high-level features such as skin, faces, and cars. The low- and high-level maps are integrated to obtain the final saliency map.

Few end-to-end learning models have been proposed for 360° saliency. The SaltiNet [Assens et al., 2018] model is initialized with the pre-trained parameters of SalNet [Pan et al., 2016], and then trained over the Salient360! dataset using the binary cross entropy (BCE) loss. The SalNet360 [Monroy et al., 2018] approach trained SalNet on the cube faces of a 360° image under CMP. Then, a fully convolutional network (FCN) is adapted to fuse the spherical coordinates of the cube faces with the extracted saliency maps. The work in [Chao et al., 2018] proposed rotating the 360° image at different CMPs with several angles, then SalGAN [Pan et al., 2017] is fine-tuned on the Salient360! dataset using these projections. Unlike previous DNN approaches, [Suzuki and Yamanaka, 2018] explicitly learns the equator bias with a layer in the proposed CNN architecture, which acts on the viewports for generating the final saliency map of the 360° image. ATSAL [Dahou et al., 2021] combines a latent attention mechanism that allows the network to focus on the most relevant parts of the input space, with expert instances of SalEMA [Linardos et al., 2019] for each patch location produced by the CMP, to learn effective features for saliency modelling.

It is clear from the above review that the models targeting HM/EM 360° image visual attention modeling share the same core concept of applying a CNN on patches from ERP/CMP projections. As outlined above, this design is conceptually limited, and is computationally demanding at the inference stage. This chapter attempt to better address these limitations.

4.3 Method

The proposed algorithm takes advantage of the geometric flexibility of the 360° data domain, i.e. the spherical representation, where the different projections represent robust views for training a differentiable parametric function $f_{\boldsymbol{\theta}}(\mathbf{x})$, with parameters θ (e.g. neural network) to maximize the mutual information among the views without any further supervision. We rely on exploiting contrastive learning-based approaches [Le-Khac et al., 2020] to learn optimal and robust representations for 360° data. To further measure the quality of the latent representations, a separate parametric

function g_ϕ (decoder), is able to decode good quality saliency maps for the downstream task. It is important to mention that the two stages are asynchronous.

4.3.1 Overview of the Approach

Suppose we are given a 360° image dataset, $\mathcal{D} = \{\mathbf{x}_1, ..., \mathbf{x}_{|\mathcal{D}|}\}$ where $\mathbf{x}_i \in \mathcal{R}^{3\times H\times W}$, and a set of transformations $\mathcal{T}$ and projections $\mathcal{P}$, with empirical probability distribution $p(\mathbf{X})$. The set $\mathcal{T}$ contains standard transformations and specifically small random crops (<5% of the image size; large crops can affect saliency [Che et al., 2019]), random jitter in color space, random conversion to gray-scale, random horizontal flips. The set $\mathcal{P}$ specifically contains projections top-to-front and bottom-to-front, from the sphere-to-plane, using the ERP projection. The aim is to learn representations that maximize the agreement between global representations of $\mathbf{x}$ (source view) and local spatial patches of $\mathbf{x}_t \sim \mathcal{T}(\mathcal{P}(\mathbf{x}))$ (augmented view) as in [Hjelm et al., 2019, Bachman et al., 2019]. However, there are some significant differences that distinguish our approach from previous works. As we are mainly inferring for a regression downstream task, we are not concerned with the exact value of the MI, as minimizing further the contrastive objective encourages clusters to form in the representation space. Thus, we aim at optimizing the feature maps across spatial locations to capture enough symmetries about the input data, with the use of both the local-to-global approach and the self-attention module.

4.3.2 Unsupervised Contrastive Module

Base encoder ($\mathbf{f}_{\boldsymbol{\theta}}$) learns a network $\mathbf{f} : \mathbf{x} \mapsto \mathbf{\Lambda}$ parameterised by $\boldsymbol{\theta}$, where $\mathbf{x} \in \mathbb{R}^{3\times 160\times 320}$ and $\mathbf{\Lambda} \in \mathbb{R}^{512\times 10\times 20}$. To be precise, $\mathbf{x}$ (Source view) is the whole panorama and $\mathbf{x}_t$ is perspective image with augmentations as depicted in Figure 4.4 with $f_\theta(\mathbf{x})$ and $f_\theta(\mathbf{x}_t)$ representing their local latent representations ($f_\theta(\mathbf{x}_t), f_\theta(\mathbf{x}) \in \mathbf{\Lambda}$). We report findings for VGG16 [Simonyan and Zisserman, 2015] and ResNet50 [He et al., 2016] as encoders. We made some changes to use ResNet-50 as an encoder and this is depicted in Figure 4.5.

Global module ($\mathbf{\Sigma}_{\boldsymbol{\sigma}}$) learns a mapping $\mathbf{\Sigma} : f_\theta(\mathbf{x}) \mapsto \mathbf{v}_\mathbf{x}$ ($\mathbf{x}$ always represents a panorama image) parameterised by $\boldsymbol{\sigma}$, where $\mathbf{v}_\mathbf{x} \in \mathbb{R}^{512}$. This module provides a compact/global representation of $\mathbf{x}$ as shown in Figure 4.4 and a detailed architecture is shown in Figure 4.5. This module remains the same of for VGG16 and ResNet50. This module can also be understood as a projection layer often used in self-supervised literature but in this case it is asymmetric (i.e., it is not applied to $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_t)$). The Global module is used during pre-training only.

Local module ($\mathbf{\Omega}_{\boldsymbol{\omega}}$) is again a non-linear mapping $\mathbf{\Omega} : \mathbf{\Upsilon}_{\mathbf{x}_t} \mapsto \Psi_{\mathbf{x}_t}$ parameterised by $\boldsymbol{\omega}$, where $\Psi_{\mathbf{x}_t} \in \mathbb{R}^{512\times\times 10\times 20}$. The architecture of the Local module consists of

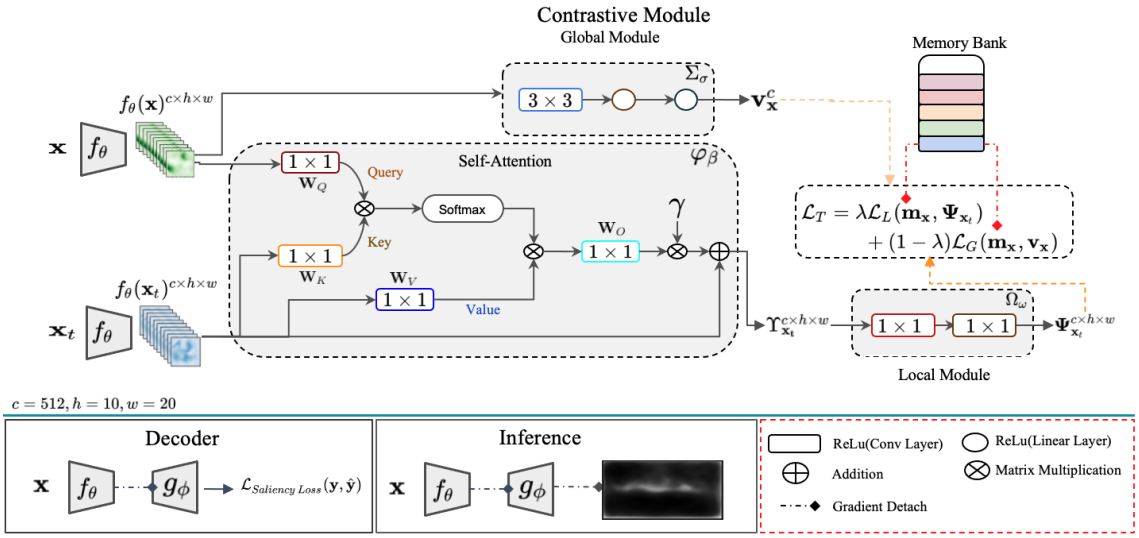


Figure 4.4: Complete pipeline for training. The **contrastive module** comprises of an encoder f_θ, a global module Σ_σ, self-attention φ_β, and local module Ω_ω trained jointly to optimize the set of parameters $\boldsymbol{\Theta} = \{\boldsymbol{\theta}, \boldsymbol{\sigma}, \boldsymbol{\beta}, \boldsymbol{\omega}\}$ in a completely unsupervised regime. **Decoder** g_ϕ is trained to optimize $\boldsymbol{\phi}$, keeping the encoder fixed (no gradient flow). **Inference** can be performed to predict the saliency on unseen data. The `ReLu (Conv Layer)` and `ReLu (Linear Layer)` in bottom right changes depending on different module (see Section 4.3.2).

$2\times$[Conv2d $\rightarrow$ ReLU] followed by a BatchNorm2D and is fixed for both VGG16 and ResNet50. Similar to the Global module, the Local module is also used during pre-training only.

Self-attention module serves as a medium to build spatial relationship between local representations $f_\theta(\mathbf{x})$ and $f_\theta(\mathbf{x}_t)$. The architecture is similar to [Wang et al., 2018a, Zhang et al., 2019] but *query* and *key* come from different sources i.e., $f_\theta(\mathbf{x})$ and $f_\theta(\mathbf{x}_t)$ respectively. Usually *key* and *query* are linear projections of same representations. This self-attention mechanism is the building block of vision transformers [Dosovitskiy et al., 2021]. To give more intuition about how self-attention works in this case; the module takes in the source ($\Lambda_\mathbf{x} = f_\theta(\mathbf{x})$) and augmented ($\Lambda_{\mathbf{x}_t} = f_\theta(\mathbf{x}_t)$) local latent representations, which has dimension $512\times10\times20$. The objective of this module is to provide a mechanism to perform feature selection between the feature maps of two views (source and augmented). To achieve this, the latent representations are transformed into `Q` (query) and `K` (key) feature spaces through weight matrices $\mathbf{W}_\text{Q}$ and $\mathbf{W}_\text{K}$ for source and augmented latent representations respectively. Attention is calculated through dot product between the two representations i.e. $\mathbf{W}_\text{Q}(\Lambda_\mathbf{x})$ and $\mathbf{W}_\text{K}(\Lambda_{\mathbf{x}_t})$. This dot product is calculated between each location "vector" (i.e. for each location $\{(:, i, j) \Big| : i \in \{1, \ldots, 10\}, j \in \{1, \ldots, 20\}\}$) across two views ($\mathbf{W}_\text{Q}(\Lambda_\mathbf{x})$ and $\mathbf{W}_\text{K}(\Lambda_{\mathbf{x}_t})$), resulting in attention weights of size 200×50*. The ouput from $\mathbf{W}_\text{K}(\Lambda_{\mathbf{x}_t})$)

*excluding batch-size for simplicity

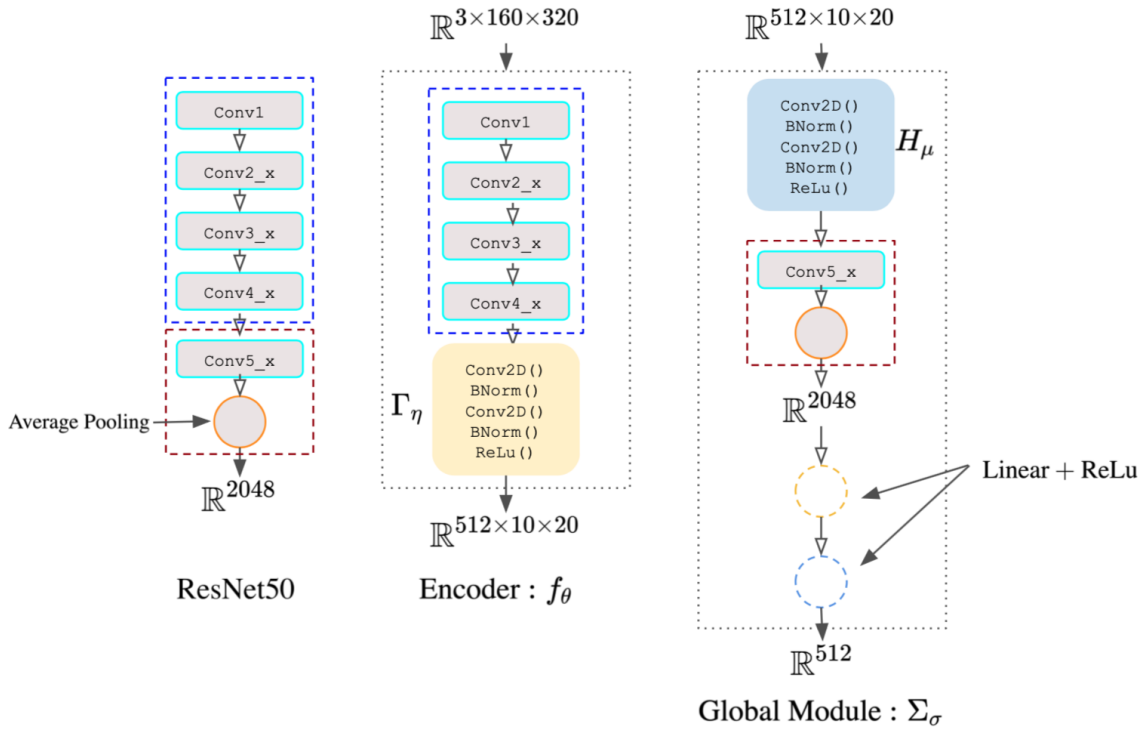


Figure 4.5: Detailed breakdown of the model architecture. Encoder $\mathbf{f}_\theta$ is derived from ResNet50 architecture as well as global module $\sum_\sigma$.

is max-pooled to reduced for spatial resolution which results in an attention matrix of size 200×50 (instead of 200×200). Intuitively, this means performing the dot product of each location vector ($\mathtt{Q}$) with a patch (in $\mathbf{W}_\mathrm{K}(\Lambda_{\mathbf{x}_t})$) because each pixel has a receptive field of 4, due to the max-pooling. Once we have these attention weights we could now perform feature selection through $\mathtt{V}$ (value) i.e. $\mathbf{W}_\mathrm{V}(\Lambda_{\mathbf{x}_t})$. Further, output from (Q,K,V) operation is projected through another weight matrix $\mathbf{W}_O$ and is multiplied by γ and added to $\Lambda_{\mathbf{x}_t}$. This residual connection modulates the value of γ in terms of the extra contribution. In our case γ achieves a value of 8.0 after 250 epochs. The choice to reduce the channel dimensions and applying pooling was done following [Zhang et al., 2019]*.

Loss function: We minimize a NCE [Gutmann and Hyvärinen, 2010] based objective as in [Tian et al., 2019a]. The idea is to run logistic regression to tell apart the target data from noise (negative sample) (Chapter 2), unlike here we developed this objective for m negative samples:

$$\begin{aligned}\mathcal{L}_{\mathrm{NCE}}(\mathbf{x}, \mathbf{x}_t) = -\mathop{\mathbb{E}}_{\mathbf{x}\sim p(\mathbf{x})}\Big\{\mathop{\mathbb{E}}_{\mathbf{x}_t\sim p(.|\mathbf{x})}\big[\log\big(P(D=1|\mathbf{x}_t,\mathbf{x})\big)\big] \\ +m\mathop{\mathbb{E}}_{\mathbf{x}_n\sim p_n(.|\mathbf{x})}\big[\log(P(D=0|\mathbf{x}_n,\mathbf{x}))\big]\Big\}.\end{aligned} \tag{4.1}$$

Optimizing $\mathcal{L}_{\mathrm{NCE}}(\mathbf{x}, \mathbf{x}_t)$ is simply minimizing the negative log-posterior probability

*Implementation is taken from [Zhang et al., 2019] and adapted to PyTorch

of label D to distinguish "positive pair" $(\mathbf{x}, \mathbf{x}_t)$ $(D = 1)$ from "negative pair" $(\mathbf{x}, \mathbf{x}_n)$ $(D = 0)$ where $\mathbf{x}_n$ is often referred to as a negative sample. A negative sample is any sample that is not typically derived from $\mathbf{x}$ and its distortion/augmentation. $p(\mathbf{x})$ and $p_n(.)$ in Equation 4.1 is the empirical data distribution and distribution of noisy samples respectively. The posterior distribution with m noise sample is given by:

$$P(D = 1|\mathbf{x}_t, \mathbf{x}) = \frac{p(\mathbf{x}_t|\mathbf{x})}{p(\mathbf{x}_t|\mathbf{x}) + m\, p_n(\mathbf{x}_n|\mathbf{x})}, \tag{4.2}$$

with $p(\mathbf{x}_t|\mathbf{x})$ being the true unknown distribution, which is approximated by a score function $s(\mathbf{x}_t, \mathbf{x}) = \exp\left(\mathbf{x}_t^T \mathbf{x}/\tau\right)$, where τ is the temperature hyper-parameter (fixed to 0.07) that modulates the distribution. This function assumes L_2 normalized vectors. Refer to [Tian et al., 2019a, Gutmann and Hyvärinen, 2012] for further details on the derivation of the NCE loss.

Memory bank: Following [Wu et al., 2018b, Misra and Maaten, 2020], we maintain a memory bank to retrieve m negative samples which is an exponential moving average of feature representations $\mathbf{v}_\mathbf{x}$(i.e. source view) denoted as $\mathbf{m}_\mathbf{x}$ that were computed in prior epochs.

The **final objective** is defined as a convex combination of both the global($\mathcal{L}_G$) and local($\mathcal{L}_L$) NCE losses:

$$\mathcal{L}(\mathbf{x}, \mathbf{x}_t) = \lambda \mathcal{L}_L(\mathbf{m}_\mathbf{x}, \Psi_{\mathbf{x}_t}) + (1 - \lambda)\mathcal{L}_G(\mathbf{m}_\mathbf{x}, \mathbf{v}_x). \tag{4.3}$$

Note: we do not directly minimize the NCE between global and local representations but instead rely on representations from memory $\mathbf{m}_x$. This first encourages similarity to memory representations encoding invariance, as shown in [Misra and Maaten, 2020], and secondly it directly maximizes the MI between global and local representations via memory representations.

$\mathcal{L}_G$ is the global NCE between two feature vectors $\mathbf{m}_x$ and $\mathbf{v}_\mathbf{x}$ (each $\in \mathbb{R}^{512}$), while $\mathcal{L}_L$ is the local NCE between a vector $\mathbf{m}_x$ and feature map $\Psi_{\mathbf{x}_t} \in \mathbb{R}^{512\times10\times20}$. In this later case the dot product in $s(\mathbf{m}_x, \Psi_{\mathbf{x}_t})$ is calculated as $\frac{1}{hw}\sum_{i=0}^{h}\sum_{j=0}^{w} \mathbf{m}_x^T \Psi(:, i, j)$, which is referred to as *local-dot* encode in [Hjelm et al., 2019]. Recall that the dot-product in the scoring function assumes L_2 normalized vectors, so $\Psi_{\mathbf{x}_t}$ is L_2 normalized along each location i.e. across $\Psi(:, i, j)$. The dimensions $(c, h, w) = (512, 10, 20)$ remains fixed across all settings.

4.3.3 Supervised Module

Problem formulation. Visual attention modelling for ODIs is the downstream task chosen to measure the quality of the representations. The motivation lies with the difficulty of the task, and the availability of benchmarks. It consists of predicting an

(head+eye) based ERP-saliency map from the input 360° image. In this setting, the ground truth saliency maps are computed by convolving each fixation or trajectory points (for all observers of one image), defined as:

$$FM_{ij} = \begin{cases} 1 & \text{if location } (i,j) \text{ is a fixation} \\ 0 & \text{otherwise,} \end{cases}$$

with a Gaussian or Kent kernel. The resulting saliency map $P \in [0,1]^{W \times H}$ can be treated as a multivariate Bernoulli distribution where each pixel is Bernoulli distributed, with a probability p to be attended, and $(1-p)$ to be discarded.

Decoder module g_ϕ : The decoder architecture is inspired from SalGAN; however, we only kept one single convolution layer per block, rather than three layers as in the original SalGAN. The main motivation for this is to avoid over-parametrization, and to show that a less complex function is able to decode the representations and provide evidence of the generality and robustness of Ψ.

Saliency loss function. The saliency task can be seen as a distance measure between the predicted saliency distribution $Y \in [0,1]^{W \times H}$, and the continuous ground truth $P \in [0,1]^{W \times H}$. The objective function must be designed to maximise the in-variance of predictive maps and give higher weights to locations with higher fixation probability. Thus, the decoder is trained to minimize the Kullback-Leibler Divergence (KLD), widely adopted for benchmarking saliency models [Bylinskii et al., 2018], the KLD between Y and P is given by:

$$\mathcal{L}_{\text{KLD}}(Y,P) = \sum_{i=1}^{W \times H} P_i \log\left(\epsilon + \frac{P_i}{\epsilon + Y_i}\right), \tag{4.4}$$

4.4 Experimental Setup

Pre-Training Dataset: We first pre-train the encoder following the unsupervised scheme. Contrastive learning requires a large amount of unlabelled data to be trained effectively. Due to the unavailability of large-scale 360° images datasets, we had to gather a new one with 90K ODIs from multiple sources. The dataset comprises of; PVS: HMEM [Xu et al., 2018a] contains 76 panoramic videos, images were sampled at a rate of 1 frame per second (fps), 360-Indoor [Chou et al., 2020] contains 3024 complex indoor scenes containing common objects and Videos gathered from YouTube playlists* (1 fps).

Downstream Dataset: We evaluate the unsupervised module's representational properties on three saliency datasets namely; **Salient360!** images [Rai et al.,

*Playlist 1, Playlist 2, Playlist 3

2017]: a small-scale dataset, consisting of (80/25) images for training and validation respectively, each recorded for at least 40 observers. It provides the head-eye saliency map obtained jointly from eye tracking and head positions in the ERP format. **Sitzmann** [Sitzmann et al., 2018] containing a total of (14/11) training/validation ODIs; the authors captured and analyzed gaze and head orientation data of 169 users. Due to the small amount of labelled static data (103 ODIs), we sampled at a rate of 1 fps from the large-scale video dataset **VR-EyeTracking** [Xu et al., 2018b]. The resulting set contains (4700/1300) 360° images.

Evaluation: Our approach is experimentally compared to five models, two state-of-the-art 2D static saliency models, UNISAL [Droste et al., 2020] and SalGAN, and three 360° specific models: ATSAL [Dahou et al., 2021], SalGAN360 [Chao et al., 2018] and SaltiNet [Assens et al., 2018]. This choice is motivated by the availability of the source code. All approaches were evaluated according to five different saliency metrics: Normalized Scanpath Saliency (NSS), Kullback-Leibler Divergence (KLD), Similarity (SIM), Linear Correlation Coefficient (CC), and AUC-Judd (AUC-J). Please refer to [Bylinskii et al., 2018] for an extensive review of these metrics.

Implementation Details: Both the contrastive encoder and the supervised decoder were implemented in PyTorch, and trained using two GPUs (RTX 3090 & RTX 2080 Ti). The contrastive encoder was optimized using SGD with a learning rate of 10^{-2}. The encoder was trained for 250 epochs using the max batch size of 80*, with negative samples fixed to 16000, $\tau = 0.07$ and $\lambda = 0.7$. Choices about total epochs and the number of negative samples are based on computation and time constraints. Training for more epochs or with large negative samples may provide a further boost in the performance [Kolesnikov et al., 2019, Misra and Maaten, 2020].

Finally, the decoder was optimized using Adam [Kingma and Ba, 2015] with a learning rate of 10^{-4} for 100 epochs. It is worth mentioning that the unsupervised encoder weights were not *fine-tuned* on the downstream task; only the randomly initialized decoder is trained on top of the frozen encoder. The main motivation for this is to set a robust evaluation procedure and to prevent the encoder from adapting its parameters to saliency specific requirements.

4.5 Results

4.5.1 Quantitative Assessment

Table 4.1 shows the comparative study with the aforementioned models according to the different saliency metrics on Salient360! and (VR-EyeTracking+Sitzman) datasets 25/1300 test 360° images. Our model is very competitive in the two

*maximum batch size that could be achieved given our constraints

Table 4.1: Comparative performance study on: Salient360! and VR-EyeTracking datasets. Training setting$_{(i)}$: trained w/o self-attention, Training setting$_{(ii)}$: trained w/ self attention. The best scores are marked in **bold** and second best in blue.

	Model	Salient360!					VR-EyeTracking+Sitzman				
		AUC-J ↑	NSS ↑	CC ↑	SIM ↑	KLD ↓	AUC-J ↑	NSS ↑	CC ↑	SIM ↑	KLD ↓
2D models	UNISAL [Droste et al., 2020]	0.701	1.404	0.389	0.435	2.519	0.783	1.918	0.276	0.242	9.044
	SalGAN [Pan et al., 2017]	0.701	1.398	0.377	0.483	1.544	0.718	1.023	0.145	0.152	10.195
360° models	ATSAL [Dahou et al., 2021]	0.777	**1.638**	**0.642**	**0.639**	**0.761**	0.822	1.613	0.239	0.191	9.796
	SalGAN360 [Chao et al., 2018]	**0.831**	1.598	0.639	0.611	0.798	0.704	1.267	0.226	0.218	7.938
	SaltiNet [Assens et al., 2018]	0.702	1.057	0.536	0.541	1.098	0.674	0.967	0.186	0.198	9.938
Training setting$_{(i)}$	VGG	0.758	1.557	0.553	0.585	0.909	0.841	1.583	0.246	0.221	7.965
	ResNet	0.756	1.524	0.520	0.54	1.039	0.833	1.545	0.232	0.203	8.574
Training setting$_{(ii)}$	VGG	0.760	1.548	0.538	0.569	0.922	0.867	1.880	0.308	0.234	7.583
	ResNet	0.769	1.601	0.584	0.591	0.849	**0.869**	**2.089**	**0.329**	**0.248**	**7.110**

Table 4.2: GPU inference time comparison of video saliency prediction methods (NVIDIA RTX 3090). All methods are reported based on the Salient360! benchmark [Xu et al., 2018b]. The best computational performance among dedicated 360° models is shown in bold. (*) represents 2D models.

Model	Runtime (s)
SalGAN360 [Chao et al., 2018]	14.330
ATSAL [Dahou et al., 2021]	0.230
SaltiNet [Assens et al., 2018]	0.450
(*) SalEMA [Droste et al., 2020]	0.020
(*) UNISAL [Pan et al., 2017]	0.010
Ours (ResNet-based)	**0.025**

datasets, and exhibits the top score for all metrics on VR-EyeTracking+Sitzman. As expected, 2D SOTA approaches fail to generalize on ODIs, which questions the effectiveness of the direct transfer of visual attention features from 2D to 360° data. *VR-EyeTracking & Sitzman:* The 1300 validation/test images were sampled from the 75 diverse test ODVs of the first dataset, mixed up with the 11 images from Sitzman, making the prediction task very challenging. It can be seen that both VGG and ResNet-based models outperform 2D saliency models by a good margin, with a significant improvement over 360° specialized models, trained in an end-to-end scheme on supervised data. The ResNet-based model trained with self-attention achieves the best results following: (KLD ↓ = 7.110). *Salient360!:* We used the 25 (360° validation) images for testing the model*. The model was not trained on this dataset, making this an *out-of-distribution* test. The proposed model produces a reasonable improvement in accuracy compared to other models, except SalGAN360 and ATSAL, which were trained on this specific dataset.

Computational load. As model efficiency is a key factor for real-world ODIs applications, Table 4.2 shows a GPU runtime comparison (processing time per 360°

*The reserved test set was unavailable due to COVID-19

Table 4.3: Comparative performance study on: VR-EyeTracking datasets. $VGG_{(i)}$/ $VGG_{(ii)}$ following training setting (i)/(ii)

Model		**VR-EyeTracking+Sitzman**				
		AUC-J ↑	NSS ↑	CC ↑	SIM ↑	KLD ↓
VGG_(i)	$\lambda = 0.5$	0.852	1.872	0.306	0.237	**7.540**
	$\lambda = 0.7$	0.841	1.583	0.246	0.221	7.965
	$\lambda = 0.9$	0.849	1.825	0.278	0.226	7.875
VGG_(ii)	$\lambda = 0.5$	0.860	**1.894**	0.307	0.231	7.648
	$\lambda = 0.7$	0.867	1.880	**0.308**	0.234	7.583
	$\lambda = 0.9$	0.860	**1.894**	0.301	**0.241**	7.648

image) of the different competitors on the 4K Salient360! ODIs. Compared with other 360° specialized models, our model exhibits a remarkable improvement, being over 8× faster than ATSAL, which is the fastest model in this category. This is an important finding from the perspective that a careful design not only leads to better performance but at the same time could lead to better efficiency. One thing to note about the encoder $\mathbf{f}_\theta$ is that it's not even a complete ResNet50 yet it performs on par with other models. This further strengthens our intuition about designing efficient self-supervised models giving better inference efficiency during test times. This is an important aspect essential for real world applications.

4.5.2 Qualitative Assessment

Figure 4.6 illustrates the prediction task on a sample of 360° images from two datasets: Salient360! and VR-EyeTracking. It can be seen that the saliency maps generated by our model (ResNet-based with self-attention) correlate well with the ground truth maps in terms of fixation distribution. Other competitors shown in the same figure overestimate saliency in general, or overly bias it to the equator/center. Furthermore, the effectiveness of the predictor in capturing the main objects in the scene can be observed. Another key point is the model's capacity to accurately detect saliency in the Zenith and Nadir without using any form of projections at inference time (see Figure 4.7). This demonstrates the effectiveness of the contrastive encoder in embedding the views as a superposition in the function weights and biases. However, it is important to point that the bias towards the equatorial region still exists which is visible in Figure 4.6.

4.6 Ablations

In this section we justify the choices by ablating key features of the procedure.

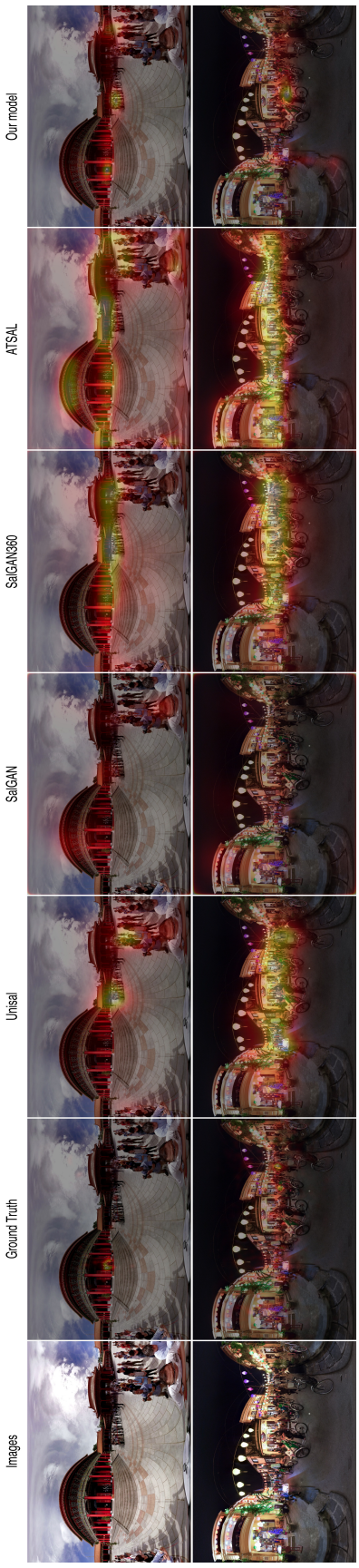


Figure 4.6: Qualitative results of our model and four other competitors on sample images from VR-EyeTracking and Salient360! datasets. It can be observed that the proposed approach is able to handle various challenging scenes well and produces consistent saliency maps.

Table 4.4: Results on Salient360! validation images for a model based on a contrastive encoder trained with/without projections.

	Salient360!				
	AUC-J ↑	NSS ↑	CC ↑	SIM ↑	KLD ↓
ResNet w/o	0.736	1.524	0.479	0.536	0.999
ResNet w/	**0.769**	**1.601**	**0.584**	**0.591**	**0.849**

What is the effect of λ in the loss functions? The total loss is a convex combination with a hyper-parameter λ, that trade-offs between the two NCE losses namely global ($\mathcal{L}_{\mathrm{G}}$) and local ($\mathcal{L}_{\mathrm{L}}$) NCE. As depicted in Table Table 4.3 we varied λ to 0.5, 0.7, and 0.9. As the results suggests increasing λ improves the performance on the downstream task. Intuitively if we look closely in 4.3, giving more emphasis to $\mathcal{L}_{\mathrm{G}}$ biases the function to learn trivial solution as $\mathbf{m_x}$ is moving average of $\mathbf{v_x}$, as result this leads to much easier classification task. However, more emphasis on $\mathcal{L}_{\mathrm{L}}$ makes the classification task (NCE optimization) more difficult because the optimizer is maximizing the agreement from different view of the same scene (object) and that too locally. This leads to better expressive power and generalisation capability. We chose $\lambda = 0.7$ for all our previous evaluations.

Training with/without self-attention: we evaluate the models to observe the effect of self-attention in the learning regime. A performance boost is observed irrespective of λ when using self-attention. Table 4.1 also shows results for models trained with and without attention. Self-attention/attention is intuitively motivated by how we humans pay attention to specific regions or parts of images and try to correlate among them. At the same time, this correlation can be extended to different images/patches of same view. In summary, it helps to infer a patch/region in an image based on this correlation (importance vectors). Given that the objective is to maximize the agreement between two views, the self-attention module serves as a mode of finding the best correlation in the two views in terms of information shared among them by performing this soft feature selection mechanism across the views.

Training with/without projections: to further showcase the importance of projections, we trained a model without any projections (only augmentations). One could also argue training without augmentations and projections will lead us to NPID [Wu et al., 2018b] ($\lambda = 0$) and this is not our objective. Table 4.4 depicts the results for this setting. Performance drops when projections are removed, validating the hypothesis that using different projections in addition to augmentation is natural for 360° images and produces representations that are more effective in downstream tasks. Intuitively by removing projections we make the feature extractor f_θ prone to exploiting low-level visual features such as color aberrations as observed in [Doersch et al., 2015, Noroozi and Favaro, 2016], and not learning useful semantic

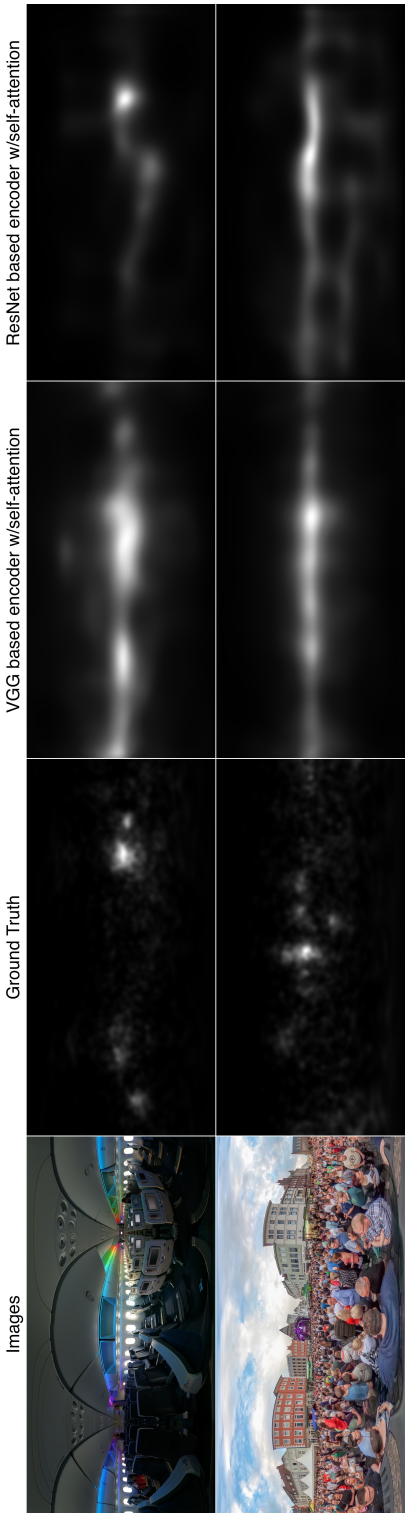


Figure 4.7: Predicted saliency maps from Salient360! samples.

representations, resulting in a performance drop on the downstream task. This phenomenon of relying on low-level features given large unlabelled datasets is often referred to as short-cuts in the unsupervised learning literature [Minderer et al., 2020, Chen et al., 2020a]. This experiment further validates our hypotheses of exploiting the top/bottom to equator projections for contrastive learning.

4.7 Discussion

2D vs 360° models. Deep learning based saliency models [Assens et al., 2018, Pan et al., 2016, Chao et al., 2018, Suzuki and Yamanaka, 2018] trained end-to-end on 360° datasets show remarkable improvements over early models adapting 2D approaches on ODIs [De Abreu et al., 2017, Maugey et al., 2017, Startsev and Dorr, 2018, Lebreton and Raake, 2018, Cornia et al., 2016, Luz et al., 2017]. This demonstrates the new constraints imposed by the spherical domain when modelling visual attention.

Supervised vs Unsupervised learning. Our approach is an opening for a new line of research exploring the subtle definition of gaze policy naturally embedded in the brain. In fact, the early research into how the human visual system functions, produced many interesting results, demonstrating that visual attention could be influenced by regions that maximize a reward in a task-driven scenario [Sprague and Ballard, 2004], which are typically the most informative regions [Itti and Baldi, 2009, Bruce and Tsotsos, 2006]. This suggests that saliency can be disentangled into low-level (e.g. color, intensity, etc.) and high-level (e.g. human faces) features [Itti and Koch, 2000, Borji and Itti, 2012]. Research in cognitive science (e.g. [Yan et al., 2018, Li, 2002, Veale et al., 2017]) indicates that low-level saliency in both humans and animals happens early in the primary visual cortex, suggesting that it can potentially be learned without supervision. We believe this could be an important future research direction.

Generalization to other 360° downstream tasks. The last decade has witnessed many works on 360° video/image processing including, visual attention, visual quality assessment (VQA), and compression [Xu et al., 2020b]. Visual attention can serve as a tool for compression approaches (e.g. saliency-aware adaptive coding [Zhu et al., 2018a, Luz et al., 2017, Liu et al., 2018]). Perceptual approaches for VQA require predicted saliency maps as weight maps [Ma and Zhang, 2008, Li et al., 2019]. Generally, the spherical characteristics of the input data mean that many of these tasks face the challenges addressed in this work. Thus, we suspect that the contrastive encoder can indeed be exploited for VQA and compression.

4.8 Conclusion

In this chapter, by exploring **H2** as introduced in Chapter 1 we are able to answer **RQ2** by proving that we can indeed efficiently design and effectively exploit contrastive self-supervised methods for a granular task like 360° (omnidirectional images) visual saliency prediction. We successfully introduce a method for modeling human visual attention with contrastive self-supervised learning, which improves the

generalization and expressive power of the model. In designing the self-supervised model for saliency prediction, instead of just performing Global-to-Global feature learning, we also rely on Global-to-Local feature learning. The approach exploits the geometric flexibility of the spherical data to learn representations that contain locally consistent information across the views. The qualitative and quantitative results on the downstream saliency task demonstrate the competitiveness of the approach. Interestingly, the experiments conducted suggest that it is possible to design efficient self-supervised models that also can provide efficiency during inference.

At this point, it is important to discuss the direction of this thesis. As this chapter provides two major research directions, one entirely exploring and developing self-supervised models for 360^0 visual saliency prediction and the other promising direction is designing efficient ways of designing self-supervised models for efficient inference in resource-constrained industrial settings. The efficiency aspect of (at least for contrastive) self-supervised learning was demonstrated in this work by carefully designing the encoder network. However, this manual designing might not always be straightforward as there is no set rules to design that. Motivated by this, in the following chapter we explore how to design efficient dynamic self-supervised models.

Chapter 5

Dynamic Channel Selection for Self-supervised Learning Models

Effectively designing self-supervised models suited for a particular downstream task is an open area of research. However, in Chapter 4 we show that effectively designing a contrastive self-supervised model for 360° image visual saliency prediction not only leads to better performance but also faster inference (specifically a small variant of the ResNet50 architecture as the encoder as well as a much simpler decoder module). However, it is difficult to hand-design the width of the channels to be retained or to decide how many layers to incorporate. To this end, in this chapter, we explore **H3** as introduced in Chapter 1 by investigating whether self-supervised models learn highly redundant channel features.

A self-supervised network that can dynamically select the important channels and remove the unnecessary ones would be very attractive in practical deployment settings. As shown in Chapters 3 and 4, DNNs pre-trained with self-supervision can achieve comparable performance on downstream tasks compared to their supervised counterparts. However, there are drawbacks to self-supervised models including their large numbers of parameters, computationally expensive training strategies, and a clear need for faster inference on downstream tasks. In this chapter, our goal is to address the latter by studying how a standard channel selection method developed for supervised learning can be applied to networks trained with self-supervision. We validate our findings on a range of target budgets t_d for channel computation on image classification tasks across different datasets, specifically CIFAR-10, CIFAR-100, and ImageNet-100. We obtain comparable performance to that of the original network when selecting all channels but at a significant reduction in computation reported in terms of FLOPs. The research that emanated from this work was published at 24^{th} Irish Machine Vision and Image Processing Conference 2022, Belfast, Northern Ireland.

5.1 Motivation

Self-supervised pre-training of convolutional neural networks, as mentioned in [Chen et al., 2020a], has almost matched the performance of supervised pre-training on the ImageNet [Deng et al., 2009] image classification task, but at a cost of a huge number of parameters and inefficient training and inference methods. In the supervised learning setting, as described in [Veit and Belongie, 2018], it is accepted that networks with dynamic data-dependent (conditional) channel computation architecture during inference can lead to enhanced representation power, adaptivity, interpretability and can greatly reduce computation cost and memory resources without compromising on the accuracy by a significant margin. This motivates us to investigate the behavior of neural networks with a channel selection mechanism trained under self-supervision. We hypothesize that self-supervised models are an ideal candidate for such dynamic network structures as they capture highly redundant channel features during pre-training. In addition, there is also a great need to explore more efficient inference methods on downstream tasks for SSL.

In order to establish the trade-off between computation and performance, there are two well-established research directions when it comes to introducing channel sparsity using dynamic structure in neural networks: channel pruning and channel conditional computation. Dynamic channel pruning, as reported in [Gao et al., 2018], estimates channel saliency measures and allows a network to learn and prioritize certain channels and ignore the irrelevant ones given a fixed target density. Models based on pruning usually learn sparsity through a three-stage pipeline i.e., pretrain-prune-finetune while in other works like [Tiwari et al., 2021] the pruning stage itself consists of two steps, namely soft pruning and hard pruning. Conditional channel computation as proposed in [Herrmann et al., 2020] learns to compute only a subset of channels in every layer for the given input and hence benefits inference time efficiency and provides an insight into dataset-specific network behavior. Both channel pruning and conditional channel computation are categorical decisions that cannot be optimized by gradient descent methods; however, using the Gumbel-Softmax trick from [Jang et al., 2017] provides a way to overcome this challenge. Adafuse [Meng et al., 2020] proposed an adaptive temporal fusion network that learns a decision policy to dynamically fuse channels from current and history feature maps (i.e. dynamically deciding which channels to keep, reuse or skip per layer and per instance) for action recognition. Notwithstanding these works, the use of dynamic networks for channel selection has to date been mostly limited to supervised learning settings only.

To the best of our knowledge, there is no study on the impact of conditional channel selection on SSL. The work described in [Caron et al., 2020b] studies the

effect of standard pruning techniques developed for supervised learning on a network trained with self-supervision. In particular, they use an iterative magnitude based pruning technique described in [Han et al., 2015], which compresses the network by alternatively minimizing a training objective and pruning the network parameters with the smallest magnitude. The weights of the resulting sub-network are reset based on a weight initialization scheme: the lottery winning ticket [Frankle and Carbin, 2018, Frankle et al., 2020]. We adopt a similar strategy but focus on exploring the application of standard conditional channel selection methods, as proposed in [Li et al., 2021], to self-supervised models during the pre-training stage and do not include any re-training. Our contributions can be summarised as follows:

1. *Do self-supervised models learn redundant channel features*? Through our exhaustive evaluation, we demonstrate that the SSL model (SimSiam) does indeed learn redundant channel features.

2. We show in Table 5.1 that exploiting this redundancy leads to a drop in computational complexity (FLOPs), reducing inference time without excessively increasing training time, as we learn from scratch and on-the-fly, unlike competing approaches [Caron et al., 2020b] that involve re-training.

3. We demonstrate that this channel selection mechanism preserves the feature quality when evaluated on the task of image classification and gives comparable performance when compared with a vanilla (no channel selection) SSL approach.

5.2 Related Works

5.2.1 Dynamic Channel Computation

Channel Pruning. Channel pruning estimates channel saliency measures and eliminates all input and output connections from unimportant channels. The approach reported in [Wen et al., 2016] added group Lasso (a regularisation technique used to enforce sparsity across groups of parameters in a model) on channel weights to the model's training loss function resulting in a reduction of the magnitude of channel weights during training. The authors in [He et al., 2018] proposed pruning channels using thresholds by setting unimportant channels to zero. Network Slimming [Liu et al., 2017] used Lasso regularisation with global thresholds. However, deep models pruned with structured sparsity methods lose their capabilities and connections permanently. As a result, dynamic channel pruning methods were devised that learn sparsity through a three-stage pipeline pretrain-prune-finetune or use pretrained models. The authors of [Gao et al., 2018] propose feature boosting and suppression (FBS) to dynamically amplify and suppress output channels computed

by convolutional layers. [Tiwari et al., 2021] presents a deterministic pruning strategy using the continuous heaviside function and *crispness loss* to identify a highly sparse subnetwork from an existing dense network.

Conditional Channel Computation. Regarding conditional computation at the channel level, the work proposed in [Lin et al., 2017] generates decisions to skip computation for a subset of output channels. The channel gating network [Hua et al., 2019] finds regions among the features that contribute less to the classification result and skips computation on a subset of the input channels for these ineffective regions. ConvAIG [Veit and Belongie, 2018] introduced a network with a hard attention mechanism that adaptively selects specific layers of importance for each input image to assemble an inference graph by specifying a target rate for each layer. The authors of [Herrmann et al., 2020] also study conditional computation at the channel level and extend ConvAIG by learning target rates for each gate by specifying the target rate for the whole network. DGNet [Li et al., 2021] proposed a dual gating mechanism by introducing sparsity along two separate dimensions, spatial and channel, in order to reduce model complexity at run time. For a more detailed background on sparsity, pruning and conditional computation, we recommend the review work presented in [Hoefler et al., 2021].

While [Caron et al., 2020b] studied the behaviour of self-supervised models under standard pruning techniques, we investigate the effect of standard channel selection methods described in DGNet [Li et al., 2021] on self-supervised models. We also analyse whether networks trained under self-supervision with channel selection can preserve performance on downstream tasks.

5.3 Method

5.3.1 Self-supervised Module

In this work we consider SimSiam [Chen et al., 2020a] as our self-supervised objective. We use ResNet18 as a base encoder (across all experiments), which takes two augmented views $\mathbf{x}_1$ and $\mathbf{x}_2$ from an anchor view $\mathbf{x}$ by applying stochastic augmentation from a set of augmentations $\mathcal{P}$. $\mathcal{P}$ comprises random resized crop, color jitter, random gray scale, Gaussian blur and random horizontal flip. These augmented views are processed through $f_{\boldsymbol{\theta}}$ to get a compact representation of $f_{\boldsymbol{\theta}}(\mathbf{x}_1), f_{\boldsymbol{\theta}}(\mathbf{x}_2) \in \mathbb{R}^{512}$. One view is further processed by a prediction MLP head (bottleneck architecture) $g_{\boldsymbol{\phi}}$ giving rise to an asymmetric architecture i.e. $\mathbf{p}_1 \triangleq g_{\boldsymbol{\phi}}(f_{\boldsymbol{\theta}}(\mathbf{x}_1)))$ and $\mathbf{z}_2 \triangleq f_{\boldsymbol{\theta}}(\mathbf{x}_2)$. As a standard practice, a base encoder is augmented with a projection head MLP i.e., $f_{\boldsymbol{\theta}} = h \circ m$, where m and h represents ResNet18 and projection layers respectively.

The SimSiam learning objective simplifies to a symmetric cosine similarity:

$$\mathcal{L}_{\texttt{SSL}} = \frac{1}{2}\mathcal{D}(\mathbf{p_1}, \texttt{SG}(\mathbf{z_2})) + \frac{1}{2}\mathcal{D}(\mathbf{p_2}, \texttt{SG}(\mathbf{z_1})), \tag{5.1}$$

where $\mathcal{D}(\mathbf{a}, \mathbf{b}) = -\mathbf{a}^T\mathbf{b}$, with $\mathbf{a}$ and $\mathbf{b}$ being L_2 normalised vectors (i.e., $\mathcal{D}(\mathbf{a}, \mathbf{b})$ is negative cosine similarity.). `SG` stands for `Stop-Grad()`.

5.3.2 Channel Selection via Gating

Preliminaries. Channel selection or conditional computation (data dependent gates) is often realized through a gating mechanism. A typical output for an input $\mathbf{x}$ from a convolutional (conv) layer l is given by $f_l(\mathbf{x}_{l-1}) \in \mathbb{R}^{C\times H\times W}$ where $f_l(\mathbf{x}_{l-1})$ consists of a convolution operation with kernel size k followed with a batch normalization layer (BN) and relu $((\cdot)_+)$ non-linearity with $\mathbf{x}_{l-1}$ being the output from the previous layer. The output from a gated convolutional network can be realized as:

$$\hat{f}_l(\mathbf{x}_{l-1}) = \pi_l(\mathbf{x}_{l-1}) \cdot \text{BN}(\texttt{conv}_l(\mathbf{x}_{l-1}))_+, \tag{5.2}$$

where $\pi_l(\mathbf{x}_{l-1}) \in \{0, 1\}^C$ * is a gate dependent on input $\mathbf{x}_{l-1}$, which decides whether to keep ("on") or discard ("off") a particular channel. This can be seen as a form of *hard attention* (mask). This masking imposes a discrete structure on the network, resulting in different computational graphs for training and inference. During training this structure is achieved through stochastic gradient descent (SGD), while during inference it works as *hard attention*. One of the main reasons for channel selection is to induce sparsity i.e. operate on a lower computational budget (fewer FLOPs) during inference. In this work we closely follow DGNet [Li et al., 2021] using ResNet18 as our base encoder.

Channel Selection (Gating). In order for gates (channel selection) to be effective, they need to estimate the importance of input features. This *importance* is often referred to as relevance/saliencies (vectors) of the input feature map (along the channels) in the literature. This relevance is crucial in order for the network to avoid trivial solutions. A simpler way is to use SE block [Hu et al., 2018], as was used in DGNet [Li et al., 2021], is to create a relevance vector. This usually requires getting a context vector $\mathbf{z} \in \mathbb{R}^C$ via global average pooling to accumulate spatial information. Finally, this context vector $\mathbf{z}$ is passed through a lightweight network to get channel attention $g_l(\mathbf{x}_{l-1})$, which can be summarized as:

$$g(\mathbf{x}_{l-1}) = \mathbf{W}_1\Big(\text{BN}\big(\mathbf{W}_0\mathbf{z}\big)\Big)_+, \quad \mathbf{W}_1 \in \mathbb{R}^{C_l\times\frac{C_{l-1}}{r}}, \mathbf{W}_0 \in \mathbb{R}^{\frac{C_l}{r}\times C_l}, \tag{5.3}$$

*(a vector of dimension equivalent to number of channels with ones and zeroes)

where r is a reduction ratio. For more details please refer to [Hu et al., 2018]. Finally, to achieve binary mask $\pi_l(\mathbf{x}_{l-1})$ we can use the channel attention $g_l(\mathbf{x}_{l-1})$ and set $\pi_l^i(\mathbf{x}_{l-1})$:

$$\pi_l^i(\mathbf{x}_{l-1}) = \begin{cases} 1, & \text{if } g_l^i(\mathbf{x}_{l-1}) \geq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{5.4}$$

This discrete selection works during inference but it breaks the computational graph during training. To make the training possible, the Gumbel-SoftMax Trick [Jang et al., 2017] is adopted. The Gumbel-Trick has been widely used as reparameterisation technique for the task of dynamic channel selection [Li et al., 2021, Herrmann et al., 2020, Veit and Belongie, 2018, Meng et al., 2020]. A gating block is introduced after the first convolution in `Basic Block` of Resnet18 following DGNet. Intuitively, the channel selection network could be interpreted as learning a policy whether to keep (compute) or discard (skip) a particular channel.

5.3.3 Optimisation

To remove unimportant channels and induce sparsity in the gating mask $\pi_l(\mathbf{x}_{l-1})$ we need to add an objective based on some budget t_d. To this end we use regularisation, a term used in DGNet [Li et al., 2021] as sparsity objective, which is a combination of sparsity and a bound regularisation term:

$$\mathcal{L}_G = \lambda \underbrace{\left(\frac{\sum_{l=1}^{L} F_l^R}{\sum_{l=1}^{L} F_l^O} - t_d \right)^2}_{\text{Sparsity}} + \gamma \mathcal{L}_{Bound},$$

where F_l^R is the average FLOPs over the batch along with FLOPs computation of the gating block, while F_l^O is the original FLOPs without a gating module. Only the blocks with gating modules take part in FLOP computation as they are responsible for any sort of sparsity introduced in the network. The purpose of $\mathcal{L}_{Bound}$ is to control early optimization as detailed in DGNet [Li et al., 2021]. It is important to note that apart from FLOPs, there are other different metrics to measure the efficiency of the network e.g., *latency, throughput, energy.* Objectively comparing efficiency can be challenging due to the diverse and often confusing metrics used for evaluation. Some metrics depend on hardware, while others do not, with some focusing on memory and compute performance and others on power consumption. For instance, latency and throughput are critical for evaluating how well a model performs in real-time applications during inference, but these are hardware dependent while FLOPs is

Table 5.1: Performance comparison of SimSiam [Chen and He, 2021] with dynamic channel selection during inference. Evaluated with k-nearest neighbors ($k = 1$) on the validation set of CIFAR-10, CIFAR-100 and ImageNet-100 across various target budgets t_d.

Dataset	**Budget** (t_d)	**Acc**%	**FLOPs**
CIFAR-10	`Baseline`	85.46%	7.03E8
	10%	76.72%	6.64E7 (90.55%↓)
	20%	78.78%	1.25E8 (82.11%↓)
	30%	80.82%	2.01E8 (71.41%↓)
	40%	81.35%	2.66E8 (62.18%↓)
	50%	81.93%	3.29E8 (53.15%↓)
	60%	82.96%	3.94E8 (43.89%↓)
	70%	**83.08**%	4.58E8 (34.76%↓)
CIFAR-100	`Baseline`	52.96%	7.03E8
	10%	46.84%	6.88E7 (90.21%↓)
	20%	49.50%	1.48E8 (78.88%↓)
	30%	50.70%	2.03E8 (71.01%↓)
	40%	52.05%	2.65E8 (62.32%↓)
	50%	52.54%	3.25E8 (53.67%↓)
	60%	53.18%	3.95E8 (43.73%↓)
	70%	**53.50**%	4.66E8 (33.69%↓)
ImageNet-100	`Baseline`	64.34%	1.81E9
	30%	56.08%	5.30E8(70.43%↓)
	40%	57.86%	7.13E8 (60.66%↓)
	50%	**60.38**%	8.78E8 (51.55%↓)

hardware independent.

Final Objective. The overall training objective is defined as: $\mathcal{L} = \mathcal{L}_{\text{SSL}} + \mathcal{L}_G$, with $\lambda = 5$ and $\gamma = 1$ across all the datasets and training regimes.

5.4 Experimental Setup

Implementation Details. We closely follow the approach in DGNet for channel selection. For training, we use SimSiam as a self-supervised model with ResNet18 as a base encoder whose objective is modified as explained in section 5.3.3. We train the model with varying target densities t_d. The implementation of SimSiam is based on the solo-learn library [da Costa et al., 2022]. The base encoder is *randomly initialized*[*] and is trained with SGD for 500 epochs (for a given target budget t_d) with a batch size of 256 on two NVIDIA 2080Ti GPUs, with a warm-start of 10 epochs following a cosine decay with base learning rate of 0.01. Since we are using a very lightweight model as our gating network, there is no significant computational

[*]default initialization in Pytorch

overhead during training. We report the inference speedup in terms of the hardware-independent theoretical metric of FLOPs and not wall-clock time as we are not using any hardware accelerators to utilize sparsity during training. Code is made available at: `https://github.com/KrishnaTarun/SSL_DGC`.

Evaluation. Training and evaluation is carried on train and validation data of CIFAR-10/100 [Krizhevsky and Hinton, 2009] and ImagNet-100* respectively. For Cifar-10/100 we train for $t_d = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$ while for ImageNet-100 we restrict t_d to only $\{0.3, 0.4, 0.5\}$ due to computational constraints. We use K-nearest neighbors (often denoted as KNN is a machine learning technique that uses proximity to classify or predict the grouping of data points) as our evaluation metric evaluated with $k = 1$. For *baseline* we train Simsiam with standard objectives without any channel selection for each of the datasets under consideration.

Results. Our main findings based on the evaluation criteria validate our initial hypothesis that self-supervised models can learn highly redundant channel features.

Table 5.1 shows that in the case of CIFAR-10, by keeping only 70% of the channels across the whole network, SimSiam achieves 83.08% accuracy on the KNN task, which is a minor drop from the baseline performance of 85.46% but at an ample reduction of 34.76% in FLOPs. Furthermore, we also find that an enormous 90.55% of FLOPs can be reduced by using only 10% of the channels across the whole network causing a drop of only 8.74% in KNN accuracy. For CIFAR-100, we found that by restricting the channel usage to only 60% over the whole network, SimSiam surpasses the baseline KNN accuracy of 52.95% by 0.22% reaching 53.18%. Additionally, FLOP computation can be reduced by 90.21% by keeping only 10% of the channels, leading to a drop of only 6.12% in KNN accuracy. On ImageNet-100, 50% of the channel usage in the entire network results in 60.38% KNN accuracy, which is 3.96% less than the baseline. However, this decrease in accuracy is compensated by ∼51.55% percent drop in FLOPs. Aside from this, we get a substantial 70.43% drop in FLOPs by fixing channel utilization to only 30% in the whole model. Therefore, channel selection can be thought of as a way to take advantage of the trade-off between performance and computation depending the downstream task and individual use case. These results also show that SSL models trained with channel selection preserve the performance in downstream tasks. Figure 5.1 shows the channel activation distribution for CIFAR-10, CIFAR-100 and ImageNet-100 datasets, revealing a deeper insight into the dataset-specific behavior of the channel selection network by visualizing how many channels in each ResNet18 blocks are always off (skipped), always on (computed), or input dependent. We notice that the significant number of channels are switched off and switched on all the time in all the three datasets and while others are input dependent. The channel distribution for CIFAR-10 and CIFAR-100 are very similar, which might

*`https://www.kaggle.com/datasets/ambityga/imagenet100`

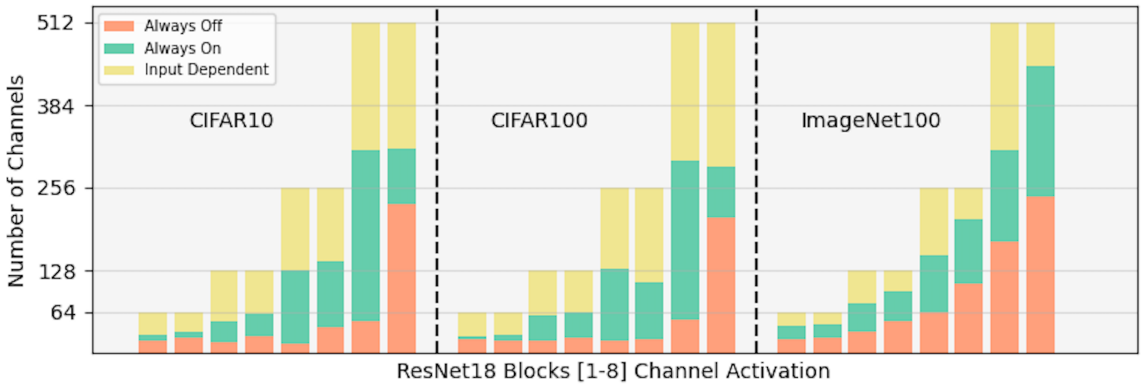


Figure 5.1: Channel distribution over validation set for $t_d = 0.5$ on CIFAR-10, CIFAR-100, ImageNet-100

be due to the fact that image statistics in both of these datasets are similar.

5.5 Discussion and Conclusion

In this chapter, we study the behaviour of self-supervised learning when integrated with channel selection networks given a global target budget for computational cost as our core objective. We validate our hypothesis (**H3**, defined in Chapter 1) that self-supervised models are an ideal candidate for such dynamic network structures as they *might* capture highly redundant channel features during pre-training. Empirical results provided interesting insights about self-supervised learning when trained with channel selection. First, self-supervised models learn highly redundant channel features that can be discarded to reduce computational overhead (Figure 5.1, Table 5.1). Second, we show that channel selection modules can significantly reduce FLOP computation and make inference more efficient (Table 5.1). Third, our results also provide intuition that representations learnt by self-supervised networks with channel selection can also be transferred to downstream tasks.

There are, however, some limitations with this work. First, we still need to evaluate the tranferability of learned representations beyond classification to other downstream tasks such as object segmentation, detection and instance retrieval to name a few. Second, the SSL training objective involves maximizing the agreement between augmented views of the same object or scenes (instance discrimination) and this forces them to have similar representations in the embedding space. In this work, we do not account for this by enforcing some consistency-aware constraints for channel selection in the training objective. One important aspect that seems to be missing in this setting is the idea of collaborative learning as we are in a Siamese setting and this setting itself seems to be underutilised for conditional computation in SSL setting. In the next chapter we build upon the findings of this chapter to incorporate dynamic computation but in a collaborative fashion to gain more

flexibility*.

*The meaning of flexibilty in this context is explained in detail in the next chapter.

Chapter 6

Unifying Self-supervised Learning and Dynamic Computation

As validated in Chapter 5, self-supervised models can learn to select channels dynamically depending upon the input, which leads to reduced computation during inference. However, the Siamese (self-supervised) setting seems to be underutilized in the literature in this context. Techniques like *knowledge distillation* (KD), *dynamic computation* (DC), and *pruning* are often used to obtain a lightweight model, which usually involves multiple epochs of fine-tuning (or distillation steps) of a large pre-trained model, making it computationally challenging. In this chapter, we present a novel perspective on the interplay between the SSL and DC paradigms. In particular, we show that it is feasible to simultaneously learn a *dense* and *gated* sub-network from scratch in an SSL setting without any additional fine-tuning or pruning steps. The co-evolution during pre-training of both dense and gated encoders offers a good accuracy-efficiency trade-off and therefore yields a generic and multi-purpose architecture for application-specific industrial settings. Extensive experiments on several image classification benchmarks including CIFAR-10/100, STL-10 and ImageNet-100, demonstrate that the proposed training strategy provides a *dense* and corresponding *gated* sub-network that achieves on-par performance compared with the vanilla self-supervised setting, but at a significant reduction in computation in terms of FLOPs, under a range of target budgets (t_d). The research that emanated from this work was published at the British Machine Vision Conference 2023 (BMVC), Aberdeen, UK.

6.1 Motivation

Motivation. Self-supervised representation learning methods [Chen et al., 2020a, Caron et al., 2020a, Chen and He, 2021, Bardes et al., 2022] are the standard

approach for training large-scale deep neural networks (DNNs). One of the main reasons for their popularity is their capability to leverage the inherent structure of data from a vast unlabeled corpus during pre-training, which makes them highly suitable for transfer learning [Goyal et al., 2021]. However, this comes at the cost of substantially larger model sizes, computationally expensive training strategies (larger training times, large batch sizes, etc.) [Chen et al., 2020b, Goyal et al., 2021] and subsequently more expensive inference times. Though such strategies are effective for achieving state-of-the-art results in computer vision, they may not be practical in resource-constrained industrial settings that require lightweight models to be deployed on edge devices.

To lessen the computational burden, it is common to extract (or learn) a *lightweight*[*] network from an off-the-shelf pre-trained model. This has been successfully achieved through techniques such as knowledge distillation (KD) [Hinton et al., 2015], pruning [Frankle and Carbin, 2018], dynamic computation (DC) [Veit and Belongie, 2018], etc. KD methods follow a standard two-step procedure of pre-training and distilling knowledge into a *student* network using a self-supervised (SS) objective [Fang et al., 2020, Abbasi Koohpayegani et al., 2020, Navaneet et al., 2021] or by incorporating both supervised and SS objectives [Tian et al.,]. While pruning based approaches heavily rely on multiple steps of pre-train $\rightarrow$ prune $\leftrightarrow$ finetune to get a lightweight network irrespective of the objective, additionally methods based on dynamic/conditional computation [Veit and Belongie, 2018, Herrmann et al., 2020] again rely on a pre-trained model to obtain a *lightweight* network while keeping the network topology intact via a *gating* mechanism. These approaches are effective but using fine-tuning to obtain a sub-network from large pre-trained models (such as Large Language Models) can be computationally expensive and cumbersome. Also, since downstream tasks are diverse and vary widely, any change in the task requires repeating the entire procedure multiple times, making it inefficient and less transferable.

Research Questions. These limitations motivated us to ask the following question: *"Can we unify the learning of a lightweight sub-network along with a dense network from scratch and in a completely self-supervised fashion?"* A straightforward way to achieve this is via an *online* KD (with self-supervised objective) [Yang et al., 2023, Bhat et al., 2021] learning paradigm which involves training teacher (f_θ) and student (g_ϕ) networks simultaneously during a pre-training stage. However, recognizing that this adds to the computational burden during pre-training (extra g_ϕ), we adopt a different route to attain the same goal but with a simpler, more efficient pre-training objective and faster inference than online KD-based methods.

This motivated us to reformulate the above research question to the one posed

[*]model with relatively less number of parameters

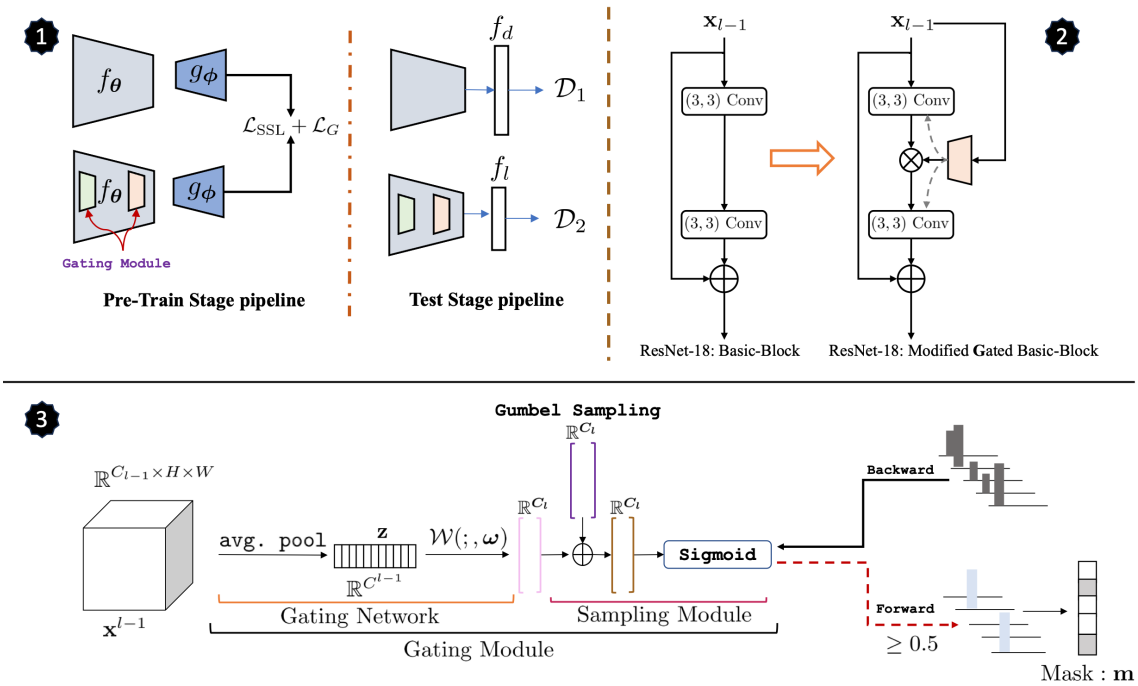


Figure 6.1: **1.** Illustration of the unification of SSL and DC during the pre-training and testing (inference) phase. f_d and f_l denote the linear layer for the dense and gated networks respectively. **Note:** dimensional size of f_d and f_l is the same, while in the figure this may look otherwise but is simply to depict the fact that empirically, the dimension of f_l is less than f_d. **2.** Illustration of the modification of the ResNet-18 basic block to accommodate the gating network during inference. **3.** This figure describes the gating module which comprises a *gating* network and *sampling* module.

in Chapter 1: "*Can we learn a single encoder (function) that could serve the dual purpose of being used as a dense and lightweight network with minimal additional overhead?*"

Our objective is to simultaneously learn a *dense* and a *lightweight* model through a unified pre-training procedure to maintain high performance on the downstream task. We achieve this by exploiting the Siamese setting (a common setting for SSL [Balestriero et al., 2023]) combined with a gating mechanism for dynamic channel selection (DCS) [Veit and Belongie, 2018, Li et al., 2021]. We opted for dynamic channel selection over KD/pruning for two main reasons: first, the gating mechanism preserves the network topology adding flexibility to the approach; second, these gating modules are computationally inexpensive. For the self-supervised objective we choose VICReg [Bardes et al., 2022] due to its symmetric nature and its ability to regularise each branch independently, as dense and sparse branches will have different statistics. Figure 6.1 (*top-left*) demonstrates this dual setting of obtaining a dense and lightweight network (derived from the dense one).

It should be noted that in this Chapter we do not follow the vocabulary of student-teacher networks. Instead, we restrict the terminology to lightweight (gated)* and a

*Note that we use the terms emphlightweight and *gated* networks interchangeably.

dense network, where encoder and gates are randomly initialized and trained jointly from scratch, with the aim that they co-evolve during pre-training. It is, however, important to mention that in this work we are not proposing any new KD objective or KD-induced learning algorithm nor any new DC objective or any pruning-based learning. Instead, we provide a novel perspective on exploiting the synergies that exist between self-supervised representation learning and dynamic computing. This approach is easily extendable to other symmetric-twins like Barlow-Twins [Zbontar et al., 2021], SimCLR [Chen et al., 2020a] or W-MSE [Ermolov et al., 2021], while it may require some adjustments for non-symmetric methods like BYOL [Grill et al., 2020], MoCo [He et al., 2020], etc., which will be explored in future work. Our main contributions are:

- We present a novel perspective on unifying the learning of *dense* and *lightweight* networks by exploiting a symmetric joint embedding architecture of the SSL paradigm.
- We demonstrate that a single encoder can be exploited as a dense as well as a lightweight network; we show in Table 6.1 and Table 6.2 that a single base encoder can serve this dual purpose. This not only reduces computational overhead during training but also gives enough flexibility to use a single network and exploit it as per its requirement.
- We demonstrate, through exhaustive experimentation that this unification preserves feature quality across different experimental settings and gives on-par performance when compared with strict baselines (see Section 6.4).

6.2 Background and Related Work

Self-supervised learning, dynamic computation and beyond: Most of the works on dynamic computation have been confined to supervised learning. Recently, [Krishna et al., 2022] used SimSiam [Chen and He, 2021] as a self-supervised objective combined with a dynamic channel gating (DGNet) [Li et al., 2021] mechanism and showed that comparable performance can be achieved under channel budget constraints. Likewise [Meng et al., 2022] used a channel gating-based dynamic pruning (CGNet) [Hua et al., 2019] augmented with CL to achieve inference speed-ups without substantial loss of performance. In a similar line of work, [Chen et al., 2021a] used iterative magnitude pruning (IMP) to obtain a winning ticket [Frankle and Carbin, 2018] for a pre-trained task (self-supervised objective) and evaluated its performance on various downstream tasks. [Pan et al., 2022] extended the work done in [Caron et al., 2020b] in a MoCo (pre-trained) setting augmented with ADMM [Zhang et al.,

2018] for systematic pruning. A self-supervised *loss* objective can serve as a tool for KD [Hinton and Dean, 2015] and model compression (MC). [Tian et al., 2019b] used a contrastive objective (along with a supervised loss for *task specific distillation*) to train a student network from a pre-trained network. Similar to [Tian et al., 2019b] but in a completely self-supervised setting [Abbasi Koohpayegani et al., 2020, Fang et al., 2020] minimizes the KL-divergence between the distribution of similarities for the teacher (pre-trained) and student networks, while SimReg [Navaneet et al., 2021] minimizes the regression loss. The authors in [Xu et al., 2020a] used a two-step strategy to train a teacher (with labels and then using an SSL head with a fixed backbone) followed by training a student using a KD loss. However, we follow a more simplistic approach through the unification of SSL (VICReg) and dynamic channel selection (DCS), where DCS maintains the network topology, making fine-tuning easier on different downstream tasks, unlike other methods that make network structure irreversible.

6.3 Preliminaries and Setup

1. VICReg as SSL objective: VICReg [Bardes et al., 2022] learns a joint embedding space governed by a loss objective, which consists of *invariance* (s) (mean squared error (MSE)), *variance* (v) and *co-variance* (c), depicted Equation 6.1. Let us consider some image dataset $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{\mathcal{D}}$ (random cropping, random horizontal flip, color jittering, grayscale (p=0.2).) and a set of transformations $\mathcal{T}$ (). An anchor image $\mathbf{x}_i \in \mathbb{R}^{H \times W \times 3}$ is augmented through transformations $t_1, t_2 \sim \mathcal{T}$ to get $\mathbf{x}_i^1 = t_1(\mathbf{x}_i)$ and $\mathbf{x}_i^2 = t_2(\mathbf{x}_i)$ respectively. Augmented views are encoded through f_θ (ResNet-18 [He et al., 2016] (R18) in this study) to get feature representations. Furthermore, these representations are mapped to an *embedding* space via *expander* (g_ϕ) where the final VICReg loss is applied between the embedding vectors $\mathbf{z}_i^1 = g_\phi(f_\theta(\mathbf{x}_i^1))$ and $\mathbf{z}_i^2 = g_\phi(f_\theta(\mathbf{x}_i^2))$. Formally the loss is defined on a batch of embedding vectors $\mathbf{Z^1} = [\mathbf{z}_1^1, \cdots, \mathbf{z}_{|\mathcal{B}|}^1]$ and $\mathbf{Z^2} = [\mathbf{z}_1^2, \cdots, \mathbf{z}_{|\mathcal{B}|}^2]$ as:

$$\mathcal{L}_{\text{VICReg}}(\mathbf{Z}^1, \mathbf{Z}^2) = \overbrace{\underbrace{\mu[v(\mathbf{Z}^1) + v(\mathbf{Z}^2)]}_{\textbf{V}\text{ariance}} + \underbrace{\nu[c(\mathbf{Z}^1) + c(\mathbf{Z}^2)]}_{\textbf{C}\text{o-Variance}}}^{\text{Regularisation Term}} + \underbrace{\eta s(\mathbf{Z}^1, \mathbf{Z}^2)}_{\textbf{I}\text{nvariance}}, \tag{6.1}$$

where $\mu = 25$, $\nu = 25$ and $\eta = 1.0$. For a detailed description of Equation 6.1 refer to [Bardes et al., 2022].

2. Gating for channel selection. The gating module comprises of a *gating network* [Li et al., 2021, Bejnordi et al.,] and a *sampling module* [Jang et al., 2017] (Figure 6.1, bottom). The *gating network* can be thought of as a lightweight network that decides the *relevance* of channels referred to as *importance* vector. To enable a

lightweight design of the gating network ($\mathcal{W}$) we follow the squeeze and excitation block design [Hu et al., 2018], similar to [Veit and Belongie, 2018, Herrmann et al., 2020]. This usually requires obtaining a context vector $\mathbf{z} \in \mathbb{R}^{C_{l-1}}$ via global average pooling to accumulate spatial information. This context vector $\mathbf{z}$ is processed through $\mathcal{W}$ to get relevance scores for each channel:

$$\mathcal{W}(\mathbf{z}, \boldsymbol{\omega}) = \mathbf{w}_2 * (\texttt{BatchNorm}(\mathbf{w}_1 * \mathbf{z}))_{\text{ReLU}}, \quad \{\mathbf{w}_1, \mathbf{w}_2\} \in \omega \tag{6.2}$$

where $*$ denotes convolution, $\mathbf{w}_1 \in \mathbb{R}^{\frac{C_{l-1}}{r} \times C_l \times 1 \times 1}$, $\mathbf{w}_2 \in \mathbb{R}^{C_l \times \frac{C_{l-1}}{r} \times 1 \times 1}$ and r is defined as reduction rate (set to 4) following [Hu et al., 2018].

Finally, to make a selection over a subset of channels, we need to map the output of $\mathcal{W}$ to a binary vector (or mask $\mathbf{m} \in \mathbb{R}^{C_l}$). This discrete selection works perfectly during inference but breaks the computational graph during training. To make training possible, the *sampling module* utilizes the Gumbel-softmax reparameterization trick [Jang et al., 2017] to make this discrete selection without breaking the computational graph. Figure 6.1 (*top-right 2*) shows the modification of ResNet18 *basic block* (during inference).

In this work we follow the setting of DGNet [Li et al., 2021] for channel selection where sparsity is induced by setting a global target budget (t_d) to optimize a loss objective :

$$\mathcal{L}_G = \lambda \left(\frac{\sum_{l=1}^{L} F_l^R}{\sum_{l=1}^{L} F_l^O} - t_d \right)^2 \tag{6.3}$$

where F_l^R is the average FLOPs over the batch along with FLOPs contribution from the gating network $\mathcal{W}$ (which is fixed for each layer), while F_l^O is the original FLOPs without a gating module, $\lambda = 5$ [Li et al., 2021] across all datasets and training regimes. Only blocks with gating modules take part in FLOP computation as they contribute to the sparsification of the network. We refer to our approach as VICReg-Dual-Gating (**VDG**).

6.3.1 Experimental Setup and Implementation Details

1. Pre-training: We closely follow the implementation of VICReg [Bardes et al., 2022] (as our self-supervised objective) suited to our computational constraints using solo-learn library [da Costa et al., 2022], while for dynamic gating we follow DGNet [Li et al., 2021] for inducing channel sparsity via gating mechanism with ResNet18. The unified framework is depicted in Figure 6.1: the two branches have a *separate batch normalization* layer following [Yu et al., 2019]. The encoder and gating networks are *randomly* initialized and trained with SGD for 500 epochs with a batch size of 512 on two NVIDIA 2080Ti GPUs, with a warmup start of 10 epochs following a cosine decay with a base learning rate of 0.3 using the LARS

optimizer [You et al., 2017]. Since we are using a very lightweight model as our gating network, there is no significant computational overhead during training, to be precise there is a slight increase in computation which amounts to 2.11% of extra model parameters (0.013% of FLOPs computation). We pre-train for a target budget, $t_d = \{10\%, 30\%, 50\%\}$ for each of the datasets except for ImageNet-100 where t_d is restricted to $\{30\%, 50\%\}$ due to computational constraints. We report the inference speedup in terms of a hardware-independent theoretical metric of FLOPs and not wall-clock time as we do not avail any hardware accelerators to utilise sparsity during training. Any scaling parameter in the loss term is derived by following the respective paper i.e., [Bardes et al., 2022, Li et al., 2021]. The code is available at `https://github.com/KrishnaTarun/Unification`.

2. Evaluation: Pre-training and evaluation is carried on the train and validation data of CIFAR-10/100 [Krizhevsky and Hinton, 2009], ImageNet100 [*] and STL-10 [Coates et al., 2011]. For pre-training with STL-10 we considered only the *un-labelled* set. We follow the standard practice of evaluating the trained encoder by freezing its weights and training a linear classifier on top of it. We trained a single linear layer for 100 epochs with a batch size of 512 on a single NVIDIA 2080Ti with a learning rate of 0.3 following step decay of 0.1 at the 60^{th} and 80^{th} epochs. We report top-1 accuracy averaged over 5 runs.

3. Baselines: To exhaustively compare the performance of the dense and gated models we consider VICReg [Bardes et al., 2022] as an SSL *dense* baseline while VICReg augmented with sparsity loss $\mathcal{L}_G$ (following Krishna *et al.* [Krishna et al., 2022]) serves as a *gated* baseline, here goal is to train a lightweight gated encoder from scratch with a self-supervised objective.

4. Comparison with self-supervised KD: To make a fair assessment of this unification, we compare the gated network's performance with KD-based methods specifically SEED [Fang et al., 2020] and SimReg [Navaneet et al., 2021] where the former was proposed to distill representational knowledge into a smaller network while the latter showed that a simple regression objective can serve as an effective tool for knowledge transfer. For SimReg: During pre-training (SSL) (500 epochs) we used VICReg [Bardes et al., 2022] with a ResNet-18 (R18: Teacher) as base encoder trained on CIFAR-100 and ImageNet-100 ($\mathcal{D}_{\text{Pretrain}}$), while distillation is performed using the SimReg objective on $\mathcal{D}_{\text{Target}}$ ($= \mathcal{D}_{\text{Pretrain}}$) by further training it for another 130 epochs. For SEED: We use a MoCo-v2 pre-trained ResNet50 (R50) encoder trained on ImageNet-1K ($\mathcal{D}_{\text{Pretrain}}$), while distillation using the SEED objective is performed for 200 epochs on $\mathcal{D}_{\text{Target}}$ (same as SimReg). Student networks are derived by sampling from R18's subspace with the number of filters for each *basic block* derived from our gating network i.e., channels are selected following policies learned

[*] `https://www.kaggle.com/datasets/ambityga/imagenet100`

Table 6.1: **Linear Evaluation**: ↑/↓ in orange font is comparison with *Baseline-1*, while blue font is comparison with *Baseline-2*. FLOPs R. denotes FLOP reduction. We report Top-1 accuracy averaged over 5 runs.

	VICReg Baseline-1 [Bardes et al., 2022]			**VICReg-Gating** Baseline-2 [Krishna et al., 2022]		**VICReg-Dual-Gating** `this work`		
Dataset	Dense	FLOPs	t_d(%)	Gated	FLOPs R.	Dense ↑	Gated ↑	FLOPs R. ↑
CIFAR-10	91.11 ± 0.03	7.03E8	10%	87.75 ± 0.03	85.92%	88.99 ± 0.04 (↓ 2.12)	88.94 ± 0.06 (↓ 2.17) (↑ 1.19)	81.49% (↓ 4.43)
			30%	89.49 ± 0.04	69.27%	90.38 ± 0.04 (↓ 0.73)	90.27 ± 0.03 (↓ 0.84) (↑ 0.78)	66.43% (↓ 2.84)
			50%	90.70 ± 0.04	51.62%	90.20 ± 0.02 (↓ 0.91)	90.40 ± 0.06 (↓ 0.71) (↓ 0.30)	49.02% (↓ 2.60)
STL-10	86.15 ± 0.10	3.33E8	10%	82.48 ± 0.15	82.85%	84.29 ± 0.21 (↓ 1.86)	83.29 ± 0.05 (↓ 2.86) (↑ 0.81)	78.34% (↓ 4.51)
			30%	84.16 ± 0.11	68.38%	84.90 ± 0.05 (↓ 1.25)	84.85 ± 0.04 (↓ 1.30) (↑ 0.69)	65.24% (↓ 3.14)
			50%	85.40 ± 0.20	49.93%	85.75 ± 0.02 (↓ 0.40)	85.72 ± 0.02 (↓ 0.43) (↑ 0.32)	48.41% (↓ 1.52)
CIFAR-100	65.86 ± 0.10	7.03E8	10%	63.12 ± 0.09	84.82%	65.21 ± 0.06 (↓ 0.65)	64.31 ± 0.08 (↓ 1.55) (↑ 1.19)	81.71% (↓ 3.11)
			30%	65.41 ± 0.09	68.68%	65.90 ± 0.10 (↑ 0.04)	65.64 ± 0.00 (↓ 0.22) (↑ 0.23)	66.83% (↓ 1.85)
			50%	65.75 ± 0.12	50.04%	66.41 ± 0.05 (↑ 0.55)	66.40 ± 0.14 (↑ 0.54) (↑ 0.65)	49.06% (↓ 0.98)
ImageNet-100	77.74 ± 0.12	1.81E9	30%	74.04 ± 0.09	67.95%	75.12 ± 0.07 (↓ 2.62)	75.04 ± 0.10 (↓ 2.17) (↑ 1.00)	64.98% (↓ 2.97)
			50%	75.83 ± 0.07	50.11%	76.42 ± 0.26 (↓ 1.32)	76.24 ± 0.12 (↓ 1.51) (↑ 0.41)	47.69% (↓ 2.42)

by our gating module for a fair comparison. The representation from the last average pooling layer is l_2 normalized and is evaluated using KNN as the evaluation criterion with $k = 1$ to report top-1 accuracy.

6.4 Results

1. Quantitative assessment. Table 6.1 compares the performance of VDG with the other two baselines for dense and gated networks. The lightweight gated network achieves improved performance across all datasets and target budgets (t_d) as compared to *Baseline-2*, with a negligible drop at $t_d = 50\%$ for CIFAR-10 only. The improved gated performance can be attributed to the fact that both dense and lightweight gated models co-learn during pre-training via weight sharing. However, the performance gain is compensated by a slightly smaller reduction in FLOPs as compared to *Baseline-2*. Improved performance of the gated network further closes the gap with a completely self-supervised dense model (*Baseline-1*). In comparison to *Baseline-1*, we observe minor drop in performance of VDG e.g., at $t_d = 10\%$ across CIFAR-10 (↓ 2.17%), STL-10 (↓ 2.86%), CIFAR-100 (↓ 1.55%), ImageNet-100 (↓ 2.7%) but with a significant reduction in the number of FLOPs. This illustrates that even under severe budget constraints our model achieves comparable performance to *Baseline1*. This drop further decreases on increasing t_d to 50%.

Another important aspect of our learning method is the performance of the *dense* (f_θ) model. Ideally, our aim is to achieve fewer fluctuations with varying t_d with a performance equivalent to a dense model as in *Baseline-1*. However, we find that the performance of the dense network (*this work*) is slightly below the performance of the dense (*Baseline-1*) for CIFAR-10/STL-10/ImageNet-100 while for CIFAR-100 the performance is better than the self-supervised dense module. This is interesting because what we achieve from this single pre-training of 500 epochs, is a single

Table 6.2: **Transfer Performance**: dense and gated under VICReg-Dual-Gating is compared with the common dense *baseline* of VICReg. ↑ / ↓ represents increment/decrement in performance. We report Top-1 *linear evaluation* accuracy averaged over 5 runs.

Dataset	**VICReg**	**VICReg-Dual-Gating**					
	[Bardes et al., 2022]	10%		30%		50%	
From → To	Dense	Dense	Gated	Dense	Gated	Dense	Gated
CIFAR-100 → STL-10	70.37	64.69 ↓	64.29 ↓	65.63 ↓	65.93 ↓	66.52 ↓	67.30 ↓
CIFAR-100 → CIFAR-10	80.08	80.56 ↑	79.92 ↓	80.23 ↑	80.24 ↑	80.16 ↑	80.09 ↑
CIFAR-100 → ImageNet-100	39.45	35.88 ↓	36.31 ↓	38.49 ↓	38.20 ↓	39.41 ↓	40.68 ↑
ImageNet-100 → STL-10	72.80	-	-	65.86 ↓	65.01 ↓	67.74 ↓	67.15 ↓
ImageNet-100 → CIFAR-10	55.12	-	-	53.38 ↓	49.41 ↓	54.01 ↓	49.80 ↓
ImageNet-100 → CIFAR-100	31.42	-	-	28.81 ↓	25.55 ↓	28.75 ↓	25.14 ↓

Table 6.3: Comparison of KD methods *students* performance with our *gated* network.

Method	$\mathcal{D}_{\text{Pretrain}}$	$\mathcal{D}_{\text{Target}}$	SSL-Pre Method (Teacher)$_{\text{epoch}}$	Student	1-NN
SimReg [Navaneet et al., 2021]	CIFAR-100	CIFAR-100	VICReg (R18)$_{500}$	R18 (10%)	37.32%
	CIFAR-100	CIFAR-100	VICReg (R18)$_{500}$	R18 (30%)	45.71%
	CIFAR-100	CIFAR-100	VICReg (R18)$_{500}$	R18 (50%)	48.99%
	ImageNet-100	ImageNet-100	VICReg (R18)$_{500}$	R18 (30%)	63.80%
	ImageNet-100	ImageNet-100	VICReg (R18)$_{500}$	R18 (50%)	65.78%
SEED [Fang et al., 2020]	ImageNet-1K	CIFAR-100	MoCO-v2 (R50)$_{800}$	R18 (10%)	31.54%
	ImageNet-1K	CIFAR-100	MoCO-v2 (R50)$_{800}$	R18 (30%)	35.29%
	ImageNet-1K	CIFAR-100	MoCO-v2 (R50)$_{800}$	R18 (50%)	38.22%
	ImageNet-1K	ImageNet-100	MoCO-v2 (R50)$_{800}$	R18 (30%)	64.38%
	ImageNet-1K	ImageNet-100	MoCO-v2 (R50)$_{800}$	R18 (50%)	67.50%
Ours (Gated)	CIFAR-100	-	VICReg (R18)$_{500}$	R18 (10%)	**56.57%**
	CIFAR-100	-	VICReg (R18)$_{500}$	R18 (30%)	**58.49%**
	CIFAR-100	-	VICReg (R18)$_{500}$	R18 (50%)	**59.83%**
	ImageNet-100	-	VICReg (R18)$_{500}$	R18 (30%)	**65.54%**
	ImageNet-100	-	VICReg (R18)$_{500}$	R18 (50%)	**67.72%**

base encoder (dense encoder) and gates (via gating modules) and their combination gives a gated lightweight network.

2. Transfer Learning: Table 6.2 compares the transfer performance of VDG (dense and gated) with VICReg. This experiment gives further insights into the quality of the learned representation in this joint setting. In general, there is a drop in performance for VICReg-Dual for both *dense* and *gated*, although the difference is not significant. However, for CIFAR-100 → CIFAR-10 *dense* and *gated* outperforms only *dense* in VICReg at a very low target budget. Even in the case when the model is pre-trained on ImageNet-100, performance is comparable. This is encouraging as this new perspective still maintains good generalization and transferability.

3. SSL-KD vs SSL-Gating: Table 6.3 compares the performance of SSL-KD methods with our SSL-Gating framework. To avoid confusion, we would like to reiterate we don't follow the student-teacher paradigm, "Student" in Table 6.3 for "Ours (Gated)" is basically a R18 (base-encoder) with gates (see Figure 6.1) and $\mathcal{D}_{\text{Pretrain}} = \mathcal{D}_{\text{Target}}$. Results in Table 6.3 are very promising as we outperform both the KD methods by a substantial margin across all budgets. This result suggests that combining gating could serve as a general recipe to obtain a lightweight network along with a *dense* network during pre-training.

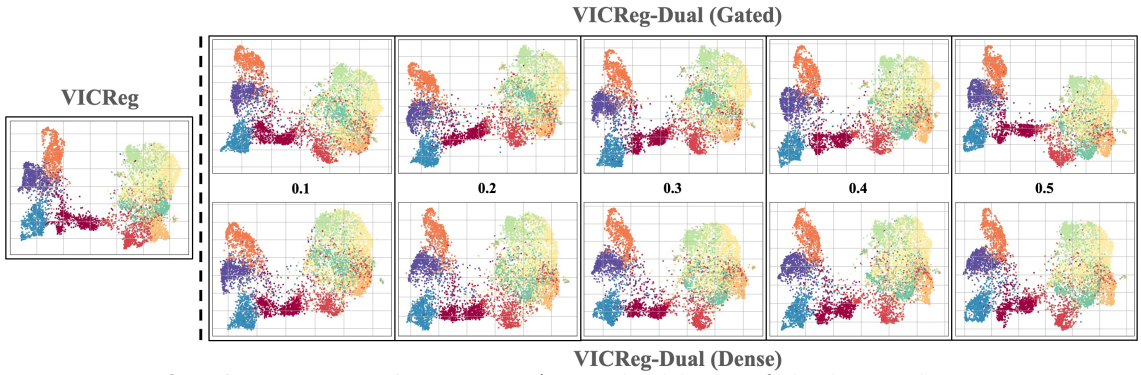


Figure 6.2: **Qualitative analysis:** UMAP embeddings of the learned representations: *lightweight* gated network (*top* row), while dense network (*bottom*) row over different target budgets t_d. This is compared with embeddings of VICReg (dense) trained without any sort of sparsity.

Table 6.4: Barlow Twins vs VICReg in dual setting.

		Barlow Twins(BT)-Dual-Gating			**VICReg-Dual-Gating**		
Dataset	$\mathbf{t}_d$	Dense	Gated	FLOPs R.	Dense	Gated	FLOPs R.
CIFAR-100	10%	53.64	53.01	80.60%	**54.92**	54.66	81.71%
	30%	55.03	54.89	66.09%	**56.34**	55.53	66.83%
	50%	55.63	55.39	47.64%	**56.83**	57.25	49.06%
STL-10	10%	73.45	73.49	76.74%	**76.61**	76.45	78.27%
	30%	75.51	76.02	64.12%	**78.55**	78.80	65.17%
	50%	76.99	77.10	48.32%	**79.69**	79.95	48.39%

4. Qualitative assessment: Figure 6.2 shows uniform manifold approximation and projection (UMAP) [McInnes et al., 2018] embeddings of the learned representations ($f_\theta(\mathbf{x}) \in \mathbb{R}^{512}$) trained using the dual-setting on the STL-10 dataset and compares it with VICReg [Bardes et al., 2022]. The learned structure is similar to dense (VICReg) at a very low budget. Furthermore, the classes appear to be visually distinct, similar to the VICReg setting, and this is observed for both the dense and gated networks of VICReg-Dual-Gating.

6.5 Additional Insights

For all experimental settings and studies discussed hereafter; models (and settings) were pre-trained for 500 epochs on CIFAR-100 and STL-10 datasets. Instead of using linear evaluation, we use KNN as the evaluation criterion with $k = 1$ to report *top-1* accuracy. In all tables, **bold** and underline are the best-performing results for the dense and gated modules, respectively. Representations are not l_2 normalised.

1. Comparison with the symmetric Barlow Twins (BT) architecture. VICReg is built upon the findings of BT [Zbontar et al., 2021] and it is straightforward to apply the dual setting to BT because it minimizes the cross-correlation (regu-

Table 6.5: Alternative base encoders. Comparing the performance of using single base encoder or 2 encoders one as dense and other as gated one.

		VICReg-Dual-Gating (1×ResNet-18)			**VICReg-Dual-Gating** (2×ResNet-18)		
Dataset	$\mathbf{t}_d(\%)$	Dense	Gated	FLOPs R.	Dense	Gated	FLOPs R.
CIFAR-100	10%	**54.92**	54.66	81.71%	53.85	52.44	85.77%
	30%	**56.34**	55.53	66.83%	55.65	55.52	70.07%
	50%	**56.83**	57.25	49.06%	55.64	55.71	52.60%
STL-10	10%	**76.61**	76.45	78.27%	74.93	74.89	82.48%
	30%	**78.55**	78.80	65.17%	76.83	76.71	69.26%
	50%	**79.69**	79.95	48.39%	77.74	78.00	51.10%

Table 6.6: Investigating the role of mean squared error (MSE).

		VICReg-Dual-Gating w/ Invariance			**VICReg-Dual-Gating** w/o Invariance		
Dataset	$\mathbf{t}_d(\%)$	Dense	Gated	FLOPs R.	Dense	Gated	FLOPs R.
CIFAR-100	10%	**54.92**	54.66	81.71%	4.06	6.31	93.77%
	30%	**56.34**	55.53	66.83%	4.84	4.89	82.72%
	50%	**56.83**	57.25	49.06%	2.79	4.15	49.05%
STL-10	10%	**76.61**	76.45	78.27%	11.79	18.02	90.71%
	30%	**78.55**	78.80	65.17%	12.25	17.89	74.96%
	50%	**79.69**	79.95	48.39%	12.81	15.72	51.91%

larisation term) to identity $\mathcal{I}$ although the loss function is entirely mutual unlike in VICReg. In Table 6.4 we compare the performance of BT augmented with *our* setting. We observe that 1-NN performance of BT is low as compared to VICReg-Dual-Gating. The drop in performance could be attributed to the fact that VICReg applies independent regularisation which are later matched through the invariance loss. This further validates the hypothesis of choosing VICReg as our objective.

2. Training with a different base encoder. In this setting we train a model with two base encoders, one w/ gate (gated) and other w/o gate (dense) i.e., (2×ResNet-18) (results in Table 6.5). An interesting observation is that VICReg-Dual-Gating with a single base encoder outperforms a more powerful setting with two base encoders (Table 6.5) although the FLOPs reduction (FLOPs R.) is higher (↑) in the setting of two different encoders. This is due to the fact that the sparsity loss ($\mathcal{L}_G$) operates solely on the un-shared branch so there is no trade-off involved, as in the case of a single base encoder which simultaneously tries to enforce sparsity and visual invariance.

3. Role of *mean squared error* in co-evolving. It's a well-known fact in SSL that these methods suffer from dimensional collapse [Hua et al., 2021, Jing et al.,]. Training without any regularisation term or any trick [Grill et al., 2020, Chen et al.,

2020a] would lead to dimensional collapse. Also, the authors in [Kim et al., 2021] showed that MSE serves as a better option for exact logit matching as compared to Kullback-Leibler (KL) divergence. So, in order to understand the role of MSE we trained a model w/o the *invariance* Equation 6.1 loss. As Table 6.6 shows, we found that there is a large performance drop if we remove the invariance term. This implies that the invariance term plays a crucial role and seems to be an important factor, not only for co-evolving, but for self-supervision.

6.6 Discussion and Conclusion

RQ4 as described in Chapter 1 of this thesis poses the question of whether it is possible to learn a single encoder (function) that could serve the dual purpose of being used as a dense and lightweight network with minimal additional overhead during pre-training. In this chapter, we present a novel perspective on unifying synergies between SSL and DC to answer this question. We exploit DC to induce sparsity into symmetric branches of self-supervised models enabling both branches to co-evolve with each other during training. In addition, this approach also allows simultaneous training of a dense and gated (sparse) sub-networks from scratch with a target budget t_d under a self-supervised training objective with minimal computational overhead via weight sharing, thereby offering a good accuracy-efficiency trade-off for a given downstream application. As a result, our single base encoder offers sufficient flexibility to serve a dual purpose to reduce excessive computational overhead, which is validated through exhaustive experimentation (Tables 6.1, 6.2, 6.3).

However, it should be noted that there are limitations to this work. First, the dense model performance degrades and its performance further fluctuates with varying t_d. We experimented with RotNet [Komodakis and Gidaris, 2018] as an extra proxy loss for the dense branch but this did not yield good performance. Second, we have not imposed any constraint on the training objective that enforces a uniform distribution of channel activations, i.e. preservation of channel diversity during inference (which could also be a solution to the first limitation). Third, it would be interesting to extend this setting to contrastive and non-symmetric architectures, which are not explored in this chapter.

Chapter 7

Conclusion

In this final chapter, we bring together all the key findings of our research to highlight our contributions and discuss their significance. We review the hypotheses and research questions that we set out in Chapter 1 and assess how well we have answered them. In this way, we summarize the important insights from our work, discuss practical implications, and suggest directions for future research. By doing so, we aim to provide a clear and concise conclusion highlighting the value of this work.

7.1 Hypotheses and Research Questions Revisited

This section revisits the research questions arising from the overarching research hypotheses presented in Chapter 1 and considers these in light of the various studies conducted and reported in this thesis.

Hypothesis 1 (H1): We hypothesize that models that are learned to encode semantic similarity among instances via discriminative learning should perform well on the task of image instance retrieval. This leads to **Research Question 1 (RQ1):** *How effectively do SSL methods encode semantic identity in comparison to SL methods for the task of image instance retrieval?* In Chapter 3, we present an extensive evaluation of contrastive self-supervised models for the task of image instance retrieval. We compare different contrastive self-supervised models across different datasets and across different settings with a supervised baseline. We find that these models, without any explicit supervisory signals, can learn to encode inherent similarity via instance discrimination and as a result, they generalize well when the target domain is different from the source domain. This is validated through various experimental results (see Tables 3.2, 3.3). Additionally, we find that low-dimension feature descriptors perform better than the true dimension (i.e. not PCA compressed - see Table 3.4).

Hypothesis 2 (H2): While self-supervised representation learning has shown success with 2D images, its application to 360° images remains underexplored. We

argue that omnidirectional images are particularly suited to such an approach due to the geometry of the data domain. This gives rise to **Research Question 2 (RQ2):** *Can we efficiently design and effectively exploit contrastive self-supervised methods for a granular task like visual saliency prediction task for* 360° *omnidirectional images?* In Chapter 4, we show that it is possible to design a self-supervised contrastive model to exploit the geometric flexibility offered in this data regime (i.e., 360°) by learning representations that are locally consistent across the different views. We find that the proposed method performs on par with supervised methods and in some cases surpasses them (see Table 4.1). Additionally, as shown in Table 4.2, we observe that our model exhibits a remarkable performance improvement, being over 8× faster than ATSAL [Dahou et al., 2021], which is the fastest model in this category. This is an important finding from the perspective that a careful design not only leads to better performance but at the same time could lead to better efficiency.

Hypothesis 3 (H3): We hypothesize that self-supervised models are an ideal candidate for (such) dynamic network structures as they capture highly redundant channel features during pre-training, which gives rise to **Research Question 3 (RQ3):** *Do self-supervised models learn highly redundant channel features, dynamically select the important channels and remove the unnecessary ones?* In Chapter 5 it is proven that self-supervised models learn redundant channel features, which further can be dynamically removed depending upon the input without significantly affecting model performance (see Table 5.1). Further, we prove that channels can be dynamically selected across the network during inference (see Figure 5.1). This validates that it is indeed possible to design self-supervised models which can be made efficient during inference.

Hypothesis 4 (H4): We hypothesize that the Siamese setting can be utilized for simultaneously training an encoder that can serve the purpose of a dense encoder as well as a sparse lightweight encoder. This prompts **Research Question 4 (RQ4):** *Is it possible to learn a single encoder (function) that could serve the dual purpose of being used as a dense and lightweight network with minimal additional overhead?* Building upon the findings in Chapter 5, in Chapter 6 we show that a single encoder can indeed serve the purpose of being used as a dense and lightweight network and demonstrates the transferability of the learned representations across different datasets improving generalizability (see Table 6.1 and Table 6.2). As a result, both a lightweight sparse network as well as a dense model are available after pre-training, which can be exploited as needed by the downstream task. Interestingly our approach surpasses self-supervised knowledge distillation methods by a significant margin, as shown in Table 6.3.

7.2 Research Contributions

The various contributions of the thesis can be summarised as follows:

- Exhaustive evaluation across different benchmarks (Chapter 3, Section 3.4.2, Table 3.2, 3.3) shows that contrastive methods can easily surpass supervised models without any explicit supervision. The work done in Chapter 3 showed that contrastive models learn features that can be used to bootstrap image retrieval engines. Of particular note is the impressive fact that contrastive pre-training did not consider the image retrieval task, yet it learned features that can be effectively used for this task. This is important in the context of utilizing self-supervised models as a fixed feature extractor.

- Chapter 4 introduced a method for modeling human visual attention with contrastive self-supervised learning, which improves generalization and expressive power compared to existing approaches. The approach exploits the geometric flexibility of the spherical data to learn representations that contain locally consistent information across the views. Through extensive evaluation in Chapter 4, Table 4.1, we show that contrastive learning can be exploited for saliency prediction and that it performs on par with fully supervised methods. Contrastive learning requires a large amount of unlabelled data to be trained effectively. Due to the unavailability of large-scale 360° images datasets, we gather a new dataset with 90K ODIs from multiple sources and make it publicly available along with the code to reproduce the results[*].

- In Chapter 5, we study the behaviour of self-supervised learning when integrated with channel selection networks given a global target budget for computational cost. Our empirical results (Table 5.1) provide interesting insights about self-supervised learning when trained with channel selection. First, self-supervised models learn highly redundant channel features that can be discarded to reduce computational overhead. Second, we show that channel selection modules can significantly reduce FLOP computation and make inference more efficient. Third, our results also provide intuition that representations learned by self-supervised networks with channel selection can also be transferred to downstream tasks. The code required to reproduce the results in Chapter 5 is publicly available[*].

- Finally in Chapter 6, we exploit dynamic computation to induce sparsity into symmetric branches of self-supervised models enabling both branches to co-evolve with each other during training. In addition, this approach also allows

[*]Available at: `https://github.com/KrishnaTarun/360_unsupervised_saliency`
[*]Available at: `https://github.com/KrishnaTarun/SSL_DGC`

simultaneous training of a dense and gated (sparse) sub-network from scratch with a target budget under a self-supervised training objective with minimal computational overhead via weight sharing. This offers a good accuracy-efficiency trade-off for a given downstream application (see Tables 6.1 and 6.2). As a result, our single base encoder offers enough flexibility to serve a dual purpose to reduce excessive computational overhead, which is validated through exhaustive experimentation. Once again, code to reproduce the results is publicly available*.

7.3 Proposal for Future Work

Notwithstanding the advances made by the research reported in this thesis, there are exciting opportunities for future work. We document some of these in the following.

- In Chapter 5 & 6, we exploit the Gumbel softmax trick for channel selection but we do not impose any constraints that enforce some diversity in the selection process. As can be seen in Figure 5.1, some channels are never selected which implies we are not exploiting the entire expressivity of the network. This merits further investigation.
- In Chapter 6, we observe a slight drop in the performance of the dense model. Further investigation is needed into a way to minimize this performance drop (in fact, ideally avoid any performance drop at all) that is consistent across different target budgets. Furthermore, this unifying paradigm needs to be studied across different approaches to self-supervised methods e.g. non-symmetric architectures e.g., `BYOL` [Grill et al., 2020], `SimSiam` [Chen and He, 2021] etc.
- A potentially extremely impactful direction for future research is to extend the work in Chapter 6 for video understanding under budget constraints across multiple video domains where huge datasets and foundation models now play a crucial role in learning global representations for videos.

7.4 Closing Remarks

Recent advances in computer vision have been significantly driven by deep learning methods, especially through self-supervised representation learning. These methods utilize extensive amounts of unlabeled data to create models that extract robust representations from visual information. This thesis has shown how self-supervised

*Available at: `https://github.com/KrishnaTarun/Unification`

visual representation learning can be used to efficiently design, and explore ways to use limited resources to maximize performance.

Self-supervised learning aims to leverage the inherent structure of data, reducing the reliance on large volumes of labelled data and offering solutions when traditional deep learning methods face limitations. A key insight from this research is that design choices in self-supervised models can lead to effective and efficient utilization of this machine learning paradigm. The approaches explored in this thesis are particularly relevant for applications where training resources are limited, both in terms of label availability and computational power and hence particularly relevant in practical industry settings.

This thesis also delves into how self-supervised models can be exploited in different target domains and non-traditional data regimes, such as scenarios with a small set of labeled data alongside a large set of unlabeled data (e.g., saliency prediction). We demonstrate how self-supervised models can be tailored to the requirements of downstream tasks without imposing excessive computational constraints during pre-training.

Bibliography

[Abbasi Koohpayegani et al., 2020] Abbasi Koohpayegani, S., Tejankar, A., and Pirsiavash, H. (2020). Compress: Self-supervised learning by compressing representations. *Advances in Neural Information Processing Systems*, 33:12980–12992. 67, 70

[Achanta et al., 2012] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282. 42

[Arandjelović and Zisserman, 2012] Arandjelović, R. and Zisserman, A. (2012). Three things everyone should know to improve object retrieval. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2911–2918. IEEE. 30

[Asano et al., 2019] Asano, Y., Rupprecht, C., and Vedaldi, A. (2019). Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations*. 12

[Assens et al., 2018] Assens, M., Giro-i Nieto, X., McGuinness, K., and O'Connor, N. E. (2018). Scanpath and saliency prediction on 360 degree images. *Signal Processing: Image Communication*, 69:8–14. 42, 48, 49, 54

[Azizpour et al., 2015a] Azizpour, H., Sharif Razavian, A., Sullivan, J., Maki, A., and Carlsson, S. (2015a). From generic to specific deep representations for visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 23

[Azizpour et al., 2015b] Azizpour, H., Sharif Razavian, A., Sullivan, J., Maki, A., and Carlsson, S. (2015b). From generic to specific deep representations for visual recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 36–45. 24

[Babenko et al., 2014] Babenko, A., Slesarev, A., Chigorin, A., and Lempitsky, V. (2014). Neural codes for image retrieval. In *European conference on computer vision*, pages 584–599. Springer. 25

[Bachman et al., 2019] Bachman, P., Hjelm, R. D., and Buchwalter, W. (2019). Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32. 13, 25, 26, 34, 40, 43

[Balestriero et al., 2023] Balestriero, R., Ibrahim, M., Sobal, V., Morcos, A., Shekhar, S., Goldstein, T., Bordes, F., Bardes, A., Mialon, G., Tian, Y., et al. (2023). A cookbook of self-supervised learning. *arXiv preprint arXiv:2304.12210*. 68

[Bardes et al., 2022] Bardes, A., Ponce, J., and LeCun, Y. (2022). VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*. 14, 15, 66, 68, 70, 71, 72, 73, 74, 75

[Battisti et al., 2018] Battisti, F., Baldoni, S., Brizzi, M., and Carli, M. (2018). A feature-based approach for saliency estimation of omni-directional images. *Signal Processing: Image Communication*, 69:53–59. 42

[Bejnordi et al.,] Bejnordi, B. E., Blankevoort, T., and Welling, M. Batch-shaping for learning conditional channel gated networks. In *International Conference on Learning Representations*. 70

[Bhat et al., 2021] Bhat, P., Arani, E., and Zonooz, B. (2021). Distill on the go: online knowledge distillation in self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2678–2687. 67

[Bishop, 2007] Bishop, C. M. (2007). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1 edition. 12

[Borji, 2018] Borji, A. (2018). Saliency prediction in the deep learning era: An empirical investigation. *arXiv preprint arXiv:1810.03716*, 10. 6, 38

[Borji and Itti, 2012] Borji, A. and Itti, L. (2012). State-of-the-art in visual attention modeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):185–207. 54

[Brock et al., 2018] Brock, A., Donahue, J., and Simonyan, K. (2018). Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*. 18

[Bruce and Tsotsos, 2006] Bruce, N. and Tsotsos, J. (2006). Saliency based on information maximization. In *Advances in neural information processing systems*, pages 155–162. 54

[Bylinskii et al., 2018] Bylinskii, Z., Judd, T., Oliva, A., Torralba, A., and Durand, F. (2018). What do different evaluation metrics tell us about saliency models? *IEEE transactions on pattern analysis and machine intelligence*, 41(3):740–757. 47, 48

[Caron et al., 2018] Caron, M., Bojanowski, P., Joulin, A., and Douze, M. (2018). Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149. 12

[Caron et al., 2020a] Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. (2020a). Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924. 12, 66

[Caron et al., 2020b] Caron, M., Morcos, A., Bojanowski, P., Mairal, J., and Joulin, A. (2020b). Pruning convolutional neural networks with self-supervision. *arXiv preprint arXiv:2001.03554*. 57, 58, 59, 69

[Chao et al., 2018] Chao, F.-Y., Zhang, L., Hamidouche, W., and Deforges, O. (2018). Salgan360: Visual saliency prediction on 360 degree images with generative adversarial networks. In *2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 01–04. IEEE. 42, 48, 49, 54

[Che et al., 2019] Che, Z., Borji, A., Zhai, G., Min, X., Guo, G., and Le Callet, P. (2019). How is gaze influenced by image transformations? dataset and model. *IEEE Transactions on Image Processing*, 29:2287–2300. 43

[Chen et al., 2014] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K. P., and Yuille, A. L. (2014). Semantic image segmentation with deep convolutional nets and fully connected crfs. *CoRR*, abs/1412.7062. 23

[Chen et al., 2021a] Chen, T., Frankle, J., Chang, S., Liu, S., Zhang, Y., Carbin, M., and Wang, Z. (2021a). The lottery tickets hypothesis for supervised and self-supervised pre-training in computer vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16306–16316. 5, 69

[Chen et al., 2020a] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020a). A simple framework for contrastive learning of visual representations. In *International*

conference on machine learning, pages 1597–1607. PMLR. 7, 13, 14, 25, 26, 27, 28, 34, 53, 57, 59, 66, 69, 77

[Chen et al., 2020b] Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. E. (2020b). Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255. 3, 67

[Chen et al., 2020c] Chen, X., Fan, H., Girshick, R., and He, K. (2020c). Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*. 13, 25, 27

[Chen and He, 2021] Chen, X. and He, K. (2021). Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758. ix, 17, 62, 66, 69, 81

[Chen et al., 2021b] Chen, X., Xie, S., and He, K. (2021b). An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9640–9649. 13

[Chen et al., 2019] Chen, Z., Li, Y., Bengio, S., and Si, S. (2019). You look twice: Gaternet for dynamic filter selection in cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9172–9180. 19

[Chopra et al., 2005] Chopra, S., Hadsell, R., and LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE. 13

[Chou et al., 2020] Chou, S.-H., Sun, C., Chang, W.-Y., Hsu, W.-T., Sun, M., and Fu, J. (2020). 360-indoor: Towards learning real-world objects in 360deg indoor equirectangular images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 845–853. 47

[Chuang et al., 2020] Chuang, C.-Y., Robinson, J., Lin, Y.-C., Torralba, A., and Jegelka, S. (2020). Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775. vi, 12

[Chum et al., 2007] Chum, O., Philbin, J., Sivic, J., Isard, M., and Zisserman, A. (2007). Total recall: Automatic query expansion with a generative feature model for object retrieval. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE. 30

[Coates et al., 2011] Coates, A., Ng, A., and Lee, H. (2011). An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth*

international conference on artificial intelligence and statistics, pages 215–223. JMLR Workshop and Conference Proceedings. 12, 72

[Cornia et al., 2016] Cornia, M., Baraldi, L., Serra, G., and Cucchiara, R. (2016). A deep multi-level network for saliency prediction. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 3488–3493. IEEE. 41, 54

[da Costa et al., 2022] da Costa, V. G. T., Fini, E., Nabi, M., Sebe, N., and Ricci, E. (2022). solo-learn: A library of self-supervised methods for visual representation learning. *Journal of Machine Learning Research*, 23(56):1–6. 62, 71

[Dahou et al., 2021] Dahou, Y., Tliba, M., McGuinness, K., and O'Connor, N. (2021). Atsal: An attention based architecture for saliency prediction in 360 videos. In Del Bimbo, A., Cucchiara, R., Sclaroff, S., Farinella, G. M., Mei, T., Bertini, M., Escalante, H. J., and Vezzani, R., editors, *Pattern Recognition. ICPR International Workshops and Challenges*, pages 305–320, Cham. Springer International Publishing. vii, 39, 42, 48, 49, 79

[De Abreu et al., 2017] De Abreu, A., Ozcinar, C., and Smolic, A. (2017). Look around you: Saliency maps for omnidirectional images in vr applications. In *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6. IEEE. 41, 54

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. 57

[Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. 3

[Devlin et al., 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics. 3

[Doersch et al., 2015] Doersch, C., Gupta, A., and Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430. 4, 11, 39, 52

[Donahue et al., 2014] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2014). Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655. PMLR. 24

[Donahue et al., 2017] Donahue, J., Krähenbühl, P., and Darrell, T. (2017). Adversarial feature learning. In *International Conference on Learning Representations*. 18

[Donahue and Simonyan, 2019] Donahue, J. and Simonyan, K. (2019). Large scale adversarial representation learning. *Advances in neural information processing systems*, 32. 18

[Dosovitskiy et al., 2021] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*. 2, 44

[Droste et al., 2020] Droste, R., Jiao, J., and Noble, J. A. (2020). Unified image and video saliency modeling. In *European Conference on Computer Vision*, pages 419–435. Springer. 48, 49

[Ermolov et al., 2021] Ermolov, A., Siarohin, A., Sangineto, E., and Sebe, N. (2021). Whitening for self-supervised representation learning. In *International Conference on Machine Learning*, pages 3015–3024. PMLR. 14, 69

[Falcon and Cho, 2020] Falcon, W. and Cho, K. (2020). A framework for contrastive self-supervised learning and designing a new approach. *arXiv preprint arXiv:2009.00104*. 24

[Fang et al., 2018] Fang, Y., Zhang, X., and Imamoglu, N. (2018). A novel superpixel-based saliency detection model for 360-degree images. *Signal Processing: Image Communication*, 69:1–7. 42

[Fang et al., 2020] Fang, Z., Wang, J., Wang, L., Zhang, L., Yang, Y., and Liu, Z. (2020). Seed: Self-supervised distillation for visual representation. In *International Conference on Learning Representations*. 67, 70, 72, 74

[Figurnov et al., 2017] Figurnov, M., Collins, M. D., Zhu, Y., Zhang, L., Huang, J., Vetrov, D., and Salakhutdinov, R. (2017). Spatially adaptive computation time for residual networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1039–1048. 19

[Frankle and Carbin, 2018] Frankle, J. and Carbin, M. (2018). The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*. 7, 18, 58, 67, 69

[Frankle et al., 2020] Frankle, J., Dziugaite, G. K., Roy, D., and Carbin, M. (2020). Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pages 3259–3269. PMLR. 58

[Gao et al., 2018] Gao, X., Zhao, Y., Dudziak, Ł., Mullins, R., and Xu, C.-z. (2018). Dynamic channel pruning: Feature boosting and suppression. In *International Conference on Learning Representations*. 19, 57, 58

[Gidaris et al., 2021] Gidaris, S., Bursuc, A., Puy, G., Komodakis, N., Cord, M., and Pérez, P. (2021). Obow: Online bag-of-visual-words generation for self-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6830–6840. 17

[Gong et al., 2014] Gong, Y., Wang, L., Guo, R., and Lazebnik, S. (2014). Multi-scale orderless pooling of deep convolutional activation features. In *European conference on computer vision*, pages 392–407. Springer. 25

[Goodfellow et al., 2014] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks. *arXiv preprint arXiv:1406.2661*. 4, 18

[Gordo et al., 2016] Gordo, A., Almazán, J., Revaud, J., and Larlus, D. (2016). Deep image retrieval: Learning global representations for image search. In *European conference on computer vision*, pages 241–257. Springer. 23, 25, 29, 31

[Gordo et al., 2017] Gordo, A., Almazan, J., Revaud, J., and Larlus, D. (2017). End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision*, 124(2):237–254. 23, 25, 30

[Gordoa et al., 2012] Gordoa, A., Rodriguez-Serrano, J. A., Perronnin, F., and Valveny, E. (2012). Leveraging category-level labels for instance-level image retrieval. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3045–3052. IEEE. 24

[Goyal et al., 2021] Goyal, P., Caron, M., Lefaudeux, B., Xu, M., Wang, P., Pai, V., Singh, M., Liptchinsky, V., Misra, I., Joulin, A., et al. (2021). Self-supervised pretraining of visual features in the wild. *arXiv preprint arXiv:2103.01988*. 67

[Goyal et al., 2022] Goyal, P., Duval, Q., Seessel, I., Caron, M., Misra, I., Sagun, L., Joulin, A., and Bojanowski, P. (2022). Vision models are more robust and

fair when pretrained on uncurated images without supervision. *arXiv preprint arXiv:2202.08360*. 3, 4

[Grill et al., 2020] Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. (2020). Bootstrap your own latent – a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284. 17, 69, 77, 81

[Guo et al., 2019] Guo, Y., Shi, H., Kumar, A., Grauman, K., Rosing, T., and Feris, R. (2019). Spottune: transfer learning through adaptive fine-tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4805–4814. 19

[Gutmann and Hyvärinen, 2010] Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304. JMLR Workshop and Conference Proceedings. 13, 24, 25, 26, 39, 45

[Gutmann and Hyvärinen, 2012] Gutmann, M. U. and Hyvärinen, A. (2012). Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13(2). 46

[Han et al., 2015] Han, S., Pool, J., Tran, J., and Dally, W. (2015). Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28. 58

[He et al., 2022] He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009. 18

[He et al., 2020] He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738. 7, 13, 25, 27, 69

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778. 2, 23, 24, 29, 43, 70

[He et al., 2018] He, Y., Kang, G., Dong, X., Fu, Y., and Yang, Y. (2018). Soft filter pruning for accelerating deep convolutional neural networks. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2234–2240. 58

[Herrmann et al., 2020] Herrmann, C., Bowen, R. S., and Zabih, R. (2020). Channel selection using gumbel softmax. In *European Conference on Computer Vision*, pages 241–257. Springer. 19, 57, 59, 61, 67, 71

[Hinton and Dean, 2015] Hinton, G. and Dean, J. (2015). Distilling the Knowledge in a Neural Network. Technical report. 70

[Hinton et al., 2015] Hinton, G., Vinyals, O., Dean, J., et al. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7). 7, 16, 18, 67

[Hjelm et al., 2019] Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. (2019). Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*. 13, 25, 26, 40, 43, 46

[Ho et al., 2020] Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851. 18

[Hoefler et al., 2021] Hoefler, T., Alistarh, D., Ben-Nun, T., Dryden, N., and Peste, A. (2021). Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *Journal of Machine Learning Research*, 22(241):1–124. 18, 19, 59

[Hu et al., 2018] Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141. 60, 61, 71

[Hua et al., 2021] Hua, T., Wang, W., Xue, Z., Ren, S., Wang, Y., and Zhao, H. (2021). On feature decorrelation in self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9598–9608. 13, 76

[Hua et al., 2019] Hua, W., Zhou, Y., De Sa, C. M., Zhang, Z., and Suh, G. E. (2019). Channel gating neural networks. *Advances in Neural Information Processing Systems*, 32. 59, 69

[Huang et al., 2018] Huang, G., Chen, D., Li, T., Wu, F., van der Maaten, L., and Weinberger, K. (2018). Multi-scale dense networks for resource efficient image classification. In *International Conference on Learning Representations*. 19

[Huang et al., 2022] Huang, W., Yi, M., Zhao, X., and Jiang, Z. (2022). Towards the generalization of contrastive self-supervised learning. In *The Eleventh International Conference on Learning Representations*. 3

[Iscen et al., 2017] Iscen, A., Tolias, G., Avrithis, Y., Furon, T., and Chum, O. (2017). Efficient diffusion on region manifolds: Recovering small objects with compact cnn representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2077–2086. 29

[Itti and Baldi, 2009] Itti, L. and Baldi, P. (2009). Bayesian surprise attracts human attention. *Vision research*, 49(10):1295–1306. 54

[Itti and Koch, 2000] Itti, L. and Koch, C. (2000). A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision research*, 40(10-12):1489–1506. 38, 54

[Jabri et al., 2020] Jabri, A., Owens, A., and Efros, A. A. (2020). Space-time correspondence as a contrastive random walk. *arXiv preprint arXiv:2006.14613*. 24

[Jain et al., 2023] Jain, S., Salman, H., Khaddaj, A., Wong, E., Park, S. M., and Mądry, A. (2023). A data-based perspective on transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3613–3622. 3

[Jang et al., 2017] Jang, E., Gu, S., and Poole, B. (2017). Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*. 20, 57, 61, 70, 71

[Jégou et al., 2010] Jégou, H., Douze, M., Schmid, C., and Pérez, P. (2010). Aggregating local descriptors into a compact image representation. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3304–3311. IEEE. 23, 24

[Jiang et al., 2015] Jiang, M., Huang, S., Duan, J., and Zhao, Q. (2015). Salicon: Saliency in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1072–1080. 41

[Jing et al.,] Jing, L., Vincent, P., LeCun, Y., and Tian, Y. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations*. 13, 76

[Jing et al., 2015] Jing, Y., Liu, D., Kislyuk, D., Zhai, A., Xu, J., Donahue, J., and Tavel, S. (2015). Visual search at pinterest. In *Proceedings of the 21th ACM*

SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1889–1898. 6

[Khosla et al., 2020] Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. (2020). Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673. 24

[Kim et al., 2021] Kim, T., Oh, J., Kim, N., Cho, S., and Yun, S.-Y. (2021). Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation. *arXiv preprint arXiv:2105.08919*. 77

[Kingma and Ba, 2015] Kingma, D. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA. 48

[Kingma and Welling, 2014] Kingma, D. P. and Welling, M. (2014). Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. 4, 17

[Kolesnikov et al., 2019] Kolesnikov, A., Zhai, X., and Beyer, L. (2019). Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1920–1929. 48

[Komodakis and Gidaris, 2018] Komodakis, N. and Gidaris, S. (2018). Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations (ICLR)*. 11, 77

[Krishna et al., 2022] Krishna, T., Rai, A. K., Djilali, Y. A., Smeaton, A. F., McGuinness, K., and O'Connor, N. E. (2022). Dynamic channel selection in self-supervised learning. In *24th Irish Machine Vision and Image Processing Conference*. 69, 72, 73

[Krizhevsky and Hinton, 2009] Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario. 2, 63, 72

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105. 2, 23, 24

[Larsson et al., 2016] Larsson, G., Maire, M., and Shakhnarovich, G. (2016). Learning representations for automatic colorization. In *European Conference on Computer Vision*, pages 577–593. Springer. 11

[Larsson et al., 2017] Larsson, G., Maire, M., and Shakhnarovich, G. (2017). Colorization as a proxy task for visual understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6874–6883. 11

[Le,] Le, A. Predicting visual saliency: Where do people look? 6

[Le-Khac et al., 2020] Le-Khac, P. H., Healy, G., and Smeaton, A. F. (2020). Contrastive representation learning: A framework and review. *IEEE Access*. 13, 26, 42

[Lebreton and Raake, 2018] Lebreton, P. and Raake, A. (2018). Gbvs360, bms360, prosal: Extending existing saliency prediction models from 2d to omnidirectional images. *Signal Processing: Image Communication*, 69:69–78. 41, 54

[LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324. 2

[Li et al., 2022] Li, A. C., Efros, A. A., and Pathak, D. (2022). Understanding collapse in non-contrastive siamese representation learning. In *European Conference on Computer Vision*, pages 490–505. Springer. 17

[Li et al., 2019] Li, C., Xu, M., Jiang, L., Zhang, S., and Tao, X. (2019). Viewport proposal cnn for 360° video quality assessment. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10169–10178. IEEE. 54

[Li et al., 2021] Li, F., Li, G., He, X., and Cheng, J. (2021). Dynamic dual gating neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5330–5339. 19, 58, 59, 60, 61, 68, 69, 70, 71, 72

[Li, 2002] Li, Z. (2002). A saliency map in primary visual cortex. *Trends in Cognitive Sciences*, 6(1):9–16. 54

[Lin et al., 2017] Lin, J., Rao, Y., Lu, J., and Zhou, J. (2017). Runtime neural pruning. *Advances in neural information processing systems*, 30. 59

[Lin et al., 2014] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer. 2

[Linardos et al., 2019] Linardos, P., Mohedano, E., Nieto, J. J., O'Connor, N. E., Giró-i-Nieto, X., and McGuinness, K. (2019). Simple vs complex temporal recurrences for video saliency prediction. In *30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*, page 182. BMVA Press. 42

[Liu et al., 2018] Liu, Y., Yang, L., Xu, M., and Wang, Z. (2018). Rate control schemes for panoramic video coding. *Journal of Visual Communication and Image Representation*, 53:76–85. 54

[Liu et al., 2017] Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., and Zhang, C. (2017). Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, pages 2736–2744. 58

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110. 24

[Luz et al., 2017] Luz, G., Ascenso, J., Brites, C., and Pereira, F. (2017). Saliency-driven omnidirectional imaging adaptive coding: Modeling and assessment. In *2017 IEEE 19th international workshop on multimedia signal processing (MMSP)*, pages 1–6. IEEE. 41, 54

[Ma and Zhang, 2008] Ma, Q. and Zhang, L. (2008). Image quality assessment with visual attention. In *2008 19th International Conference on Pattern Recognition*, pages 1–4. IEEE. 54

[Maddison et al., 2017] Maddison, C., Mnih, A., and Teh, Y. (2017). The concrete distribution: A continuous relaxation of discrete random variables. In *Proceedings of the international conference on learning Representations*. International Conference on Learning Representations. 20

[Maugey et al., 2017] Maugey, T., Le Meur, O., and Liu, Z. (2017). Saliency-based navigation in omnidirectional image. In *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE. 41, 54

[McGill and Perona, 2017] McGill, M. and Perona, P. (2017). Deciding how to decide: Dynamic routing in artificial neural networks. In *International Conference on Machine Learning*, pages 2363–2372. PMLR. 19

[McInnes et al., 2018] McInnes, L., Healy, J., Saul, N., and Großberger, L. (2018). Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29). 75

[Meng et al., 2022] Meng, J., Yang, L., Shin, J., Fan, D., and Seo, J.-S. (2022). Contrastive dual gating: Learning sparse features with contrastive learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12247–12255. 69

[Meng et al., 2020] Meng, Y., Panda, R., Lin, C.-C., Sattigeri, P., Karlinsky, L., Saenko, K., Oliva, A., and Feris, R. (2020). Adafuse: Adaptive temporal fusion network for efficient action recognition. In *International Conference on Learning Representations*. 57, 61

[Menon et al., 2020] Menon, A. K., Jayasumana, S., Rawat, A. S., Jain, H., Veit, A., and Kumar, S. (2020). Long-tail learning via logit adjustment. In *International Conference on Learning Representations*. 3

[Mikolov et al., 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*. 3

[Minderer et al., 2020] Minderer, M., Bachem, O., Houlsby, N., and Tschannen, M. (2020). Automatic shortcut removal for self-supervised representation learning. In *International Conference on Machine Learning*, pages 6927–6937. PMLR. 53

[Misra and Maaten, 2020] Misra, I. and Maaten, L. v. d. (2020). Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717. 40, 46, 48

[Mohedano et al., 2016] Mohedano, E., McGuinness, K., O'Connor, N. E., Salvador, A., Marques, F., and Giró-i Nieto, X. (2016). Bags of local convolutional features for scalable instance search. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pages 327–331. 23, 25

[Monroy et al., 2018] Monroy, R., Lutz, S., Chalasani, T., and Smolic, A. (2018). Salnet360: Saliency maps for omni-directional images with cnn. *Signal Processing: Image Communication*, 69:26–34. 42

[Navaneet et al., 2021] Navaneet, K. L., Koohpayegani, S. A., Tejankar, A., and Pirsiavash, H. (2021). Simreg: Regression as a simple yet effective tool for self-supervised knowledge distillation. In *British Machine Vision Conference (BMVC)*. 67, 70, 72, 74

[Neill, 2020] Neill, J. O. (2020). An overview of neural network compression. *arXiv preprint arXiv:2006.03669*. 18

[Netzer et al., 2011] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*. 2, 3

[Nister and Stewenius, 2006] Nister, D. and Stewenius, H. (2006). Scalable recognition with a vocabulary tree. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2161–2168. Ieee. 24

[Noroozi and Favaro, 2016] Noroozi, M. and Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer. 3, 4, 11, 52

[Oord et al., 2018] Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*. 13, 25

[Oquab et al., 2023] Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al. (2023). Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*. 4

[Ozbulak et al., 2023] Ozbulak, U., Lee, H. J., Boga, B., Anzaku, E. T., Park, H., Van Messem, A., De Neve, W., and Vankerschaver, J. (2023). Know your self-supervised learning: A survey on image-based generative and discriminative training. *arXiv preprint arXiv:2305.13689*. vi, 4, 26

[Pan et al., 2017] Pan, J., Ferrer, C. C., McGuinness, K., O'Connor, N. E., Torres, J., Sayrol, E., and Giro-i Nieto, X. (2017). Salgan: Visual saliency prediction with generative adversarial networks. *arXiv preprint arXiv:1701.01081*. 42, 49

[Pan et al., 2016] Pan, J., Sayrol, E., Giro-i Nieto, X., McGuinness, K., and O'Connor, N. E. (2016). Shallow and deep convolutional networks for saliency prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 598–606. 42, 54

[Pan et al., 2022] Pan, S., Qin, Y., Li, T., Li, X., and Hou, L. (2022). Momentum contrastive pruning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2646–2655. 69

[Pathak et al., 2016a] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. (2016a). Context encoders: Feature learning by inpainting. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2536–2544. 3

[Pathak et al., 2016b] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. (2016b). Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2536–2544. 11

[Perronnin and Dance, 2007] Perronnin, F. and Dance, C. (2007). Fisher kernels on visual vocabularies for image categorization. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE. 23, 24

[Perronnin and Larlus, 2015] Perronnin, F. and Larlus, D. (2015). Fisher vectors meet neural networks: A hybrid classification architecture. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3743–3752. 25

[Perronnin et al., 2010] Perronnin, F., Liu, Y., Sánchez, J., and Poirier, H. (2010). Large-scale image retrieval with compressed fisher vectors. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3384–3391. IEEE. 24

[Philbin et al., 2007] Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE. 6, 24, 29

[Philbin et al., 2008] Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2008). Lost in quantization: Improving particular object retrieval in large scale image databases. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE. 29

[Radenović et al., 2018] Radenović, F., Iscen, A., Tolias, G., Avrithis, Y., and Chum, O. (2018). Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5706–5715. 29

[Radenović et al., 2015] Radenović, F., Jégou, H., and Chum, O. (2015). Multiple measurements and joint dimensionality reduction for large scale image search with short vectors. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 587–590. 24

[Radenović et al., 2016] Radenović, F., Tolias, G., and Chum, O. (2016). Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *European conference on computer vision*, pages 3–20. Springer. 23, 25

[Rai et al., 2017] Rai, Y., Gutiérrez, J., and Le Callet, P. (2017). A dataset of head and eye movements for 360 degree images. In *Proceedings of the 8th ACM on Multimedia Systems Conference*, pages 205–210. 1, 37, 48

[Rana et al., 2019] Rana, A., Ozcinar, C., and Smolic, A. (2019). Towards generating ambisonics using audio-visual cue for virtual reality. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2012–2016. vii, 39

[Ren et al., 2016] Ren, S., He, K., Girshick, R., and Sun, J. (2016). Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149. 23

[Revaud et al., 2019] Revaud, J., Almazán, J., Rezende, R. S., and Souza, C. R. d. (2019). Learning with average precision: Training image retrieval with a listwise loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5107–5116. 23, 25, 29, 31

[Russakovsky et al., 2015] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252. 2, 25, 29

[Scardapane et al., 2024] Scardapane, S., Baiocchi, A., Devoto, A., Marsocci, V., Minervini, P., and Pomponi, J. (2024). Conditional computation in neural networks: principles and research trends. *arXiv preprint arXiv:2403.07965*. vi, 20

[Sermanet et al., 2018] Sermanet, P., Lynch, C., Chebotar, Y., Hsu, J., Jang, E., Schaal, S., Levine, S., and Brain, G. (2018). Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1134–1141. IEEE. 24

[Sharif Razavian et al., 2014] Sharif Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813. 24

[Simonyan and Zisserman, 2015] Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*. 2, 23, 24, 43

[Sitzmann et al., 2018] Sitzmann, V., Serrano, A., Pavel, A., Agrawala, M., Gutierrez, D., Masia, B., and Wetzstein, G. (2018). Saliency in vr: How do people

explore virtual environments? *IEEE transactions on visualization and computer graphics*, 24(4):1633–1642. 37, 48

[Sivic and Zisserman, 2003] Sivic, J. and Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *Computer Vision, IEEE International Conference on*, volume 3, pages 1470–1470. IEEE Computer Society. 24

[Song and Ermon, 2019] Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32. 4

[Sprague and Ballard, 2004] Sprague, N. and Ballard, D. (2004). Eye movements for reward maximization. In *Advances in neural information processing systems*, pages 1467–1474. 54

[Startsev and Dorr, 2018] Startsev, M. and Dorr, M. (2018). 360-aware saliency estimation with conventional image saliency predictors. *Signal Processing: Image Communication*, 69:43–52. 41, 54

[Suzuki and Yamanaka, 2018] Suzuki, T. and Yamanaka, T. (2018). Saliency map estimation for omni-directional image considering prior distributions. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2079–2084. IEEE. 42, 54

[Szegedy et al., 2015] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9. 2

[Thakur et al., 2011] Thakur, S., Paul, S., Mondal, A., Das, S., and Abraham, A. (2011). Face detection using skin tone segmentation. In *2011 World Congress on Information and Communication Technologies*, pages 53–60. IEEE. 42

[Tian et al., 2019a] Tian, Y., Krishnan, D., and Isola, P. (2019a). Contrastive multiview coding. *CoRR*, abs/1906.05849. 45, 46

[Tian et al., 2019b] Tian, Y., Krishnan, D., and Isola, P. (2019b). Contrastive representation distillation. In *International Conference on Learning Representations*. 13, 70

[Tian et al.,] Tian, Y., Krishnan, D., Research, G., and Isola, P. CONTRASTIVE REPRESENTATION DISTILLATION. Technical report. 67

[Tian et al., 2020a] Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., and Isola, P. (2020a). What makes for good views for contrastive learning? *Advances in neural information processing systems*, 33:6827–6839. 24

[Tian et al., 2020b] Tian, Y., Yu, L., Chen, X., and Ganguli, S. (2020b). Understanding self-supervised learning with dual deep networks. *arXiv preprint arXiv:2010.00578*. 17

[Tiwari et al., 2021] Tiwari, R., Bamba, U., Chavan, A., and Gupta, D. (2021). Chipnet: Budget-aware pruning with heaviside continuous approximations. In *International Conference on Learning Representations*. 19, 57, 59

[Tolias et al., 2016] Tolias, G., Sicre, R., and Jégou, H. (2016). Particular object retrieval with integral max-pooling of cnn activations. In *ICLR 2016-International Conference on Learning Representations*, pages 1–12. 23, 25, 27, 28

[Veale et al., 2017] Veale, R., Hafed, Z. M., and Yoshida, M. (2017). How is visual salience computed in the brain? insights from behaviour, neurobiology and modelling. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 372(1714):20160113. 54

[Veit and Belongie, 2018] Veit, A. and Belongie, S. (2018). Convolutional networks with adaptive inference graphs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–18. 7, 18, 19, 57, 59, 61, 67, 68, 71

[Wang and Jiang, 2015] Wang, S. and Jiang, S. (2015). Instre: a new benchmark for instance-level object retrieval and recognition. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 11(3):1–21. 29

[Wang et al., 2018a] Wang, X., Girshick, R., Gupta, A., and He, K. (2018a). Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803. 44

[Wang et al., 2018b] Wang, X., Yu, F., Dou, Z.-Y., Darrell, T., and Gonzalez, J. E. (2018b). Skipnet: Learning dynamic routing in convolutional networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 409–424. 19

[Wen et al., 2016] Wen, W., Wu, C., Wang, Y., Chen, Y., and Li, H. (2016). Learning structured sparsity in deep neural networks. *Advances in neural information processing systems*, 29. 58

[Wu et al., 2020] Wu, W., He, D., Tan, X., Chen, S., Yang, Y., and Wen, S. (2020). Dynamic inference: A new approach toward efficient video action recognition.

In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 676–677. 19

[Wu et al., 2018a] Wu, Z., Nagarajan, T., Kumar, A., Rennie, S., Davis, L. S., Grauman, K., and Feris, R. (2018a). Blockdrop: Dynamic inference paths in residual networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 19

[Wu et al., 2018b] Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. (2018b). Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742. 13, 24, 25, 27, 46, 52

[Xu et al., 2020a] Xu, G., Liu, Z., Li, X., and Loy, C. C. (2020a). Knowledge distillation meets self-supervision. In *European Conference on Computer Vision*, pages 588–604. Springer. 70

[Xu et al., 2020b] Xu, M., Li, C., Zhang, S., and Le Callet, P. (2020b). State-of-the-art in 360 video/image processing: Perception, assessment and compression. *IEEE Journal of Selected Topics in Signal Processing*, 14(1):5–26. 38, 54

[Xu et al., 2018a] Xu, M., Song, Y., Wang, J., Qiao, M., Huo, L., and Wang, Z. (2018a). Predicting head movement in panoramic video: A deep reinforcement learning approach. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2693–2708. 47

[Xu et al., 2018b] Xu, Y., Dong, Y., Wu, J., Sun, Z., Shi, Z., Yu, J., and Gao, S. (2018b). Gaze prediction in dynamic 360 immersive videos. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5333–5342. ix, 37, 48, 49

[Yan et al., 2018] Yan, Y., Zhaoping, L., and Li, W. (2018). Bottom-up saliency and top-down learning in the primary visual cortex of monkeys. *Proceedings of the National Academy of Sciences*, 115(41):10499–10504. 54

[Yandex and Lempitsky, 2015] Yandex, A. B. and Lempitsky, V. (2015). Aggregating local deep features for image retrieval. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1269–1277. 23, 25

[Yang et al., 2023] Yang, C., An, Z., Zhou, H., Zhuang, F., Xu, Y., and Zhang, Q. (2023). Online knowledge distillation via mutual contrastive learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 67

[Yang et al., 2017] Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O., and Li, H. (2017). High-resolution image inpainting using multi-scale neural patch synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6721–6729. 4

[Yang et al., 2016] Yang, J., Parikh, D., and Batra, D. (2016). Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5147–5156. 12

[You et al., 2017] You, Y., Gitman, I., and Ginsburg, B. (2017). Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*. 72

[Yu et al., 2019] Yu, J., Yang, L., Xu, N., Yang, J., and Huang, T. (2019). Slimmable neural networks. In *International Conference on Learning Representations*. 71

[Zbontar et al., 2021] Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. (2021). Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR. 14, 15, 69, 75

[Zhang et al., 2019] Zhang, H., Goodfellow, I., Metaxas, D., and Odena, A. (2019). Self-attention generative adversarial networks. In *International conference on machine learning*, pages 7354–7363. PMLR. 44, 45

[Zhang et al., 2016a] Zhang, R., Isola, P., and Efros, A. A. (2016a). Colorful image colorization. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 649–666. Springer. 4

[Zhang et al., 2016b] Zhang, R., Isola, P., and Efros, A. A. (2016b). Colorful image colorization. In *European Conference on Computer Vision*, pages 649–666. Springer. 11

[Zhang et al., 2018] Zhang, T., Ye, S., Zhang, K., Tang, J., Wen, W., Fardad, M., and Wang, Y. (2018). A systematic dnn weight pruning framework using alternating direction method of multipliers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 184–199. 70

[Zhao et al., 2023] Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al. (2023). A survey of large language models. *arXiv preprint arXiv:2303.18223*. 3

[Zhu et al., 2018a] Zhu, C., Huang, K., and Li, G. (2018a). An innovative saliency guided roi selection model for panoramic images compression. In *2018 Data Compression Conference*, pages 436–436. IEEE. 54

[Zhu et al., 2018b] Zhu, Y., Zhai, G., and Min, X. (2018b). The prediction of head and eye movement for 360 degree images. *Signal Processing: Image Communication*, 69:15–25. 42

[Zhuang et al., 2020] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76. 3