Dublin City University

School of Electronic Engineering

Saliency modelling for 2D and 360° visual data: deep learning models and performance evaluation

Yasser Abdelaziz Dahou Djilali, MEng

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

Supervised by:

Prof. Noel E. O'Connor and Dr Kevin McGuinness

July 2024

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, and that 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.

Signature:

Yasser Abdelaziz Dahou Djilali

Student ID: 19213990

Date: 1^{st} of December 2023

SUMMARY

This thesis, *"Saliency modeling for 2D and 360° visual data: deep learning models and performance evaluation"* presents research in the area of visual attention modeling, specifically saliency prediction for both 2D and 360° data. The improvements seen in saliency prediction research since the successful integration of deep learning rely on the availability of labeled human fixations. This research aims to study current methods, propose better solutions, point out the flaws and limitations of existing approaches, and critically reconsider the saliency learning paradigm. Our contributions are as follows:

- We analyze the self-attention maps of self-supervised pretrained Vision Transformers (ViTs) on saliency prediction datasets in Chapter 3. We find that smoothing the internal attention maps with a few convolutional filters can achieve reasonable saliency maps with acceptable metric scores. We investigate the relationship between a transformer's self-attention mechanism and saliency.

- In chapter 4, we present the Saliency TRansformer (SalTR) model, a novel approach for learning saliency prediction using only the fixation maps. We discuss how we can forecast the saliency as a set prediction and how we can leverage the vanilla transformer decoder to solve it.

- In chapter 5, we target 360° data, a recently introduced type of multimedia that provides the user with an immersive experience (sometimes referred to as panoramic, spherical, or omnidirectional video). The content of omnidirectional

videos is rendered to cover the entire 360° × 180° viewing space along both axes. We propose ATSal, a model that combines an attention mechanism with expert instances for each patch location in the omnidirectional scene, to learn effective features for saliency modeling.

- In chapter 6, we leverage contrastive learning for 360° scenes. In fact, despite the success of self-supervised representation learning on planar data, to date, it has not been studied on 360° images. In this chapter, we extend recent advances in contrastive learning to learn latent representations that are sufficiently invariant to be highly effective for spherical saliency prediction as a downstream task. We argue that omnidirectional images are particularly suited to such an approach due to the geometry of the data domain. 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 good quality saliency distributions from the encoder embeddings.

- Finally in chapter 7, we discuss the evaluation process for this type of data, and show that a simple constant predictor, i.e. the average map across the training sets of ground truth saliency maps, can fool classical saliency metrics and achieve results on par with specialized models. Thus, we propose a new probabilistic metric based on the independent Bernoulli assumption that is more suited to the 360° saliency task.

LIST OF PUBLICATIONS

Publications

This thesis is based on the following publications:

- **Djilali, Yasser Abdelaziz Dahou**, Kevin McGuinness, and Noel O'Connor. "Vision Transformers are inherent saliency learners." Presented at BMVC 2023.

- **Djilali, Yasser Abdelaziz Dahou**, Kevin McGuinness, and Noel O'Connor. "Learning Saliency From Fixations" Accepted at WACV2023.

- **Djilali, Yasser Abdelaziz Dahou**, Marouane Tliba, Kevin McGuinness, and Noel O'Connor. "ATSal: An Attention Based Architecture for Saliency Prediction in 360 Videos." In Pattern Recognition. ICPR: Virtual Event, January 10–15, 2021, Proceedings, Part III, pp. 305-320. Cham: Springer International Publishing, 2021.

- **Djilali, Yasser Abdelaziz Dahou**, Tarun Krishna, Kevin McGuinness, and Noel E. O'Connor. "Rethinking 360deg Image Visual Attention Modelling With Unsupervised Learning." In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 15414-15424. 2021.

- **Djilali, Yasser Abdelaziz Dahou**, Kevin McGuinness, and Noel E. O'Connor. "Simple baselines can fool 360deg saliency metrics." In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3750-3756. 2021.

The author of this thesis has further contributed to the following publications:

- **Yasser Abdelaziz Dahou Djilali**, Sanath Narayan, Eustache LeBihan, Haithem Boussaid, Ebtessam Almazrouei, Merouane Debbah. "Do VSR Models Generalize Beyond LRS3?" Accepted at WACV2023.

- **Dahou Djilali, Yasser Abdelaziz**, Mohamed Sayah, Kevin McGuinness, and Noel E. O'Connor. "On the Use of 3D CNNs for Video Saliency Modeling." In Computer Vision, Imaging and Computer Graphics Theory and Applications: 15th International Joint Conference, VISIGRAPP 2020 Valletta, Malta, February 27–29, 2020, Revised Selected Papers, pp. 453-476. Cham: Springer International Publishing, 2022.

- Krishna, Tarun, Ayush K. Rai, **Yasser AD Djilali**, Alan F. Smeaton, Kevin McGuinness, and Noel E. O'Connor. "Dynamic Channel Selection in Self-Supervised Learning." IMVIP 2022.

- **Djilali, Yasser Abdelaziz Dahou**, Mohamed Sayah, Kevin McGuinness, and Noel E. O'Connor. "3DSAL: An Efficient 3D-CNN Architecture for Video Saliency Prediction." In 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications.

- Tliba, Marouane, Mohamed Sayah, and **Yasser Abdelaziz Dahou Djilali**. "2d-based saliency prediction framework for omnidirectional–360° video." In 11th International Conference of Pattern Recognition Systems (ICPRS 2021), vol. 2021, pp. 31-37. IET, 2021.

- Achab, Mastane, Reda Alami, **Yasser Abdelaziz Dahou Djilali**, Kirill Fedyanin, Eric Moulines, and Maxim Panov. "Distributional deep Q-learning with CVaR regression." In Deep Reinforcement Learning Workshop NeurIPS 2022.

- **Djilali, Yasser Abdelaziz Dahou**, Kebin Wu, Kevin McGuinness, Ebtessam Almazrouei, Merouane Debbah, and Noel O'Connor. "Augmenting Object De-

tection Supervised Training with Pixel-level Contrastive Learning." Accepted at BMVC-W 2022.

- **Yasser Abdelaziz Dahou Djilali**, Sanath Narayan, Haithem Boussaid, Ebtessam Almazrouei, Merouane Debbah. "Lip2Vec: Efficient and Robust Visual Speech Recognition via Latent-to-Latent Visual to Audio Representation Mapping." Presented at ICCV 2023.

- Achab, Mastane, Reda Alami, **Yasser Abdelaziz Dahou Djilali**, Kirill Fedyanin and Eric Moulines. "One-Step Distributional Reinforcement Learning." Accepted at TMLR.

- Achab, Mastane, Reda Alami, **Yasser Abdelaziz Dahou Djilali**, Kirill Fedyanin and Eric Moulines. "One-Step Distributional Reinforcement Learning." Accepted at TMLR.

- Achab, Mastane, Reda Alami, **Yasser Abdelaziz Dahou Djilali**, Kirill Fedyanin and Eric Moulines. "One-Step Distributional Reinforcement Learning." Accepted at TMLR.

Contents

SUMMARY iii

LIST OF PUBLICATIONS v

Abstract 1

Glossary 5

1 Introduction 7
1.1 Human Visual System (HVS) . . . 7
1.2 Artificial Intelligence and Deep Learning . . . 10
1.2.1 Saliency in this context . . . 11
1.3 Scope and Research Questions . . . 13

2 Literature review 17
2.1 Introduction . . . 17
2.2 Background and setup . . . 18
2.2.1 How are models evaluated? . . . 18
2.2.2 Datasets . . . 23
2.3 Static Saliency Models . . . 28
2.3.1 Heuristic Approaches. . . . 29
2.3.2 Data-Driven Approaches . . . 39
2.4 Dynamic Saliency Models . . . 42
2.5 Saliency Prediction for 360° Video/Image . . . 44
2.5.1 360° heuristic approaches . . . 45
2.5.2 360° Deep learning approaches . . . 46
2.6 Visual attention from a neuroscience perspective . . . 47
2.7 Conclusion . . . 52

I Learning with 2D data 53

3 Vision Transformers for image saliency prediction 57
3.1 Introduction . . . 57
3.2 Background . . . 60
3.2.1 Transformers . . . 61
3.2.2 (Vision) Transformers (ViTs) . . . 66
3.2.3 How ViTs really work? . . . 69
3.2.4 Connections between Vision Transformers and CNNs . . . 71
3.2.5 Attention in Psychology, Neuroscience, and Machine Learning 72

3.3 Are ViTs inherently saliency learners? . . . 73
3.3.1 ViTs Self-Attention Visualisation . . . 74
3.3.2 Salient Attention . . . 77
3.3.3 Obtaining the saliency maps from ViT Attention/Relevance maps . . . 79
3.4 Experiments . . . 80
3.5 Results . . . 82
3.5.1 Supervised vs. Self-Supervised self-attention maps . . . 85
3.5.2 Saliency for low-level features . . . 86
3.6 Ablation study . . . 90
3.7 Conclusion . . . 93

4 Learning saliency from fixations only 95
4.1 Introduction . . . 95
4.2 Saliency Transformers (SalTR) . . . 96
4.2.1 SalTR components . . . 96
4.2.2 Learning dynamics . . . 98
4.3 Experimental setting . . . 100
4.4 Results . . . 102
4.4.1 Saliency for low-level features. . . . 104
4.4.2 Failure cases. . . . 105
4.5 Ablation study . . . 106
4.6 SalTR for scanpath prediction . . . 110
4.7 Ideas on how SalTR can be extended to satisfy more constraints on saliency . . . 111
4.8 Conclusion . . . 113

II Learning with 360° data 115

5 Saliency for 360° Images/Videos 119
5.1 Introduction . . . 119
5.2 Background . . . 122
5.2.1 Equirectangular Projection (ERP) . . . 122
5.2.2 Cubemap Projection (CMP) . . . 124
5.3 Why this is important? . . . 126
5.3.1 Foveated Rendering . . . 126
5.4 Method . . . 128
5.4.1 Attention mechanism . . . 129
5.4.2 Expert models . . . 132
5.4.3 Loss function for the attention stream . . . 133
5.5 Experiments . . . 134
5.5.1 Results . . . 135
5.5.2 Limitations . . . 139
5.6 Conclusion . . . 140

6 Learning implicit geometrical features for 360° images. 143
6.1 Introduction . . . 143
6.2 Related prior work . . . 144

6.3 Method . . . 146
6.3.1 Overview of the Approach . . . 147
6.3.2 Unsupervised Module . . . 148
6.3.3 Supervised Module . . . 150
6.4 Experimental Setup . . . 152
6.4.1 Results . . . 154
6.4.2 Additional results . . . 157
6.5 Ablation study. . . . 162
6.6 Architecture Details . . . 164
6.6.1 Self-Attention . . . 165
6.7 Conclusion . . . 167

7 Benchmarking saliency prediction for 360° images/videos 169
7.1 Introduction . . . 169
7.2 Works investigating saliency metrics . . . 171
7.3 Evaluation setup . . . 173
7.3.1 Saliency formulation . . . 176
7.4 Experiments and results . . . 180
7.5 Recommendations for designing the metrics . . . 182
7.6 Conclusion . . . 184

8 Conclusion 185

References 189

Abstract

"Saliency modelling for 2D and 360° visual data: deep learning models and performance evaluation"

Yasser Abdelaziz Dahou Djilali

This thesis explores the field of visual attention modeling, focusing on saliency prediction for both 2D and 360° visual data. The research is grounded in integrating deep learning techniques with human fixation data, aiming to refine current methods and address existing approaches' limitations. Key contributions include the analysis of self-attention maps in Vision Transformers for improved saliency prediction and developing a transformer-based model for image saliency. Then, we target the field of 360° data and introduce ATSal, which integrates attention mechanisms with expert instances for omnidirectional scene analysis. Additionally, the thesis explores the application of contrastive learning to 360° scenes, presenting an unsupervised framework that effectively learns general representations. The final chapter introduces a new probabilistic metric suited for 360° saliency. This comprehensive study advances the understanding and application of deep learning in visual attention modeling.

Glossary

AI Artificial Intelligence.

AUC Area Under the Curve.

BCE Binary Cross Entropy.

BERT Bidirectional Encoder Representations from Transformers.

BiRNN Bidirectional Recurrent Neural Network.

BMS Boolean Map Saliency.

BU Bottom-up.

CC Correlation Coefficient.

CKA Centered Kernel Alignement.

CMP Cubemap Projection.

CNN Convolutional Neural Network.

CPC Contrastive Predictive Coding.

DCT Discrete Cosine Transform.

DINO DIstillation with NO labels.

DL Deep Learning.

DVA Dynamic Visual Attention.

ECSF The Extended Contrast Sensitivity Function.

EM Eye movements.

EMD Earth Mover's Distance.

ERP Equirectangular Projection.

FOV Field-of-view.

GANs Generative Adversarial Networks.

GBVS Graph-Based Visual Saliency.

HM Head movements.

HMD Head Mounted Display.

HSIC Hilbert-Schmidt Criterion.

HVS Human Visual System.

ICL Incremental Coding Length.

IG Information Gain.

IQA Image quality assessment.

KLD Kullback-Leibler Divergence.

LRP Layer-wise Relevance Propagation.

LSB Least Significant Bit.

LSTM Long short-term memory.

MI Mutual Information.

MLP Multi-layer perceptron.

NCE Noise Contrastive Estimation.

NLP Natural language processing.

NSS Normalized Scanpath Saliency.

ODI omnidirectional images.

ODV omnidirectional videos.

RNN Recurrent Neural Networks.

ROC Receiver Operating Characteristic.

ROI Region-of-interest.

s-AUC Shuffled Area Under the Curve.

SalTR Saliency Transformers.

SGT Ground-truth Saliency Map.

SIM Similarity Metric.

SNR Signal- to-Noise Ratio.

Sp Predicted Saliency Map.

TD Top-down.

ViT Vision Transformer.

VQA Video quality assessment.

Chapter 1

Introduction

1.1 Human Visual System (HVS)

Human attention is the key underpinning consideration in saliency research. The terms "attention" and "saliency" are highly correlated in most psychological definitions and are often used interchangeably in the computer vision community. However, attention is an old concept investigated first in psychology and neuroscience and most recently as a computer vision problem in computer science [Mancas et al. 187]. Early advances in psychology [James 127] related attention to the focalization and concentration of consciousness as opposed to the concept of distraction. [Sohlberg and Mateer 252] framed the question of attention into five main levels that hierarchically build upon one another:

- **Focused attention**: This refers to the capacity to concentrate on a specific piece of information or a particular task while ignoring other distractions. It is about zeroing in on something specific in the environment and giving it full conscious awareness;

- **Sustained attention**: This is about maintaining focus on a single activity or stimulus over an extended period. It involves staying vigilant and continuously applying one's cognitive resources to the task at hand, like reading a book or watching a movie from start to finish without getting distracted;

- **Selective attention**: This type involves focusing on certain stimuli while ignoring others. It is like being in a noisy room but only paying attention to the conversation one is engaged in, effectively tuning out other conversations or background noise;

- **Alternating attention**: This is the ability to switch the focus back and forth between different tasks, which might have varying cognitive demands. It is similar to multitasking but involves shifting the full attention from one task to another, like switching between writing a thesis and watching football;

- **Divided attention**: This is the most complex form of attention, where a human attempts to process multiple tasks simultaneously. It requires distributing one cognitive resource across several activities, like listening to music while writing a thesis;

The feature integration theory proposed by [Treisman and Gelade 266] stands out as the pioneering work describing which visual features direct human attention. Indeed, this formed the starting point of many computational models. Furthermore, computer vision approaches mainly consider the selective mechanism when modeling attention, where the subtle definition of saliency would relate to the gaze policy on the scene , i.e., characterizing some subsets of the space on which a human subject would focus. The term "salient" emerged in the context of bottom-up computations [Koch and Ullman 142, Itti et al. 124], while attention refers more broadly to scene-driven bottom-up (BU) or expectation-driven top-down (TD) mechanisms. It should be noted that computational models have been proposed to imitate the attentional mechanisms of the human visual system (HVS) for both dynamic and static scenes.

The key questions when modeling attention are how, when, and why human behavior selects pertinent image regions. Inspired by anatomical findings and the early understanding of how the human visual system functions, many interesting results were derived demonstrating that visual attention could be influenced by the regions maximizing a reward in a task-driven scenario [Sprague and Ballard 253],

which are typically the most informative or surprising regions [Itti and Baldi 122, Bruce and Tsotsos 32]. This, in turn, led to pioneering seminal works in saliency prediction [Itti et al. 124, Tsotsos et al. 267, Milanese 194, Rosenholtz 235, Salvucci 240, Boccignone and Ferraro 20, Rimey and Brown 234]. For an extensive survey on eye movements, readers are referred to [Privitera and Stark 223]. Despite the deep thinking behind these methods and the elegant mathematical formulation of the mathematical prediction models, all of these approaches share the paradigm of learning the representation in one step, which does not reflect how humans process the sensory data in multiple cognitive layers. Consequently, the performances of these approaches were far from the "infinite humans" baselines [Judd et al. 135].

Predicting where people look in images and videos is useful in various applications across several domains (e.g., computer vision, robotics, assistive systems, human-computer interaction, and defense). Some example applications include gaze-aware compression and summarization, activity recognition, object segmentation, recognition and detection, image captioning, question answering, advertisement, novice training, patient diagnosis, and surveillance. The work of [Mishra and Aloimonos 197] proposed an algorithm that uses the fixation information to find the bounding contour and achieves the segmentation of an object. [Ma et al. 185] introduced a saliency-based method for both image Quality Assessment (IQA) and Video Quality Assessment QA (VQA) by adding weights related to saliency features to these original IQA or VQA criteria. For Image Matching, [Siagian and Itti 248] computed saliency as a measure of interest at every image location, efficiently directing the time-consuming landmark-identification process toward the most likely candidate locations in the image. For image rendering, [DeCarlo and Santella 67] developed a system that identifies the meaningful elements of an image using a model of human perception and a record of a user's eye movements in looking at the photo; the system renders a new image using transformations that preserve and highlight these visual elements. [Marchesotti et al. 190] used saliency for image thumbnailing, whereby they refine a saliency map to extract thumbnails. The work of [Jacobson et al. 125]

used saliency for motion-compensated frame interpolation. Indeed, a discriminant saliency classifier is utilized to determine which regions of the motion field are most important to a human observer. These regions are refined using multistage motion vector refinement, which promotes motion vector candidates based on their likelihood given a local neighborhood. For regions that fall below the saliency threshold, frame segmentation is used to locate regions of homogeneous color and texture via normalized cuts. [Sadaka and Karam 239] leveraged saliency for a super-resolution algorithm to reduce computational complexity, where the active pixels lying in the attended regions are processed at higher accuracy by the Super-Resolution method relative to pixels in other regions. Saliency has been also used for object detection [VOCUS 272, Navalpakkam and Itti 203, Butko and Movellan 33, Ehinger et al. 82]. This brief review shows that saliency prediction can be considered a key enabling technology for many other tasks. It is thus highly desirable to have robust, high-performance video saliency prediction models.

1.2 Artificial Intelligence and Deep Learning

The desire to understand human cognition has spawned a variety of scientific disciplines. Cognitive science, neuroscience, and the study of machine learning and artificial intelligence (AI) are the most prominent examples of scientific fields that study the human mind (cognitive science), its physical neural substrate (neuroscience), and the replication of some of its behavior and core algorithms in artificial machines (machine learning and AI).

The work in this thesis is situated in the field of deep learning, which is one of the most widely pursued branches in AI research. Deep learning deals with the question of how we can build systems and design algorithms that learn from data and experience (e.g., by interacting with an environment), which is in contrast to the traditional approach in computer vision, where systems are explicitly programmed to follow a precisely outlined sequence of instructions. The problem of learning is commonly approached by fitting a model to data with the goal that this learned

model will generalize to new data.

Traditionally, many machine learning models are built on top of sets of features extracted using a pre-defined procedure from the raw data format. For example, such features could be word occurrence statistics in natural language sentences or pixel statistics in image data. Developing sophisticated feature extractors is often referred to as feature engineering, culminating in developing popular feature detection algorithms such as SIFT [Lowe 181].

Deep learning, an approach that has enjoyed immense popularity in the past decade, instead addresses the learning problem by jointly learning representations of the raw input data and a predictive model for the task at hand. This is usually achieved by stacking multiple 'layers' of differentiable non-linear transformations and training such a model in an end-to-end fashion using gradient descent techniques. The resulting class of models is often referred to as deep neural networks.

1.2.1 Saliency in this context

Saliency intuitively characterizes some parts of a scene, such as objects or regions that appear to an observer to stand out relative to their neighboring parts. The term saliency is often considered in the context of bottom-up computations [Itti et al. 124], [Koch and Ullman 142]. **Gaze**, a coordinated motion of the eyes and head, has often been used as a proxy for attention in natural behavior (see [Hayhoe and Ballard 109]). For instance, a human has to interact with surrounding objects and control his/her gaze to perform a task while moving in the environment.

To model human visual attention computationally, researchers gather human fixation from observers using an eye-motion tracker. Participants are required to view the stimuli (image or video) wearing the tracking device. Finally, each stimulus is dedicated to a map of fixations defined as:

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

The ground truth saliency maps are computed by convolving each fixation or trajectory point (for all observers of one stimulus) with a Gaussian or Kent kernel. A Gaussian kernel, based on the Gaussian (or normal) distribution, smooths data points such that each point contributes to the estimate with a weight decreasing with distance from the point, following the shape of a normal distribution curve. This kernel is particularly effective for data in Euclidean spaces, commonly used in kernel density estimation and regression, and is used for 2D saliency. On the other hand, the Kent kernel, also known as the Fisher-Bingham distribution, is used for directional data on a sphere, providing a way to handle data on spherical surfaces by incorporating concentration parameters that define the elliptical contours of the distribution. The Kent kernel is useful in fields such as geostatistics and meteorology, where directional data is prevalent. This type of kernel is usually used for spherical saliency. The resulting saliency map $S_{GT} \in [0,1]^{W \times H}$ 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. The computer vision task is developing algorithms to predict an average saliency map given an input image.

Figure 1.1: Sample images from the MIT300 benchmark (top), the fixation locations from the 39 observers (middle), and the corresponding saliency maps (bottom) (Figure taken from [Borji and Itti 26])

Thanks mainly to advances in deep learning, the development of computational

models of human attention has received renewed research interest in recent years. The common paradigm includes the use of a convolutional neural network (CNN) auto-encoder to map the image to a given representation and then decode it to a predicted saliency map. In this thesis, we argue for the introduction and design of further deep learning models for the saliency prediction task to reflect the wider set of methods and learning paradigms suitable for this task. One way to achieve this is by considering the recent advances in self-supervised learning techniques or by using an inductive bias-free architecture such as the transformers [Vaswani et al. 269], which will be of central interest in this thesis.

1.3 Scope and Research Questions

This research aims to improve deep learning models for saliency prediction for 2D and 360° visual data. The current success and the good performance of saliency models are strongly dependent on the benchmark used. The main challenges are to identify the current flaws in the existing approaches theoretically and practically, study the direct extension of visual features from 2D to 360° data, and ultimately find a better and more appropriate formulation of the saliency task. The following research questions guide the contributions of this thesis:

Research Question 1: How do the Vision Transformers (ViTs) align with human visual attention? Can self-attention maps from these transformers be translated into saliency maps?

In Chapter 3, we comprehensively examine how Vision Transformers (ViTs) utilize local and global spatial information. The findings highlight a compelling similarity between the penultimate self-attention maps produced by Vision Transformers and conventional saliency maps. Building on this understanding, we introduce a procedure that employs tensor decomposition to effectively retrieve the most informative attention maps. These self-attention maps can be conveniently transformed into saliency maps using a shallow convolutional neural network decoder. Furthermore, we present a comparative analysis of Vision Transformers, drawing distinctions between

those trained under supervised classification tasks and those under a self-supervised regime. This comparison reveals the superior adaptability of the self-supervised models in terms of generalizing to saliency. In contrast, their supervised counterparts predominantly concentrate on the most distinguishable objects in the scene.

Research Question 2: Can we optimize for discrete fixation maps to reflect actual human saccades? Can transformers be tailored to predict fixation points by learning directly from fixation maps?

In the saliency prediction literature, the majority of models optimize for the continuous saliency maps to learn useful representations. This choice is motivated by the unstable training dynamics when optimizing for discrete fixation maps. However, such architectures do not emulate the data collection methodology; instead, they lean towards an implicit learning process using the refined saliency map. In Chapter 4, following the use of transformers for saliency, we introduce the "Saliency TRansformer (SalTR)". Hence, we frame the saliency prediction as a set prediction task with the objective of predicting a collection of spatial fixation points. To this end, we leverage a transformer encoder-decoder design with a set of learned fixation queries. This strategy sets our approach apart from existing saliency predictors and offers a fresh perspective on saliency prediction.

Research Question 3: Can we combine local and global features for learning better 360 video saliency?

We address this question by merging local features from expert networks with global features over the equirectangular scene by utilizing a selective attention mechanism. The proposed ATSal integrates an attention stream for global saliency extraction and expert instances designed for different patches of the 360° scene.

Research Question 4: How can we learn a unified representation for 360° image saliency?

For the 360° image analysis, Cube Map Projections present a unique challenge. Usually, when an image undergoes this projection method, it is projected into distinct faces, each treated as separate images. This results in models learning disparate

representations for each face, increasing the computational load and diminishing the overall efficiency. Such an approach not only segregates the global context inherent in 360° imagery but also amplifies the model's processing time. We aim to bridge the gaps between these isolated cube faces by leveraging a contrastive learning technique. By maximizing the Mutual Information (MI) across the CMP faces, we aspire to learn a unified representation for the entire sphere. The design implicitly embeds the geometric specifications in the model weights and offers a clear computational advantage.

Research Question 5: Are the classical metrics suitable for the 360° saliency?

The challenge of evaluating saliency prediction models in 360° environments, a task distinct from traditional 2D contexts, is the applicability of classical saliency metrics, developed primarily for 2D scenarios, which could be questioned in the spherical domain. The primary focus of this question is to assess whether metrics that have proven effective in 2D environments can accurately capture the complexities of viewer interactions and fixation patterns in 360 scenes. To address this, Chapter 7 explores a novel probabilistic metric grounded in the independent Bernoulli assumption, designed explicitly for 360° content. This metric aims to evaluate better a model's ability to predict human fixations in panoramic settings.

Aside from the principal contributions elucidated in this dissertation, Chapter 2 introduces foundational topics on visual saliency, laying the groundwork for comprehensive discussions and analyses in the subsequent chapters. These foundational topics are important in understanding the field of saliency prediction and the novelties presented in this research. After presenting the core of our work, we conclude and outline interesting future directions.

Chapter 2

Literature review

2.1 Introduction

In this chapter, the discussion first centers around saliency datasets and the metrics used to assess progress and compare models. The review then covers important works related to attention modeling for static and dynamic scenes, considering heuristic and data-driven approaches. Additionally, works extending saliency models to 360° data are reviewed. The last decade has witnessed remarkable progress in saliency prediction, with many methods presenting and achieving remarkable performances on the recently introduced benchmarks. Deep learning-based methods, in particular, have yielded a boost in performance. Researchers leverage existing Convolutional Neural Network (CNN) architectures and repurpose them to predict saliency, often introducing some architectural novelties to meet the task requirements. These models are trained end-to-end on fixation datasets, formulating saliency as a regression problem to alleviate the issue of less annotated fixation data compared to the deep learning standards. Indeed, the model's encoder is often pretrained on large image recognition datasets such as ImageNet [Deng et al. 68] to obtain useful representations at the latent space level. Subsequently, it is fine-tuned on saliency datasets.

2.2 Background and setup

Understanding both the theoretical background and the practical methodologies employed in exploring visual attention models is important. In this section, we provide a foundational overview of the principles and techniques used to evaluate saliency models. The focus is on the metrics, their properties, what differentiates them, and when each can be considered. Then, datasets used for training and evaluation are described for multiple modalities.

2.2.1 How are models evaluated?

In this thesis, the term Estimated/Predicted Saliency Map (S_p) refers to a predicted saliency map of a model, while Ground-truth Saliency Map (S_{GT}) denotes a map built by combining recorded eye fixations from all subjects or by combining tagged salient regions by human subjects for each image. S_p represents the model's prediction given an input image, and the aim is to quantitatively evaluate the quality of S_p, by comparing it to a Ground Truth saliency map (S_{GT}). This evaluation can be conducted in multiple ways depending on the assumptions made. One approach is to consider them as probability distributions and use the Kullback-Leibler Divergence (KLD) to measure the probabilistic distance between the respective distributions. Another approach is to consider S_p as a binary classifier and use signal detection theory analysis (Area Under the ROC Curve (AUC) metric) to assess the performance of this classifier. Alternatively, S_p and S_{GT} can be viewed as random variables, and measures like the Correlation Coefficient (CC) or Normalized Scanpath Saliency (NSS) can be used to measure their statistical relationship. Another perspective is to view S_{GT} as a sequence of eye fixations (scanpath) and compare this sequence with the sequence of fixations chosen by a saliency model. From a different viewpoint, evaluation measures for saliency can be classified into two categories: location-based and distribution-based set of metrics. We discuss each category in what follows:

Location-based metrics are used to evaluate saliency models based on their

ability to classify locations of visual fixations accurately. For instance, **AUC-Judd** [Judd et al. 137] is a variant of the Area Under the Curve (AUC) metric designed for saliency modeling, where true and false positive rates are calculated by setting a threshold and then determining the proportion of saliency map values above this threshold at pixels where fixations occur (true positives) versus where they do not (false positives). True/false positives are saliency map values above a threshold at fixated/non-fixated pixels.

Alternatively, **AUC-Borji** [Borji et al. 28] modifies the previous approach by using a uniform random sample of image pixels as the basis for negative samples. False positives are saliency map values exceeding the threshold at these randomly chosen non-fixated pixels. This metric was proposed to address potential biases in selecting negative samples, offering a more balanced and robust evaluation of saliency models.

The shuffled AUC metric, **sAUC** [Borji et al. 29], introduces another variation where negatives are sampled from fixation locations in different images. This method counters the central bias present in many images—since fixations tend to cluster towards the center when averaged over many images, leading to a central Gaussian distribution [Tatler 260]. This central bias is a common challenge in saliency modeling and sAUC aims to provide a fairer comparison by accounting for it.

With the Normalized Scanpath Saliency (**NSS**, [Peters et al. 221]), performance of a saliency map model is defined to be the average saliency value of fixated pixels in the normalized (zero mean, unit variance) saliency maps (i.e., the average z-score of the fixated saliency values), and is defined as follows:

$$NSS(S_p, \mathcal{M}_{GT}) = \frac{1}{N} \sum_i \overline{S_{p_i}} \times \mathcal{M}_{GT_i} \tag{2.1}$$

where, $N = \sum_i \mathcal{M}_{GT_i}$ and $\overline{S_{p_i}} = \frac{S_p - \mu(S_p)}{\sigma(S_p)}$. Zero value NSS means random prediction, positive NSS measures correspondence above random, and negative NSS means anti-correspondence.

In summary, each location-based metric uncovers aspects of human visual at-

tention and the inherent biases in image composition and viewing patterns. While AUC-Judd focuses on the direct alignment with human fixations, AUC-Borji aims to provide a more relaxed evaluation by incorporating randomness in negative sampling. Unlike AUC-based metrics, NSS offers a statistical Contextual Variability by normalizing saliency maps to zero mean and unit variance, allowing for consistent comparisons across diverse images. It also directly emphasizes fixated points, clearly showing a model's ability to capture human attention. NSS is sensitive to both correct and incorrect identifications of fixation points, ensuring a comprehensive evaluation of model performance. Moreover, its normalization process reduces the impact of central bias, making it a robust and interpretable metric in saliency assessment.

Distribution-based metrics. In this category, models are configured to be probabilistic, allowing saliency maps to approximate the underlying distribution rather than fixation locations directly. Various metrics are used to evaluate the accuracy of model predictions against continuous ground-truth saliency maps.

Used as a general entropic distance-like function, the Kullback–Leibler divergence (KLD) is an oriented statistical distance measured between two densities p and q, where p represents the ground truth and q represents the model-predicted saliency maps. KLD expects the inputs to be valid probability distributions, typically achieved by applying softmax over the saliency map pixels. The formula for Kullback–Leibler divergence, denoted as $\mathrm{KL}(S_{GT}||S_p)$, is commonly used to evaluate saliency maps. It is based on the assumption that the ground truth saliency S_{GT} is a categorical distribution over the pixels in the image, represented as $S_{GT} = (S_{GT_1}, \ldots, S_{GT_N})$, where $S_{GT_i} \geq 0$ and $S_{GT}^T \mathbf{1} = 1$. Here, $\mathbf{1}$ is a vector of ones and S_{GT}^T denotes the transpose of the vector S_{GT}, ensuring that the sum of all elements in S_{GT} equals 1. Similarly, given a softmax prediction $S_p = (S_{p_1}, \ldots, S_{p_N})$, where $S_{p_i} \geq 0$ and $S_p^T \mathbf{1} = 1$, the KL-divergence between the predicted distribution and the ground truth is calculated as follows:

$$\mathrm{KL}(S_{GT}||S_p) = \sum_{i=1}^{N} S_{GT_i} \log \frac{S_{GT_i}}{S_{p_i}} \tag{2.2}$$

Regularization constants are often added to the formula for numerical stability.

The correlation coefficient (**CC**, [Jost et al. 134]) measures the correlation between the model saliency map and the empirical saliency map after normalizing both saliency maps to have zero mean and unit variance. This is equivalent to measuring the Euclidean distance between the predicted and normalized empirical saliency maps. The expected Euclidean distance to a random variable is minimized by its expectation value. Therefore, the optimal saliency map with respect to CC is the expected normalized empirical saliency map. It can be calculated as follow:

$$CC(S_p, S_{GT}) = \frac{\sigma(S_p, S_{GT})}{\sigma(S_p) \times \sigma(S_{GT})} \tag{2.3}$$

where $\sigma(S_p, S_{GT})$ is the covariance of S_p and S_{GT}. CC values close to 1 occur at locations where both the saliency map and ground truth saliency map have values of similar magnitudes. Also, CC is symmetric and penalizes false positives and negatives equally.

The Similarity metric (**SIM**, [Judd et al. 136]) normalizes the model saliency map and the empirical saliency map to be probability vectors (in the same way as the Kullback–Leibler divergence) and sums the pixel-wise minimum of two saliency maps, viewed as histograms. As opposed to the CC-metric, which can be interpreted as measuring the l_2-distance between normalized saliency maps. SIM effectively measures the l_1-distance between saliency maps as follows:

$$SIM(S_p, S_{GT}) = \sum_i \min(S_{GT_i}, S_{p_i}) = 1 - \frac{1}{2}||S_p - S_{GT}||_1 \tag{2.4}$$

A SIM value of one indicates that the distributions are the same, while zero means no overlap.

Each of these metrics offers a different perspective: KLD emphasizes prediction accuracy, CC focuses on linear correlation, and SIM on distribution overlap. This variety in evaluation approaches ensures a comprehensive assessment of how well saliency models replicate human visual attention in a probabilistic framework.

What metric to consider?

A recent study from [Kummerer et al. 154] argued that no model could win across all metrics and presented novel definitions and predictions for saliency models. A saliency model is defined as a predictor of the fixation probability density $p(x, y|I)$ given an image I. In contrast, using ground-truth data, a saliency metric is considered a performance measure for a saliency map. A saliency map, $S_{p,metric}(x, y, I)$, is derived from the model density and is specific to the metric used.

By formulating saliency models probabilistically, they are treated as posterior densities over all possible events. The saliency metric, on the other hand, is treated as a utility function. Given these premises, the saliency map is chosen to maximize the expected utility. Based on the fixation density predicted by a model, the authors propose using expected utility maximization to derive the saliency map expected to yield the highest performance in a particular metric. This process entails evaluating a saliency metric, which requires a saliency map $S(x, y|I)$ for a stimulus I and ground truth fixation data $\mathcal{M}_{GT} = (x_1, y_1), ..., (x_n, y_n)$. This can be expressed as $K[s(x, y|I); \mathcal{M}_{GT}]$ for a given metric K. The selection of the saliency map that is expected to achieve the highest performance for the metric involves solving the optimization problem:

$$\max_{S(x,y|I)} E_{\mathcal{M}_{GT}}[K[\mathcal{M}_{GT}, S(x, y \mid I)]] \tag{2.5}$$

where $E_{\mathcal{M}_{GT}}$ represents the expectation over the fixation distribution $\mathcal{M}_{GT}$. This optimization problem transforms $p(x, y|I) \rightarrow S_K(x, y|I)$ from fixation densities to derived metric-specific saliency maps.

In this work [Kummerer et al. 154], the authors then provide exact or approximate solutions for six widely used metrics, which include AUC, sAUC, NSS, IG, CC, SIM, KL-Div. For each metric, the authors derive the saliency map that should be expected to yield the highest performance. These deterministic transformations depend only on the model's density prediction. No optimization using ground truth data is

necessary. Thus, the results reported in this thesis follow the metrics formulation as presented by [Kummerer et al. 154].

2.2.2 Datasets

In the field of machine learning, datasets play a critical role and serve as the foundation for any computational model. These datasets consist of examples and their corresponding outcomes or labels, forming the basis for training, validating, and testing algorithms across various tasks. This principle holds across all downstream tasks, and it is particularly pertinent in the context of saliency prediction. In saliency prediction, the goal is to train models to accurately predict where humans are likely to look within a visual scene.

Eye-tracking datasets are essential in this domain, containing natural images used as visual stimuli and eye movement data collected through eye-tracking devices. However, what is crucial to note is the specific purpose or task assigned to viewers during data collection. For instance, viewers may be tasked with free-viewing, where they observe images without any specific instructions in a free-viewing scenario, or they may be given a particular visual search task, directing them to locate specific objects or features within the images, such as sports, etc. The nature of the task can significantly influence the patterns of eye movements observed. These datasets typically comprise hundreds or thousands of images, observed by numerous human subjects. As participants view the images, their eye positions are continuously monitored and recorded in the context of image coordinates.

Image datasets.

In Table 2.1, we provide an illustrative overview of sample datasets commonly used for the validation and training of saliency models. Several heuristic approaches were adopted in the early days of the saliency model development. These methods, as discussed in studies such as those by [Le Meur et al. 160, Navalpakkam and Itti 203, Kootstra et al. 145, Murray et al. 202, Bruce and Tsotsos 32, Garcia-Diaz et al.

91, Seo and Milanfar 246, Goferman et al. 96, Gao and Vasconcelos 90], relied on relatively small size datasets. Notable among these datasets were those presented by [Stark and Choi 254, Ouerhani et al. 211, Torralba 265]. These datasets served as the foundation for early heuristic models, allowing researchers to validate the results of this model's category and assess the performance.

As the field of saliency model development evolved and the potential of machine learning and deep learning became more apparent, a shift occurred towards data-driven approaches, as exemplified by [Borji 23]. In this context, "data-driven" refers to an approach where models are trained using large-scale datasets, allowing them to learn patterns directly from the data rather than relying solely on handcrafted features or rules. This contrasts earlier heuristic approaches based on manually designed features or rules. Data-driven approaches leverage the power of machine learning algorithms to automatically extract relevant features and relationships from the data, leading to more flexible and effective models.

The shift towards data-driven approaches brought with it a demand for larger datasets, offering greater complexity and diversity, that would be more suitable for training and fine-tuning Convolutional Neural Networks (CNNs) [Krizhevsky et al. 148]. The three primary datasets that emerged during this period are SALICON [Jiang et al. 130], MIT300 [Bylinskii et al. 34], and CAT2000 [Borji and Itti 25]. These datasets—SALICON, MIT300, and CAT2000—have become the foundation of contemporary deep learning-based saliency models due to their extensive scale and the wide variety of image types they contain. The comprehensive nature of these datasets has allowed researchers to delve deeper into the complexities of visual attention and has significantly contributed to advancements in saliency prediction.

Video datasets.

In Table 2.2, we present a detailed overview of the key datasets that have been utilized extensively in the validation and training of saliency models within the context of video content. These datasets have evolved over time, tracking the advancements in

Datasets	No. Images	No. Viewers	Year	Features
[Stark and Choi 254]	15	7	1996	Used a video camera to capture the stimulus size measuring 15 x 20cm. The images used consisted of terrain photographs, landscapes, and paintings.
[Ouerhani et al. 211]	7	30	2003	Eye position was tracked using a video-based system consisting of a headset with infrared eye cameras and a screen-monitoring head camera, enabling the computation of gaze position at 250 Hz with a gaze-position accuracy of 0.5°1.0° relative to the stimulus position, followed by parsing the eye tracking data in real-time for fixations and saccades, excluding microsaccades and short fixations.
[Torralba 265]	36	24	2006	The study involved a people search task with stimuli containing varying numbers of people, as well as painting and mug searches using the same set of images, with eye-tracking performed using a Generation 5.5 SAI Dual Purkinje Image Eyetracker, and mean target sizes specified for each category.
MIT [Judd et al. 137, Bylinskii et al. 34]	1003	15	2009	Images from the Flicker creative commons and LabelMe datasets. A one-second gray screen was displayed between each pair of images to ensure consistent viewing conditions. The initial fixation data was excluded from the analysis. The participants in the study were between the ages of 18 and 35.
[Ehinger et al. 82]	912	14	2009	This study explores the predictability and modeling of human search fixations, utilizing a dataset of 45,144 fixations recorded during the search of 912 real-world scenes, investigating the consistency of search behavior across individuals and the efficacy of computational models in guiding visual search.
CAT2000 [Borji and Itti 25]	4000	120	2009	The dataset consists of images from 20 categories, each containing stimuli with specific characteristics, such as action, affective, art, black & white, cartoon, fractal, indoor, inverted, jumbled, line drawings, low resolution, noisy, object, outdoor man-made, outdoor natural, pattern, random, satellite, sketch, and social, collected from various sources and databases, and eye tracking experiments were conducted with 120 observers using an infrared Eyelink eye-tracking device to record their eye movements while viewing the images.
SALICON [Jiang et al. 130]	20000	64	2015	A novel mouse-contingent paradigm was designed to compare mouse-tracking data with eye-tracking data, utilizing multi-resolution images generated in real-time. The experiment was conducted with 16 subjects in a controlled laboratory environment, followed by deploying the mouse-tracking experiments on the Amazon Mechanical Turk (AMT) crowd sourcing marketplace to collect large-scale attentional data.

Table 2.1: Saliency datasets for static images

research methodologies and the shift towards data-driven strategies. The early phase of research was marked by smaller datasets, which nevertheless provided a foundation for innovative techniques and heuristic approaches in saliency prediction, as reflected in the datasets by [Le Meur et al. 159] and [Itti and Baldi 122]. The dataset by [Le Meur et al. 159] utilized a dual-Purkinje eye tracker, a specialized eye-tracking device capable of precisely capturing eye movements, to generate 'priority maps' that indicated the visually significant regions in various video clips for a human subject. The work by [Itti and Baldi 122] focused on exploring the 'surprise hypothesis', suggesting that surprising elements within scenes capture attention.

The development of the UFC sports [Mathe and Sminchisescu 191] and the Hollywood-2 datasets [Mathe and Sminchisescu 191] marked a shift in the nature of the research, with larger and more complex datasets that incorporated a greater variety of content and viewer interactions. For instance, the UFC sports dataset divided

Datasets	No. Videos	No. Viewers	Year	Features
[Le Meur et al. 159]	7	25	2007	In this study, subjects' eye movements were tracked using a dual-Purkinje eye tracker to create 'priority maps' that revealed the visually important regions for an average observer while they watched a variety of video clips during a free-viewing task; this raw eye-tracking data was further processed and refined through several improvements to the model, which included building two chromatic saliency maps and implementing temporal saliency computation.
[Itti and Baldi 122]	50	8	2005	The study tested the surprise hypothesis, which posits that surprise attracts human attention in natural scenes, by recording eye movements of eight observers watching a variety of video clips, processing these clips through computational metrics to produce dynamic response maps, and subsequently analyzing these metrics by comparing them to randomly sampled data; this process indicated that human observers tend to non-uniformly gaze at select regions that elicit high metric responses.
UFC sports [Mathe and Sminchisescu 191]	150	19	2012	The videos cover nine sports actions were collected, with 16 volunteers split into an 'active' group, who identified actions in each video, and a 'free-viewing' group who watched without a specific task. Eye movements were recorded using a high-speed eye tracker under binocular viewing conditions, and stimuli were displayed in sessions to prevent fatigue, with data from any sessions with calibration errors discarded.
Hollywood-2 [Mathe and Sminchisescu 191]	1707	19	2012	It features 12 classes of actions from 69 Hollywood movies split into distinct training and test sets used to study the gaze patterns of humans performing a recognition task.
DHF1K [Wang et al. 277]	1000	17	2018	A large-scale dataset for studying gaze in free-viewing of videos, consisting of 1k diverse videos from YouTube, categorized into 150 classes, with eye-tracking annotations from 17 observers, which not only covers varied motion patterns and numerous objects but also illuminates scene and human factors, and is split into training, validation, and testing sets for public use and benchmarking.

Table 2.2: Saliency datasets for static videos

participants into active and free-viewing groups, and high-speed eye tracking under binocular viewing conditions was used to record eye movements. The Hollywood-2 dataset provided insight into gaze patterns during a recognition task using 12 classes of actions from 69 Hollywood movies.

The emergence of machine learning and deep learning led to the requirement for even larger and more diverse datasets that would enable comprehensive model training and fine-tuning. The DHF1K dataset [Wang et al. 277] is an excellent example of this shift, comprising 1000 diverse videos sourced from YouTube. It incorporates a wide variety of motion patterns, objects, scene factors, and human factors, making it ideal for studying gaze in the free-viewing of videos. This dataset serves as an essential tool for current and future developments in the field of video saliency prediction.

360° Image/Video datasets

The development and validation of saliency models for 360° images/videos require specialized datasets different from 2D data, as capturing and analyzing visual

attention in panoramic environments is inherently complex. Table 2.3 offers an overview of key datasets in this domain. Unlike traditional 2D imagery, gathering data for 360° environments often involves using head-mounted displays (HMDs) that can track both the head movement (HM) and the eye movement (EM) of subjects. This dual tracking is crucial for understanding how viewers interact with and process panoramic scenes, complicating data-gathering.

In Table 2.3, we provide a comprehensive overview of key datasets used in the validation and training of saliency models specifically for 360° images/videos. It can be seen that the size of these datasets is dramatically smaller compared to the ones in Table 2.1. This difference in size can be attributed to several factors inherent to the nature of 360° vs 2D imagery:

- **Data Collection Complexity:** Collecting 360° data is more challenging due to the necessity of specialized equipment such as head-mounted displays (HMDs). In contrast, 2D datasets can be compiled using simpler setups, including standard monitors and simpler eye-tracking systems.

- **Viewer Interaction:** In 360° environments, viewer interaction is more dynamic and complex. Participants can change their viewpoint by moving their heads, which requires simultaneous tracking of head and eye movements. This contrasts with 2D datasets, where viewer interaction is typically limited to eye movements on a static screen.

- **Data Processing and Analysis:** The analysis of 360° data is more complex due to the spherical nature of the images. This necessitates advanced algorithms for processing and interpreting head and eye movement data, which is not required in the case of 2D images.

- **Experimental Set-Up and Time:** Setting up experiments for 360° data collection is often more time-consuming and resource-intensive. Each session requires calibration and adjustment for individual users, whereas 2D data

collection can be more standardized and less personalized, even using only a mouse without any eye-tracking device.

- **Viewer Experience and Fatigue:** The experience of viewing 360° content can be more intense and potentially fatiguing for participants, leading to shorter sessions and, consequently, smaller datasets.

In short, it can be seen that this area is still limited by the hardware quality preventing large scale adoption of such a technology. However, despite these challenges, 360° datasets provide unique features that cannot be captured by traditional 2D datasets, particularly in the context of virtual reality and augmented reality applications.

Dataset	Modality	Dataset Size	Subjects	HM/EM	Summary
[De Abreu et al. 66]	Image	21 of 4096×2048 resolution	32	HM	The participants in the study were evenly divided into two groups. One group viewed the 360° images for 10 seconds, while the other group observed the same images for 20 seconds using the Oculus Rift DK2 head-mounted display (HMD).
[Gutiérrez et al. 104]	Image	98 of varying resolutions	63	HM/EM	Using the Oculus Rift DK2 head-mounted display (HMD) and the Sensomotoric Instruments (SMI) eye-tracker, each image was viewed by 40 subjects. The final maps were aggregated over viewers.
[Sitzmann et al. 250]	Image	22 of 4K resolution	169	HM/EM	The authors designed three distinct experiment conditions: the VR stand, the VR seated, and the desktop. For the desktop condition, data collection employed the Tobii EyeX eye-tracker, and participants interacted with mouse-controlled desktop panorama viewers displayed on a conventional 2D monitor. Each experimental setup allowed for diverse perspectives and interactions with the 360 content, enhancing the data collected for the research.
[David et al. 65]	Video	19 of 3840×1920 resolution	57	HM/EM	The experiment was conducted using an HTC VIVE VR headset equipped with an SMI eye-tracker. The gaze data was processed to create saliency maps and scanpaths for both head and eye movements and head-only movements.
[Xu et al. 286]	Video	208 of more than 4k	58	HM/EM	The experiment involved subjects wearing an HTC Vive head-mounted display seated on a swivel chair and freely turning around to explore the entire scene. The HM data, in terms of longitude and latitude, and eye fixation data were recorded.

Table 2.3: Saliency datasets for 360 images/videos

2.3 Static Saliency Models

This section provides a comprehensive review of approaches for still image saliency prediction.

2.3.1 Heuristic Approaches.

Saliency prediction for images has been widely studied during the last few decades. As pioneers, [Itti et al. 124] derived bottom-up visual saliency using center-surround differences across multi-scale image features. The conspicuity maps are obtained from linearly summed and normalized feature maps where three features are considered: color C, orientation l, and intensity I:

$$C_I = f_I, \quad C_C = \mathcal{N}(\sum_{l \in L_C} f_l), \quad C_O = \mathcal{N}(\sum_{l \in L_O} f_l). \tag{2.6}$$

where $\mathcal{N}(.)$ is the map normalisation operator. The final saliency map is the average of the three conspicuity maps: $S = \frac{1}{3} \sum_{k \in \{I,C,O\}} C_k$. A more advanced bottom-up saliency approach based on other HVS features, namely, contrast sensitivity functions, perceptual decomposition, visual masking, and center-surround interactions, was proposed in [Le Meur et al. 160]. Indeed, approaches from [Le Meur et al. 160, Navalpakkam and Itti 203, Kootstra et al. 145, Murray et al. 202, Bruce and Tsotsos 32, Garcia-Diaz et al. 91, Seo and Milanfar 246, Goferman et al. 96, Gao and Vasconcelos 90] are mostly cognitive-based models relying on computing multiple visual features such as color, edge, and orientation at multiple spatial scales to produce a saliency map. [Navalpakkam and Itti 203] proposed a model for optimizing object detection speed in visual search, modeled as a problem of maximizing the Signal-to-Noise Ratio (SNR). The SNR is defined as the ratio of expected salience (visual prominence) of the target over distractors:

$$SNR = \frac{E[ST(A)]}{E[SD(A)]} \tag{2.7}$$

where $E[ST(A)]$ and $E[SD(A)]$ denote the expected salience of the target and distractors, respectively. The expectation is calculated over various random variables, such as the visual features of the target/distractors, spatial configuration, and internal noise. Then, the salience for each feature dimension is calculated as a linear

combination of bottom-up saliences weighted by top-down influences:

$$S_j(x, y, A) = \sum (g_{ij} \cdot s_{ij}(x, y, A)) \tag{2.8}$$

here, $S_j(x, y, A)$ represents the overall perceived salience for a feature dimension j, computed at the location (x, y) in the search array A. Also, g_{ij} represents the weight assigned to the bottom-up salience $s_{ij}(x, y, A)$ for feature dimension j at location (x, y) in the search array A, influenced by top-down factors. These weights are determined based on the relevance or importance of each feature dimension in the search task and may be learned from data or specified based on prior knowledge. The overall saliency is obtained by integrating this information across all feature dimensions:

$$S(x, y, A) = \sum (g_j \cdot S_j(x, y, A)) \tag{2.9}$$

The expected saliency of the target and distractors can be computed in terms of their salience in each saliency map within the feature dimensions, leading to an expression for SNR in terms of these expected saliences. To maximize the SNR, optimal top-down weights (gains) are found by differentiating the SNR with respect to these weights and setting the derivatives to zero. This leads to the following equations for adjusting the weights that maximize the SNR:

$$\frac{\partial}{\partial g_{ij}} SNR = \frac{SNR_{ij}}{SNR} - \frac{1}{\alpha_{ij}} \tag{2.10}$$

$$\frac{\partial}{\partial g_j} SNR = \frac{SNR_j}{SNR} - \frac{1}{\alpha_j} \tag{2.11}$$

where α_{ij} and α_j are positive normalization terms. $\frac{\partial}{\partial g_{ij}} SNR$ and $\frac{\partial}{\partial g_j} SNR$ are the derivatives of the SNR with respect to the gains g_{ij} and g_j. They provide the rules for adjusting these gains to maximize the SNR.

[Kootstra et al. 145] investigated the role of symmetry in visual attention by

developing several symmetry saliency models and comparing them to human eye-tracking data. Three saliency models were created: the isotropic symmetry operator, the radial symmetry operator, and the color symmetry operator. The isotropic symmetry operator calculates the degree of symmetry at a given position (x, y) based on the gradients of the intensity in surrounding pixels. Each pixel pair contributes to the local symmetry according to:

$$c(i, j) = m_i \cdot m_j \cdot d(i, j, \sigma) \cdot p(i, j) \tag{2.12}$$

where m_i is the magnitude of the gradient at point i, $d(i, j, \sigma)$ is a Gaussian weighting function on the distance between the two pixels, and $p(i, j)$ is a symmetry measurement. The overall symmetry value is given by summing up the contributions of all pixel pairs within a certain radius. The radial symmetry operator promotes the contributions of pixel pairs with dissimilar orientations, allowing for the detection of symmetrical patterns with multiple axes of symmetry:

$$M_p^{rad} = \sum_{(i,j)\in\Gamma(p)} c(i, j) \cdot \sin^2(\phi_p - \phi(i, j)) \tag{2.13}$$

the color symmetry operator is an adaptation of the isotropic symmetry operator, considering gradients of pixels not only in one color channel but also between different channels:

$$M_{x,y}^{col} = \sum_{(i,j)\in\Gamma(x,y)} \sum_{k,k'\in K} c(i, j, k, k') \tag{2.14}$$

Each operator was expanded into a multi-scale symmetry saliency model to accommodate the fact that the human visual system processes visual information on multiple spatial scales. The symmetry maps were normalized and then combined into a symmetry saliency map.

[Murray et al. 202] extended the color induction model from [Otazu et al. 210] to estimate visual saliency. Initially, the wavelet transform [Pathak 219] is used to estimate local oriented contrast C_{ij} at each wavelet coefficient:

$$C_{ij} = \sqrt{\sum_{\theta} W_{ij}^{\theta 2}} \tag{2.15}$$

where W_{ij}^{θ} represents the wavelet coefficients. Contrast energy E_{ij} in the neighborhood of each coefficient is calculated by summing the squares of coefficients in a window w around the location (i, j):

$$E_{ij} = \sum_{m,n \in w} C_{mn}^2 \tag{2.16}$$

The center-surround effect is modeled by calculating the ratio R_{ij} of energies of central E_{Cij} and surround regions E_{Sij}:

$$R_{ij} = \frac{E_{Cij}}{E_{Sij}} \tag{2.17}$$

The extended contrast sensitivity function (ECSF) is used to weight the center-surround contrast energy, as shown in Equation 2.17:

$$S_{ij} = R_{ij} \cdot ECSF(\omega_{ij}, R_{ij}) \tag{2.18}$$

where ω_{ij} is the spatial frequency and S_{ij} is the final response. The perceived image channel is obtained by performing an inverse wavelet transform on the wavelet coefficients, now weighted by the ECSF response. The final saliency map is computed from the perceived image channels. Parameters in the intensity and color functions of ECSF are optimized using psychophysical data. The overall process results in an image saliency estimation from a generalized low-level visual representation, which is combined with the saliency maps for each channel into a final saliency map.

The authors of [Bruce and Tsotsos 32] argued that the primate visual system is optimized for sparse representation of image statistics, supporting independent neural firing assumptions. In their work, saliency is determined by evaluating the self-information of each local image patch, using a representation based on independent components. In this context, self-information for each local image patch refers to

quantifying the unexpectedness or uniqueness of each patch's content relative to the underlying statistical model used for representation. Independent component analysis (ICA) [Lee and Lee 166] is performed on a large sample of 7×7 RGB patches from natural images to establish a suitable basis. For a given image, the probability of observing the RGB values of a patch is evaluated by considering the likelihood of each corresponding basis coefficient independently.

$$P(W_1 = V_1, W_2 = V_2, \ldots, W_n = V_n) = \prod_{i=1}^{n} P(W_i = V_i) \tag{2.19}$$

A Gaussian kernel density estimate is employed to estimate the likelihood of the components of W.

$$p(W_{i,j,k} = V_{i,j,k}) \approx \frac{1}{L} \sum_{s,t \in \Omega} w(s,t) \exp\left(-\frac{(V_{i,j,k} - V_{s,t})^2}{2\sigma^2}\right) \tag{2.20}$$

The saliency map is then convolved with a Gaussian to account for the peripheral vision drop-off. This approach distinguishes between two key concepts: self-information and entropy. Self-information, denoted as $-\log(p(X))$, where $p(X)$ is the probability of observing a specific vector of RGB values, measures the surprise or uncertainty associated with observing that particular event. On the other hand, entropy describes the extent to which a feature is uniformly distributed across the dataset.

[Garcia-Diaz et al. 91] employ Gabor-like filters to decompose the image, utilizing color and local energy as two feature dimensions. The local energy is calculated by applying a bank of log-Gabor filters to the luminance component, represented as:

$$\text{logGabor}(\rho, \alpha; \rho_i, \alpha_i) = e^{-\frac{(log(\rho/\rho_i))^2}{2(log(\sigma\rho_i/\rho_i))^2}} e^{-\frac{(\alpha-\alpha_i)^2}{2\sigma\alpha^2}} \tag{2.21}$$

The color information is obtained through a multiscale decomposition of the opponent color components (a, b) using a bank of log Gaussian filters. The distinctiveness of local responses is measured by decorrelating the multiscale responses via Principal Component Analysis (PCA) [Abdi and Williams 1], and computing the Hotelling's T2 statistic [Hotelling 115], which quantifies the statistical distance of each pixel

from the global average in the decorrelated space.

$$T_i^2 = (\mathbf{x}_i - \bar{\mathbf{x}})' W^{-1} (\mathbf{x}_i - \bar{\mathbf{x}}) \tag{2.22}$$

where $\mathbf{x}_i$ is the feature vector associated with each pixel, $\bar{\mathbf{x}}$ is the global average feature vector, and W is the covariance matrix. Finally, the model integrates these measures into a single saliency map with an overall computational complexity of $O(Nlog(N))$, where N is the number of pixels.

The algorithm proposed by [Goferman et al. 96] for image saliency incorporates color, pattern, and positional information. First, single-scale saliency is computed, defined as the mean dissimilarity to the K most similar patches, denoted as $S_i^r = 1 - \exp\left(-\frac{1}{K}\sum_{k=1}^{K} d(p_i^r, q_k^r)\right)$. This is then extended to multiple scales, and the saliency at a pixel is computed as the mean across different scales, denoted as $\bar{S}i = \frac{1}{M}\sum r \in RS_i^r$. High-level factors such as face detection are incorporated for final saliency map enhancement.

The discriminant saliency model [Gao and Vasconcelos 90], which defines salient attributes as those that best distinguish a visual concept from others. It requires a set of visual classes and aims to maximize discrimination between the class of interest and all other classes. Feature selection techniques, such as maximizing marginal diversity are used to design optimal saliency detectors. Biologically plausible models based on multiresolution image decomposition and non-linearity are employed to achieve saliency detection. The experiments utilize discrete cosine transform (DCT) features, which have proven effective in large-scale recognition problems.

The authors of [Seo and Milanfar 245] proposed to use local regression kernels as features that capture the underlying local structure of the images exceedingly well, even in the presence of significant distortions. Then, they used a nonparametric kernel density estimation for such features, which resulted in a saliency map consisting of a local "self-resemblance" measure, indicating the likelihood of saliency. To formalize saliency at each pixel, they let the binary random variable y_i denote whether a pixel position $x_i = [x_1, x_2]_i^T$ is salient or not as follows:

$$y_i = \begin{cases} 1, & \text{if } x_i \text{ is salient} \\ 0, & \text{otherwise} \end{cases}$$

where $i = 1, \ldots, M$, and M is the total number of pixels in the entire image. The final saliency map is defined at pixel position x_i as a posterior probability $P(y_i = 1|F)$ as follows:

$$S_i = P(y_i = 1|F)$$

where the feature matrix $F_i = [f_{1i}, \ldots, f_{Li}]$ at the pixel of interest x_i contains a set of feature vectors (f_i) in a local neighborhood, where L is the number of features in that neighborhood. In turn, the larger collection of features $F = [F_1, \ldots, F_N]$ is a matrix containing features from the center and a surrounding region. N is the number of feature matrices in the center+surround region. Using Bayes' theorem:

$$S_i = P(y_i = 1|F) = \frac{p(F|y_i = 1)P(y_i = 1)}{p(F)}$$

[Schauerte and Stiefelhagen 241] devised a quaternion-based spectral saliency prediction model. Quaternions are a 4D non-commutative algebra H over the real numbers, defined as $q = a + bi + cj + dk \in H$ where $a, b, c, d \in \mathbb{R}$. The model maps an image with at most four color components to a quaternion matrix, $IQ = I_4 + I_1 i + I_2 j + I_3 k$, which may also take into account the relative importance of color components via a quaternion component weight vector $w = [w_1 w_2 w_3 w_4]^T$. The model also introduces Fourier and Cosine transforms with quaternions. The Eigenaxis and Eigenangle spectral saliency are computed using spectral residuals with log amplitude and low-pass filtering, leading to the Eigen Spectral Residual (EigenSR) saliency map. This model allows for the consideration of multiple scales by combining the spectral saliency of the image at different scales.

This quaternion-based model is particularly adept at handling images with up to four color components (e.g., red, green, blue, and alpha for transparency) by mapping these components to a quaternion matrix. The matrix is denoted as

$IQ = I_4 + I_1 i + I_2 j + I_3 k$, where I_1, I_2, I_3, and I_4 represent the intensity values of each color component. The model also considers the relative importance of each color component using a weight vector $w = [w_1 w_2 w_3 w_4]^T$. Additionally, the model incorporates mathematical techniques such as Fourier and Cosine transforms but extends them into the quaternion domain. These transformations help calculate what is known as the 'Eigenaxis' and 'Eigenangle,' which are used to determine spectral saliency. Spectral saliency is computed using a method called 'spectral residuals', which involves modifying the log amplitude spectrum of an image and then applying a low-pass filter. The result is the 'Eigen Spectral Residual (EigenSR) saliency map', which highlights the most visually striking parts of the image. Furthermore, this model is designed to evaluate saliency at multiple scales, enhancing its ability to detect salient features that appear at different sizes within the image. This multi-scale analysis is crucial for applications where understanding the focus of visual attention is necessary.

The Dynamic Visual Attention (DVA) [Hou and Zhang 117] is a computational approach that simulates attention mechanisms in the primary visual cortex using Sparse Feature Representation and Incremental Coding Length. This model considers each image patch as a linear combination of sparse coding basis functions (features). The activity of each feature is evaluated through its incremental Coding Length, defined as the entropy gain of the ensemble during that feature's activity increment. From a dataset of 120,000 8×8 RGB image patches derived from natural scenes, a set of 192 basis functions (the sparse basis A) was extracted. The filter functions $W = A^{-1}$ are then computed, providing the sparse representation of an image patch $s = Wx$, where x is the vectorized image. The activity ratio for the i^{th} feature is given by $p_i = \frac{\sum_k |w_i x_k|}{\sum_i \sum_k |w_i x_k|}$. The ICL for each feature is calculated as $ICL(p_i) = -H(p) - p_i \log p_i$, where $H(p)$ is the entropy of the probability distribution of feature activities. The model identifies a set of salient features $S = \{i | ICL(p_i) > 0\}$ and redistributes energy according to feature saliency, with $d_i = \frac{ICL(p_i)}{\sum_{j \in S} ICL(p_j)}$ for $i \in S$. The resulting saliency map M is computed as $m_k = \sum_{i \in S} d_i w_i x_k$. The saliency of

an image patch is context-dependent, determined by the distribution of p, which is obtained by sampling the environment over space and time, thereby maximizing the efficiency of information transmission.

BMS [Zhang and Sclaroff 292] is a bottom-up saliency model based on the Boolean Map Theory of visual attention. The saliency of an image I is modeled as the mean attention map $\bar{A}$ over all randomly generated Boolean maps B:

The Boolean Map Saliency (BMS) model, as proposed by [Zhang and Sclaroff 292], is a bottom-up saliency approach based on the Boolean Map Theory of visual attention. The saliency of an image I is modeled as the mean attention map $\bar{A}$ over all randomly generated Boolean maps B:

$$\bar{A} = \int A(B)p(B|I)dB \quad (1) \tag{2.23}$$

where boolean maps are generated by randomly thresholding the input image's feature maps. If we denote a feature map of I as $\phi(I)$, and the threshold as θ, each Boolean map Bi can be generated as:

$$B_i = \text{THRESH}(\phi(I), \theta), \quad \phi \sim p_\phi, \quad \theta \sim p_\theta \quad (2) \tag{2.24}$$

The attention maps $A(B)$ are computed for each Boolean map B_i based on a Gestalt principle of figure-ground segregation. Each attention map is then normalized to create the mean attention map $\bar{A}$, which is post-processed to produce the final saliency map S.

The Graph-Based Visual Saliency (GBVS) model introduced by [Harel et al. 107], is designed to identify significant, informative locations in an image I through the computation of feature maps (s1), activation (s2), and normalization and combination (s3) steps. An activation map $A : [n]^2 \to \mathbb{R}$ is computed from a feature map $M : [n]^2 \to \mathbb{R}$, where locations in I that are unusual in their neighborhood correspond to high activation values. The authors propose a dissimilarity metric $d((i,j),(p,q))$ based on the logarithmic distance between $M(i,j)$ and $M(p,q)$. Using this metric,

the model constructs a fully connected directed graph GA with nodes corresponding to the lattice of M and directed edge weights $w1((i,j),(p,q))$ proportional to their dissimilarity and closeness in the domain of M. The equilibrium distribution of a Markov chain on this graph is used as an activation measure derived from pairwise contrast, reflecting regions of high dissimilarity. The "normalization" step is stated as a goal of concentrating mass on activation maps. A similar Markovian approach is used, constructing a graph GN with nodes labeled with indices from $[n]^2$, and each node is connected to every other node. The weight $w2((i,j),(p,q))$ is proportional to the activation of (p,q) and the closeness of (i,j) to (p,q). The equilibrium distribution of a Markov chain on GN allows mass to flow preferentially to nodes with high activation, resulting in a mass concentration algorithm by construction.

Other approaches have incorporated Bayesian models, built on top of cognitive models, to introduce a layer of prior knowledge such as scene context or gist. These models employ a probabilistic method, specifically Bayes' rule, for integrating various factors. For example, the work of [Torralba 265] presents a model that integrates global scene attributes into the attentional decision process, suggesting that understanding the overall scene helps predict where viewers direct their attention. Similarly, [Oliva et al. 207] explored how top-down control in object detection is guided by an understanding of the scene structure, enhancing visual search efficiency. [Zhang et al. 294] introduced the SUN framework, which uses natural scene statistics to inform a Bayesian model of visual saliency, thereby adapting more flexibly to varied visual environments. [Itti and Baldi 122] proposed a model of 'Bayesian surprise' that quantifies how data affects an observer's beliefs, effectively predicting attentional deployment by measuring the degree of surprise experienced by the observer. These Bayesian approaches unify many aspects of visual processing by systematically incorporating prior knowledge and probabilistic reasoning to enhance the prediction and interpretation of attentional focus. Integrating such comprehensive models underscores the complexity of attentional mechanisms, illustrating a sophisticated

$[n]^2$ denotes the set of all pairs of integers from 1 to n, representing the pixel coordinates of an n x n image.

Table 2.4: Comparative performance study on MIT300 and CAT2000.

Models	MIT300					CAT2000				
	s-AUC	NSS	CC	KLD	SIM	s-AUC	NSS	CC	KLD	SIM
ITTI [Itti et al. 124]	0.53	0.40	0.13	1.49	0.33	0.59	1.19	0.43	1.02	0.45
SUN [Zhang et al. 294]	0.62	0.76	0.27	1.28	0.39	0.58	0.81	0.29	2.33	0.39
SSR [Seo and Milanfar 245]	0.64	0.81	0.29	1.52	0.41	0.59	0.85	0.30	1.74	0.41
QSS [Schauerte and Stiefelhagen 241]	0.66	0.91	0.33	1.14	0.42	0.61	0.83	0.30	1.21	0.40
[Hou et al. 116]	0.66	0.99	0.37	1.08	0.42	0.60	1.08	0.38	1.10	0.42
DVA [Hou and Zhang 117]	0.65	1.01	0.37	1.11	0.45	**0.62**	1.42	0.50	**0.99**	**0.49**
[Goferman et al. 95]	0.64	1.01	0.38	1.07	0.43	0.59	1.09	0.39	1.10	0.44
AIM [Bruce and Tsotsos 31]	0.66	0.88	0.34	1.24	0.40	0.60	0.96	0.34	1.37	0.40
RARE [Riche et al. 233]	0.67	1.15	0.42	1.00	0.45	0.61	1.09	0.39	1.14	0.43
BMS [Zhang and Sclaroff 292]	**0.69**	1.15	0.41	1.02	0.44	**0.62**	1.18	0.41	1.20	0.46
GBVS [Harel et al. 107]	0.62	**1.24**	**0.47**	**0.88**	**0.48**	0.58	**1.23**	**0.44**	1.01	0.45

merger of cognitive theory with Bayesian inference.

In Table 2.4, we present the performance outcomes of heuristic approaches on two benchmark datasets: MIT300 [Bylinskii et al. 34] and CAT2000 [Borji and Itti 25]. Notably, the Graph-Based Visual Saliency (GBVS) model [Harel et al. 107] emerges as the most effective model within this category. However, the Dynamic Visual Attention (DVA) model [Hou and Zhang 117] exhibits marginally superior performance on the CAT2000 test set.

2.3.2 Data-Driven Approaches

Data-driven approaches, known as pattern classification models, fundamentally involve learning from recorded eye-fixations or annotated salient maps. [Kienzle et al. 140] pioneered a non-parametric, bottom-up method to infer saliency directly from human eye fixation data. This model employs a Support Vector Machine (SVM) that determines saliency-based solely on local intensity values, marking the first such approach to eschew any prior assumptions about Human Visual System (HVS) features traditionally used to encode saliency. Similarly, [Judd et al. 137] developed a method that trains a linear SVM on a dataset comprising 1003 labeled images, utilizing an array of low, mid, and high-level image features.

In more recent developments, deep learning-based static saliency models, such as those proposed in these works [Huang et al. 121, Kümmerer et al. 151, Liu et al. 176, Kruthiventi et al. 150, Cornia et al. 57, Pan et al. 213, Pan et al. 212], have

demonstrated significant enhancements. These advances have been driven by the robust capabilities of deep neural networks and the availability of extensive datasets for static scenes, such as those detailed in the MIT Saliency Benchmark [Bylinskii et al. 35], SALICON [Jiang et al. 130], and CAT2000 by [Borji and Itti 25]. Integrating these large-scale datasets has been instrumental in refining and enhancing the accuracy of saliency prediction models.

The authors of [Pan et al. 213] and [Kruthiventi et al. 150] were the first to use CNNs for saliency prediction when introducing eDN and DeepFix, respectively. DeepFix initialized the first 5 convolution blocks with VGG-16 weights, and then two novel Location-based Convolutional (LBC) layers were added to capture semantics at multiple scales. Pan et al. [Pan et al. 212] used Generative Adversarial Networks (GANs) [Goodfellow et al. 97] to build the SalGAN model. The network consists of a generator model whose weights are learned by back-propagation computed from a binary cross entropy (BCE) loss over existing saliency maps. The resulting prediction is processed by a discriminator network trained to solve a binary classification task between the saliency maps generated at the generative stage and the ground truth ones in a min-max game, using the following adversarial loss:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{BCE}} + L(D(q, p_{model}), 1). \tag{2.25}$$

The objective is to optimize $L(D(I, S), 1)$, where $D(I, S)$ is the probability of fooling the discriminator network by producing saliency maps that look very similar to the Ground truth coming from the data distribution.

[Kummerer et al. 155] introduced two models, one using high-level features based on object recognition (DeepGaze II) and another using low-level features based on image contrast, to predict human fixations on images. The study finds that a significant portion of fixations are better explained by the low-level model, suggesting that contrast can be as vital as object presence for predicting where people look in images. Then, authors of [Linardos et al. 172] systematically assess the impact of various ImageNet backbones on spatial saliency prediction, initially used

in DeepGaze II architecture with a VGG19 backbone. They found that replacing VGG19 with a ResNet50 backbone improves performance from 78% to 85%, but further replacements with more recent models yield no additional gains. Their analysis reveals a substantial variation in generalization to unseen datasets, with overconfidence observed in fixation predictions. By integrating multiple backbones, they develop DeepGaze IIE, which exhibits a significant leap in performance, reaching SoTA scores on the MIT1003 benchmark [Kummerer et al. 154].

[Ding et al. 71] proposed SalFBNet, a feedback convolutional network for saliency prediction, consisting of feed-forward feature learning, feature feedback, and saliency feature decoding. Feedback connections enable contextual information aggregation. A fixation-based loss, sFNE, is introduced for better training; it computes the squared Euclidean distance between the two saliency distributions. It encourages the model to focus more on accurately predicting saliency at the regions where fixations occur. Pre-trained models are used to transfer saliency knowledge through pseudo-saliency labels, improving model performance. [Jia and Bruce 128] presents an Efficient Multi-Level Network (EML-NET) utilizing deep CNN models like DenseNet-161 [Huang et al. 119] and NasNet-Large [Zoph et al. 303] for feature extraction. The encoding stage compresses predictions into one feature map to save memory. The decoding stage aligns and concatenates features from multiple layers efficiently. The saliency metrics, Kullback-Leibler Divergence (KLD), modified Pearson's Correlation Coefficient (CC), and modified Normalized Scanpath Saliency (NSS), are combined to form the loss function for training. The modified metrics improve the network's performance and are used for training, while the original metrics are used for benchmark evaluation. [Huang et al. 121] proposed SALICON, the model optimizes an objective function based on the saliency evaluation metrics from two parallel streams at different image scales. [Liu and Han 175] trained the deep spatial contextual long-term recurrent convolutional network (DSCLRCN), incorporating global spatial interconnections and scene context modulation.

It can be seen that these approaches rely mainly on the power of data and have

Table 2.5: Comparative performance study on MIT300 and CAT2000.

Models	MIT300					CAT2000				
	s-AUC	NSS	CC	KLD	SIM	s-AUC	NSS	CC	KLD	SIM
DeepGaze IIE [Linardos et al. 172]	0.79	2.53	0.82	0.35	0.70	0.67	2.11	0.82	0.34	0.71
DeepGaze II [Kruthiventi et al. 150]	0.78	2.34	0.77	0.42	0.66	0.65	1.96	0.80	0.38	0.69
SALICON [Thomas 262]	0.74	1.70	0.56	0.78	0.52	0.65	1.68	0.64	0.80	0.62
SalGAN [Pan et al. 212]	0.74	1.86	0.67	0.76	0.59	0.64	1.46	0.57	0.94	0.54
Unisal [Droste et al. 79]	0.78	2.37	0.79	0.41	0.67	–	–	–	–	–
ML-Net [Cornia et al. 57]	0.74	1.97	0.66	0.80	0.58	0.63	1.33	0.50	0.94	0.53
SalFBNet [Ding et al. 71]	0.79	2.47	0.81	0.42	0.69	–	–	–	–	–
EML-NET [Jia and Bruce 128]	0.75	2.49	0.79	0.84	0.68	–	–	–	–	–
MSI-Net [Kroner et al. 149]	0.78	2.31	0.78	0.42	0.67	–	–	–	–	–
TranSalNet [Jianxun et al. 131]	0.75	2.41	0.81	1.01	0.69	–	–	–	–	–
GazeGAN [Che et al. 44]	0.73	2.21	0.76	1.34	0.65	–	–	–	–	–
CASNet II [Fan et al. 87]	0.74	1.99	0.71	0.59	0.58	–	–	–	–	–
SAM-VGG [Cornia et al. 58]	0.73	1.96	0.66	1.27	0.60	–	–	–	–	–
DVA [Wang and Shen 276]	0.73	1.93	0.66	0.63	0.58	–	–	–	–	–
ICF [Kummerer et al. 155]	0.70	1.61	0.59	0.71	0.56	–	–	–	–	–
eDN [Vig et al. 271]	0.62	1.14	0.45	1.14	0.41	–	–	–	–	–

less mathematical depth than classical methods. However, as shown in Table 2.5, deep models achieve results closer to the human baseline on the SALICON [Jiang et al. 130], MIT300 [Bylinskii et al. 34], and CAT2000 [Borji and Itti 25] datasets.

2.4 Dynamic Saliency Models

Models for dynamic scenes need to take into account egocentric movements or the motion of the scene being observed, as well as accounting for the short amount of time available to view a video frame ($\sim 1/30$ sec) compared to static images (typically 5 seconds), that affect the saliency distribution per frame. Dynamic saliency is usually formulated as bottom-up feature extraction with an ad-hoc temporal domain adaptation from previous time points [Borji and Itti 27]. This, in addition to the need for high computational and memory resources, means that video saliency prediction is a more challenging research question than static saliency prediction [Borji 23]. In contrast to static stimuli, few hand-crafted dynamic approaches have been proposed [Boccignone 19, Marat et al. 188, Oliva et al. 207, Le Meur et al. 159, Garcia-Diaz et al. 92, Zhang and Sclaroff 292, Guo et al. 102, Rudoy et al. 237]. Boccignone [Boccignone 19] proposed a non-parametric Bayesian technique, namely variational inference on a mixture of Dirichlet processes. Based on HVS biology, [Marat et al. 188] extracted

two signals from a video corresponding to the outputs of the retina, parvocellular and magnocellular, to predict spatiotemporal saliency. [Le Meur et al. 159] extended [Le Meur et al. 160] to the spatio-temporal domain by combining chromatic, achromatic, and temporal priors. These methods all apply some linear/nonlinear algorithms to fuse spatial and temporal features. However, such a straightforward approach to combining spatial and temporal information often forces the model to lose the intrinsic correlation between these two complementary aspects.

Deep Saliency Models. More recent works have demonstrated the potential of learning saliency from large eye-fixation data distributions [Bak et al. 11, Jiang et al. 129, Wang et al. 277, Adel Bargal et al. 3, Lai et al. 156, Droste et al. 79, Linardos et al. 173, Min and Corso 195, DAHOU DJILALI et al. 62]. These are mainly based on two distinct network modules to deal with spatial and temporal fields separately. These works exhibit strong performance and show the potential of using neural networks for the video saliency problem. Bak et al. [Bak et al. 11] were the first to leverage deep learning when they used a two-stream CNN architecture for video saliency prediction. Video frames and motion maps were fed to the two streams. [Wang et al. 277] proposed a CNN-LSTM network architecture with an attention mechanism to explicitly encode static saliency information, thus allowing the LSTM to focus on learning a more flexible temporal saliency representation across successive frames. [Panagiotis et al. 214] introduced a temporal regularization for their previous model SalGAN [Pan et al. 212] for static saliency prediction. In terms of architecture, they added a convolutional LSTM layer on top of the frame-based saliency prediction to adapt it for dynamic saliency prediction. SalEMA [Linardos et al. 173] added a conceptually simple exponential moving average of an internal convolutional state to the SalGAN network [Pan et al. 212], arguing that saliency tends to be relatively consistent across successive frames, arguing that the use of ConvLSTM is not necessary. [Lai et al. 156] proposed a composite attention mechanism that learns multi-scale local and global features priors end-to-end, designed in a two-stream approach with models learning appearance and motion features separately. UNISAL [Droste et al. 79]

proposed four novel domain adaptation techniques to enable strong shared features: Domain-Adaptive Priors, Domain-Adaptive Fusion, Domain-Adaptive Smoothing, and Bypass-RNN. UNISAL achieved the state-of-the-art results on the DHF1K benchmark [Wang et al. 277] (e.g. AUC-J: 0.901, NSS:2.776).

2.5 Saliency Prediction for 360° Video/Image

360° scenes, referred to as panoramic, spherical, or omnidirectional images/videos (ODI/ODV), is a new type of multimedia that provides the user with an immersive experience. The content of panoramic scenes is rendered to cover the entire 360 × 180 viewing space. Humans, however, naturally focus on the most attractive and interesting field-of-views (FoV) while ignoring others in their visual field. Despite significant advances in the field of saliency modeling on a fixed 2D viewport (see [Itti and Koch 123, Borji 23] for a comprehensive review), saliency prediction methods on 360° video/image are still in their infancy. A recent survey conducted by Xu et al. [Xu et al. 285] reviewed many works for predicting the Head and eye movements saliency maps of 360° images and video that model the probability distribution of viewports/regions of interest (RoIs) of multiple subjects. Rai et al. [Rai et al. 230] derived the volcano-like distribution of EM within the viewport. Results show that eye-gaze fixations are quasi-isotropically distributed in orientation, typically far away from the center of the viewport. Furthermore, ODV presents some statistical biases, as investigated in [Xu et al. 286]. Human attention on 360° video is biased toward the equator and the front region, known as the equator bias, which could be leveraged as priors in modeling. Along with the statistical bias, a subject's attention is driven by the most salient objects in the scene.

This section reviews important works related to attention modeling for 2D 360° video/image. These are primarily concerned with predicting the HM/EM saliency maps in 360° images/video and can be grouped into heuristic and data-driven approaches.

2.5.1 360° heuristic approaches

The heuristic approaches encode saliency on the 360° sphere using handcrafted features. The pioneer works by [Bogdanova et al. 22, Bogdanova et al. 21] generate the spherical static saliency map by combining chromatic, intensity, and three cue conspicuity maps after normalization through multiscale analysis on the sphere. They build the motion pyramid on the sphere by applying block matching and varying the block size. Finally, the two maps are fused to produce the dynamic saliency map on the sphere. Unfortunately, no quantitative results were provided in [Bogdanova et al. 22, Bogdanova et al. 21] since the HMD was unavailable back then. RM3 [Battisti et al. 13] for 360° images, combines low-level features hue, saturation, and GBVS features, with the high-level features present in each viewport. Unlike [Bogdanova et al. 22, Bogdanova et al. 21, Battisti et al. 13], Fang et al. [Fang et al. 88] proposed the extraction of low-level features of color, texture luminance, and boundary connectivity from the super-pixels at multiple levels segmented from an equirectangular projection (ERP) image.

Other authors have adapted existing 2D saliency models into 360° video/images. This approach, however, suffers from geometric distortion and border artifacts. [De Abreu et al. 66] introduced a fused saliency map (FSM) approach to HM saliency prediction on 360° images; they adapted SALICON [Huang et al. 121] (a 2D image saliency prediction model) to 360° images using ERP. [Lebreton and Raake 162] extended the 2D Boolean Map Saliency (BMS [Zhang and Sclaroff 292]) and Graph-Based Visual Saliency (GBVS [Harel et al. 108]) models to integrate the properties for equirectangular images, naming their approaches BMS360 and GBVS360 respectively. [Maugey et al. 192] applied a 2D saliency model on each face generated under the cubemap projection (CMP). Recently, it has become easier to collect HM and EM data, and thus, many end-to-end saliency prediction approaches for 360° video/images have emerged.

2.5.2 360° Deep learning approaches

Trained on recently published datasets [Xu et al. 286, Xu et al. 287, David et al. 65, Zhang et al. 296], several DNN-based saliency prediction approaches for 360° video have been proposed [Zhang et al. 296, Xu et al. 286, Cheng et al. 50, Suzuki and Yamanaka 258, Lebreton et al. 161, Nguyen et al. 204, Zhang and Chen 293, Hu et al. 118]. [Nguyen et al. 204] fine-tuned the PanoSalNet 2D static model on 360° video datasets to predict HM saliency map of each frame without considering the temporal dimension. The predicted saliency is enhanced by a prior filter based on an apriori statistical bias. [Cheng et al. 50] proposed a DNN-based spatial-temporal network consisting of a static model and a ConvLSTM module to adjust the outputs of the static model based on temporal features. They also used a "Cube Padding" technique in the convolution, pooling, and convolutional LSTM layers to keep connectivity between the cube faces by propagating the shared information across the views. [Lebreton et al. 161] extended BMS360 [Lebreton and Raake 162] to V-BMS360 by adding a temporal bias and optical flow-based motion features. [Hu et al. 118] proposed a deep learning-based agent for automatic piloting through 360° sports videos. At each frame, the agent observes a panoramic image and has the knowledge of previously selected viewing angles. The developed online policy allows a recurrent neural network to shift the current viewing angle to the next. [Chao et al. 43] finetuned SalGAN [Pan et al. 212] on the Salient360! image dataset with a new loss function combining three saliency metrics. . [Qiao et al. 224] proposed a Multi-Task Deep Neural Network (MT-DNN) model for head movement prediction; the center of each viewport is spatio-temporally aligned with eight shared convolution layers to predict saliency features.

Unlike all previous approaches, [Zhang et al. 296] proposed a spherical convolutional neural network, spherical U-NET, trained following the teacher forcing technique, for applying a planar saliency CNN to a 3D geometry space, where the kernel is defined on a spherical crown, and the convolution involves the rotation of the kernel along the sphere. The model input includes one frame and the predicted

saliency maps of several previous frames to allow for improved modeling of dynamic saliency. The authors also defined a spherical MSE (S-MSE) loss function for training the spherical U-Net to reduce the non-uniform sampling of the ERP. Furthermore, instead of using supervised learning to learn saliency from data, [Xu et al. 286] applied deep reinforcement learning (DRL) to predict HM positions by maximizing the reward of imitating human HM scanpaths through the agent's actions. The reward, which measures the similarity between the predicted and ground-truth HM scanpaths, is estimated to evaluate the action made by the DRL model. The reward is then used to make a decision on the action through the DRL model, i.e., the HM scanpath in the current frame.

2.6 Visual attention from a neuroscience perspective

Visual attention was extensively studied within neuroscience, with implications for how we model human visual behavior in computational systems. In this section, we take a different perspective on the saliency modeling problem, particularly focusing on the implications for dataset and ground truth design and whether the common procedure of creating saliency datasets, as discussed earlier, aligns with human perception. Do these ground truth saliency maps capture the intricacies of human perception and attentional engagement, as revealed by neuroscientific studies? This is the question we study here.

The recent study by [Friedman and Robbins 89] explores the prefrontal cortex (PFC) and its central role in cognitive control and executive functions, which are integral to managing attention and behavior in response to complex environments. Their research delves into the neural mechanisms through which the PFC modulates attentional processes, emphasizing its ability to maintain and adjust activity patterns representing goals and the methods for achieving them. This modulation is crucial for prioritizing cognitive resources and focusing attention in ways aligned with these goals,

a dynamically adjusted process based on feedback from other brain regions. Their findings suggest that the PFC's architectural diversity and extensive connectivity allow it to integrate a wide range of sensory inputs and internal states, making it uniquely suited to support complex behaviors, including decision-making, problem-solving, and planning. By managing the interactions between attentional focus and goal-directed behavior, the PFC plays a pivotal role in filtering out irrelevant stimuli and enhancing the salience of pertinent aspects of the environment. This aspect of the PFC's function can be paralleled in computational models of visual saliency, where the challenge is to predict what elements of visual scenes will draw attention. Understanding the PFC's role in these processes provides valuable insights for designing algorithms that more accurately mimic human attentional behaviors, potentially leading to more effective and adaptive saliency modeling techniques.

The paper by [Chun and Turk-Browne 53] explores the dynamic interplay between attention and memory, detailing how memories and prior experiences influence attentional processes. This research is particularly relevant to understanding how individuals' attention is selectively drawn to aspects of their environment based on what they remember or find familiar, which can vary significantly across different contexts and individuals. The insights from this study are valuable for computational models of visual attention or saliency. They suggest that adaptive models, which can incorporate elements of past interactions or learned preferences, may more accurately reflect human-like attention patterns. This could lead to advancements in how saliency models predict areas of interest in visual scenes, making them more personalized and context-aware.

The study by [Wang et al. 275] investigates the relationship between visual salience and visual importance through psychophysical experiments and computational analysis. Their research addresses whether saliency maps, as defined in the previous sections, align with maps of visual importance, which reflect subjective human ratings concerning the significance of various objects in the scene. In their experiments, human subjects rated each object within images on a scale reflect-

ing relative importance to create the importance maps. Then, their experiments also involved tracking participants' gaze patterns across these same images without specific tasks, to create saliency maps. Their findings suggest that while saliency and importance are correlated, they are not as closely linked as might be assumed. Saliency maps accurately identified primary subjects within the images but were less successful in recognizing objects of secondary importance (Figure.1 in [Wang et al. 275]). On the same line, the findings of [Elazary and Itti 83] suggest a significant correlation between the predicted saliency from a computational model and the objects labeled by humans as interesting. Specifically, on the used data, they report that in 43% of images, the most salient location identified by the model coincided with a labeled object, significantly higher than the chance level of 21%. Additionally, the top three salient locations included a labeled object in 76% of cases.

Their analysis revealed that early gaze positions predominantly fixate on main subjects, suggesting a potential strategy within the visual system to locate key elements of a scene quickly. This observation provides insight into how initial visual processing might prioritize significant components of an image, which could influence both natural perception and computational models aiming to mimic this aspect of human vision. This study underscores the complexity of visual attention and highlights the need to consider both salience and importance when studying visual perception, which not reflected in the commonly used datasets.

On the point of recording conditions and setups, the work of [Engelke et al. 86] investigates the consistency of fixation density maps generated from eye-tracking data across different independent laboratories. The main focus is on how variations in experimental setup affect the generated ground-truth maps. They revealed that despite different experimental conditions across three labs, the obtained fixation maps exhibited high similarity. Their research demonstrates that time and image content significantly influence the similarity of results, with longer viewing times and specific content types leading to more consistent realizations across different settings.

To discuss the idea of perceived interest and overt visual attention in natural

images, [Engelke and Le Callet 85] explored the relationship between these two facets of visual perception through some experiments. They conducted a region-of-interest (ROI) selection experiment alongside an eye gaze tracking experiment to analyze how perceived interest (determined through region-of-interest selections by participants) aligns with where people actually look (as tracked through eye movements) in various natural images.

Their results revealed a significant correlation between perceived interest and overt visual attention, indicating that the areas people find interesting are generally the same areas they tend to look at. This relationship was found to vary depending on the content of the images and the duration for which the images were presented during the eye-tracking experiment. Particularly, it was noted that eye-tracking data could predict the primary ROI with considerable accuracy, especially when the ROI was clearly dominant over other parts of the image.

The work [Le Callet and Niebur 157] discusses the analogies between overt and covert attention mechanisms, highlighting their implications for multimedia applications. Overt attention is measurable through eye-tracking techniques and involves directing the gaze toward significant elements within a scene, as commonly replicated in the existing saliency datasets. Conversely, covert attention does not involve moving the eyes but focuses on processing important peripheral details, which is crucial for applications requiring situational awareness, such as driving assistance systems. [Le Callet and Niebur 157] also discusses integrating visual attention models into multimedia technologies, proposing that such integration can significantly optimize content delivery. For instance, understanding attention distribution across a scene can inform adaptive compression algorithms, enhance the rendering quality of images and videos, and personalize user interfaces to align with individual viewing behaviors.

On the temporal dynamics of attention, [Wieser and Keil 280] investigates this aspect by measuring event-related potentials and compares these with behavioral data to examine the correlation between psychophysical and physiological measurements of

attention. This study emphasizes the need to consider temporal characteristics when designing datasets for saliency models, highlighting that brain activity in response to stimuli is delayed compared to initial stimulus presentation.

Additionally, the work of [Wieser and Keil 280] on sustained attention in the visual cortex explores how dynamics in the visual cortex can predict target detection performance during distraction, demonstrating the significant role of attentional shifts over time. This study provides empirical evidence on how visual attention fluctuates and impacts perception, suggesting that ground truth in saliency datasets should reflect these dynamic aspects of attention.

The integration of sensory modalities could significantly enhance human perceptual processing, a topic extensively explored in neuroscience and psychology. For instance, [Maravita et al. 189] research on multisensory integration highlights how tactile and auditory cues can influence spatial awareness and body schema, emphasizing the importance of these cues in enhancing perceptual accuracy and speed of processing in various tasks. This finding is essential for understanding how individuals navigate and interact in environments with multiple sensory inputs, suggesting implications for the design of virtual and augmented reality systems where sensory congruence can lead to more immersive experiences.

Furthering this exploration, [Calvert 38, Kaya and Elhilali 138] provide insights into how auditory cues can direct visual attention in dynamic scenes. It discusses the neural adaptations that occur when attention shifts between sensory modalities, noting changes in brain connectivity that enhance cognitive flexibility and multisensory integration. This study also demonstrates that auditory cues can significantly guide visual attention in complex scenes, indicating that sounds can alert viewers to important visual events, enhancing the detection and interpretation of relevant visual information. These studies underscore the potential of auditory cues to modify viewer engagement and attention in multimedia applications, thereby influencing how saliency is perceived and modeled in environments where multiple sensory modalities are engaged.

In summary, the neuroscience perspective provides invaluable insights into the complexities of human visual attention, challenging and guiding the development of computational saliency models that aim to emulate the intricate mechanisms of human perception. It is worth noting that the current datasets do not reflect these aspects and are mostly built on raw fixation recordings.

2.7 Conclusion

Saliency prediction is an active research area that has attracted the interest of researchers from different fields over the last two decades. Early models hand-engineered visual features such as color, intensity, etc., to build methods capable of producing a saliency map from an input image. Deep learning-based models have shifted the paradigm to learning from human fixations and have shown impressive performances. Additionally, the recently introduced omnidirectional images and videos present a challenge, as geometric considerations constrain them. Saliency modeling in this field requires significant improvements to meet the required performance standards. Furthermore, benchmarking saliency models remains an open research question due to inconsistent rankings when varying the metrics. Thus, the consensus among the research community is that the metrics measure different things [Riche et al. 232, Bylinskii et al. 37] and that it is conceptually impossible to determine the best-performing model without considering the different metrics. Moving forward, it will be crucial to integrate insights from neuroscience to develop more sophisticated models that mimic and deeply understand human visual and cognitive processes, potentially impacting how visual attention is quantified and predicted in complex, dynamic environments.

Part I —

Learning with 2D data

MOTIVATION AND SUMMARY

Transformers, initially developed for natural language processing tasks, have been successfully adapted for various applications in computer vision. These adaptations, known as Vision Transformers (ViTs), leverage the transformer architecture to address tasks ranging from simple image classification to complex semantic segmentation. This part of the thesis explores the application of Vision Transformers to build models for predicting visual saliency on 2D images.

In Chapter 3, we investigate how off-the-shelf Vision Transformers align with the human visual system. This involves studying their self-attention maps and designing techniques to adapt them into saliency maps for various images.

Then, in Chapter 4, we propose Saliency Transformers (SalTR), which focuses on parallel decoding in Transformers to learn saliency directly from fixation maps. This approach addresses the challenge of optimizing discrete fixation maps. Unique fixation prediction is enforced by a global loss mechanism using bipartite matching combined with a transformer encoder-decoder architecture.

Chapter 3

Vision Transformers for image saliency prediction

3.1 Introduction

The last decade has witnessed remarkable progress in saliency prediction, with many methods achieving impressive performances on recently introduced benchmarks, particularly those based on deep learning, which have significantly boosted performance. Indeed, training a Convolutional Neural Network's (CNN) auto-encoder has become the de facto approach for visual attention modeling, as discussed in Chapter 2. The success of deep learning in visual attention modeling has primarily relied on CNNs [Huang et al. 121, Pan et al. 212, Borji 24]. With its inherent spatial equivariance, the convolution operation has enabled impressive results on visual attention datasets. Notably, recent studies have advanced the frontiers in many critical computer vision tasks, such as classification [Dosovitskiy et al. 78], object detection [Carion et al. 39], and semantic segmentation [Strudel et al. 256], by leveraging Transformer neural networks [Vaswani et al. 269] at a large scale in both model and dataset sizes.

Interestingly, the attention block, universal in nature, is shared among Vision

This chapter is based on our BMVC 2023 [Djilali et al. 74]

Transformers (ViT), language [Devlin et al. 69, Brown et al. 30], audio [Baevski et al. 9, Radford et al. 226], and many other modalities [Zhao et al. 297]. Unlike convolution, the self-attention mechanism treats the input sequences as a fully connected dense graph and parses information across nodes. Hence, symmetries and structures are learned from the data rather than being explicitly incorporated using image-specific inductive biases [Cordonnier et al. 56, Khan et al. 139, Raghu et al. 227]. Furthermore, hybrid ViT-CNN models have become an active area of research [d'Ascoli et al. 81, Yuan et al. 290, Liu et al. 178, Carion et al. 39]. This raises two important questions: Are Vision Transformers inherently learning saliency maps? Are the self-attention maps focusing on the salient regions of the input image? Analyzing the self-attention maps of pretrained ViTs on saliency prediction datasets, we find that smoothing the internal attention maps with a few convolutional filters can produce reasonable saliency maps with acceptable metric scores. We explore how this phenomenon arises, finding that self-attention promotes the early aggregation of global information. It associates highly attended features in higher layers, compares their dependencies, and draws analogies over recurring patterns. This suggests that ViTs initially perform a feature search, followed by a conjunction search that combines multiple features sharing higher mutual information.

This inductive-bias-free nature of transformers raises several fundamental questions about their learning paradigm and its relationship to human visual attention: 1) How does a Vision Transformer's learning paradigm relate to the feature integration theory [Treisman and Gelade 266] that guides human visual attention? 2) Do Vision Transformers inherently assign high attention mass to salient areas? 3) If so, are there methods to extract a saliency map from these self-attention maps?

The focus of this chapter is to explore and analyze the analogies between self-attention maps and saliency maps and to investigate ways in which Vision Transformers can be utilized as off-the-shelf saliency models. It is clear from the above introduction that deep learning-based approaches targeting image visual attention modeling have little in common with classical methods. These classical methods

often rely on predefined features and simpler statistical models that closely align with early cognitive theories, such as Gestalt laws. In contrast, deep learning methods leverage complex network architectures that learn features directly from data without predefined assumptions, focusing more on predictive accuracy and less on mimicking human cognitive processes. As a result, these methods have significantly diverged from the initial cognitive motivations aimed at modeling human visual processing in a more interpretable and biologically inspired manner.

On the learning side, [Tuli et al. 268] proposed the Class-wise Jensen-Shannon (JS) distance to produce an error-consistency metric very close to Cohen's κ [Geirhos et al. 93]. This metric was used to assess whether CNNs or ViTs correlate more with human vision from an error consistency point of view. They concluded that ViTs exhibit a higher shape bias and are largely more consistent with human errors. The study by [Zhou et al. 298] found that, while both ViTs and CNNs demonstrate functional parallels with the brain's ventral stream—particularly in how they process visual information hierarchically—neither model paradigm is optimal for replicating the human visual system entirely. These parallels are often revealed through layer-wise activation patterns in the models that correspond to progressive stages of visual processing observed in the ventral stream, from basic feature detection to complex object recognition. This chapter investigates whether ViTs are inherent saliency learners by exploring their analogies with cognitive-based saliency models. We also examine the effects of supervised versus self-supervised ViTs on the saliency task. Specifically, our contributions are:

- By analyzing how local and global spatial information is utilized, we demonstrate that the penultimate self-attention maps of ViTs are similar to saliency maps.

- Furthermore, we discover that retrieving salient attention maps using a tensor decomposition method yields the most informative self-attention maps tensor, which can be easily mapped to a saliency map using a shallow convolutional neural network decoder.

- We explore the main differences between ViTs trained on supervised classification tasks and those trained under a self-supervised regime. Supervised training involves using labeled data where each image is associated with a specific class label, teaching the model to focus and recognize features directly related to those labels. This often leads to models that concentrate on the most salient objects in the scene, which are typically central to classifying the image. In contrast, self-supervised training does not rely on explicit labels but instead uses the input data as the label, typically through tasks like predicting one part of the image from another or reconstructing the image from corrupted inputs. This type of training encourages the model to understand broader and more general features of the scene, leading to better generalization on saliency tasks where understanding the overall context and subtler cues is crucial. Our findings suggest that ViTs under a self-supervised regime can better capture a wide range of salient features across various contexts compared to their supervised counterparts, which often focus on the most visually prominent features linked directly to specific class labels.

3.2 Background

The field of deep learning has rapidly evolved in terms of architectural design, giving rise to specialized architectures tailored for specific tasks across various domains such as computer vision [Ren et al. 231, Chen et al. 46], natural language processing (NLP) [Mikolov et al. 193, Chung et al. 54], speech recognition [Povey et al. 222], translation [Bahdanau et al. 10, Cho et al. 51], and reinforcement learning (RL) [Mnih et al. 200, Schulman et al. 243]. Each domain presented its own set of challenges and unique data characteristics, which led to the development of neural network architectures specifically designed for those areas. For instance, convolutional neural networks (CNNs) [Krizhevsky et al. 148] became the standard architecture for computer vision tasks, while recurrent neural networks (RNNs) [Rumelhart et al. 238], including Long Short-Term Memory (LSTM) networks [Hochreiter and Schmidhuber 114],

predominantly served the NLP field.

Initially, the landscape of deep learning research was characterized by isolated communities focused primarily on their specific domains, with limited cross-domain interactions concerning deep learning architectures. However, as research progressed at an exponential pace, it became increasingly difficult to stay up-to-date on developments across different fields. This fragmentation hindered knowledge transfer and led to missed opportunities for leveraging insights and experiences from diverse communities. Nevertheless, the introduction of the Transformer architecture [Vaswani et al. 269] marked a significant shift. Although it took some time for the transformer to gain widespread adoption, it has become a pivotal element in machine learning research. In this section, we explore the details of Transformers and their adaptations to computer vision (Vision Transformers [Dosovitskiy et al. 78]), providing a comprehensive overview of their architecture and training details.

3.2.1 Transformers

The Transformer architecture built on the work by [Bahdanau et al. 10], in the context of translation, aimed to improve approaches based on Recurrent Neural Networks (RNNs). The innovative idea by cites bahdanau2014neural was to have a model with two "cursors," one moving through the source sequence (encoded by a Bidirectional Recurrent Neural Network (BiRNN) [Schuster and Paliwal 244]) and another moving through the target sequence. A BiRNN enhances the traditional RNN by processing data in both forward and reverse directions, allowing it to capture context from past and future states effectively. This dual-direction processing is particularly beneficial in tasks like translation, where understanding the entire input sequence is crucial for accurate output.

The cursor trajectories in this model would be marginalized out using dynamic programming [Bellman 15]. The first attempt by [Bahdanau et al. 10] implemented two cursors moving synchronously at the same time (effectively hard-coded diagonal attention). The authors proposed teaching the RNN decoder to search where to place

the cursor in the source sequence, a concept inspired by translation exercises typically used in language learning, where one's gaze shifts back and forth between the source and target texts during translation. Consequently, the soft search was expressed as softmax, followed by the weighted averaging of BiRNN states. This methodology was initially published under the **RNNSearch** name, and later, **Yoshua Bengio** renamed the approach to (attention).

The 'Attention is All You Need' paper [Vaswani et al. 269] argued that this tiny module of attention was primarily responsible for the performance improvements observed in the Bahdanau et al. model. Therefore, they discarded everything else, retaining only the attention module. This attention-only model allowed for processing inputs based on sets of tokens rather than strict sequences, enabling it to focus on relevant parts of the input with varying degrees of emphasis. This architecture has proved remarkably versatile and successfully adapted across various domains.

Definition of attention: Following the terminology used by [Graves et al. 99], attention involves the process of averaging **values** that correspond to **keys** matching a given **query**. The intuition behind the attention mechanism is similar to focusing one's mental resources on specific parts of information while ignoring others, analogous to how human attention works. In this context, the **query** represents the current item or area of focus, the **keys** represent different pieces of information available from the input, and the **values** are the actual data points that are relevant to the keys. Matching the query to the keys determines which values are most relevant, which are then averaged to produce the output. Consequently, the keys used for computing attention and the values subjected to averaging are distinct tensors. This selective focus allows neural networks to prioritize certain inputs over others dynamically, enhancing the model's ability to handle large and complex datasets by focusing on the most informative parts of the data.

Given a query sequence $Q \in \mathbb{R}^{T \times D}$, a key sequence $K \in \mathbb{R}^{T' \times D}$, where D is the feature dimension, and a value sequence $V \in \mathbb{R}^{T' \times D'}$, compute an attention matrix $A \in \mathbb{R}^{T \times T'}$ by matching Q's to K's, and weight V with it to get the result sequence

$Y \in \mathbb{R}^{T \times D'}$.

$$\forall i, \quad A_i = \text{softmax}\left(\frac{KQ_i}{\sqrt{D}}\right), \quad Y_i = V^\top A_i, \tag{3.1}$$

The queries and keys have the same dimension D, and there are as many keys T' as there are values. The result Y has as many rows T as there are queries of the same dimension D' as the values.

Encoder: The input tokens, denoted as X, are initially embedded and then sent to an attention mechanism, usually multi-headed. To avoid the over-smoothing bottleneck, the residual connections connect the layers such as: for an input $X = (x_1, \ldots, x_n)$ where each $x_i \in \mathbb{R}^d$:

$$x_i = x_i + \sum_{j=1}^{n} K_{ij} W^V x_j \tag{3.2}$$

Here, $K = \text{softmax}(A)$ and $A_{ij} = (W^Q x_i)^T (W^K x_j)$. After the self-attention mechanism, a layer normalization is applied, which balances and adjusts each dimension. This step is critical due to the potential imbalance caused by the attention mechanism, which might prioritize only a few spots. The normalized outputs at each position are then passed to an MLP consisting of a linear layer, followed by a skip connection and another normalization layer. The full encoder comprises N stacks of the described block. This produces an encoded output $Hx \in \mathbb{R}^{T_x \times D}$, with T_x indicating the number of input tokens and D specifying the attention vector dimensionality. **Decoder:** After encoding the input, the decoder generates the output by applying a cross-attention. Initially, the decoder's attention block processes all previously generated tokens, $y_{1:t-1}$, and computes an encoding $Hy \in \mathbb{R}^{T_y \times D}$. This specific block employs masked attention to ensure that the output at time t can only attend to locations in Y preceding t. The cross-attention block gives attention to the input's encoding, Hx, and the outputs generated earlier, Hy. A joint representation, denoted as Z, is formed by

$$Z = \text{Attn}(Q = Hy, K = Hx, V = Hx) \tag{3.3}$$

This evaluates how the output aligns with the input. This combined encoding, Z, then undergoes processing by an MLP layer. The entire decoder follows this block structure N times. Upon completing the decoder's operations, the output is transformed into a sequence of T_y output logits via a final linear layer. **The mystery of positional encoding:** Because there are no recurrent connections, the transformer has no way to know where in the sequence each token has occurred (i.e., the transformer is permutation invariant unless additional information is provided). The authors proposed a Sinosoial positional encoding, where each token in a position j is the sines and cosines with a period T as follows: **The mystery of positional encoding:** In the Transformer architecture, there are no recurrent connections, which traditionally help sequential models like Recurrent Neural Networks (RNNs) to recognize the order of input data. Consequently, the transformer is inherently permutation invariant, meaning it lacks any mechanism to inherently distinguish the order of tokens in the sequence unless additional information is provided. To address this, the authors of the Transformer model introduced a method called *sinusoidal positional encoding*. This technique involves assigning each position in the sequence a unique value representing its location. Positional encoding is implemented by applying sine and cosine functions of different frequencies to each position j of the sequence. Specifically, each position j is encoded with a set of sine and cosine values, where each value corresponds to a sinusoidal function with a different period T. Mathematically, this can be expressed as:

$$PE(pos, j) = \begin{cases} \sin\left(\frac{pos}{10000^{\frac{j}{d_{\text{model}}}}}\right) & \text{if } j \text{ is even} \\ \cos\left(\frac{pos}{10000^{\frac{j-1}{d_{\text{model}}}}}\right) & \text{if } j \text{ is odd} \end{cases} \tag{3.4}$$

where:

- pos is the position in the sequence.

- d_{model} is the dimension of the model.
- j is the dimension within d_{model}.

This ensures that each token in the input sequence is supplemented with a distinctive set of values that reflects its position in the sequence, thus allowing the model to maintain an awareness of the order of the sequence elements throughout the processing. These positional encodings are then added to the input embeddings, providing the necessary context that allows the transformer to consider the order of the tokens as it processes and learns from the data. This approach is crucial for language understanding and translation tasks, where the sequence order significantly impacts meaning.

Where did this come from? It may initially seem puzzling how a combination of sine and cosine functions could represent the position or order of elements in a sequence. However, the concept becomes apparent when we consider the representation of numbers in a binary system. Just as in binary encoding, where each bit changes at a different rate (the least significant bit changes with every increment, and the next bit changes every two increments), sinusoidal functions exhibit a similar pattern of variation. Imagine representing a position using binary digits, as shown in Figure 3.1. One would notice that the rate of change in binary digits forms a pattern: the least significant bit alternates with every increment in position, the second least significant bit changes every two increments, etc. The idea in Figure 3.4 can be seen as a continuous analog of this pattern. By employing sinusoidal functions—sine for one dimension and cosine for the next—each position j in the sequence is encoded uniquely. These functions are selected because their periodic nature allows them to simulate the binary pattern of change smoothly and continuously. The use of $10000^{2i/d_{model}}$ in Equation 3.4 scales the wavelengths of the sinusoids across dimensions, ensuring that each dimension of the positional encoding correlates to a sinusoid with a different frequency. This method, akin to how bits represent numbers in binary, allows each token to carry a unique signature of its position, enabling the model to maintain positional awareness and effectively handle sequences.

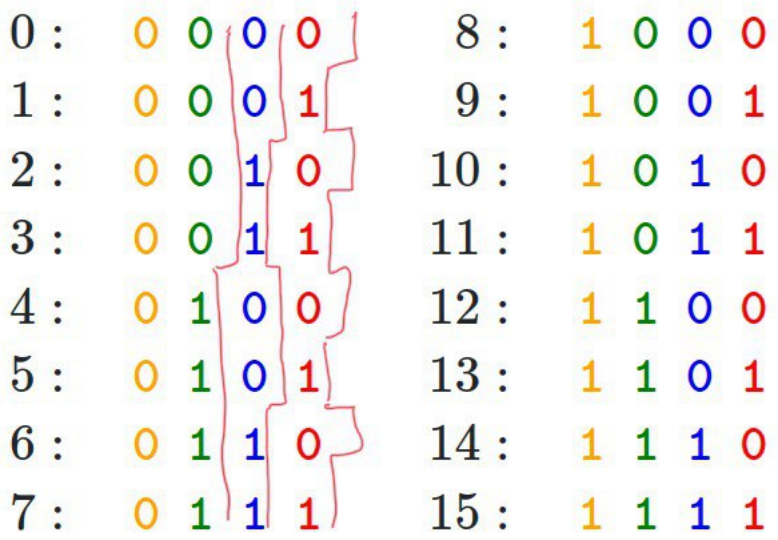


Figure 3.1: **Binary analog of sinusoidal position encoding used in transformers.** We can identify waves of high/low frequencies by looking at the least significant bit.

Self-Attention vs. Cross-Attention: Self-attention is employed within the transformer's encoder. The keys, values, and queries are derived from the same input sequence. This mechanism enables the encoder to capture the internal structure of the input sequence, integrating information across all positions and establishing a comprehensive representation. On the other hand, Cross-attention, utilized in the transformer's decoder, is distinct in that the queries come from the previous decoder layer. At the same time, the keys and values originate from the output of the encoder. This architecture allows each element in the decoder to attend to the entire output of the encoder. In the case of machine translation, cross-attention enables each word in the target sequence to focus on the relevant words in the source sequence, facilitating translation by considering the full context of what has been encoded.

3.2.2 (Vision) Transformers (ViTs)

The Vision Transformer (ViT) represented a shift in the field of computer vision [Dosovitskiy et al. 78] as the main competitor to the dominant Convolutional Neural Networks (CNNs). In adapting the Transformer [Vaswani et al. 269] for vision tasks and to overcome the dimension bottleneck, an image is first segmented into a series of

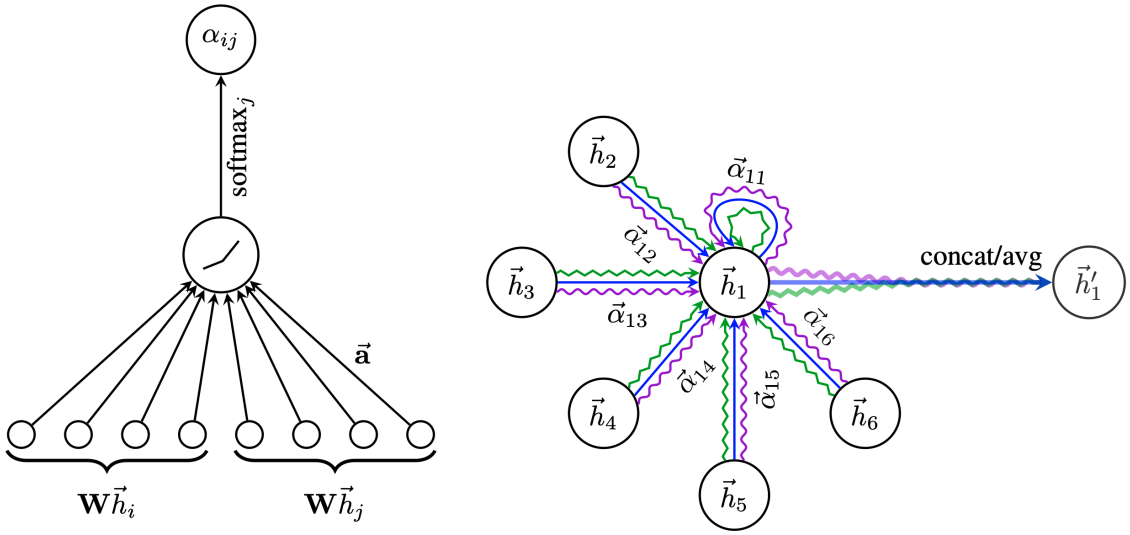
patches. These patches are akin to the soft tokens in NLP and are linearly embedded into vectors. These embeddings, representing the patches, are then processed through the Transformer encoder, much like words in a text sequence, following the BERT design [Devlin et al. 69].

Convolutional Neural Networks (CNNs) are architected with a strong inductive bias toward recognizing spatial hierarchies in images. This bias is an intrinsic characteristic of CNNs, stemming from their convolutional layers that employ localized receptive fields through filters. These filters adeptly capture local features such as edges and textures at initial layers and progressively combine them into more complex patterns in the deeper layers. This localized processing approach, when paired with the down-sampling effect of pooling layers, leads to two prominent biases: translation invariance—meaning the ability to recognize patterns regardless of their position in the visual field—and the establishment of a spatial hierarchy that abstracts image details into higher-level features.

These inherent biases of CNNs enable them to achieve remarkable performance on various vision tasks with comparatively fewer training samples relative to models like Vision Transformers (ViTs). The efficiency arises because the architecture is predisposed to exploit the spatial structure of image data, hence requiring less external information to form accurate representations. Nonetheless, this same strong inductive bias can also be a limitation in scenarios with abundant data. Since CNNs are designed to recognize and generalize using patterns from fewer examples, adding more data does not proportionally enhance their performance. Beyond a certain point, the model's performance plateaus; the network does not extract new insights because its architecture restricts it to the hierarchical spatial features it has already learned to identify. In contrast, models with weaker inductive biases, like ViTs, may continue to scale and improve as they consume more data, potentially capturing more complex patterns and relationships that CNNs may not inherently leverage due to their design.

By relying on self-attention mechanisms, Vision Transformers (ViTs) process an

Figure 3.2: **Left:** The attention mechanism $a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$ employed by our model, parametrized by a weight vector $\vec{a} \in \mathbb{R}^{2F'}$, applying a LeakyReLU activation. **Right:** An illustration of multi-head attention (with $K = 3$ heads) by node 1 on its neighborhood. Different arrow styles and colors denote independent attention computations. The aggregated features from each head are concatenated or averaged to obtain $\vec{h}'_1$.



entire image as a sequence of flattened patches (i.e., tokens). Unlike Convolutional Neural Networks (CNNs), ViTs do not inherently impose a spatial hierarchy or locality in processing the image data. This design choice does not encode translation invariance or local receptive fields into the architecture. Instead, ViTs learn to dynamically focus on different parts of the image, irrespective of spatial proximity, by considering the relationships between all pairs of patches in the sequence. This enables ViTs to learn global and long-range dependencies within the image. Intuitively, this allows the model to scale their understanding of the data with increased training samples, as they are not limited to the local patterns learned through the local filters of CNNs. Empirical evidence supporting this can be found in studies that show ViTs achieving improved performance on image recognition tasks as the dataset size increases [Dosovitskiy et al. 78]. Such studies have demonstrated that ViTs can outperform CNNs, especially in data-rich environments, indicating their superior capacity to leverage larger datasets to capture complex patterns across the image.

3.2.3 How ViTs really work?

The best way to think of Vision Transformers is as a message-passing procedure over a set of nodes (the patch of an image). The input to the layers is a set of node features, $\mathbf{h} = \{\vec{h}_1, \vec{h}_2, \ldots, \vec{h}_N\}$, $\vec{h}_i \in \mathbb{R}^D$, where N is the number of patches per image, and D is the number of features in each node. The layer produces a new set of node/patch features given the information aggregated from the neighbors. In this setting, it's a fully connected graph, meaning each patch is connected to all the other patches in the image. In order to obtain sufficient expressive power to transform the input features into higher-level features, at least one learnable linear transformation is required. To that end, as an initial step, a shared linear transformation, parametrized by a weight matrix, $W \in \mathbb{R}^{D \times D}$, is applied to every patch. We then perform self-attention on the nodes—a shared attentional mechanism $a : \mathbb{R}^{D \times D} \to \mathbb{R}$ computes attention coefficients:

$$e_{ij} = a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j) \quad (1) \tag{3.5}$$

This equation indicates the importance of patch j's features to node i. In its most general formulation, the model allows every patch to attend to every other patch, dropping all structural information. The image structure is injected into the mechanism by performing positional encoding on the input patches. Then, to make coefficients easily comparable across different patches, they are normalized using the softmax function:

$$\alpha_{ij} = \text{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in N_i} \exp(e_{ik})} \tag{3.6}$$

Indeed, the attention mechanism a is a single-layer feedforward neural network, parametrized by a weight vector $\vec{a} \in \mathbb{R}^{2F'}$, and applying the Gaussian Error Linear Unit (GELU [Hendrycks and Gimpel 112]) nonlinearity. Fully expanded out, the coefficients computed by the attention mechanism (illustrated by Figure 3.2 (left))

may then be expressed as:

$$\exp\left(\text{LeakyReLU}\left(\vec{a}^T\left[\mathbf{W}\vec{h}_i \,||\, \mathbf{W}\vec{h}_j\right]\right)\right) \tag{3.7}$$

where ^{T} represents transposition and $||$ is the concatenation operation. Once these coefficients are obtained, they are used to compute a linear combination of the features corresponding to them to serve as the final output features for every patch (after potentially applying a nonlinearity, σ):

$$\vec{h}'_i = \sigma\left(\sum_{j\in N_i} \alpha_{ij}\mathbf{W}\vec{h}_j\right). \tag{3.8}$$

To stabilize the learning process of self-attention, Vision Transformers also employ multi-head attention, similarly to [Vaswani et al. 269]. Specifically, K independent attention mechanisms execute the transformation of Equation 3.8, and then their features are concatenated, resulting in the following output feature representation:

$$\vec{h}'_i = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j\in N_i} \alpha_k\mathbf{W}_k\vec{h}_j\right) \tag{3.9}$$

where $\|$ represents concatenation, α_k are normalized attention coefficients computed by the k-th attention mechanism (a_k), and $\mathbf{W}_k$ is the corresponding input linear transformation's weight matrix. Note that, in this setting, the final returned output, $\vec{h}'$, will consist of KD features (rather than D) for each patch. The output features are:

$$\vec{h}' = \frac{1}{K}\sum_{k=1}^{K} \sigma\left(\sum_{j\in N_i} \alpha_k\mathbf{W}_k\vec{h}_j\right). \quad (6) \tag{3.10}$$

The aggregation process of a multi-head graph attentional layer is illustrated by Figure 3.2 (right), where multiple arrows come off a node.

3.2.4 Connections between Vision Transformers and CNNs

The work of [Raghu et al. 227] studied whether ViTs and CNNs see the world similarly. For this, they used the Centered Kernel Alignment (CKA), which has shown its relevance in understanding and comparing the information encoded by different neural network layers [Kornblith et al. 146]. Formally, CKA relies on two sets of data $\mathbf{X} \in \mathbb{R}^{p \times N}$ and $\mathbf{Y} \in \mathbb{R}^{q \times N}$ through their corresponding kernels $\mathbf{K} = k(\mathbf{X}^\top, \mathbf{X}) \in \mathbb{R}^{N \times N}$ and $\mathbf{L} = \ell(\mathbf{Y}^\top, \mathbf{Y}) \in \mathbb{R}^{N \times N}$ where k, ℓ are some kernel functions applied on the columns of $\mathbf{X}$ and $\mathbf{Y}$ respectively (e.g., linear or RBF kernels). Therefore, the CKA is computed in terms of $\mathbf{K}$ and $\mathbf{L}$ as:

$$\mathrm{CKA}(\mathbf{K}, \mathbf{L}) = \frac{\mathrm{HSIC}(\mathbf{K}, \mathbf{L})}{\sqrt{\mathrm{HSIC}(\mathbf{K}, \mathbf{K})\,\mathrm{HSIC}(\mathbf{L}, \mathbf{L})}}, \tag{3.11}$$

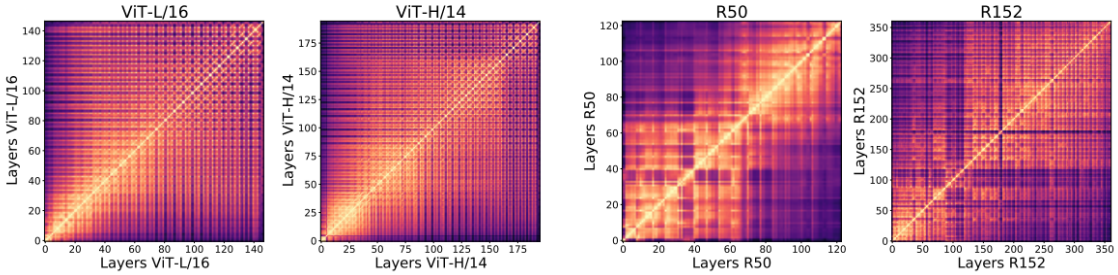
where $\mathrm{HSIC}(\cdot, \cdot)$ is the Hilbert-Schmidt Independence Criterion [Gretton et al. 100, Ma et al. 186] defined as:

$$\mathrm{HSIC}(\mathbf{K}, \mathbf{L}) = \frac{1}{(N-1)^2}\,\mathrm{tr}\left(\mathbf{KCLC}\right), \tag{3.12}$$

with $\mathbf{C} = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^\top$ the centring matrix. We refer the reader to [Kornblith et al. 146] for broader properties and studies of the CKA metric on neural network representations.

As shown in Figure 3.3, ViTs and Residual Networks (ResNet [He et al. 111]) process visual information differently. The authors of [Raghu et al. 227] examined ViT's self-attention layers, which gather information from different spatial locations, contrasting with ResNet's fixed, local receptive fields in early layers. The study finds that even in ViT's lower layers, there is a mix of attention to local and global areas, whereas ResNets focus only locally in these layers. This difference becomes more pronounced in higher layers, where ViT's attention is entirely global. The work of [Raghu et al. 227] also shows that the amount of training data affects ViT's ability to focus locally in early layers.

Figure 3.3: The structure of representations in ViTs and convolutional networks is fundamentally different. ViTs display a uniform similarity across their entire structure, making them practically indistinguishable at different layers. On the other hand, the ResNet models show blokinness effect, where lower layers form a blob of similarity, namely low-level features, and last layers capture more high-level features. (Figure taken from [Raghu et al. 227])



Additionally, the study compares the features learned by ViTs and ResNets, revealing that ViTs access to more global information leads to different features than those computed by ResNet's local fields. Finally, they concluded that a ViT can approximate a Resnet at nearly 70%

3.2.5 Attention in Psychology, Neuroscience, and Machine Learning

The parallels between Transformers and neural processes in the human brain offer intriguing insights into how modern deep learning models may mirror biological information processing systems. As presented earlier, transformers process inputs by dynamically focusing on different data segments. Is this similar to how the human visual system employs selective attention to optimize perceptual tasks? We discuss these aspects here.

The work of [Toneva and Wehbe 264] introduces a novel approach to interpreting neural network models used in natural language processing through the lens of human brain activity. This work leverages brain imaging data of subjects reading complex natural texts to analyze and compare the linguistic representations in the commonly used NLP model Bidirectional Encoder Representations from Transformers (BERT) [Devlin et al. 69]. Then, they aligned neural network embeddings with brain activity

data to evaluate how different models process language-related information across various layers, context lengths, and attention types.

Their findings highlight notable differences in how these models handle context in linguistic processing, demonstrating that transformer models exhibit a more dynamic interaction between layer depth, attention mechanisms, and context length compared to other models. Additionally, they propose modifications to BERT that align more closely with brain data, showing that such alignment enhances the model's performance on syntactic NLP tasks and suggests a deeper similarity between how machines and the human brain process language. The seminal work of [Lindsay 174] explores the nature of attention and the underlying mechanisms that enhance cognitive and computational efficiencies. In this analysis, the author explains how attention in biological systems allocates limited cognitive resources more efficiently, allowing for heightened processing of relevant stimuli. This aspect of attention is paralleled in machine learning, where attention mechanisms prioritize essential data features, thus optimizing computational resources for tasks such as image recognition and language processing.

Furthermore, this work reviews the implementation of biological attention principles within machine learning models, highlighting how these models incorporate selective focus mechanisms that improve performance by minimizing the influence of irrelevant data and enhancing the ability to generalize from complex datasets.

The work concluded that machine learning models implement the simplest form of attention far behind the complexity of the human visual system. Hence, integrating more sophisticated models of attention from neuroscience into machine learning potentially leading to machine learning systems capable of effective multitasking.

3.3 Are ViTs inherently saliency learners?

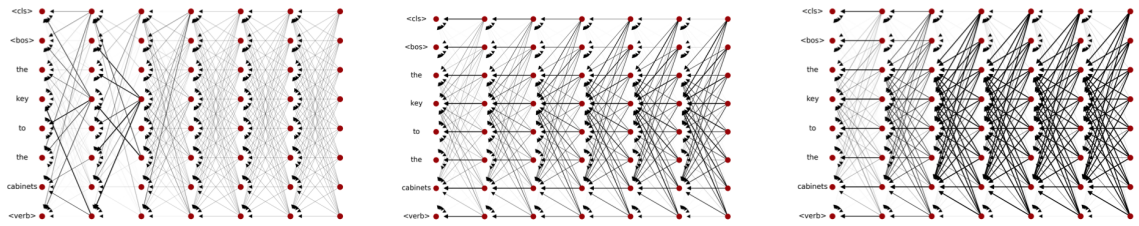
In this section, we explore the connections between Vision Transformers and saliency modeling by investigating the extent to which this architecture is intrinsically biased towards salient aspects in images through multiple experiments. This involves

examining whether the self-attention mechanism in vision transformers, which allows them to weigh and integrate various input image patches differently, inherently encourages the model to identify and fixate on salient features within the data. We begin with a detailed analysis of current techniques for visualizing self-attention, such as attention rollout and Layer-wise Relevance Propagation (LRP). Subsequently, we propose a mechanism to select a subset of self-attention maps and learn a shallow decoder to generate the final saliency map. The subsequent sections will delve deeper into the concept of salient attention in vision transformers and discuss the development of saliency maps in their self-attention, providing a comprehensive overview of their role in saliency learning.

3.3.1 ViTs Self-Attention Visualisation

Quantifying and interpreting the self-attention matrices of inner Vision Transformers has recently become an active research area, with many methods being proposed. The most common visualization technique involves treating the pairwise attention values between different patches as relevancy scores (i.e., raw attention) [Caron et al. 42, Xu et al. 283, Carion et al. 39]. These scores can be obtained from the last layer or by combining multiple layers. However, this method suffers from a blurring effect and tends to overemphasize the role of irrelevant tokens. An alternative approach is the Attention Rollout method [Abnar and Zuidema 2], which reassigns all attention scores by considering the pairwise attentions and assuming that attentions are linearly combined into subsequent contexts. This method appears to improve results over using a single attention layer. Addressing the issue of losing relevancy when employing a self-attention layer remains a challenge, as a naive propagation rule could lead to negative contributions that induce noise in the relevancy map. The authors in [Chefer et al. 45] have guided the relevancy propagation with a rule applicable to both positive and negative attributions. They calculate the scores for all attention heads using the Layer-wise Relevance Propagation (LRP) method [Binder et al. 18], and to eliminate negative contributions, the method integrates

Figure 3.4: Visualisation of attention weights. It can be seen that as we delve deeper into the model, the embeddings become increasingly contextualized. This highlights the importance of tracing the attention weights to the input layer.



(a) Embedding attentions (b) Attention rollout (c) Attention flow

relevancy and gradient information throughout the attention graph. We highlight the main methods:

Attention Rollout [Abnar and Zuidema 2]: The attention graph illustrates the flow of information between positions across different layers via edges, where the weights of these edges indicate the proportion of information transfer. Attention rollout is computed by recursively multiplying the attention weight matrices of the underlying layers. The lowest layer is designated as the input layer for computing input attention. In a Transformer with L layers, the task involves computing the attention that connects every position in a specific layer l_i to all positions in a prior layer l_j, with the condition that $j < i$. The attention graph consists of paths connecting node v at position k in l_i to node u at position m in l_j. A sequence of edges forms these paths, and the weight of each edge reflects the portion of information flow between the two nodes. The transfer of information from v to u via a specific path is quantified by multiplying the weights of that path's edges. Since multiple paths may exist between two nodes, the total information transmitted from v to u is the sum of contributions from all these paths. To calculate attention from l_i to l_j, the attention weight matrices of all preceding layers are multiplied. The following equation describes this process:

$$\tilde{A}(l_i) = \begin{cases} \sum A(l_i)\tilde{A}(l_{i-1}), & \text{if } i > j \\ A(l_i), & \text{if } i = j \end{cases} \tag{3.13}$$

here, $\tilde{A}$ is the attention rollout, whereas A is the raw attention.

Partial LRP [Voita et al. 273] was the first method to utilize Layer-wise Relevance Propagation (LRP) [Binder et al. 18], based on the observation that averaging attention heads is suboptimal since the relevance of attention heads varies across each layer [Voita et al. 273]. However, its application was limited because it did not propagate relevance scores back to the input, thus providing only partial information about the relevance of each head. It is important to note that the relevance scores were not directly evaluated but used instead to visualize the relative importance of attention heads and prune less relevant ones.

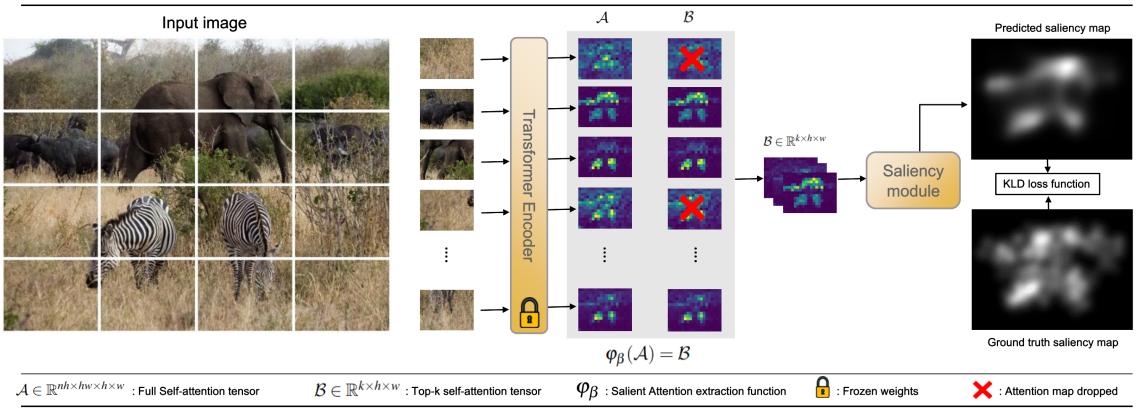
Relevance LRP [Chefer et al. 45] propagates relevance and gradients with respect to a specific class. The inputs for this process are the feature map and weights for layer l_i. This relevance propagation follows the principles of Deep Taylor Decomposition [Montavon et al. 201] and adheres to the conservation rule across two successive layers as described in [Montavon et al. 201]. Notably, the outcome of the rollout is fixed, given an input sample, irrespective of the target class being visualized. Moreover, it only considers pairwise attention scores. However, Relevance LRP requires classification logits and thus cannot be applied to self-supervised methods. In essence, the attention map $A(i)$ is analyzed for its gradients $\nabla A(i)$ and relevance $R(n_i)$ related to a target class. The cumulative output $C \in \mathbb{R}^{s \times s}$ represents the product of weighted attention relevance across all blocks:

$$\bar{A}(i) = I + \mathrm{E}_h(\nabla A(i) \circ R(n_i)) \tag{3.14}$$

$$C = \bar{A}(1) \cdot \bar{A}(2) \cdot \ldots \cdot \bar{A}(B) \tag{3.15}$$

In these equations, $\circ$ denotes the Hadamard product and E_h represents the mean across the 'heads' dimension. This method specifically focuses on the positive values of the gradients-relevance product, akin to emphasizing positive relevance. The identity matrix I is added to each block's weighted attention relevance $\bar{A}(i)$ to offset self-inhibition due to skip connections in the Transformer block.

Figure 3.5: Complete pipeline for training. The input image is processed with a Vision Transformer model to get the self-attention maps tensor $\mathcal{A}$. φ_β serves as a medium to produce the Salient Attention maps tensor $\mathcal{B}$ by selecting top-k informative modes. Then, $\mathcal{B}$ is fed to the Saliency Module to predict the visual saliency map.



3.3.2 Salient Attention

In visual attention modeling, salient regions within an image may correspond to objects or regions that capture the viewer's interest. However, these regions are not necessarily limited to objects; they could be any feature or pattern that draws attention, as discussed in [Borji and Itti 26]. Traditional methods for extracting relevance maps in Vision Transformers (ViTs) often focus solely on the $[CLS]$ token from classification models, as this token is believed to summarize the overall self-attention tensor. Consequently, the relevance map is derived from the row $C_{[CLS]} \in \mathbf{R}^s$, which assigns a relevance score to each token based on its impact on the classification task.

Moreover, as illustrated in Figure 3.5, we propose a novel approach agnostic to specific tokens when selecting the ViT's self-attention maps highlighting salient locations. The **Vision Transformer backbone** ($\mathbf{f_\theta}$) transforms the input image $\mathbf{x} \in \mathbb{R}^{C \times H \times W}$ into a representation $\mathbf{\Lambda}$. It processes the image in patches $x_p \in 1 \dots N$, where N is the number of patches and each patch $x_p \in \mathbb{R}^{C \times h \times w}$. We focus on the last layer's complete self-attention tensor $\mathcal{A} \in \mathbb{R}^{nh \times hw \times h \times w}$, excluding the $[CLS]$ token (the complete tensor would otherwise include $h \cdot w + 1$ elements), where nh represents the number of heads and hw the total number of tokens.

Our goal is to discern the salient positions within the second dimension and retrieve their corresponding self-attention maps. To accomplish this, we employ tensor decomposition, specifically the Candecomp-Parafac (CP) decomposition via Alternating Least Squares (ALS) [Kolda and Bader 143], which deconstructs $\mathcal{A}$ into a set of factor matrices, as shown in Equation 3.16:

$$\mathcal{A} \approx \sum_{r=1}^{R} \lambda_r \left(\mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \right) \tag{3.16}$$

here, R is the rank of the decomposition, and $\mathbf{a}_r$, $\mathbf{b}_r$, $\mathbf{c}_r$ are the factor matrices with dimensions corresponding to $h \cdot w$, h, and w, respectively. The symbol $\circ$ denotes the outer product between vectors.

The ALS algorithm iteratively updates each factor matrix until convergence is reached or a maximum number of iterations is completed. Post-convergence, the factor matrices enable the reconstruction of the original tensor or extracting salient information about the data's structure. By evaluating the factor matrices, we can pinpoint which modes carry the most information about $\mathcal{A}$, as indicated by higher weights in the decomposition. We then choose the top-k informative modes based on their weights for further examination. This method allows us to streamline the tensor's dimensionality and concentrate on the most pertinent information, likely corresponding to salient locations, facilitating the generation of a class-independent relevance map. The final tensor $\mathcal{C} \in \mathbb{R}^{k \times h \times w}$ is then obtained. Specifically, after decomposition, we can calculate the weights for each mode as follows:

$$w_r = \|\mathbf{a}_r\| \cdot \|\mathbf{b}_r\| \cdot \|\mathbf{c}_r\| \tag{3.17}$$

We define the index set $\mathcal{I}_k$ to include the indices of the top k weights:

$$\mathcal{I}_k = \{\text{indices of the top } k \text{ weights in } \{w_1, w_2, \ldots, w_R\}\} \tag{3.18}$$

Finally, the relevance map tensor $\mathbf{C}$ is constructed by selecting the slices from $\mathcal{A}$ corresponding to the indices in $\mathcal{I}_k$:

$$\mathbf{C} = \mathcal{A}(\mathcal{I}_k, :, :) \tag{3.19}$$

3.3.3 Obtaining the saliency maps from ViT Attention/Relevance maps

To obtain the saliency map, we train a small convolutional neural network (CNN)-based saliency module on top of the Attention/Relevance maps obtained using the various methods explained earlier. The saliency module takes these maps as input and learns to predict the corresponding saliency map, highlighting the most visually important regions in the input image.

Vision Transformer backbone ($\mathbf{f_\theta}$). This network maps the input image $\mathbf{x} \in \mathbb{R}^{C \times H \times W}$ to a representation $\mathbf{\Lambda}$. We select the ViT [Dosovitskiy et al. 78] model, which inherits a BERT-like architecture. The input is a sequence of all non-overlapping patches of size 16×16 of the input image, followed by flattening and linear layers to produce a sequence of embeddings. Similar to BERT, a classification token $[CLS]$ is appended at the beginning of the sequence and used for classification. We define ($\boldsymbol{\varphi_\beta}$) that serves as a medium to extract the Attention/Relevancy map $\mathcal{C} \in \mathbb{R}^{k \times h \times w}$ (k is the k-modes in Salient Attention, and the number of heads for all other methods). $\boldsymbol{\varphi_\beta}$ can be instantiated with any method from the pool of methods explained earlier: [Raw Attention, Attention Rollout, Partial LRP, Relevance LRP, Salient Attention]. These methods can be grouped into two folds: attention maps and relevance. Each has different properties and assumptions about the architecture and propagation of information in the network. We briefly describe each baseline in the following section and the different experiments for each domain.

The attention-map baselines are class-agnostic by definition and include roll-out [Abnar and Zuidema 2], which produces an explanation that considers all the attention maps computed along the forward pass. A more straightforward method is raw attention, i.e., using the attention map of block 1 to extract the relevance scores. The proposed saliency-based attention map is an extension of the raw attention by

retrieving the attention maps for the salient locations using the CP method and is not necessarily tied to the salient object in the scene. Unlike attention-map-based methods, relevance propagation methods consider the information flow through the entire network and not just the attention maps. These baselines include the partial application of LRP that follows [Bach et al. 7]. It is arguable whether these techniques might be practically class-agnostic [Chefer et al. 45] proved that the LRP method's visualizations remain roughly constant for distinct target classes. The Relevance LRP [Chefer et al. 45] is the most effective class-specific method as it relies on the propagation rule that considers both positive and negative contributions and provides fine-grained class-specific visualizations. **Saliency module ($\boldsymbol{\Omega_w}$)** is a non-linear mapping $\boldsymbol{\Omega_w} : \mathcal{C}^{k\times w\times h} \mapsto S_{\mathbf{p}}$ parameterized by $\boldsymbol{\omega}$, where $S_{\mathbf{p}} \in \mathbb{R}^{H\times W}$, is the predicted saliency map. The architecture of the saliency module consists of 4 blocks of [Conv2d $\rightarrow$ ReLU $\rightarrow$ Upsample].

Saliency loss function. The saliency task can be seen as a distance measure between the predicted saliency distribution $S_{\mathbf{p}} \in [0,1]^{W\times H}$, and the continuous ground truth $S_{\mathbf{GT}} \in [0,1]^{W\times H}$. The objective function must be designed to maximize the invariance of predictive maps and give higher weights to locations with higher fixation probability. Thus, the saliency module ($\boldsymbol{\Omega_w}$) is trained to minimize the Kullback-Leibler Divergence (KLD), widely adopted for benchmarking saliency models [Bylinskii et al. 37], the KLD between $S_{\mathbf{p}}$ and $S_{\mathbf{GT}}$ is given by:

$$\mathcal{L}_{\mathrm{KLD}}(S_{\mathrm{GT}}, S_{\mathrm{p}}) = \sum_{i=1}^{W\times H} S_{\mathrm{GT}i} \log\left(\epsilon + \frac{S_{\mathrm{GT}i}}{\epsilon + S_{\mathrm{p}_i}}\right) \tag{3.20}$$

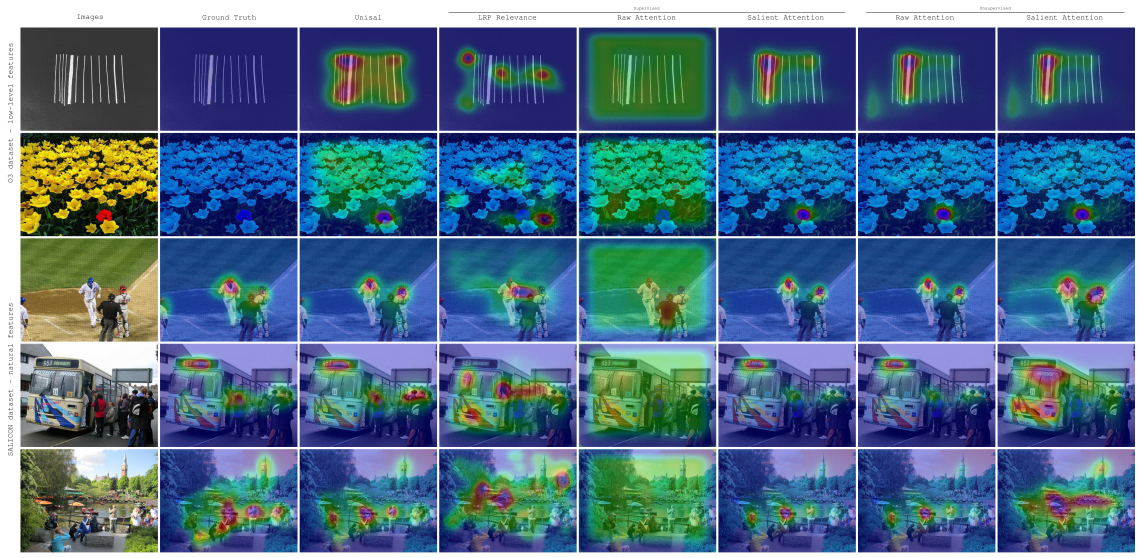
3.4 Experiments

Training. We experiment with the ViT-small/16 [Dosovitskiy et al. 78] model trained in a supervised manner on ImageNet [Deng et al. 68] and in a self-supervised setting (i.e., DIstillation with NO labels (DINO) [Caron et al. 42]). We train the CNN saliency module on the 10k/5k train/validation splits of the image saliency

Table 3.1: Comparative performance study on Salicon and MIT300. Supervised denotes initializing the ViT with classification weights on ImageNet, whereas unsupervised means using DINO self-supervised weights. Our approach using Salient locations achieves favorable gains across different settings.

	Models	Salicon					MIT300				
		SIM	s-AUC	CC	NSS	KLD	SIM	s-AUC	CC	NSS	KLD
	ITTI [Itti et al. 124]	0.37	0.61	0.20	–	–	0.46	0.13	0.44	1.11	0.95
	GBVS [Harel et al. 107]	0.44	0.63	0.42	–	–	0.48	0.62	0.47	1.24	0.88
	Salicon [Jiang et al. 130]	–	–	–	–	–	0.51	0.73	0.56	1.70	0.78
	CASNet [Fan et al. 87]		–	–	–	–	0.58	0.73	0.70	1.98	0.58
	EML-NET [Jia and Bruce 128]	0.79	0.74	0.89	2.05	0.52	0.74	0.67	0.78	2.48	0.84
	MSI-Net [Kroner et al. 149]	0.80	0.74	0.90	2.01		0.67	0.74	0.77	2.30	0.42
	TranSalNet [Lou et al. 180]	–	–	–	–	–	0.68	0.74	0.80	2.41	1.01
	SalGAN [Pan et al. 212]		0.75	0.76	2.47	-	0.63	0.73	0.67	1.86	0.75
	UNISAL [Droste et al. 79]	0.77	0.73	0.87	2.45	-	0.67	0.78	0.78	2.36	0.41
	DeepGaze [Kummerer et al. 155]	–	–	–	–	–	0.66	0.77	0.77	2.33	0.42
Supervised	Relevance LRP	0.72	0.71	0.75	2.01	0.59	0.61	0.71	0.73	1.94	0.70
	LRP	0.57	0.68	0.72	1.41	0.84	0.53	0.63	0.71	1.6	0.78
	Partial LRP	0.70	0.68	0.71	1.86	0.64	0.58	0.69	0.70	1.75	0.64
	Raw attention	0.55	0.66	0.61	1.58	0.98	0.54	0.70	0.72	1.39	0.81
	Rollout	0.58	0.69	0.67	1.45	0.80	0.55	0.72	0.73	1.42	0.75
	Salient Attention	0.71	0.69	0.73	2.12	0.57	0.62	0.72	0.70	2.14	0.72
Unsupervised	Raw attention	0.73	0.71	0.81	2.24	0.61	0.63	0.72	0.71	2.06	0.71
	Salient Attention	0.78	0.74	0.86	2.41	0.42	0.65	0.74	0.76	2.38	0.51

Figure 3.6: Qualitative results of the different models on sample images from Salicon and O3 datasets. It can be observed that the proposed approach can handle various challenging scenes well and produces consistent saliency maps



dataset Salicon [Jiang et al. 130] to evaluate the proposed framework. We use the aforementioned methods to obtain a final Attention/Relevancy map [1], which is fed to the CNN saliency module. It is worth mentioning that the ViT encoder weights were not fine-tuned on the saliency task; only the randomly initialized saliency module is trained on top of the frozen encoder. The motivation is to establish a robust evaluation procedure and prevent the encoder from adapting its parameters to saliency-specific requirements.

Evaluation setting. We compared our approach against state-of-the-art methods as listed in [Wang et al. 277] and included newer models with available implementations from [Kümmerer et al. 152]. We also conducted tests on the MIT300 benchmark [Judd et al. 135], which presents more challenges than the Salicon test set. In line with recommendations from [Kümmerer et al. 152, Bylinskii et al. 36], we employed the following evaluation metrics: Similarity Metric (SIM), shuffled AUC (s-AUC), Linear Correlation Coefficient (CC), Normalized Scanpath Saliency (NSS), and the Kullback-Leibler Divergence (KLD) [Bylinskii et al. 36]. The more recent formulations of these metrics were adopted from [Kümmerer et al. 152].

Technical details. We adopted the implementations of ViT [Dosovitskiy et al. 78] from the official DINO repository[2]. For the saliency module, we considered four blocks with 32 input convolution filters (59k parameters). This module was trained on the 10k images of Salicon. The training was conducted using a single NVIDIA RTX3090 24GB. All variants were trained for 30 epochs using the AdamW [Loshchilov and Hutter 179] optimizer. We employed a warm-up period of 6 epochs and a cosine learning rate scheduler with a maximum learning rate set to 10^{-3}.

3.5 Results

State-of-the-art comparison. We compared the proposed approach to state-of-the-art image saliency models on the Salicon, MIT1300, and CAT2000 validation

[1] LRP
[2] Github

sets. Tables Table 3.1 and Table 3.2 show the performance comparisons in terms of the five metrics for the respective validation sets. Our method, 'Salient Attention,' performs favorably against existing saliency models. Notably, it outperforms various methods of attention and relevancy extraction. In this context, "end-to-end" refers to models that process raw input images directly and output saliency maps, thereby integrating all steps of feature extraction, learning, and prediction into a single automated workflow. For the supervised setting, the relevance LRP and the salient locations-based approach show the best performance, followed by partial LRP and Rollout. The raw attention-based approach shows the worst performance among supervised models. Surprisingly, initializing the ViT backbone with self-supervised weights and training the saliency module on both the Raw Attention and Salient Locations maps exhibits scores on par with end-to-end methods such as SalGAN [Pan et al. 212] and EML-NET [Jia and Bruce 128]. This is likely due to the ViT encoder adapting the parameters to fit the classification task, corrupting the raw attention maps, and ending in what is known as supervision collapse [Jing et al. 132].

Saliency for Low-level Feature Images. State-of-the-art saliency models effectively capture high-level features such as cars, humans, etc. However, these models often struggle to adequately represent a range of other critical features that are significant in studying human visual attention, a subject extensively explored in psychology and neuroscience. One key aspect of this is visual search, which relates closely to Feature Integration Theory (FIT) [Treisman and Gelade 266, Kotseruba et al. 147]. According to FIT, the human brain processes regions differing markedly in one feature dimension—such as color, intensity, or orientation—in parallel. These are considered low-level features and are fundamental mechanisms in the Human Visual System.

Furthermore, we conducted evaluations of the performance of UNISAL and our ViT-based approaches on samples of low-level attention using images from a recently proposed dataset [Kotseruba et al. 147]. The aim is to understand the differences in how saliency exploration is performed when the self-attention mechanism promotes

global connectivity between the image patches. As demonstrated in Figure 3.6, UNISAL generates high-quality saliency maps that closely align with the ground truth for natural images from Salicon. In these images, high-level features such as human faces, buses, and monuments are prominent ($3^{rd}, 4^{th}, 5^{th}$ rows in Figure 3.6). The human visual system utilizes a combination of bottom-up and top-down processes to perform attention tasks. However, this complex interaction might not be fully captured in fixation or saliency datasets, leading to a potential misalignment with the definition of visual saliency. Consequently, while end-to-end deep learning-based models can learn to predict saliency effectively, they might not fully adhere to its subtle definitions as they primarily focus on explicit, salient features that are easily annotated in datasets.

In fact, early computational models of the visual human system, such as those cited in [Navalpakkam and Itti 203, Kootstra et al. 145, Murray et al. 202, Bruce and Tsotsos 32, Garcia-Diaz et al. 91, Seo and Milanfar 246, Goferman et al. 96, Gao and Vasconcelos 90], were predominantly cognitive-based and relied on computing multiple visual features like color, edge, and orientation at various spatial scales to produce saliency maps. These models attempted to mimic the human visual system's processing more closely as part of the model design. Given this context, could self-supervised Vision Transformers (ViTs) bridge this gap by leveraging their ability to reason over recurring features and drawing dependencies and similarities across different parts of the input? This approach could align more closely with human visual attention's cognitive and perceptual processes by recognizing underlying patterns and dependencies that are not explicitly labeled in training datasets.

In fact, UNISAL [Droste et al. 79] fails to respond to simple features. For example, considering a color (2^{nd} row in Figure 3.6), UNISAL did not capture the red flower as the most salient object. In contrast, the ViT succeeded in doing so, as this pattern is solved with the global nature of the self-attention mechanism. Also, UNISAL does not discriminate the larger shape (1^{st} row in the figure). Furthermore, a differently shaped object to others should capture the viewer's attention, but all approaches fail

to do so. This suggests that ViTs correlate with the gaze data and incorporate visual system characteristics as important priors induced by the self-attention mechanism.

3.5.1 Supervised vs. Self-Supervised self-attention maps

As shown in Figure 3.7[3], self-attention heads from the last layer are utilized to analyze the raw attention in both self-supervised and supervised DINO versions. In the context of self-attention mechanisms, "heads" refer to multiple independent attention mechanisms that run in parallel. Each head learns a different aspect of the data, allowing the model to focus on various features or parts of the input simultaneously. This multi-head approach helps improve the robustness and expressiveness of the attention mechanism.

In our investigation, we specifically examine the attention map when employing the [CLS] token. When comparing the attention maps between the two approaches, notable differences emerge. Through self-supervised learning, the attention mechanism demonstrates a superior ability to capture elements within the images that consistently elicit attention according to the training data. The attention maps generated by the self-supervised DINO model exhibit remarkable detail and clarity. These maps effectively highlight regions that the model has learned to recognize as significant. Consequently, while the model highlights these areas precisely, enabling effective focus on relevant regions, the definition of relevance is shaped by the training context and the inherent data biases. On the other hand, in the supervised DINO model, the attention maps appear to be relatively less refined. The supervised approach, relying on labeled data, exhibits a relatively coarser representation of object attention. The attention maps produced by the supervised DINO model lack the same level of sharpness and fine-grained focus as those generated through self-supervised learning. This might be explained by the model exploring more spurious correlations to solve the downstream classification task.

This disparity in attention map quality suggests that self-supervised DINO models

[3]Figure taken from the DINO [Caron et al. 42] paper

Table 3.2: Results on the CAT2000 validation set.

Models		CAT2000				
		SIM	s-AUC	CC	NSS	KLD
	SalGAN	0.54	0.63	0.56	1.46	0.93
	DeepGaze	0.68	0.64	0.79	1.96	0.38
Unsupervised	Raw attention	0.62	0.60	0.75	1.84	0.69
	Salient Attention	0.65	0.62	0.76	1.90	0.63

excel in capturing the inherent structure and salient features of objects within the images. By learning representations from the data, the self-supervised approach uncovers meaningful patterns and relationships without explicit human annotations. Consequently, the resulting attention maps reflect a stronger alignment with the underlying objects and their contextual significance. This justifies the results in Table 3.1, where saliency is easily captured from self-supervised DINO.

3.5.2 Saliency for low-level features

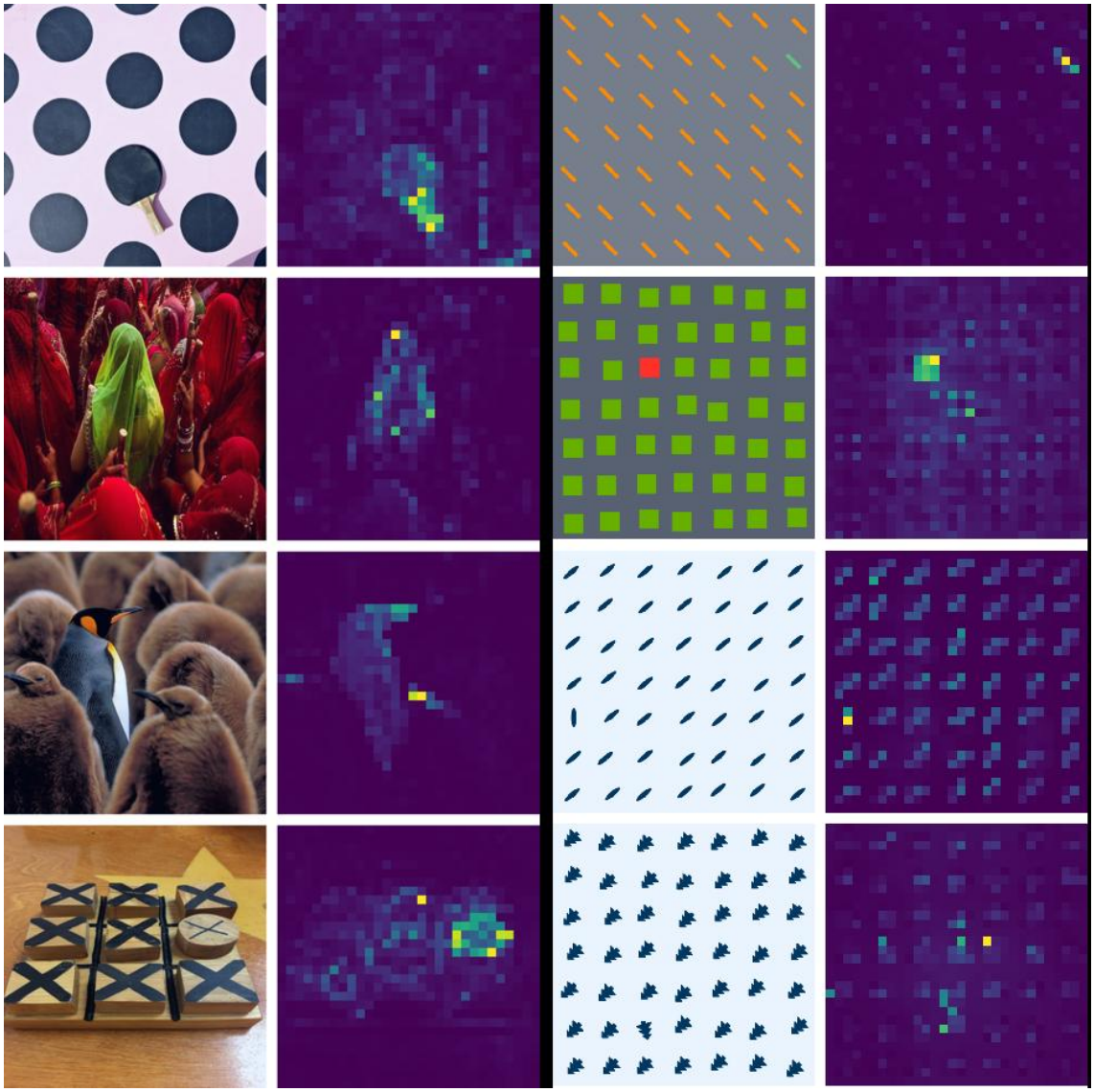
As demonstrated in Figure 3.8, the attention maps produced by Vision Transformers capture high-level semantic information and shed light on low-level visual features. The model can capture the intricate interplay of these features, resulting in attention maps that provide a detailed representation of the visual content. This suggests a notable capacity of ViTs to attend to low-level features such as color, intensity, and shape features that are foundational to the human visual system.

These observations raise the possibility that ViTs can selectively attend to relevant information, allowing for efficient processing and analysis of visual inputs. For instance, the model's focus on color may aid in distinguishing and categorizing objects based on their hues, while attention to intensity features could enhance edge detection and contour perception. Similarly, attending to shape features might facilitate recognizing and understanding complex object structures.

Such capabilities suggest that Vision Transformers could emulate some aspects of hierarchical processing observed in the human visual system, as posited by Feature Integration Theory. However, it is important to acknowledge that these claims are

Figure 3.7: Self-attention heads from the last layer for the raw attention. We look at the attention map when using the $[CLS]$ token for self-supervised vs supervised DINO. It can be seen that the self-supervised captures the objects better and produces sharper attention maps.

Figure 3.8: Self-attention heads from the last layer for the raw attention. We look at the attention map when using the $[CLS]$ token for self-supervised on the O3 and P3 datasets.



based on empirical observations from specific datasets and model configurations. Further research is needed to validate these findings across diverse visual tasks and conditions. Claiming that ViTs fully replicate or understand the intricacies of human vision would be an overstatement without substantial corroborative evidence. This discussion is intended to highlight potential areas of alignment between Vision Transformers and human visual processing, not to assert equivalence or superiority.

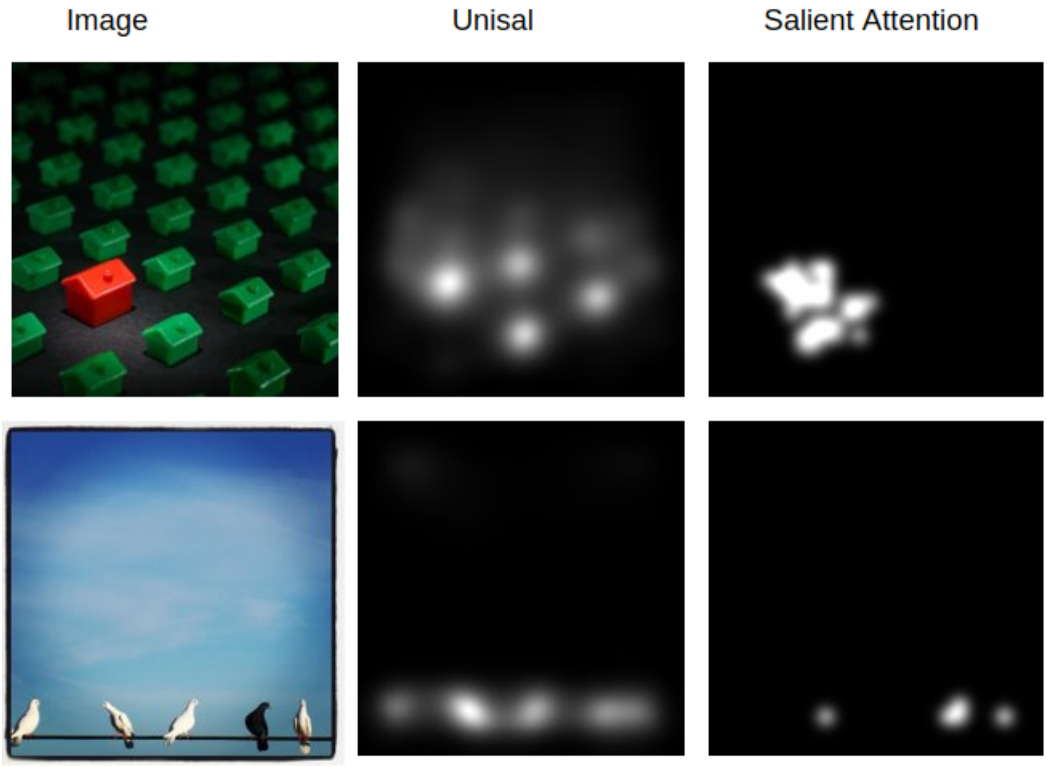
As shown in Figure 3.9, the first row features a grid of green cubes with one distinct red cube among them. This test aimed to assess the model's ability to highlight uniquely colored objects within a uniform field. The UNISAL saliency model appears to highlight several cubes but does not distinctly mark the red cube, indicating generalized attention that does not focus specifically on the unique feature. In contrast, our "Salient Attention" approach produces a more focused saliency map, highlighting the red cube. This result indicates that our approach effectively captures the saliency of uniquely featured elements in the scene, aligning with the test's objective.

The lower part of the image presents a scene with several white pigeons and one black bird against a clear blue background. Here, the objective was similar: to determine if the saliency model can isolate and emphasize an anomalous feature— in this case, the black bird among white pigeons. The UNISAL's saliency map diffusely highlights all the birds without distinct emphasis on the black bird. Conversely, the "Salient Attention" map clearly marks the black bird, demonstrating a superior ability to detect and highlight unique features within the context of the image. This again confirms that our method aligns well with the test objectives, effectively identifying and emphasizing unique or standout elements in complex visual scenes.

The comparison illustrates that The ViT seems to be more effective in detecting unique, low-level features ("the red cube" and "the black pigeon") in these images, as opposed to the UNISAL method, which provides a more generalized saliency detection without emphasizing the unique elements. This indicates that transformer-based models may be better suited for tasks that require identifying distinctive features

within a visual scene.

Figure 3.9: Comparing the saliency maps of UNISAL and our method on images highlighting clear low-level features.



3.6 Ablation study

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

Other ViT variants

In addition to the ViT-small/16 model that was evaluated, we also explored several other variants. This included ViT-small/8, ViT-base/16, and ViT-base/8, each offering different representational capacities. These models had been pre-trained on large-scale datasets such as ImageNet [Deng et al. 68] in a self-supervised learning paradigm, similar to the ViT-small/16 model used in the main experiments. By investigating a range of ViT variants, we aimed to gain insights into the performance characteristics and generalization abilities of different ViT configurations when integrated into the proposed saliency framework. We experimented with Raw

Table 3.3: Comparative performance study on Salicon. Unsupervised stands for using DINO self-supervised weights.

Models		Salicon					MIT300				
		SIM	s-AUC	CC	NSS	KLD	SIM	s-AUC	CC	NSS	KLD
ViT-S/16	Raw attention	0.72	0.72	0.80	2.20	0.60	0.64	0.71	0.70	2.08	0.69
	Salient Attention	0.78	0.73	0.84	2.42	0.41	0.66	0.75	0.76	2.39	0.52
ViT-S/8	Raw attention	0.66	0.64	0.71	1.86	0.68	0.68	0.69	0.65	1.78	0.84
	Salient Attention	0.74	0.71	0.80	2.13	0.49	0.60	0.72	0.72	2.02	0.53
ViT-B/16	Raw attention	0.73	0.71	0.81	2.24	0.61	0.63	0.72	0.71	2.06	0.71
	Salient Attention	0.78	0.74	0.86	2.41	0.42	0.65	0.74	0.76	2.38	0.51
ViT-B/8	Raw attention	0.70	0.68	0.77	2.01	0.58	0.60	0.71	0.67	2.01	0.70
	Salient Attention	0.77	0.72	0.82	2.23	0.45	0.62	0.74	0.75	2.31	0.53

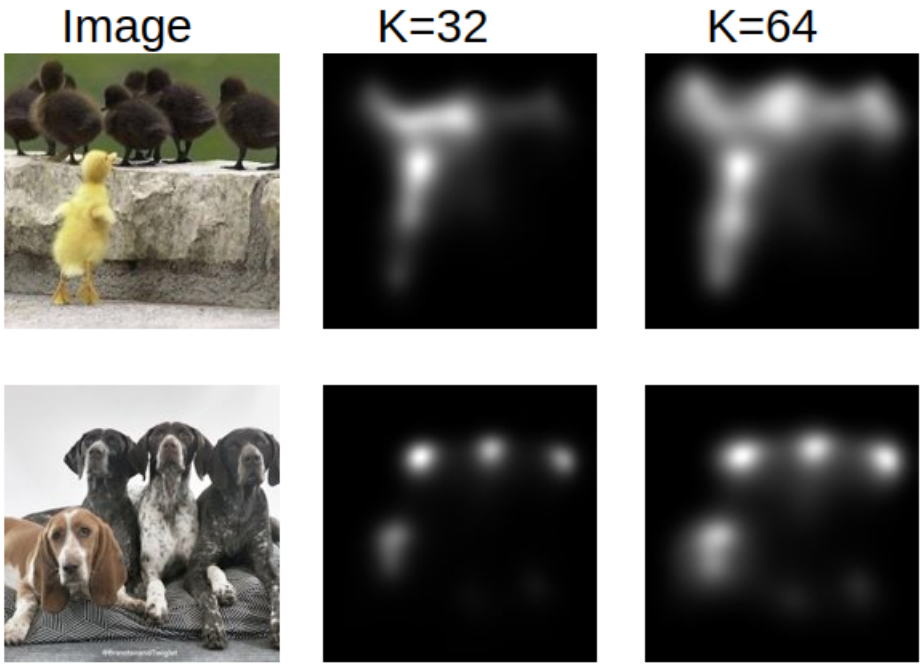
Attention and Salient Attention and reported the results on the Salicon validation set in Table 3.3. It appeared that the patch size was more important than the model size. Indeed, the ViT-S/16 surpassed the ViT-B/8 in both the Raw Attention and Salient Attention evaluations. This might be explained by the fact that smaller patches tend to result in overestimated saliency maps because the number of total patches is higher in this case.

The effects of the number of Salient Locations. Table 3.4 presents the performance of the Salient Locations based framework when the number of top k-modes is varied. We observe that 32 is the optimal value, and higher values degrade the performance since the added modes do not carry any useful information and may act as noise. As shown in Figure 3.10, for the image of the ducklings, the saliency map using 32 modes reveals a focused bright spot on the yellow duckling, indicating high saliency or importance in this region. This suggests that the K=32 setting is precise in identifying the standout features (i.e., the yellow duckling—as the primary region of interest). The background, including the black ducklings, is appropriately deemphasized, reflecting a lower saliency. In contrast, the K=64 saliency map shows a more dispersed saliency map, with lighter spots over the black ducklings, suggesting an overestimation of saliency in these areas. This implies that while K=64 captures the general distribution of features, it also assigns more mass to the salient areas, thus potentially lowering the metric scores that evaluate the precision of saliency detection.

The second row Figure 3.10 depicts a quartet of dogs lying closely together on a patterned fabric. Three dogs have black and white spotted fur, and the fourth has a distinct brown and white coat. Similarly, for the dogs, the K=32 saliency map isolates the faces of the dogs with sharp salient bobs, accurately predicting salient regions without unnecessary diffusion of saliency to the surroundings. This sharpness in the K=32 map shows precision in highlighting the salient features—the dogs' faces—and effectively ignoring the less significant patterns of the fabric beneath them. However, the K=64 saliency map for the dogs shows a spread of brightness that extends beyond the dogs' faces to the surrounding areas, including the fabric.

In summary, the K=32 saliency maps provide a more focused and accurate representation of salient features in the images. In contrast, the K=64 maps demonstrate a broader, less precise distribution of saliency, which could be less effective for tasks that require fine-grained saliency discrimination.

Figure 3.10: Comparing the impact of adding modes to the input of the saliency module. It can be seen that K=32 is sharper and does not overestimate saliency, whereas K=64 captures the distribution but puts more mass on salient areas, hence, ends in worse metric scores.



Is the saliency module required? Research in cognitive science (e.g., [Yan et al. 288, Li 171]) has shown that low-level saliency detection in both humans and

Table 3.4: Impact of the number of selected modes k. Performance comparison when k is varied. Thirty-two modes is the optimal number for the saliency task.

Models		**Salicon**					**MIT300**				
		SIM	s-AUC	CC	NSS	KLD	SIM	s-AUC	CC	NSS	KLD
Salient Locations	k=16	0.73	0.70	0.76	2.12	0.52	0.61	0.71	0.73	1.94	0.70
	k=32	0.78	0.74	0.86	2.41	0.42	0.65	0.74	0.76	2.38	0.51
	k=64	0.70	0.69	0.72	1.90	0.60	0.57	0.69	0.70	1.75	0.64
	k=128	0.71	0.68	0.73	1.93	0.59	0.55	0.70	0.73	1.84	0.61
	k=256	0.68	0.62	0.67	1.55	0.71	0.51	0.66	0.65	1.33	0.78

animals occurs early in the primary visual cortex. However, most state-of-the-art approaches predominantly utilize a supervised learning paradigm. Our study explored an alternative approach by attempting to generate saliency maps directly from the raw attention of the [CLS] token in a self-supervised Vision Transformer.

Specifically, we converted the raw attention map into a discrete fixation map $\mathcal{F}$ by applying a threshold $\lambda = 0.9$ to retain only the most relevant attention nodes. Subsequently, we applied a Gaussian filter to smooth the fixation map. Each pixel value was replaced by a weighted average of its neighbors in this process, with weights given by a Gaussian distribution. The degree of smoothing was controlled by the parameter σ, resulting in a predicted continuous saliency map G_p. Although this fully unsupervised approach produced visually appealing saliency maps, the quantitative performance did not meet the benchmark standards set by other methods on the Salicon validation set. Specifically, the Kullback-Leibler Divergence (KLD) score was 1.12, and the Normalized Scanpath Saliency (NSS) score was 1.46. These results indicate that while the unsupervised approach has potential, it still lags behind supervised methods in terms of accuracy and consistency.

3.7 Conclusion

Motivated by the research question of whether Vision Transformers (ViTs) can be connected to visual attention modeling. This chapter investigates how the emerging properties of ViTs could be utilized for saliency detection with minimal supervision. We examined existing visualization methods for ViTs and introduced a

class-agnostic approach for selecting the self-attention maps of salient locations in ViTs. This approach facilitates the identification of informative modes within the attention tensor. Both qualitative and quantitative results have demonstrated the competitiveness of this approach. We believe these findings may reveal analogies between the learning dynamics of ViTs and the human visual system. However, it is clear that the method still requires supervision to learn the saliency module. While we have achieved baseline scores across various metrics, more refined methods could potentially solve the task in a fully unsupervised manner. Similar to Relevance Layer-wise Relevance Propagation (LRP) [Chefer et al. 45], this would involve designing specific propagation rules and conservation laws tailored for the saliency task. One recent attempt is the zero-shot semantic segmentation method based on the pre-trained vision-language model CLIP [Radford et al. 225], which utilizes text embeddings as queries. Given the similarities between saliency and semantic segmentation, language-supervised models might offer zero-shot saliency capabilities.

Currently, the method proposed in this chapter is limited to using a frozen pretrained ViT and does not optimize the transformer weights due to the complexity introduced by the use of patches, which makes decoding a saliency map challenging. We will address this limitation in the following chapter by introducing the Saliency Transformer (SalTR) model. Also, we will attempt to make connections to the top-down vs bottom-up attention, where this bias could be induced in the architecture of the model.

Chapter 4

Learning saliency from fixations only

4.1 Introduction

As reviewed in Chapter 2, saliency prediction models predominantly rely on continuous saliency maps for training. These maps represent each pixel's probability of attracting human attention and are crucial for performing supervised learning [Pan et al. 213, Kruthiventi et al. 150, Pan et al. 212]. However, these models often struggle with optimizing for discrete fixation maps, which are used to mark the exact locations where human saccades occur. While these models effectively generate high-quality saliency maps, their architectures do not replicate the natural data collection process. Instead, they attempt to implicitly learn from post-processed saliency maps, which may not perfectly capture the underlying mechanisms of human visual attention.

To overcome this limitation, and inspired by [Carion et al. 39], we propose a novel approach that learns saliency prediction solely from fixation maps without relying on continuous saliency annotations. The objective is to make the model adaptable to the downstream task, allowing for adjustments in post-processing to meet specific

This chapter is based on our WACV 2024 publication [Djilali et al. 74]

task requirements based on access to the fixation points. For example, in applications such as adaptive image and video compression or progressive transmission, one could adjust the number of learned fixation queries and the smoothing parameter in the Gaussian filter according to the desired compression level. This controllable/steerable property distinguishes our model from existing approaches. Our method, the Saliency Transformer (SalTR), leverages parallel decoding in transformers to directly predict fixation points. The contributions of this chapter are as follows:

- Introduction of Saliency TRansformer (SalTR), a novel approach for saliency prediction using transformers and fixation maps, demonstrating its effectiveness on benchmark datasets.
- In the subsequent sections, we delve into the details of our analyses and the design and evaluation of SalTR, contributing to a deeper understanding of visual attention modeling and its implications for computer vision tasks.

4.2 Saliency Transformers (SalTR)

Following the design of DETR [Carion et al. 39], SalTR is an End-to-end saliency predictor that includes a ResNet backbone [He et al. 111], Transformer encoder and decoder [Vaswani et al. 269], and a fixations prediction head. The decoder part is adapted to solve the saliency task, as shown in Figure 4.1. Given an image, representations are extracted using a CNN backbone and a Transformer encoder to enrich the features. Then, fixation queries are fed to the Transformer decoder to search for fixation locations given the image features through cross-attention. The motivation behind this design is to learn solely from fixations. Hence, the number of queries used fully controls the saliency density.

4.2.1 SalTR components

Given an image dataset, $\mathcal{D} = \{\mathbf{x}_1, ..., \mathbf{x}_{|\mathcal{D}|}\}$ where each $\mathbf{x}_i \in \mathbb{R}^{C \times H \times W}$ represents an image, the goal is to predict a set of spatial fixation points $\bar{f}_i = (\bar{x}_i, \bar{y}_i)$. These points

serve as the basis for constructing the fixation map $\mathcal{M}_p$, which initially assigns a value of 1 at each fixation point and 0 elsewhere, effectively defining its domain as the set of all pixel coordinates in the image, and its range as binary values $\{0, 1\}$. We then smooth this map using a Gaussian filter with a standard deviation σ, to transform it into the final continuous saliency map. This smoothing process alters the range of $\mathcal{M}_p$ from $\{0, 1\}$ to the continuous interval $[0, 1]$, representing the likelihood of each pixel being a point of visual focus.

However, some significant differences distinguish our approach from previous works. Unlike traditional methods, our model predicts only the fixation points and does not utilize continuous ground-truth saliency maps during training. Therefore, we aim to mimic the way visual attention datasets are created, incorporating the use of both fixation queries and parallel decoding. The key modules of our approach are outlined in the following sections.

CNN Encoder ($\mathbf{f}_\theta$): The encoder, parameterized by $\boldsymbol{\theta_e}$, is a network $\mathbf{f}_\theta : \mathbf{x} \mapsto \boldsymbol{\Gamma}$ implemented using a ResNet50 backbone [He et al. 111]. This is followed by a two-layer 1×1 convolutional projection head with batch normalization and ReLU activation, which reduces the channel dimension from 2048 to 256.

Transformer Encoder: $\boldsymbol{\Omega_\omega}$ maps $\boldsymbol{\Gamma}^{c \times h \times w}$ to $\boldsymbol{\gamma}^{c \times h \times w}$. $\boldsymbol{\Gamma}$ is first reshaped into a sequence of size $c \times hw$ and then augmented with 2D positional encodings [Bello et al. 16]. The multi-head self-attention layers parse messages across $\boldsymbol{\Gamma}$ channels to capture contextual information. This functions as a smoothing prior, causing pixels sharing the same semantic class to attend to each other regardless of their spatial position in the image, effectively ignoring structural information. Furthermore, $\boldsymbol{\gamma}$ represents the smoothed transformation of $\boldsymbol{\Gamma}$, ensuring spatial coherence in the vicinity of a given pixel i and semantic coherence for pixels j that are further away but share the same class of semantic patterns.

Transformer Decoder: The decoder processes a fixed number of embeddings (i.e., fixation queries) of size 256 using multi-headed cross-attention mechanisms, where the keys and values are derived from the image features. Each embedding

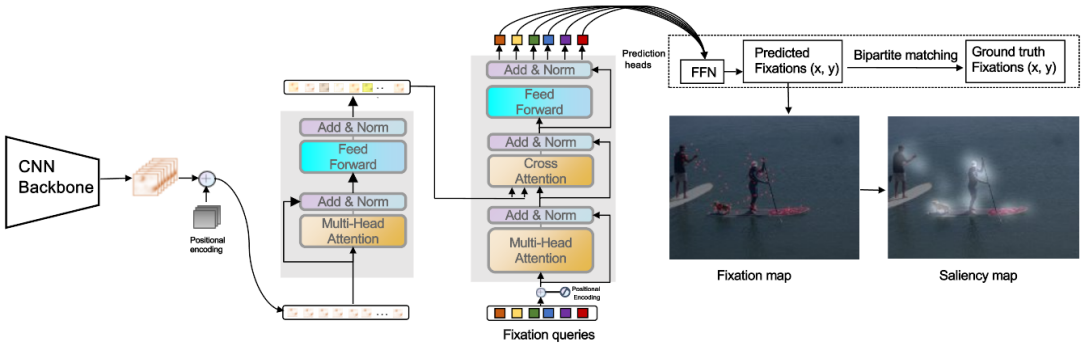


Figure 4.1: Complete pipeline for training. The CNN backbone produces the latent representation given an input image. The transformer encoder enhances this representation for a suitable decoding. The queries in the transformer decoder cross-attention are a fixed number of fixation queries, that attend to the image features. The prediction head maps the output embedding to a spatial fixation location.

Γ_i is processed simultaneously across multiple attention heads in this multi-headed cross-attention mechanism. Denoting the number of heads as H, for each head h, the embedding Γ_i interacts with image features $\mathcal{F}$ through a cross-attention operation. This operation is formalized as:

$$\text{Attention}_h(\Gamma_i, \mathcal{F}) = \text{softmax}\left(\frac{Q_h(\Gamma_i) \cdot K_h(\mathcal{F})^T}{\sqrt{d_k}}\right) V_h(\mathcal{F}) \tag{4.1}$$

where Q_h, K_h, V_h are the query, key, and value projection matrices for head h, and d_k is the scaling factor usually set as the dimension of the key vectors. The output embeddings are then mapped to a fixation point $\bar{f}_i = (\bar{x}_i, \bar{y}_i)$ using a 3-layer MLP. Basically, the fixation queries would attend to the image features to predict where to saccade. Clearly, however, a naive implementation ends in queries attending to the same location (i.e., the most salient ones). We show in the following section how to avoid this from happening.

4.2.2 Learning dynamics

Following the experimental setup used in creating saliency datasets, we denote the decoder fixation queries as $\mathcal{F}_q = F_{q0}, ..., F_{qN}$. These queries simulate the viewer's attention, i.e. where they attend to the image features, and saccade to a spatial

position that maximizes the attention task. Furthermore, we do not want the queries to predict the same ground truth fixations. Thus, the loss assigns a unique matching using the Hungarian algorithm between the prediction and ground truth fixations and then minimizes the l_1 distance for the respective matched positions.

Denote $f_i = (x_i, y_i)$ as a sample from the ground truth set, and the corresponding prediction as $\bar{f_i} = (\bar{x}_i, \bar{y}_i)$. To find a bipartite matching between these two sets, as in [Carion et al. 39], we search for a permutation of N elements $\phi \in S_N$ with the lowest cost:

$$\hat{\phi} = \arg\min_{\phi \in S_N} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{\text{match}}(f_i, \hat{f}_{\phi(i)}), \tag{4.2}$$

where $\mathcal{L}_{\text{Match}}(f_i, \hat{f}_{\phi(i)})$ is a pairwise matching cost between the ground truth f_i and a prediction with index $\phi(i)$. The final loss is:

$$\mathcal{L} = \sum_{i=1}^{N} \left|\left| f_i - \hat{f}_{\hat{\sigma}(i)} \right|\right|_1 + \alpha \mathcal{L}_{\text{NSS}}(\mathcal{M}_p, \mathcal{M}_{gt}), \tag{4.3}$$

where $\mathcal{M}_{gt} \in \{0, 1\}^{H \times W}$ is the ground truth fixation, and the predicted fixation map $\mathcal{M}_p$ is obtained using:

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

The normalized scanpath saliency loss (NSS) is defined as follows:

$$\mathcal{L}_{\text{NSS}}(\mathcal{M}_p, \mathcal{M}_{gt}) = \frac{1}{S} \sum_{i} \hat{\mathcal{M}}_{p_i} \times \mathcal{M}_{gt_i}, \tag{4.4}$$

where $S = \sum_i \mathcal{M}_{gt_i}$ and $\hat{\mathcal{M}}_p = \frac{\mathcal{M}_p - \mu(\mathcal{M}_p)}{\sigma(\mathcal{M}_p)}$, and i refers to the i-th pixel, and S represents the total count of fixated pixels. A score of 0 indicates chance, while a positive NSS value indicates agreement between the maps beyond chance. Thus, we aim at minimizing the negative value of $\mathcal{L}_{\text{NSS}}$, by setting α to a negative value (-0.2).

4.3 Experimental setting

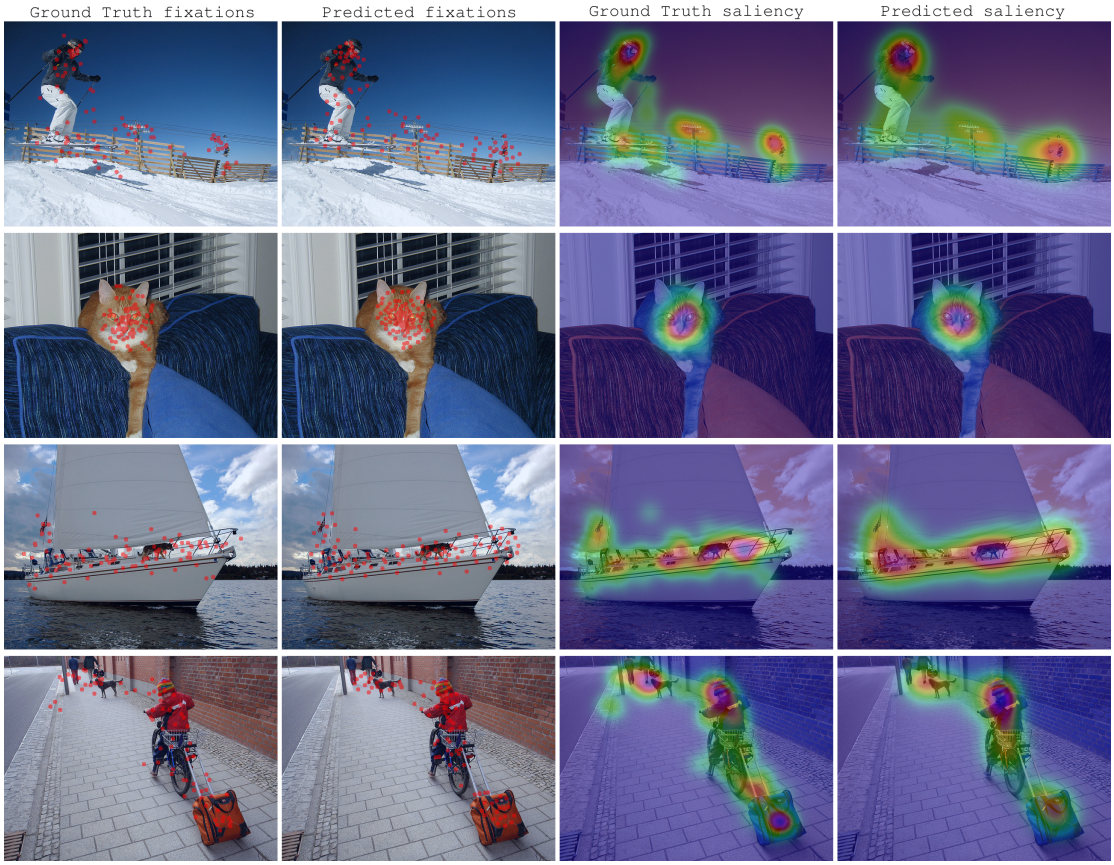


Figure 4.2: Qualitative results of our model on sample images from SALICON. It can be observed that the proposed approach is able to handle various challenging scenes well and produces consistent fixation/saliency maps.

Training. To evaluate the proposed framework, we train SalTR on the 10k/5k train/validation splits of the image saliency dataset Salicon [Jiang et al. 130].

Sampling the target fixations. The ground truth fixation map $\mathcal{M}_{gt}$ contains a large number of fixation points (i.e., up to 500) gathered across a batch of subjects (16 viewers per image for the Salicon dataset). Hence, it is necessary to select N points to match the number of fixation queries in the transformer decoder. Moreover, N should be significantly smaller than 500 for computational efficiency. For simplicity, we adopt a uniform sampling of N samples out of S points of $\mathcal{M}_{gt}$ to obtain $f_i = (x_i, y_i)$, where $i \in \{1, \ldots, N\}$. This guarantees a diverse set of samples from different viewers.

Accelerating the training. We observed that SalTR is difficult to optimize, and suffers from slow convergence, i.e., more than 100 epochs are needed to obtain

comparable performance to baselines. Following the object detection literature [Dai et al. 64, Sun et al. 257, Wang et al. 279, Wang et al. 279], several hypotheses can be proposed to account for this. First, the attention weights are uniformly assigned to all pixels in the feature maps at initialization, hence attending to meaningless locations that do not contribute to the feature propagation mechanism. Second, the discrete bipartite matching is unstable under stochastic optimization, as the same query is matched with different objects across epochs. Lastly, the decoder cross attention is under-optimized in the early training, resulting in noisy contextual information for the queries.

Addressing these issues, we draw inspiration from the concept of deformable convolution, as described in prior works [Dai et al. 63], the approach of [Zhu et al. 302]. We propose adapting the deformable attention mechanism for SalTR, termed 'Deformable SalTR'. This approach incorporates a translation term into the transformer attention formula, enabling a focused and sparse spatial sampling by directing attention to a limited set of chosen locations, known as reference points. This gating mechanism thus approximates the full self-attention process, leveraging the inherent locality bias and excluding the computation of long-term dependencies, which are less relevant for the saliency task.

In Deformable SalTR, given a feature map Γ, for each query element indexed by q with a content feature F_q and a 2D reference point p_q, the deformable attention feature is computed as follows:

$$\text{DeformAttn}(F_q, p_q, \Gamma) = \sum_{m=1}^{M} \sum_{k=1}^{K} A_{mqk} \cdot W_m^0 x(p_q + \Delta p_{mqk}) \tag{4.5}$$

here, m indexes the attention head, k indexes the sampled keys, and K represents the total number of sampled keys, which is significantly smaller than the total spatial locations HW. The terms Δp_{mqk} and A_{mqk} denote the sampling offset and attention weight for the k^{th} sampling point in the m^{th} attention head, respectively. The attention weights A_{mqk}, normalized such that their sum equals 1, are obtained through a softmax operation applied to the last MK channels of a linear projection

of the query feature F_q. The sampling offsets Δp_{mqk}, represented as 2D real numbers, are derived from the first $2MK$ channels of the same projection.

This deformable attention module, particularly suited for convolutional feature maps, significantly reduces the complexity and improves the convergence of SalTR. When the number of keys K is relatively small, the complexity of this module becomes linear with respect to the spatial size, making it efficient and scalable for large images and helping the overall convergence. Hence, we adapt the deformable attention mechanism, termed Deformable SalTR, to our settings.

Evaluation setting. We compare against the SoTA methods listed in [Wang et al. 277, Kümmerer et al. 152]. Moreover, we test on the MIT1300 benchmark [Judd et al. 135], which is more challenging than the Salicon test set. As suggested in [Kümmerer et al. 152, Bylinskii et al. 36], we use the following evaluation metrics: Similarity Metric (SIM), shuffled AUC (s-AUC), Linear Correlation Coefficient (CC), Normalized Scanpath Saliency (NSS), and the Kullback Leibler Divergence (KLD) [Bylinskii et al. 36]. We adopt the more recent metrics formulation from [Kümmerer et al. 152].

Technical details. SalTR is implemented in PyTorch [Paszke et al. 216] and trained using a single NVidia RTX3090 24GB GPU. We consider two configurations: SMALL with 3 transformer encoder-decoder layers and BASE with 6 layers, both with a ResNet-50 backbone. The number of fixation queries is set to 100. Hence, 100 fixation points are sampled from the ground truth fixation map. SalTR variants are trained for 100 epochs using the AdamW [Loshchilov and Hutter 179] optimizer, whereas the Deformable SalTR is trained for 40 epochs. We employ a warmup of 10 epochs and a Cosine learning rate scheduler with a maximum lr set to 10^{-3}.

4.4 Results

State-of-the-art comparison. We compare the proposed approach to SoTA image saliency models on the Salicon and MIT300 validation sets. Table 4.1 shows the performance comparison in terms of the five metrics for the respective validation sets.

We observe that our method performs favorably against existing approaches. Table 4.1 offers a detailed comparative analysis, revealing how various models, including the proposed method, fare on metrics. Certain models display higher performance in specific metrics, indicating their strengths. For instance, EML-NET and MSI-Net showcase high scores in SIM and CC on Salicon, while Def-SalTR-Base excels across multiple metrics in MIT1300 and CAT2000 datasets.

Salicon. SalTR-Base achieves similar scores to UNISAL, whereas the SalTR-Small showed a slightly worse performance, highlighting the importance of the transformer decoder depth to optimize appropriately for the unstable bipartite matching. Using the same amount of compute, Deformable SalTR-Small matches the UNISAL performance due to the effective gated attention mechanism allowing for a smoother optimization. Conversely, Deformable SalTR-Base exhibits SoTA scores across the five metrics on par with the best model in these test examples.

MIT300. While the model was not trained on this dataset. The proposed models produce excellent accuracy scores compared to other models, except UNISAL and DeepGaze, which were trained on this specific dataset.

Figure 4.2 illustrates the predictions on a sample of the Salicon validation set. It can be seen that the fixation maps generated by our model (SalTR-Base) correlate well with the ground truth fixation maps in terms of fixation distribution. Also, we smooth our fixation maps with a Gaussian filter to obtain the continuous saliency map, giving the approach the flexibility to set the standard deviation parameters to fit the downstream application. This demonstrates the effectiveness of the fixation queries in playing the role of human subjects.

Furthermore, we visualize the decoder self-attention maps for a set of a randomly selected queries with their respective fixation predictions in Figure 4.6. We notice that each query's attention is highly local, where it attends to granular details of the image (such as the Giraffe's mouth for query 13 and the person's leg for query 36). We believe that given well-structured keys and values from the encoder latent representations; the queries focus on the refined granularities to predict the fixation

points. This design is intuitive and is the closest to the experimental setting in obtaining the saliency datasets.

4.4.1 Saliency for low-level features.

State-of-the-art saliency models effectively capture high-level features such as cars and humans. However, these approaches may not adequately capture other crucial features that describe aspects of human visual attention, which have been extensively investigated in neuroscience. Visual search is one of the most prominent processes shaping human attention [Treisman and Gelade 266, Kotseruba et al. 147]. It involves the brain's parallel processing of regions that differ significantly in one feature dimension, such as color, intensity, or orientation. These are considered low-level features which serve as the basic mechanisms of the human visual system. We evaluated the performance of UNISAL and our SalTR model on samples of low-level attention using images from a recently proposed dataset [Kotseruba et al. 147]. The aim is to understand the main differences in how saliency exploration is performed when the self-attention mechanism promotes global connectivity between the image patches.

The model's performance on real images is quite commendable, as illustrated in Figure 4.3. It successfully identifies the targeted feature and assigns a higher saliency density to it while also maintaining attention to other regions in the image, demonstrating an acceptable distribution level in its focus. However, defining an ideal saliency map for such images is complex. It is generally agreed that humans initially focus on the most visually prominent or 'bottom-up' features within the first moments of observation. Subsequently, exploratory eye movements typically begin, covering other areas of interest in the image. Capturing this dual-phase attention process is a challenging task that current models have yet to master comprehensively.

On the other hand, our model's performance on synthetic images, as displayed in Figure 4.4, is noticeably weaker. The model struggles with understanding and interpreting the abstract features typically presented in synthetic imagery. This

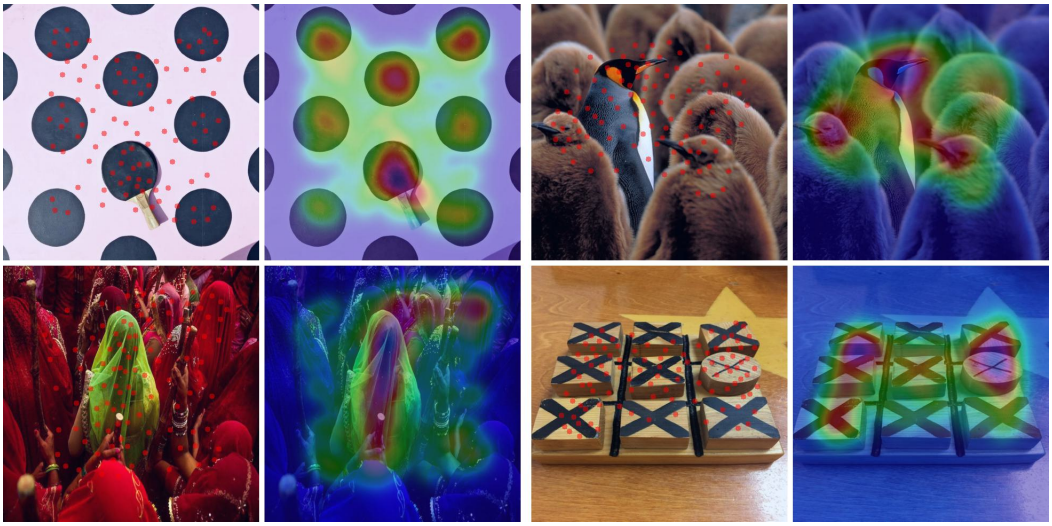


Figure 4.3: Predictions on the low-level features P3 dataset.

shortcoming is possibly attributed to the learning process. Specifically, the Salicon training set, which our model was trained on, does not include any synthetic images. The model's ability to generalize from the real images in the training set to synthetic images in the test set appears to be a significant hurdle.

In light of these observations, future efforts might focus on including a more diverse range of image types in the training set, particularly synthetic images. Furthermore, more research is needed to improve the model's ability to mimic the human attention process more accurately, especially regarding the transition from bottom-up to exploratory attention. An interesting question is whether the fixation queries in SalTR could bridge this gap by reasoning over the recurring features and drawing dependencies/similarities.

4.4.2 Failure cases.

In an effort to delve deeper into the capabilities of SalTR, we have identified certain anomalies within the Salicon validation set, specifically images marked by the highest Kullback-Leibler Divergence (KLD) value of 1.51 and the lowest Normalized Scanpath Saliency (NSS) score of 1.09. As visualized in Figure 4.5, we have pinpointed two areas where the model did not perform as expected, which we refer to as *failure modes*. In the top image, the model fails to accurately focus the prediction around

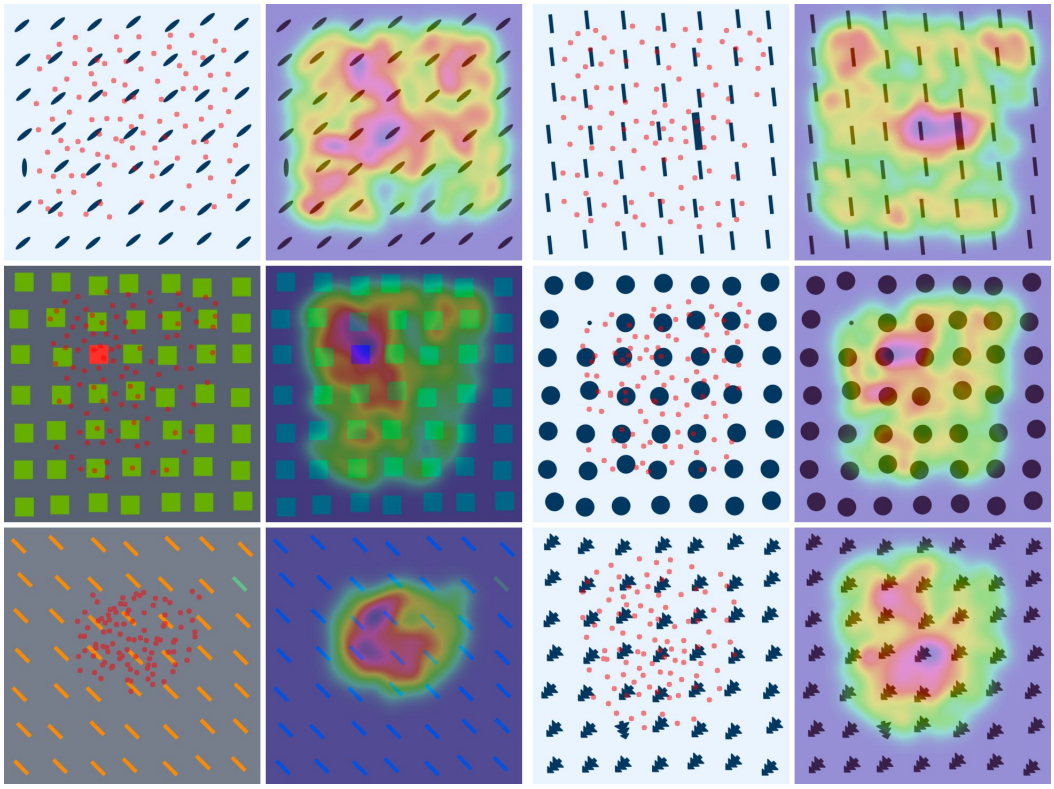


Figure 4.4: Predictions on the low-level features O3 dataset.

the human, which is considered a small object in the scene. The lack of hierarchical decoding in our framework may contribute to this failure, suggesting the need for decoding multiple resolutions of the latent representations to allow SalTR to adapt to fixation regions of varying scales. A second observed failure occurs in the last image, where the Ground Truth fixations are concentrated around the rabbit in the center. The Hungarian matching algorithm, which forces distinct predictions, tends to favor more dispersed predictions. The decoder typically mitigates this behavior when the input image lacks other potential salient regions. However, when the decoder identifies other informative areas in such scenarios, it distributes fixations uniformly across these regions.

4.5 Ablation study

Losses importance. The bipartite matching loss appears intuitive to avoid fixation queries collapse. We validate this hypothesis by eliminating the Hungarian matching

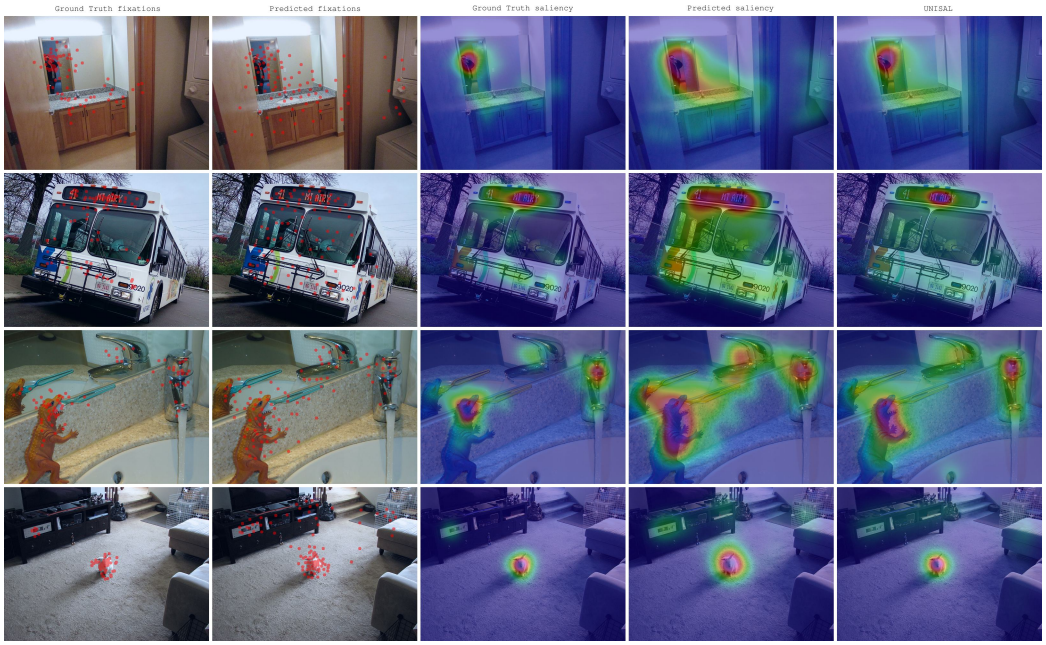


Figure 4.5: Images from the Salicon validation set where SalTR obtains high KLD scores.

Table 4.1: Comparative performance study on Salicon, MIT1300 and CAT2000.

Models	Salicon					MIT1300					CAT2000				
	SIM	s-AUC	CC	NSS	KLD	SIM	s-AUC	CC	NSS	KLD	SIM	s-AUC	CC	NSS	KLD
ITTI [Itti et al. 124]	0.37	0.61	0.20	–	–	0.46	0.13	0.44	1.11	0.95	0.45	0.59	0.43	1.19	1.02
GBVS [Harel et al. 107]	0.44	0.63	0.42	–	–	0.48	0.62	0.47	1.24	0.88	0.45	0.58	0.44	1.23	1.01
Salicon [Jiang et al. 130]	–	–	–	–	–	0.51	0.73	0.56	1.70	0.78	0.65	0.64	0.80	1.68	0.62
CASNet [Fan et al. 87]		–	–	–	–	0.58	0.73	0.70	1.98	0.58	–	–	–	–	–
EML-NET [Jia and Bruce 128]	0.79	0.74	0.89	2.05	**0.52**	**0.74**	0.67	0.78	2.48	0.84	–	–	–	–	–
MSI-Net [Kroner et al. 149]	**0.80**	0.74	**0.90**	2.01		0.67	0.74	0.77	2.30	0.42	–	–	–	–	–
TranSalNet [Lou et al. 180]	–	–	–	–	–	0.68	0.74	**0.80**	2.41	1.01	–	–	–	–	–
SalGAN [Pan et al. 212]		0.75	0.76	**2.47**	-	0.63	0.73	0.67	1.86	0.75	0.54	0.64	0.57	1.46	0.94
UNISAL [Droste et al. 79]	0.77	0.73	0.87	1.95	-	0.67	0.78	0.78	2.36	0.41	–	–	–	–	–
DeepGaze [Kummerer et al. 155]	–	–	–	–	–	0.66	0.77	0.77	2.33	0.42	**0.69**	0.65	**0.80**	1.96	0.38
SalTR-Small	0.75	0.71	0.84	1.79	0.98	0.61	0.70	0.70	1.98	0.59	0.67	0.64	0.76	1.81	0.48
SalTR-Base	0.78	0.75	0.87	1.97	0.74	0.65	0.74	0.75	2.22	0.46	0.68	0.66	0.79	1.91	0.40
Def-SalTR-Small	0.77	0.73	0.86	1.88	0.82	0.64	0.75	0.76	2.09	0.49	0.66	0.65	0.79	1.80	0.42
Def-SalTR-Base	0.79	**0.77**	0.89	2.12	0.62	0.69	**0.79**	**0.80**	**2.45**	**0.40**	0.66	**0.67**	**0.80**	**1.98**	**0.36**

loss and only use the final loss in Equation 4.3 without any assignments. As expected, the model learns the saliency dataset center bias. In other words. the fixation queries tend to all focus on the most salient features of the input image while ignoring the rest. This is an artifact of the cross-attention optimization, where a shortcut over the features dominates learning more diverse and rich predictions. Figure 4.7 illustrates the results of the SalTR model trained without incorporating the Hungarian matching loss. As can be observed, the predictions lack diversity and are all centered around the same region, akin to a duplicated query trained for 100 epochs. The lack of variety in the predicted regions suggests that the model picks a specific feature or

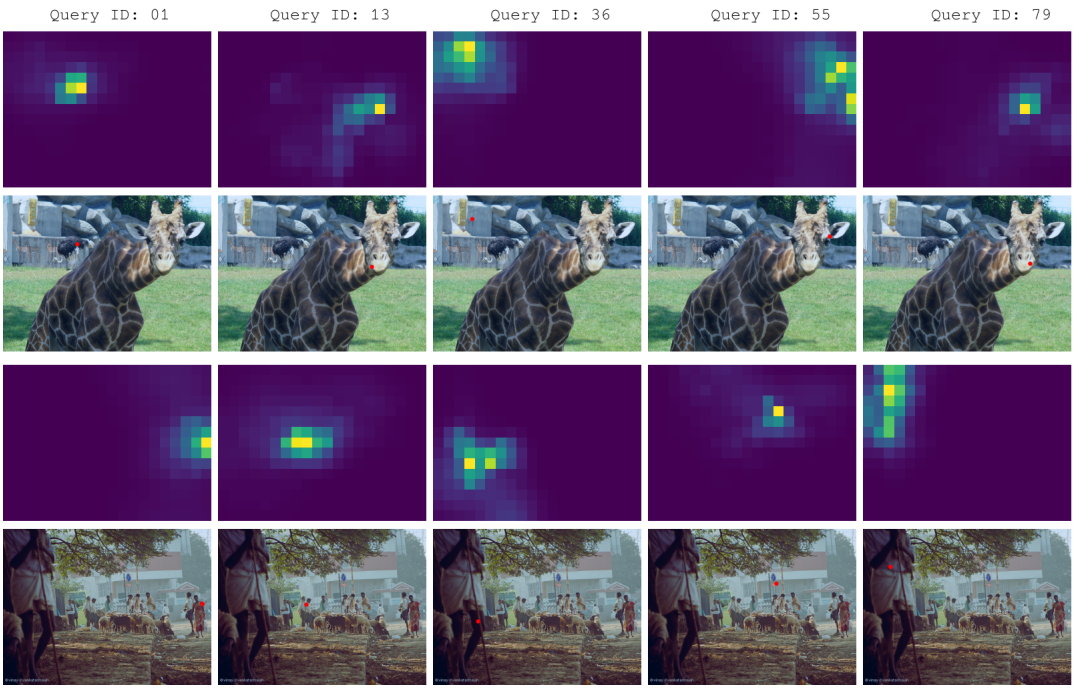


Figure 4.6: Decoder self-attention for a set of fixation queries. It can be seen that queries attend to spatial locations consistent with their predicted fixation points. Predictions are made with SalTR-Base on a validation set of images.

set of features in the training data. This behavior strongly suggests the crucial role of the Hungarian matching loss in facilitating SalTR's ability to make diverse predictions. Without it, the model's ability to generalize well across varying input data seems to be significantly hindered. A training duration of 100 epochs does not alleviate the observed issue.

Object Detection vs. Saliency Prediction. We attempted to initialize the SalTR model with DETR weights, which had been trained on the COCO dataset for object detection. We only trained the MLP head on the saliency dataset with the aim of measuring the alignment between the two downstream tasks. The model could not achieve the baseline scores (i.e., KLD: 2.5, NSS: 0.9). We observed that the fixations were mostly overestimated around the centers of the objects, suggesting a strong bias towards object-centric areas.

While object detection frameworks like DETR identify objects as regions of interest, saliency prediction often targets broader perceptual cues that draw human attention. Although objects can be salient, the most salient regions in an image are

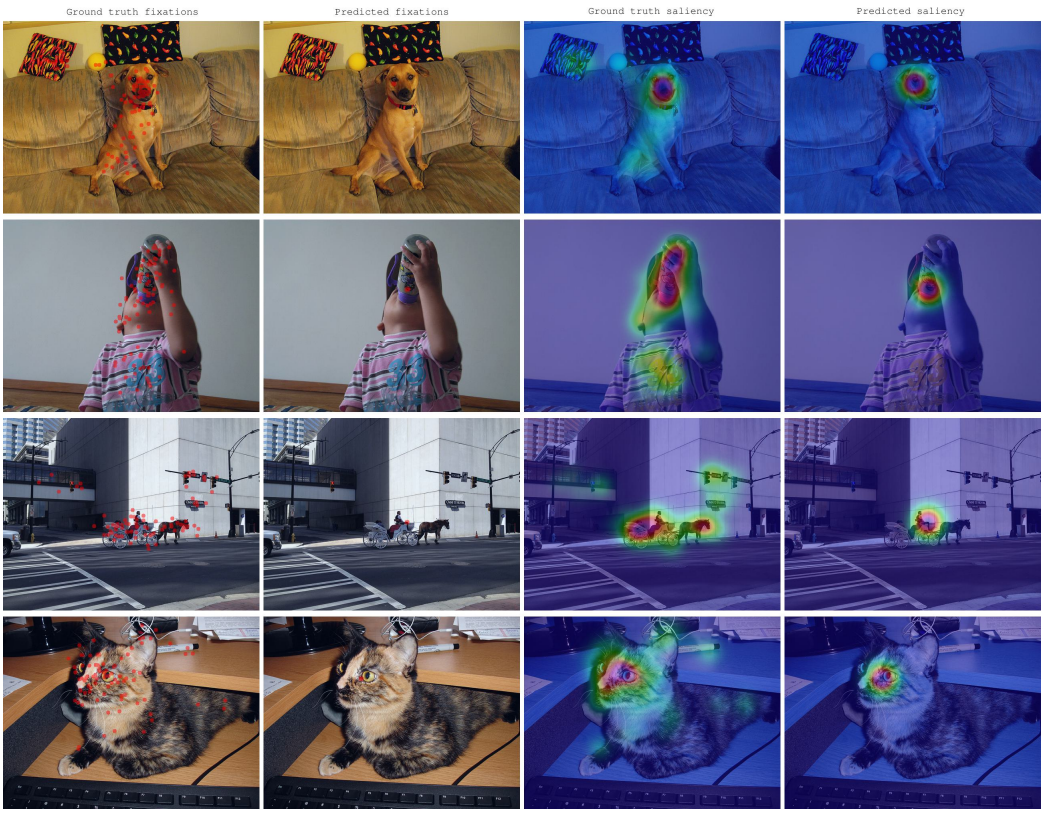


Figure 4.7: Images from the Salicon validation set where the model was trained without any Hungarian matching loss.

not always aligned with objects. They could instead be distinct features or patterns that catch the viewer's attention, as discussed by [Borji and Itti 26]. This distinction likely contributed to the underperformance of the SalTR model when initialized with DETR weights, as the model's focus on object centers did not necessarily correlate with the actual distribution of human fixations across various types of salient features. This discrepancy highlights the fundamental differences in the goals of object detection and saliency prediction, explaining the failure of this setup.

Varying the number of fixation queries. We investigate the optimal number of target fixations for better results. As shown in Table 4.2, we vary the number of queries N, hence the number of target fixations S. We observe that 100 is the optimal value, and higher values result in diminishing returns in the performance since a higher number of queries makes the optimization harder.

The impact of the Gaussian smoothing. Table 4.3 represents the impact of the standard deviation (σ) of a Gaussian filter applied to the fixation map for

Table 4.2: Impact of the number of fixation queries. Performance comparison when it is varied. 100 is the optimal number of queries for the saliency task. SalTR-Base is shown.

SalTR		Salicon				
		SIM	s-AUC	CC	NSS	KLD
Number of queries	50	0.75	0.70	0.82	1.55	1.04
	100	0.78	0.75	0.87	1.97	0.74
	150	0.79	0.74	0.88	1.92	0.71
	200	0.78	0.74	0.86	1.94	0.78

the SalTR-Base model. It can be observed that the $\sigma = 19.0$ generally offers the best performance across most metrics. Notably, the model achieves the highest SIM, CC, and NSS scores at $\sigma = 19.0$. Furthermore, we can observe that the KLD is the metric highly affected by the smoothing parameters, whereas the CC is more or less the same after the 19.0 value.

All vs single subject. To the best of our knowledge, SalTR is the first approach for learning saliency prediction from fixations only. However, it still does not fully replicate the experimental setting, as the fixation queries predict a single location, whereas human subjects may attend to multiple locations. We attempted to sample a single subject as a target so all the queries simulate a single human with multiple predicted fixations. This design resulted in visually appealing saliency maps, but the scores did not match the baselines as the predictions were mostly sparse. This potentially highlights an issue with the metrics requiring an over-estimated saliency map to match the ground truth map aggregated over a batch of subjects.

4.6 SalTR for scanpath prediction

Scanpath prediction refers to the process of estimating the timely trajectory of fixation points for a human subject when viewing a visual stimulus. The SalTR design allows for manipulating the Transformer decoder mask. Indeed, we used the full mask for parallel decoding previously; clearly, however, we can use a causal mask and add an end-of-sequence (i.e., position $(0, 0)$) to the target fixations for the

Table 4.3: Impact of the number standard deviation. Performance comparison when the Gaussian smoothing parameter is varied. SalTR-Base is shown.

	Salicon				
	SIM	s-AUC	CC	NSS	KLD
5.0	0.42	0.50	0.54	1.33	1.63
10.0	066	0.70	0.77	1.77	0.99
19.0	0.78	0.75	0.87	1.97	0.74
30.0	0.76	0.72	0.87	1.78	1.16

auto-regressive decoding. By doing so, SalTR can naturally handle ordered fixation predictions (i.e. scanpaths). The prediction head outputs the fixation points and their duration in seconds. We train SalTR-Base using the 10k Salicon images; for each image, we select all the human subject's scanpaths, each serving as its own separate training example. The model should converge on the correct distribution over scanpaths given the image. We finetune the decoder only for 20 epochs.

Results. The Multi-Match (MM) [Dewhurst et al. 70] measure is used as the main metric for ranking scanpath prediction models. Our approach obtains an MM score of 0.93 on the 5k Salicon examples (PathGAN [Assens et al. 4]: 0.96). This is achieved by averaging the individual results against all human viewer scanpaths for a single image. Figure 4.8 shows the quantitative results on sample images from Salicon. Our model demonstrates a clear and consistent tracking of the ground truth, indicating its ability to capture human visual attention accurately.

4.7 Ideas on how SalTR can be extended to satisfy more constraints on saliency

As elaborated in Section 2.6 of Chapter 2, the complexity of visual attention is often underrepresented in existing saliency datasets. A critical issue is that these datasets typically compile fixations over an extended period, blending bottom-up (stimulus-driven) and top-down (goal-directed) influences on saliency. Learning from such data pushes models not to distinguish mechanisms that govern human attention,

as the datasets fail to differentiate between the spontaneous, reflexive responses to visual stimuli and the search-oriented engagements that characterize top-down processing. Consequently, the resulting data provide a diluted representation of attention, complicating the task of developing saliency models replicating human visual behavior. This entanglement presents significant challenges in saliency modeling, an issue that merits thorough investigation and analysis.

The model we presented in this chapter, SalTR, offers a level of flexibility by having control over the predictions using the fixation queries. However, an extension could involve incorporating temporal dynamics directly into the fixation predictions to refine this approach further and address the entanglement of bottom-up and top-down saliency processes. By integrating time stamps with fixation points, the model could differentiate between reflexive responses to sudden stimuli and prolonged fixations driven by cognitive assessments or search tasks. This temporal aspect would allow the model to account for the dynamism in human attention, offering a more granular understanding and prediction of salient areas in visual scenes that satisfy the known hypothesis about the task in neuroscience.

Moreover, leveraging a specialized set of queries for different saliency types could enhance this aspect. One could design a set of specialized queries, categorized by their respective attentional focus—whether bottom-up or top-down could be dynamically selected through a mechanism akin to a mixture of experts [Shazeer et al. 247]. Each query type would be expertly tuned to capture specific aspects of visual attention, such as sudden changes in the visual field or areas of prolonged interest that might relate to the observer's intentions or task objectives.

Implementing these extensions would require modifications to the existing architecture of SalTR. The decoder, for instance, could be adapted to handle multiple sets of queries, each set trained to specialize per property of visual attention. Then, potentially, another neural network layer or a gating mechanism [Lee et al. 164] would determine each query type's contribution to the final prediction, effectively allowing the model to choose the most relevant features for any given visual input.

This approach not only mimics the complex interplay of various cognitive processes involved in human visual attention, but it also has the potential to enhance the model's utility across different applications where understanding and investigating saliency predictions is required.

4.8 Conclusion

We present a novel approach for saliency prediction in images, named Saliency TRansformer (SalTR) [Djilali et al. 74], which leverages parallel decoding in transformers to learn saliency solely from fixation maps. Unlike existing models that rely on continuous saliency maps, our approach directly predicts fixations by treating saliency prediction as a set prediction problem. We conducted experiments on the Salicon and MIT300 benchmarks and achieved remarkable performance compared to SoTA methods. Our approach not only replicates the data collection pipeline used in generating saliency datasets but also eliminates the need for continuous saliency annotations. Furthermore, we extended our approach to the scanpaths prediction problem and demonstrated its effectiveness. Overall, our approach offers a promising direction for saliency prediction, focusing on the discrete fixation maps and directly predicting fixation points. It opens up possibilities for application-guided saliency prediction, as per the flexibility offered in our design.

Given the promising performance of deep learning models on saliency for planar data, a natural question to ask is whether this extends to the 360° images/videos given their geometrical properties and the different inductive biases of the human visual system. We will explore this setting — and related questions — in the second part of this thesis.

Figure 4.8: Scanpaths for a set of images from the Salicon validation set.

Part II —

Learning with 360° data

MOTIVATION AND SUMMARY

The world around us and our understanding of it is not restricted to 2D data: human visual attention is developed in the 360° scenes, where we can turn our heads and compile our understanding of the world from different parts of the scene. This represents the extended definition of saliency beyond a fixed viewport, where the sensory observation is positioned in the 3D space.

This part of the thesis explores how we can develop models capable of capturing subjects' head and eye movements within 360° scenes.

In Chapter 5, we target explicitly incorporating the statistical properties of 360° data in the model architecture and designing a framework that captures these characteristics.

Chapter 6 aims to implicitly capture the statistical properties at the learning stage by having a general representation of 360 data using self-supervised techniques.

In the final chapter of the thesis, Chapter 7, the interest is around better evaluating 360 saliency° models where the objective is to design suitable metrics for this domain to better compare models.

Chapter 5

Saliency for 360° Images/Videos

5.1 Introduction

Having explored modeling saliency for planar 2D data in Chapters 3 and 4, we now extend our investigation to the domain of 360° data in this chapter. The rising influence of virtual reality environments, augmented reality applications, and panoramic scenes underscore the importance of 360° data as a significant modality. Transitioning from flat, planar 2D imagery to 360° data essentially stretches the existing dimensions of width and height across a sphere, enabling a full panoramic view for the human subjects.

Unlike the constrained field of view offered by 2D data, 360° content surrounds the viewer, demanding attention from all directions and offering a comprehensive immersive experience. This type of content is mapped onto a sphere with a full 360-degree azimuth and typically a 180-degree elevation range, which covers from the nadir to the zenith. However, this expansive nature poses unique challenges in predicting visual saliency. Traditional 2D data might assume viewers' attention is primarily directed toward the center or other predefined regions of interest. In contrast, 360° data allows viewers to look anywhere along the extended dimensions of width and height, now wrapped into a spherical format. This removes the usual

This chapter is based on our ICPR 2021 publication [Dahou et al. 61]

biases present in planar views and introduces complexity in predicting where a viewer might direct their gaze. Hence, predicting visual attention in such environments requires a reevaluation of the assumptions and methodologies traditionally applied in 2D saliency modeling.

In this space, understanding and predicting the viewer's head and eye movements is a critical task. Unlike conventional 2D media, which consider eye movements over a fixed field-of-view (FOV), 360° environments require accounting for both head and eye movements. This necessity arises because the expansive field of view offered by 360° content often exceeds the eye's effective field of view, compelling viewers to rotate their heads to explore extensive areas of the content. Therefore, predicting saliency in such environments necessitates an integrated approach that considers these two components of motion.

Head movements, which are generally slower and more deliberate, provide indications about broader areas of interest or overarching scene narratives. In contrast, eye movements in 360° settings typically involve both quick saccades and more focused fixations. Saccades are rapid, ballistic movements of the eyes that quickly shift the point of fixation from one part of the visual field to another, while fixations are periods when the eyes are relatively stationary and detailed information is gathered about the object of focus. Distinguishing between these types of eye movements is crucial for applications like virtual reality (VR), where head movements can control the orientation of the camera and eye-tracking data can fine-tune the details within the viewer's current field of view.

Accurately predicting these movements not only enhances the viewer's immersive experience but also optimizes content delivery by dynamically adjusting the resolution or streaming rates based on the predicted regions of interest. Furthermore, one of the principal challenges associated with 360° data is its representation. Unlike the straightforward Cartesian grid layout of 2D data, 360° content cannot be directly represented on a flat plane without some form of transformation. This necessitates specific projection methods to map the spherical data onto a 2D plane for processing.

Additionally, conventional convolutions, which are fundamental to many deep learning architectures such as CNNs, inherently assume a local, grid-like neighborhood (usually a small square in images). However, on a spherical surface, this locality is distorted due to the curvature of the sphere.

To effectively perform convolutional operations on spherical data, it is necessary to redefine these operations to take into account the spherical geometry. This requirement complicates the design and potentially affects the efficiency of the network, as standard convolutional neural network architectures are not naturally equipped to handle the distortions introduced by projecting spherical data onto a plane. In the next section, we will discuss the two widely used projection methods that address these challenges, which are adapting the data layout and convolution operations to preserve spatial relationships on a sphere.

From a modeling perspective, as stated in Chapter 2, saliency prediction studies on 360° video/image are still limited. This is in part due to the comparatively small amount of research that has investigated the visual attention features that affect human perception in panoramic scenes. Also, as shown by [Rai et al. 230, Xu et al. 286, Xu et al. 285], eye-gaze fixations are quasi-isotropically distributed in orientation, with stronger statistical biases. In fact, human attention on 360° video is biased toward the equator and the front region, known as the equator bias, which could be leveraged prior in modeling. Along with the statistical bias, a subject's attention is driven by the most salient objects in the scene. It has been shown that a smaller number of closer objects capture more human attention [Sitzmann et al. 250].

We attempt to explore the effect of these statistical biases and priors about the 360° saliency. The proposed approach, ATSal combines global visual features over the entire view with features derived from local patches in consecutive frames. At the time of the publication of this work, ATSal results in state-of-the-art accuracy on standard benchmarks. We investigate the details of predicting saliency in 360° data, considering the aforementioned challenges and evaluating how established saliency models for 2D data might be adapted or need rethinking for this domain.

5.2 Background

Spherical data requires specialized projection methods to accurately represent it on a two-dimensional plane because the surface of a sphere cannot be directly mapped onto a flat surface without distortion. The reason one can't directly apply conventional convolution operations on spherical data is due to the non-Euclidean nature of a sphere's surface. In standard convolution, as used in image processing or Convolutional Neural Networks, the operation assumes a grid, Euclidean geometry where the pixels or data points are regularly spaced. However, on a spherical surface, this regular spacing is not maintained, leading to distortions and inaccuracies if standard convolution is applied directly (i.e.filter deformation).

Hence, two widely adopted projection methods in the domain are equirectangular projection and cubemap projection. Equirectangular and cubemap projections tackle this issue by transforming the spherical data into a format more suitable for standard processing techniques. Equirectangular projection maps the sphere onto a rectangle, maintaining angles but distorting areas and distances, especially near the poles. Cubemap projection, on the other hand, maps the sphere onto the six faces of a cube, which can then be unfolded into a flat surface. This method tends to better preserve local shapes and reduces distortion compared to equirectangular projection, but it introduces crops where the cube faces meet. In essence, these projections are necessary to adapt spherical data into a form where conventional image processing techniques, like convolution, can be effectively and accurately applied without the complexities and distortions introduced by the sphere's curvature. We cover the details of these projection methods in this section.

5.2.1 Equirectangular Projection (ERP)

As depicted in Figure 5.1, the equirectangular projection method linearly maps the longitude (ϕ) and latitude (θ) of a point on a sphere to the x and y coordinates on a 2D plane [Snyder 251]. This projection unfolds the sphere onto a rectangle,

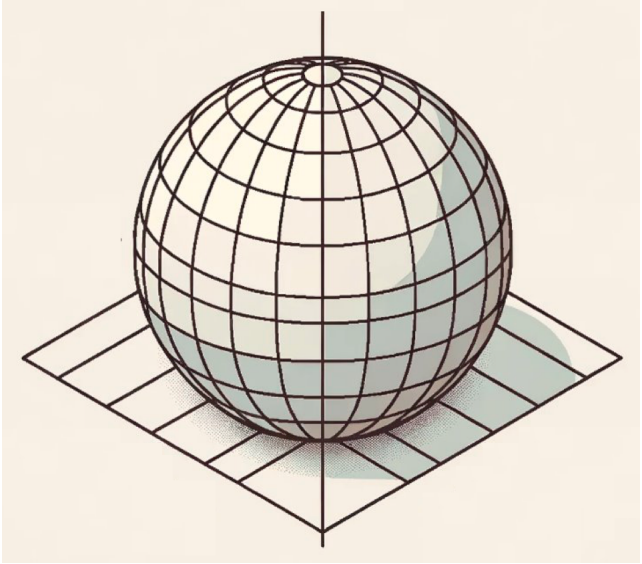


Figure 5.1: A central unit sphere next to its corresponding rectangular 2D representation. Straight lines demonstrate the mapping from the sphere to the rectangle.

maintaining constant spacing between lines of latitude and longitude but varying the spacing between points on those lines. The mapping is defined by the following equations:

$$x = \frac{w}{2\pi}\phi$$
$$y = \frac{h}{\pi}(\theta + \frac{\pi}{2})$$

where:

- w is the width of the 2D plane (or image), representing the complete 360° of longitude.
- h is the height of the 2D plane (or image), representing 180° of latitude from the North Pole to the South Pole.

The point P on the sphere, with a latitude θ (ranging from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$) and a longitude ϕ (ranging from $-\pi$ to π), is thus mapped to a Cartesian grid on the 2D plane.

The equirectangular projection, one of the simplest forms of map projection, is valued for its computational efficiency and simplicity. It has been widely used in

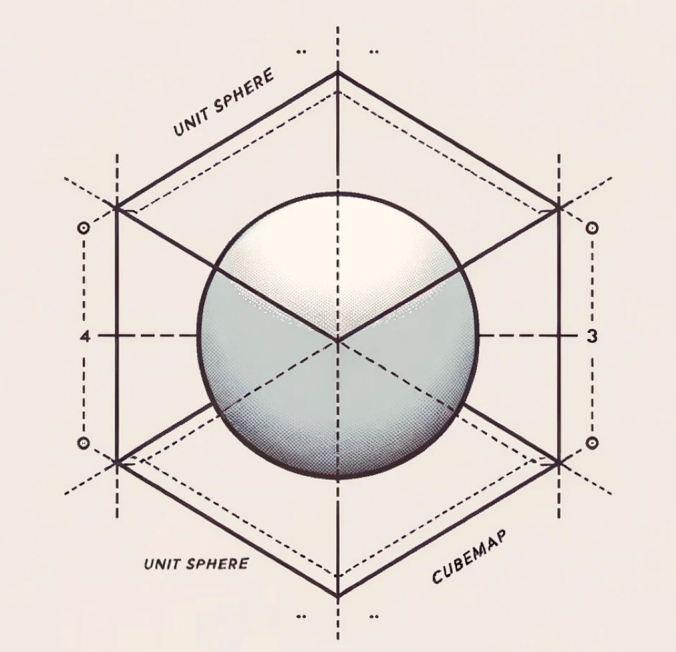


Figure 5.2: A vector illustration demonstrating the mapping from the unit sphere to the cube's faces in the Cubemap Projection (CMP).

applications such as panoramic photography, VR content creation, and geographical information systems. Despite its utility, the equirectangular projection does introduce significant distortion, particularly at the poles. This distortion is due to the projection's preservation of angles, which results in a stretching effect as one moves towards the poles. A small change in latitude near the poles can result in a disproportionately large change in the vertical position on the mapped 2D plane.

5.2.2 Cubemap Projection (CMP)

The cubemap projection method projects the spherical domain onto the six faces of a surrounding cube (see Figure 5.2) [Snyder 251]. It's a versatile representation commonly employed in computer graphics, particularly in environment mapping, where reflection calculations require a panoramic view of the surroundings.

Imagine a unit sphere centered at the origin of a 3D coordinate system. A point P on this sphere can be represented by its Cartesian coordinates (X, Y, Z) or by its spherical coordinates (r, θ, ϕ), where r is the radius (which is 1 for a unit sphere), θ is the elevation angle (ranging from the pole at $-\frac{\pi}{2}$ to the pole at $\frac{\pi}{2}$), and ϕ is the

azimuthal angle (longitude).

In the cubemap projection, this sphere is enclosed within a cube. Each point on the sphere maps to one of the six faces of the cube. The vector from the center of the sphere to a point P intersects a point Q on one of the cube's faces. For the positive X face (right) of the cube:

$$\begin{aligned} X &= 1 \\ U &= \frac{Z}{X} \\ V &= \frac{-Y}{X} \end{aligned}$$

Similarly, for the negative X face (left):

$$\begin{aligned} X &= -1 \\ U &= \frac{-Z}{X} \\ V &= \frac{-Y}{X} \end{aligned}$$

Equivalent calculations apply for the Y and Z faces. The U, V coordinates for a specific face are then rescaled and transformed to map them to the pixel grid of that face's 2D representation. If each face's image has a resolution of $w \times w$, the mapping is given by:

$$\begin{aligned} u_{\text{pixel}} &= \frac{w}{2}(U + 1) \\ v_{\text{pixel}} &= \frac{w}{2}(V + 1) \end{aligned}$$

u_{pixel} and v_{pixel} denote the pixel coordinates on the 2D image of the cube face.

A significant advantage of the cubemap projection is its more uniform sampling of the sphere compared to equirectangular projection. This reduces distortion, particularly in regions near the poles. However, the cubemap format does have challenges, such as the seams at the edges where the cube faces meet. To address

Figure 5.3: Source [Patney et al. 220]. A classroom scene with eye fixation at the yellow reticle. (Left) The perceptually validated target foveated image. (Right) Their proposed foveated rendering system avoids shading up to 70% of the pixels and closely matches the frequency content of our target by using pre-filtered shading terms, contrasts preservation, and applying a new temporal antialiasing that improves temporal stability by order of magnitude (providing stability similar to a temporally antialiased non-foveated renderer). The original version of the classroom scene is courtesy of Christophe Seux.

these, boundary conditions can be implemented, allowing for continuity and restoring locality at the cube's edges. These conditions enable seamless transitions across faces, which is crucial for preserving the visual integrity of features that span multiple faces. Despite this, certain operations, like rotations and data augmentations, can be more complex in a cubemap format than in a continuous 2D representation.

5.3 Why this is important?

Saliency detection in omnidirectional images holds significant promise for enhancing user experience and system performance across various domains. This section explores the diverse applications of saliency models.

5.3.1 Foveated Rendering

[Patney et al. 220] investigated the potential of foveated rendering to significantly enhance performance in virtual reality environments by reducing the graphical detail in the peripheral vision where human visual acuity is lower. The work details the implementation and evaluation of a foveated rendering system that aims to closely match the frequency content of target images through techniques such as pre-filtered

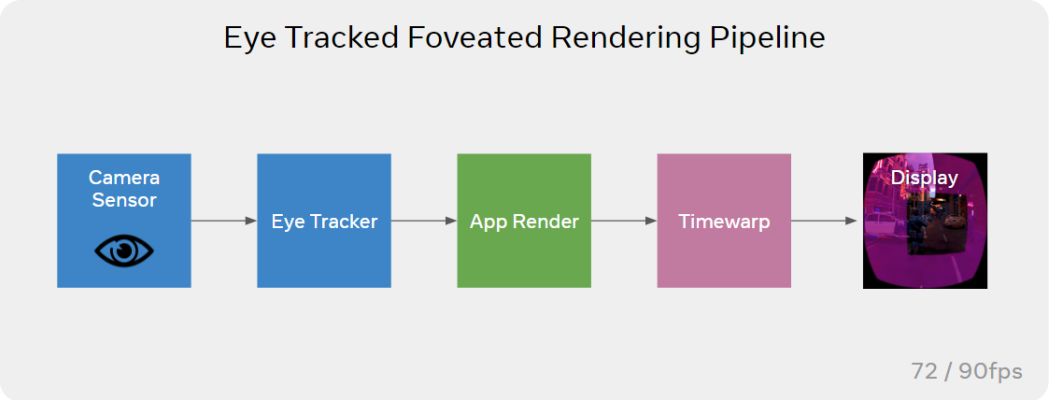


Figure 5.4: source. The common pipeline for using eye tracking for foveated rendering

shading terms, contrast preservation, and advanced temporal antialiasing to improve stability (see Figure 5.3).

The research involved a series of user studies to assess the perceptual impact of various foveation techniques and the effectiveness of the proposed rendering system. These studies confirmed that applying a post-process contrast enhancement enabled subjects to tolerate much larger blurs in peripheral vision without noticing differences from non-foveated rendering. Moreover, their work proposes a novel approach to foveated rendering that reduces shading computations by up to 70% while maintaining visual quality close to that of full-resolution rendering.

For many companies, Eye Tracked Foveated Rendering (ETFR) enhances GPU efficiency in virtual reality by dynamically adjusting the rendering resolution based on the user's gaze, tracked in real-time as shown in Figure 5.4. This method focuses on high-resolution rendering of the foveal region where the user is looking and reduces resolution in peripheral vision, resulting in significant GPU savings.

This example application demonstrates the usage of saliency for industrial settings. While we do not attempt to solve a direct application explicitly, the works proposed in this thesis can very well be applied to foveated rendering.

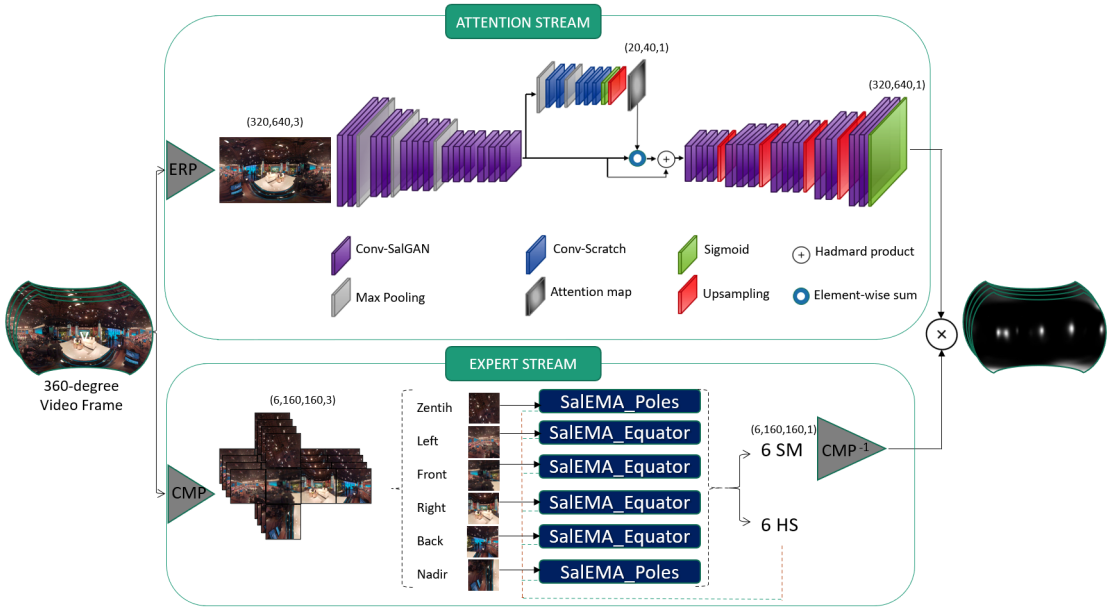


Figure 5.5: Network Architecture of ATSal. The attention stream is for extracting global saliency. The expert model operates on local patches. The final saliency is obtained by pixel-wise multiplication between the two maps generated by each stream.

5.4 Method

In this section, we describe the ATSal framework (see Figure 5.5). The design consists of an attention model, expert models, and the fusion method. The approach operates in two parallel streams that are fused later to obtain the final saliency map. One stream is dedicated to extracting global attention statistics via the attention mechanism applied to the ERP frame. This explicitly captures static saliency information, allowing the second stream to learn more effective temporal features. Indeed, the two-stream design is a tailored solution that leverages structured approaches to address known challenges in visual attention modeling, such as the need to separately and effectively process static and dynamic information.

Following the objective of capturing the inherent statistical properties of 360° data, experts are trained to specialize per region in the sphere. This incorporates two instances of SalEMA on the cube faces, **SalEMA_Poles** for the Zenith and Nadir, and **SalEMA_Equator**, specialized for the equator viewports. This approach is motivated by findings from research that investigated the properties of panoramic

scenes, which argued that the distribution of fixations is highly correlated with the locations of the viewports [Qiao et al. 224]. Thus, we adapted the state-of-the-art 2D saliency model SalEMA [Linardos et al. 173] weights to meet these specific requirements.

5.4.1 Attention mechanism

Despite the considerable research interest in saliency prediction for 360° stimuli, to the best of our knowledge, no previous studies have specifically leveraged the global attention effect inherent in omnidirectional scenes to predict saliency for both head and eye movements. To address this, we have implemented an attention mechanism within a parallel processing stream capable of encoding the global statistics of a 360° image or video. This is achieved through the introduction of a dense attention mask at the latent space of the network, which features an expanded receptive field to capture broader visual contexts. This attention mask is designed to process the feature map in a way that enhances the model's sensitivity to globally relevant visual cues. Specifically, the mask undergoes a pixel-wise multiplication with the feature map, integrating attentional focus directly into the spatial processing of the image. Subsequently, a pixel-wise summation across channels is performed to consolidate the feature map into a final latent variable representation.

Why an attention mechanism? Prior to Vision Transformers, many studies [Wang et al. 274, Yang et al. 289, Tao et al. 259, Chen et al. 47] demonstrated the effectiveness of attention mechanisms in various computer vision tasks, such as semantic segmentation. In these approaches, attention is adaptively learned in a joint and task-oriented manner, enabling the network to concentrate on the most relevant parts of the input space. In a similar manner, the latent attention module we implemented extracts robust spatial saliency information. The term "latent" here refers to the model's capability to utilize hidden representations. This module was trained on both the Salient360! [Rai et al. 229] and Sitzman 360 [Sitzmann et al. 250] image saliency datasets under direct supervision. Such training enhances the

model's ability to generalize prediction performance across different 360° saliency datasets (see Figure 5.5). Given an input ERP image $X \in \mathbb{R}^{640\times320\times3}$, this specific shape helps preserve the initial spherical characteristics of the data distribution, which has a ratio of 2:1. To effectively capture global attention, a larger receptive field is necessary to propagate each pixel-wise response across a field of 640×320. This condition is essential for ensuring that global attention spans the entirety of the image. The adoption of this approach is motivated by the need to emphasize and extract salient features that are integral yet not explicitly visible, enhancing the model's interpretive and predictive capabilities.

- As investigated in [Oliva and Torralba 206], human attention is driven by complementary pathways, which compete on global versus local features. Global features such as semantics, pose, and spatial configuration, called scene layout or "gist," guide early visual processing. In a panoramic scenario, the cube map prediction forces the model to lose the global contextual information while considering each face separately. Through the attention mechanism, the auto-encoder is encouraged to learn more explicit global static representations. The global and local features are disentangled for robust predictions.

- To ensure that the encoder's receptive field covers the entire input, we made several modifications to the VGG-16 architecture. Initially, the last two max pooling layers were removed to better preserve spatial frequencies, and the stride factor and kernel size of the max pooling layer-3 were changed from 2 to 4. These adjustments increased the initial VGG-16 receptive field from 212×212 to 244×244. However, this was still insufficient to cover the entire input $X \in \mathbb{R}^{640\times320\times3}$. We further extended the $conv5$ layer by integrating an attention mechanism that includes several dilated convolution layers interspersed with pooling and up-sampling operations. This modification enlarged the receptive field to 676×676. Given the latent variable:

$$z_1 = f_\theta(X) \in \mathbb{R}^{20\times40\times512}, \tag{5.1}$$

Table 5.1: Architecture of the attention mechanism

Layer (type)	Output shape	# Parameters
MaxPool2D_1	$1 \times 512 \times 10 \times 20$	0
Conv2D_1	$1 \times 64 \times 10 \times 20$	294,976
ReLU	$1 \times 64 \times 10 \times 20$	0
Conv2D_2	$1 \times 128 \times 10 \times 20$	73,856
ReLU	$1 \times 128 \times 10 \times 20$	0
MaxPool2D_2	$1 \times 128 \times 5 \times 10$	0
Conv2D_3	$1 \times 64 \times 5 \times 10$	73,792
ReLU	$1 \times 64 \times 5 \times 10$	0
Conv2D_4	$1 \times 128 \times 5 \times 10$	73,856
ReLU	$1 \times 128 \times 5 \times 10$	0
Conv2D_5	$1 \times 1 \times 5 \times 10$	129
Upsample	$1 \times 1 \times 20 \times 40$	0
Sigmoid	$1 \times 1 \times 20 \times 40$	0
Total Parameters: 516,609		

where f_θ is the encoder, the resulting attention mechanism produces a dense attention map $M \in [0,1]^{20\times40}$, effectively addressing the need for a more extensive receptive field to capture the full spatial extent of the input.

The attention module is activated with a sigmoid function, which relaxes the sum to one constraint often used in softmax-based neural attention. The feature map z_1 is further enhanced by:

$$z^c = (1 + M) \circ z_1^c, \tag{5.2}$$

where $c \in \{1, ..., 512\}$ is the channel index and M operates as a feature wise selector on z_1, with the residual connection to keep all the useful information [He et al. 111]. The optimal latent variable $z \in \mathbb{R}^{20\times40\times512}$ is fed to the decoder to learn the predicted saliency map $S_{p_1} \in \mathbb{R}^{640\times320}$. Table 5.1 summarizes the architecture of the attention mechanism.

5.4.2 Expert models

All previous works learn the same network parameters for the six faces of the cube. However, the saliency density is mostly represented in the equator, forcing the network to overestimate fixations near the poles, which is one of the main reasons the prediction performance drops. In short, the expert stream of the proposed framework instantiates two expert versions from SalEMA [Linardos et al. 173] and combines their results through the inverse projection to predict the final saliency. The main point of the expert stream is to predict 360° dynamic saliency viewports based on both spatiotemporal content and viewport location. Accordingly, each viewport is predicted independently with shared weights.

The input frame in ERP format X is first projected to cube map format resulting in $x_i, i \in \{1, ..6\}$. Where $x_{0,1,2,3}$ represents the front, left, right, and back views, and $x_{4,5}$ are the Zenith and Nadir views. The original SalEMA was trained on the DHF1K dataset using the binary cross entropy (BCE) and a SalGAN encoder-decoder with an added exponential moving average (EMA) recurrence. The exponential weighted average takes the state S_t output by a convolutional layer a time t. The output E_t is then propagated to the next layers using a convex combination of the state S_t and the previous outputs:

$$E_t = \alpha S_t + (1 - \alpha) E_{t-1}. \tag{5.3}$$

The initial α was set at 0.1. We further changed the training protocol of SalEMA by adjusting each corresponding cube face into a batch size of 80 for the poles while maintaining the initial batch size of 10 for the equators. This decision was influenced by computational considerations to fit the GPU memory, which necessitated a balance between batch size and effective learning. Additionally, Binary Cross-Entropy was retained as the objective function for fine-tuning SalEMA. This choice is supported by the relatively low motion present in each viewport, as discussed in [Bao et al. 12]. **SalEMA_Poles** was fine-tuned on batches of the faces $x_{4,5}$ to capture the underlying features on these viewports, while **SalEMA_Equator** was adapted to faces $x_{0,1,2,3}$.

Table 5.2: Attention model results on Salient360! validation images.

	Salient360!				
	AUC-J ↑	NSS ↑	CC ↑	SIM ↑	KLD ↓
Attention model	0.837	1.680	0.631	0.629	0.801

The local saliency for each cube face generated through this process is then inversely projected back into the ERP format using CMP^{-1}. We denote $S_{p_2} \in \mathbb{R}^{640\times320}$ as the saliency map generated after this inverse projection. The final saliency map, S_p, is obtained after performing pixel-wise multiplication of S_{p_1} and S_{p_2}:

$$S_p = S_{p_1} \circ S_{p_2} \in \left[0,1\right]^{640\times320}. \tag{5.4}$$

5.4.3 Loss function for the attention stream

According to [Bylinskii et al. 37], the saliency metrics cover different aspects of the saliency map. Thus, we define the loss function as a combination between the Kullback-Leibler Divergence (KLD) and the Normalized Scanpath Saliency (NSS). We denote the predicted saliency $S_{p_1} \in \left[0,1\right]^{640\times320}$, the fixation map as $\mathcal{M}_{GT} \in \{0,1\}^{640\times320}$, the dense mask at the middle of the bottleneck $M \in \left[0,1\right]^{20\times40}$, and the ground truth continuous saliency map obtained after blurring S_{GT} with a Gaussian filter ($\sigma = 9.35°$) as $S_{GT_1} \in \left[0,1\right]^{640\times320}$. $S_{GT_2} \in \left[0,1\right]^{40\times20}$ is the down-sampled version. The attention stream loss function is defined as follows:

$$\begin{aligned}\mathcal{L}_{\mathcal{T}}(S_{p_1}, M, \mathcal{M}_{GT}, S_{GT_1}, S_{GT_2}) = \alpha_1\mathcal{L}_{\text{KL}}(S_{p_1}, S_{GT_1}) + \\ \alpha_2\mathcal{L}_{\text{NSS}}(S_{p_1}, \mathcal{M}_{GT}) + \beta\mathcal{L}_{\text{KL}}(\mathcal{M}_{GT}, S_{GT_2})\end{aligned} \tag{5.5}$$

where $\alpha_2 = \beta = 0.2$, and $\alpha_2 = 0.8$ through some ablation studies. $\mathcal{L}_{\text{KL}}$ is chosen as the primary loss:

$$\mathcal{L}_{\text{KL}}(S_p, S_{GT}) = \sum_i S_{GT_i} \log\left(\epsilon + \frac{S_{GT_i}}{\epsilon + S_{p_i}}\right). \tag{5.6}$$

$\mathcal{L}_{\text{NSS}}$ is derived from the NSS metric, which is a similarity metric. We, therefore, optimize its negative:

$$\mathcal{L}_{NSS}(S_p, \mathcal{M}_{GT}) = -\frac{1}{N}\sum_i \bar{S_{p_i}} \times \mathcal{M}_{GT_i}, \tag{5.7}$$

where $N = \mathcal{M}_{GT}^T \mathbf{1}$ and $\bar{S_{p_i}} = S_{p_i} - \mu(S_{p_i}))/\sigma(S_{p_i})$.

5.5 Experiments

Training. The attention model was trained in two stages. First, to encode static saliency, we trained the attention model on 360° images from the Salient360! and Sitzman datasets. Due to the small amount of labeled static data (103 omnidirectional images), we applied some common data augmentation techniques: mirroring, rotations, and horizontal flipping [Cubuk et al. 59]. The resulting augmented dataset contains 2368 360° images. Table 5.2 shows the results for the Salient360! validation set. For dynamic stimuli, we consider two settings: training both streams on (i) Video-Salient360! and (ii) VR-EyeTracking. For Video-Salient360!, we used 15 omnidirectional videos for training and 4 for validation. For VR-EyeTracking, the training and testing sets include 140 and 75 videos, respectively. We evaluate our model on the test set of VR-EyeTracking and the validation sets[1] (images and videos) of Salient360!, in total 79 omnidirectional videos with over 75,000 frames.

Competitors. ATSal is compared with seven models. Three state-of-the-art 2D video saliency models: UNISAL [Droste et al. 79], 3DSAL [DAHOU DJILALI et al. 62], SalEMA [Linardos et al. 173], and four 360° specialized models: U-NET [Zhang et al. 296], CP360 [Cheng et al. 50], SalGAN360 [Chao et al. 43], and Two-Stream [Zhang and Chen 293]. This choice was motivated by the availability of the code base and model weights. The U-NETs inference protocol was, however, not discussed in the original paper since the authors use the teacher forcing technique to train the model. Here, we use the predicted saliency maps at $t-5$ as input to

[1] due to the test set being unavailable because of COVID-19

the model at a time t. All models 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).

Technical details. The attention model is implemented in PyTorch[Paszke et al. 217]. The Adam [Kingma and Ba 141] optimizer was used for the training with a learning rate of 10^{-5}. The attention model was trained end-to-end using an NVIDIA GTX 1080 and a 3.90 GHz Core-i7 7820 HK Intel processor. SalEMA experts were fine-tuned with a modified input size of 160×160.

Dataset processing. The VR-EyeTracking dataset, recorded by [Zhang et al. 296], includes 208 diverse content 4K Omnidirectional Videos (ODVs). Head and eye saccades were obtained from 45 participants. Despite the original dataset not publishing fixation and saliency maps publicly, we processed the gaze recordings to obtain 2048×1024 fixation maps. Saliency maps were generated by convolving each fixation point (from all observers for each video) with a Gaussian filter. The Gaussian filter's $\sigma = 9.35°$ was specifically chosen because it approximates the angular size that corresponds to the effective receptive field of human vision, which is often considered to be around 105 degrees in width. This size effectively covers the area within which detailed visual information is processed during natural viewing conditions, ensuring that the saliency maps reflect realistic attention distributions. The processed dataset is now available at the Github repository.

5.5.1 Results

Table 5.3 shows the comparative study with the aforementioned models according to the different saliency metrics on Salient360! and VR-EyeTracking datasets (4/75) test ODVs. Our model is very competitive in the two datasets: ATSal exhibits the best score for all metrics. Surprisingly, 2D approaches achieve mostly the same results as 360° specialized models; this points to the potential to consider the direct transfer of the well-verified visual attention features from 2D to 360° but also perhaps reflects the lack of large-scale well-annotated 360° datasets. We invite the reader to watch

Table 5.3: Comparative performance study of various saliency models on the Salient360! and VR-EyeTracking datasets. Notably, the results highlight the superior performance of specialized 360° models and our proposed model in handling the complex spatial distribution of saliency in 360° environments.

	Models \ Dataset	Salient360!					VR-EyeTracking				
		AUC-J ↑	NSS ↑	CC ↑	SIM ↑	KLD ↓	AUC-J ↑	NSS ↑	CC ↑	SIM ↑	KLD ↓
2D models	Unisal	0.746	1.258	0.206	0.190	8.524	0.764	1.798	0.314	0.281	6.767
	SalEMA	–	–	–	–	–	0.772	1.790	0.320	0.284	6.532
	3DSal	0.656	0.898	0.192	0.139	10.129	0.679	1.229	0.228	0.232	8.317
360° models	U-NET	0.725	0.864	0.129	0.122	9.810	0.818	1.430	0331	0.247	7.070
	CP360	0.819	1.094	0.175	0.213	9.313	0.735	0.914	0.171	0.179	8.879
	Salgan360	–	–	–	–	–	0.704	1.267	0.236	0.238	7.625
	Two-Stream	0.787	1.608	0.265	0.179	8.252	0.827	1.906	0.346	0.254	7.127
Training setting (i)	Attention	0.827	1.397	0.222	0.168	8.225	0.795	1.338	0.255	0.229	7.405
	Experts	0.870	2.438	0.366	0.227	6.990	0.754	1.292	0.221	0.224	7.840
	Ours	**0.881**	**2.580**	**0.363**	**0.252**	**6.327**	0.801	1.590	0.265	0.255	6.571
Training setting (ii)	Attention	0.796	1.299	0.201	0.168	7.973	0.836	1.791	0.342	0.302	5.664
	Experts	0.807	1.564	0.242	0.223	7.679	0.778	1.388	0.259	0.211	7.407
	Ours	0.837	1.764	0.285	0.255	7.382	**0.862**	**2.185**	**0.363**	**0.312**	**5.561**

Table 5.4: Comparative performance study on: Salient360!, VR-EyeTracking datasets.

	Models \ Dataset	Salient360!				
		AUC-J ↑	NSS ↑	CC ↑	SIM ↑	KLD ↓
2D models	Unisal	0.746	1.258	0.206	0.190	8.524
	3DSal	0.656	0.898	0.192	0.139	10.129
360° models	U-NET	0.725	0.864	0.129	0.122	9.810
	CP360	0.819	1.094	0.175	0.213	9.313
	Two-Stream	0.787	1.608	0.265	0.179	8.252
AtSal	Attention	0.827	1.397	0.222	0.168	8.225
	Experts	0.870	2.438	0.366	0.227	6.990
	Combined	**0.881**	**2.580**	**0.363**	**0.252**	**6.327**

the qualitative results of the proposed method in the form of a video available here.

Performance study

VR-EyeTracking. The 75 diverse test ODVs of this dataset makes the prediction task very challenging. ATSal outperforms all competitors according to all five metrics due to the combination of global features with the attention stream, but also due to the local patches encoding the temporal domain predicted by the experts model.

Salient360!. On both distributions of the Salient360! For datasets (images and videos), the proposed model gains a substantial quantitative advantage in accuracy compared with other models. This demonstrates the capacity of our model to handle

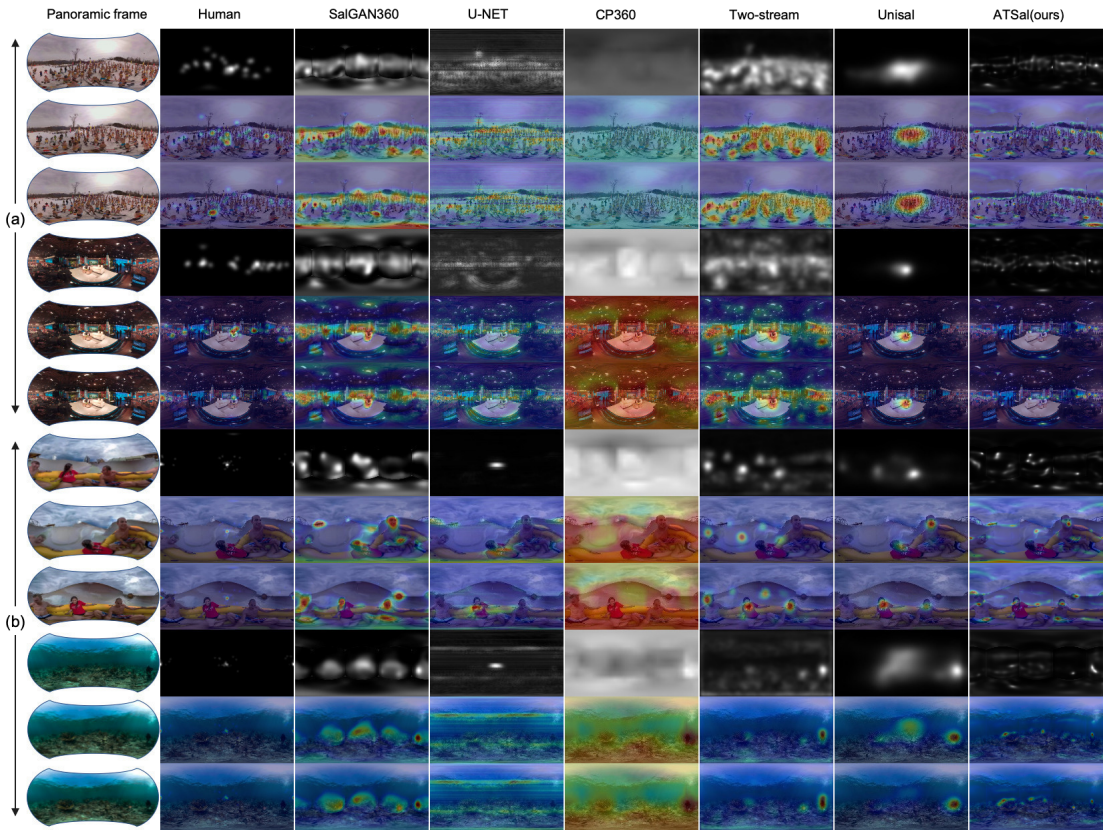


Figure 5.6: Qualitative results of our ATSal model and five other competitors on sample frames from VR-EyeTracking and Salient360! datasets. It can be observed that the proposed ATSal is able to handle various challenging scenes well and produces consistent omnidirectional video saliency results.

a larger set of scenarios.

Encouraged by the positive results of the attention model trained on static images, and later fine-tuned for the Salient360! and VR-EyeTracking ODVs. The attention model shows its ability to capture global saliency, so we evaluated the model components separately on both datasets. Table 5.4 indicates the competitiveness of ATSal with respect to other competitors, even without the integration of the expert model. A possible explanation could relate to smaller motion effects in ODVs compared to 2D videos, where viewport locations are the most prominent features.

Figure 5.6 illustrates the predictions on a sample of 360° frames from two datasets: Salient360! and VR-EyeTracking. It can be seen that the generated saliency maps with ATSal are more comprehensive and look remarkably similar to the ground truth maps in terms of saliency distribution. Other competitors shown in the same figure overestimate saliency in general or overly bias it to the equator/center. Furthermore,

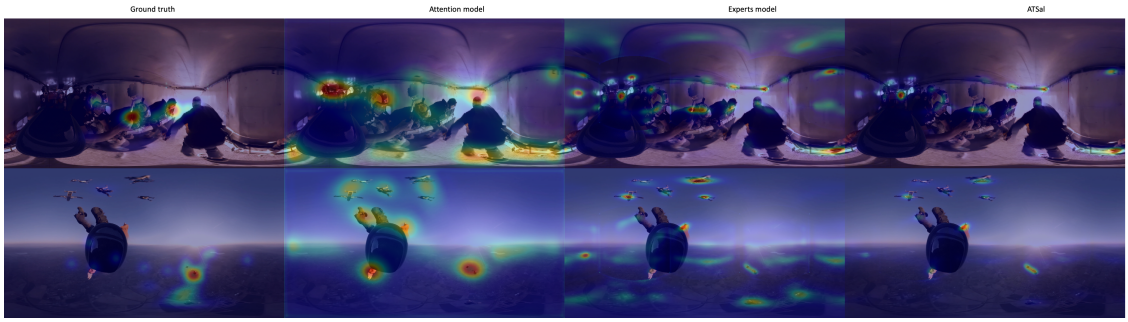


Figure 5.7: Overlaid saliency of the attention model, experts model, and the final ATSal. It can be seen that the attention model captures more global saliency, while the experts focus on local patches. ATSal gathers both features for the final prediction.

the effectiveness of ATSal in capturing the main objects in the scene is observed.

For more qualitative results, Figure 5.7 shows the overlaid saliency maps on sample frames from Salient360! for the attention model, expert models, and ATSal. Two key points can be seen from these figures:

- Figure 5.6: ATSal predicts consistent saliency at the very beginning of each video due to the attention mechanism modeling the spatial features, while other competitors center the saliency around the equator. As the scene progresses, ATSal ignores some static regions and only focuses on other moving viewports. In the video (b), all models only detect faces near the equator, while ATSal also attends to other parts of the scene.

- Figure 5.7: The saliency maps using only the attention model are sparse and cover all the possible outcomes in the scene. The larger receptive field in the attention model provides it with a global perspective on regions that might be salient. The expert models act as a saliency selector through pixel-wise multiplication. This approach guarantees that saliency is not overestimated but also forces the model to generate more focused and consistent saliency regions.

Computational load. Model efficiency is a key factor for real-time 360° video applications like streaming. Table 5.5 shows a GPU runtime comparison (processing time per 360° frame) of the different competitors on the 4K VR-EyeTracking ODVs.

Table 5.5: GPU inference time comparison of video saliency prediction methods (NVIDIA GTX 1080). All methods are reported based on the VR-EyeTracking benchmark. The best computational performance among dedicated 360° models is shown in bold.

Model	Runtime (s)
SalGAN360	17.330
Two-stream	2.061
U-Net	1.802
(*) 3DSAL	0.018
(*) SalEMA	0.010
(*) UNISAL	0.020
Attention (ours)	0.050
Experts (ours)	0.170
ATSal (ours)	**0.230**

(*) 2D models.

Compared with other 360° specialized models, ATSal gains a remarkable margin, being over 8× faster than U-Net, which is the fastest among this category.

5.5.2 Limitations

Here, we describe several limitations of the presented model.

Memory requirements In its current design, the ATSal framework demands high memory capacity. As the framework processes multiple streams and expert models in parallel and considers the dense mask learning process within the attention mechanism, the memory requirements can grow substantially with the size of the input data. Thus, deploying the framework on standard GPU configurations might pose challenges for live stream applications or high-resolution inputs.

Complex architecture One prominent challenge posed by the ATSal framework is the inherent complexity of its architecture. A solution would ideally be both effective and streamlined. ATSal, however, introduces multiple parallel streams, dedicated expert models for different viewports, and intricate processes like projection and inverse projection to and from the CMP format. While enhancing accuracy, each of these components also adds layers of computational and conceptual intricacy. Such complexity not only makes the model more resource-intensive but also poses

representation questions. A simpler design might offer easier interpretability, faster training times, and more straightforward adaptability to evolving requirements. This limitation will be addressed in Chapter 6 with the use of self-supervised learning.

5.6 Conclusion

In this chapter, we reviewed attempts to solve the visual attention modeling problem for 360° data, which involves predicting both the head and eye movements of users exploring panoramic scenes. We noted that deep learning-based saliency models [Assens et al. 5, Pan et al. 213, Chao et al. 43, Suzuki and Yamanaka 258] trained end-to-end on 360° datasets show remarkable improvements over earlier models that adapted 2D approaches to Omnidirectional Images (ODIs) [De Abreu et al. 66, Maugey et al. 192, Startsev and Dorr 255, Lebreton and Raake 162, Cornia et al. 57, Luz et al. 182], demonstrating the new constraints imposed by the spherical domain when modeling visual attention.

This motivated us to design the attention-based model **ATSal**. The framework explicitly captures global visual static saliency information through an attention mechanism at the middle of the bottleneck, with an enlarged receptive field that propagates contextual pixel-wise information across the entire frame space. This design allows the expert models to learn more effective local region-based saliency, while the temporal domain is augmented using a simple exponential moving average (EMA). We performed evaluations on the Salient360! and VR-EyeTracking datasets and compared our results with previous 2D and 360° static and dynamic models. Since our original publication [Dahou et al. 61], ATSal has been extended, with notable improvements including the use of transformer-based attention [Huang et al. 120] and leveraging the audio modality as additional information [Zhu et al. 300].

However, it is important to acknowledge the inherent architectural limitations of ATSal, where performance might be confounded by the higher number of parameters due to multiple instances of expert models. This design, involving multiple parallel streams and expert models, adds layers of computational and conceptual complexity,

opening questions for simplifying the model's architecture and exploring advanced representation learning procedures to bridge this gap, as outlined in Chapter 6.

Chapter 6

Learning implicit geometrical features for 360° images.

6.1 Introduction

In the previous chapter, we presented the ATSal and discussed its main limitations. To further advance saliency prediction for 360° data, it becomes imperative to leverage more refined deep learning methods that are consistent with the unique geometric attributes inherent to omnidirectional scenes. This necessitates a shift from traditional, fully supervised frameworks to methodologies that focus on learning better abstract representations. Furthermore, despite the success of self-supervised representation learning on planar data, its application to 360° images remains largely unexplored. In this chapter, we extend recent advances in contrastive learning to learn latent representations that are sufficiently invariant and effective for spherical saliency prediction. Here, saliency prediction is conceptualized as a downstream task that utilizes the robust features extracted through contrastive learning.

We argue that omnidirectional images are particularly well-suited to this approach due to the geometric characteristics of their data domain. To test this hypothesis, we designed a contrastive self-supervised framework that effectively maximizes mutual

This chapter is based on our ICCV 2021 publication [Djilali et al. 73]

information between different views from both the equator and the poles. We demonstrate that the decoder in this framework can learn high-quality saliency distributions from the encoder embeddings. This approach does not involve an architectural change but rather a shift in the learning procedure, acknowledging the inherent geometric flexibility and exploiting the robustness of different projections to build a more comprehensive understanding of saliency in omnidirectional data.

Our model shows favorable comparisons with fully supervised learning methods on the Salient360!, VR-EyeTracking, and Sitzman datasets. This performance is achieved using an encoder trained in a completely unsupervised manner coupled with a relatively lightweight supervised decoder (which has 3.8 times fewer parameters compared to a ResNet50 encoder). We believe that integrating supervised and unsupervised learning methods offers a significant step toward more flexible formulations of human visual attention, where traditionally considered upstream tasks like saliency prediction effectively become downstream applications of foundational unsupervised learning processes.

6.2 Related prior work

As we reviewed works about saliency for 360° data in Chapter 2, here we review the set of important works related to contrastive learning in general.

Contrastive learning has demonstrated its ability to exploit large-scale unlabeled data for learning more general and robust features about the data. Self-supervised learning [Jing and Tian 133] refers to solving a pretext task for which pseudo labels are derived from the data attributes. The assumption is that if the model solves these pretext tasks, it can only be done through learning good representations. The design of these pretext tasks depends on what kind of semantic similarity one wants to reinforce in the learned representation so that it would encourage generalization to downstream tasks. This includes generative methods such as image colorization [Zhang et al. 295], image super-resolution [Ledig et al. 163], image inpainting [Pathak et al. 218], and of course, image generation via generative

adversarial nets [Goodfellow et al. 97, Zhu et al. 301]. Similarly, to capture spatial structure in image patches, pretext tasks like context prediction [Doersch et al. 76], jigsaw puzzles [Noroozi and Favaro 205], and predicting image rotations [Gidaris et al. 94] have also been recently proposed to learn more robust representations.

In the context of videos, a similar notion of a pretext task holds, where temporal coherence is also exploited along with spatial structure. Works such as [Misra et al. 199, Rahaman et al. 228, Lee et al. 165] defined a pretext task to predict the order of the sequence or sort the correct sequence in video frames.

Contrastive learning [Le-Khac et al. 158, Jaiswal et al. 126] involves learning by comparison where the final objective is often based on some variations of Noise Contrastive Estimation (NCE). This objective is a statistical technique used to estimate the parameters of a probabilistic model. The idea in contrastive learning contexts is that the network should differentiate positive samples, which are augmentations of the anchor, from negative samples, which are other unrelated samples in the dataset.

One of the pioneering works in contrastive learning space, Contrastive Predictive Coding (CPC) [Oord et al. 208], demonstrated that minimizing an NCE objective is equivalent to maximizing mutual information (MI) between samples, which they termed InfoNCE. This approach has been extended to various modalities and frameworks to learn better features for images.

Following these foundational approaches, recent models like SimCLR [Chen et al. 48], and Momentum Contrast (MoCo) [He et al. 110, Chen et al. 49] have refined these ideas. SimCLR focuses on maximizing the agreement between different augmented views of the same data, whereas MoCo utilizes a dynamic dictionary with a queue and a moving-averaged encoder to facilitate learning without requiring extensive memory banks. In summary, all these models try to enforce invariance to geometric distortions (augmentation) through instance classification (or maximizing agreement), and, in doing so, they exploit semantic (context) similarity and spatial structure among different variations of data samples to learn better representations.

There has been tremendous progress in this direction of unsupervised learning

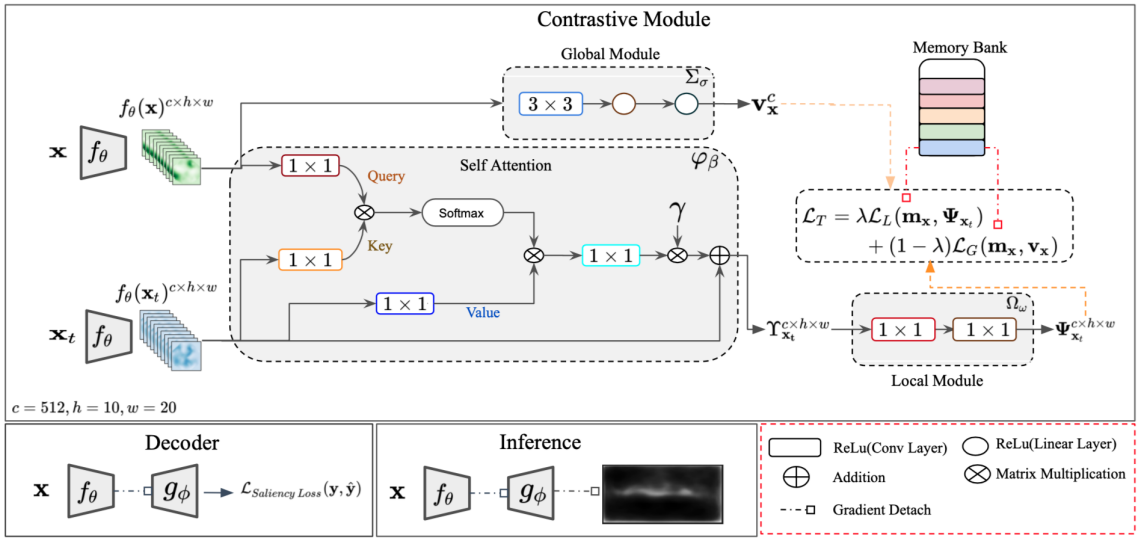


Figure 6.1: Complete pipeline for training. The **contrastive module** comprises of an encoder f_θ, a global module Σ_σ, self-attention φ_β, and local module Ω_ω trained jointly to optimize the set of parameters $\boldsymbol{\Theta} = \{\boldsymbol{\theta}, \boldsymbol{\sigma}, \boldsymbol{\beta}, \boldsymbol{\omega}\}$ 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 of unseen data. The `ReLu (Conv Layer)` and `ReLu (Linear Layer)` in the bottom right change depending on the different modules (see Section 6.3.2).

[Caron et al. 41, Grill et al. 101, Caron et al. 40, Misra and Maaten 198, Bautista et al. 14] in recent times. However, the scope of most of these methods has been limited to recognition as a downstream task (refer to [Le-Khac et al. 158, Jaiswal et al. 126] for an extensive review). In this chapter, we take a step forward and extend the approach of mutual information estimation to regression, specifically to the task of saliency prediction, which is more fine-grained than recognition. Given that this domain of 360^o data conceptually provides new sets of choices, i.e. signals for Zenith and Nadir regions, it is particularly suited for contrastive learning.

6.3 Method

The algorithm leverages the geometric flexibility inherent in the 360° data definition domain, i.e., the spherical representation, where the different projections represent robust views for training a differentiable parametric function $f_\theta : X \to \Lambda$ (depicted as $f_\theta(\mathbf{x})$ in Figure 6.1), with parameters θ (e.g., a neural network), to maximize the mutual information among the views without any further supervision. This training

approach uses the unique vantage points provided by spherical data to enhance the model's ability to generalize across varied orientations and distortions. The encoder is optimized to detect the polar regions, i.e., views, pushing the convolution filters to exploit larger groups of symmetries, including spherical transformations and rotations because the translation symmetry preserved by a CNN is not sufficient to detect the distorted objects in the polar regions. We argue that the only information shared between the views is task-relevant, and there is no irrelevant noise, as the three views can fully reconstruct the sphere. Furthermore, we rely on exploiting contrastive learning-based approaches [Le-Khac et al. 158] to learn optimal and robust representations for 360° data. To further measure the quality of the latent representations, a separate parametric function $g_\phi : \Lambda \rightarrow y$ (decoder) is able to decode good quality saliency maps for the downstream task. It is worth mentioning that the two stages are asynchronous.

6.3.1 Overview of the Approach

Suppose we are given a 360° image dataset, $\mathcal{D} = \{\mathbf{x}_1, ..., \mathbf{x}_{|\mathcal{D}|}\}$ where $\mathbf{x}_i \in \mathcal{R}^{H \times W \times 3}$, 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. 44]), 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. 113, Bachman et al. 8]. However, there are some significant differences that distinguish the proposed 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 the contrastive objective further encourages clusters to form in the representation space. Thus, we aim to optimize 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.

6.3.2 Unsupervised Module

Base encoder (f_θ): We experiment and report findings for VGG16 [Simonyan and Zisserman 249] and ResNet50 [He et al. 111] as encoders. The VGG16 encoder is similar to SalGAN [Pan et al. 212]. While for ResNet50, we consider up to `Conv_4x`[1] followed by [`Conv2d, BatchNorm2d, Conv2d, BatchNorm2d, ReLu`] as our base encoder.

Global module (Σ_σ). The encoder output is fed into this module to get a compact global visual representation of the input that would capture the global features of the scene. In case of VGG16 this module consists of a *conv block* [`Conv, BatchNorm2d, ReLu, MaxPool`] (kernel size = 3×3) followed by $2\times$[`Linear,Relu`] layers as depicted in Figure 6.1. For ResNet50, the *conv block* is replaced with [`Conv2d, BatchNorm2d, Conv2d, BatchNorm2d, ReLu`] followed by the remaining block of ResNet50, i.e., `Conv_5x` along with the `Adaptive pooling layer` keeping the rest the same. The global module outputs a representation $\mathbf{v_x} \in \mathbb{R}^{512}$.

Local module (Ω_ω). Consists of $2\times$[`Conv2d,ReLu`] layers, followed by a `BatchNorm2d` layer to get the local representation of each location across spatial dimension of $\mathbf{x}_t$, which is $\Psi_{\mathbf{x}_t} \in \mathbb{R}^{c\times h\times w}$. The architecture of the local module is fixed for both VGG16 and ResNet50. The primary objective of this module is to break the symmetry of the architecture, thereby having a more effective and efficient convergence of the neural network during training.

Self-attention module (φ_β). This module is similar to [Wang et al. 278, Zhang et al. 291], but in this case, the query comes from different views, whereas key and value are derived from the single source, more specifically $\mathbf{x}_t$ as depicted in Figure 6.1. γ in Figure 6.1 is a scalar trainable parameter initialized with zero as was used in [Zhang et al. 291] to exploit non-local evidence. For details about this module and

[1]This notation is the same as in the ResNet [He et al. 111] paper.

the ResNet-based encoder, please refer to Section 6.6.

Loss function We minimize a NCE [Gutmann and Hyvärinen 105] based approximation as in [Tian et al. 263], defined as:

$$\begin{aligned}\mathcal{L}_{\text{NCE}} = -\mathop{\mathbb{E}}_{\mathbf{x}\sim p(\mathbf{x})}\Big\{\mathop{\mathbb{E}}_{\mathbf{x}_t\sim p(.|\mathbf{x})}\big[\log\big(P(D=1|\mathbf{x}_t,\mathbf{x})\big)\big] \\ +m\mathop{\mathbb{E}}_{\mathbf{x}_n\sim p_n(.|\mathbf{x})}\big[\log(P(D=0|\mathbf{x}_n,\mathbf{x}))\big]\Big\}.\end{aligned} \tag{6.1}$$

Optimizing $\mathcal{L}_{\text{NCE}}$ is simply minimizing the negative log-posterior probability of labels 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 (or augmentation). $p(\mathbf{x})$ and $p_n(.)$ in Equation 6.1 are the empirical data distribution and distribution of noisy samples, respectively. The posterior distribution is:

$$P(D=1|\mathbf{x}_t,\mathbf{x}) = \frac{p(\mathbf{x}_t|\mathbf{x})}{p(\mathbf{x}_t|\mathbf{x}) + m\,p_n(\mathbf{x}_n|\mathbf{x})}, \tag{6.2}$$

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. 263, Gutmann and Hyvärinen 106] for further details on the derivation of the NCE loss.

Memory bank. Following [Wu et al. 282, Misra and Maaten 198], we maintain a memory bank to retrieve m negative samples, which is an exponential moving average of feature representations $\mathbf{v}_\mathbf{x}$ that were computed in prior epochs. Also, the representations $\mathbf{m}_\mathbf{x}$ saved in the memory bank correspond to the source view $\mathbf{x}$ and are L_2 normalized after each update. The number of negative samples is fixed at 16k throughout this chapter. The necessity of a memory bank arises from its ability to provide a rich repository of feature representations that the model can use to distinguish between different data points effectively. This is important because, in contrastive learning, the model learns to differentiate between similar (positive) and

dissimilar (negative) samples. Hence, the memory bank provides a rich repository of feature representations that the model can use to distinguish between different data points effectively as an augmentation of the batch size in some optimization steps. By having access to a large and varied set of negative samples, the model can better learn a more robust feature representations.

The final objective is defined as a convex combination of both the global and local NCE losses:

$$\mathcal{L}(\mathbf{x}, \mathbf{x}_t) = \lambda \mathcal{L}_L(\mathbf{m}_\mathbf{x}, \Psi_{\mathbf{x}_t}) + (1 - \lambda)\mathcal{L}_G(\mathbf{m}_\mathbf{x}, \mathbf{v}_x). \tag{6.3}$$

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 198], and the second 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 $\mathbf{x}$ (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. 113]. 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 $(c, h, w) = (512, 10, 20)$ stay fixed across all settings.

6.3.3 Supervised Module

Problem formulation. Visual attention modeling for omnidirectional images is the downstream task of interest for this thesis; hence, we measure the quality of the representations on the saliency task. It consists of predicting a head and 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 point (for all

observers of one image), defined as:

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

This convolution operation is performed in a 2D spherical space to maintain isotropy. In this spherical setting, using an isotropic Gaussian on an equirectangular map does not preserve isotropy when back-projected onto a sphere. While a Kent distribution could offer a more accurate representation due to its orientation and concentration parameters, the Gaussian kernel is often preferred in practical applications. This preference is supported by its computational simplicity and the extensive empirical evidence demonstrating its effectiveness in various visual attention models, as discussed in the literature [David et al. 65]. This balance of computational efficiency and empirical validation makes the Gaussian kernel a pragmatic choice for generating saliency maps in 360° visual attention studies.

Decoder module. As stated in the previous chapter, human attention is driven by both global and local features. In ODIs, the CMP forces the model to lose the global contextual information while considering each cube face separately. Through the contrastive encoder, more explicit global features are learned inherently as a superposition in the encoding function weights [Domingos 77]. Thus, the convolution filters are more responsive to the signal connected to the poles. Therefore, we argue that the latent representations $\Lambda \in \mathbb{R}^{512\times10\times20}$ lie within a feasible manifold to be decoded into saliency maps. The decoder architecture is inspired by 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 $S_p \in [0,1]^{W\times H}$, and the continuous ground truth $S_{GT} \in [0,1]^{W\times H}$. The objective function must be designed to maximize

Table 6.1: Comparative performance study on: Salient360! and VR-EyeTracking datasets. Training setting (i): trained w/o self-attention, Training setting (ii): trained with 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 ↓
Baseline: infinite humans		0.788	2.09	1.0	1.0	0.0	0.985	3.421	1.0	1.0	0.0
2D models	UNISAL [Droste et al. 79]	0.701	1.404	0.389	0.435	2.519	0.783	1.918	0.276	0.242	9.044
	SalGAN [Pan et al. 212]	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. 60]	0.777	**1.638**	**0.642**	**0.639**	**0.761**	0.822	1.613	0.239	0.191	9.796
	SalGAN360 [Chao et al. 43]	**0.831**	1.598	0.639	0.611	0.798	0.704	1.267	0.226	0.218	7.938
	SaltiNet [Assens et al. 5]	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**
	Finetuned	0.769	1.601	0.584	0.591	0.849	**0.869**	**2.089**	**0.329**	**0.248**	**7.110**

Table 6.2: Comparative performance study on: Salient360! and VR-EyeTracking datasets. Training setting (i): trained w/o self-attention, Training setting (ii): trained with self-attention. The best scores are marked in **bold** and second best in blue.

Model		Salient360!				
		AUC-J ↑	NSS ↑	CC ↑	SIM ↑	KLD ↓
Baseline: infinite humans		0.788	2.09	1.0	1.0	0.0
2D models	UNISAL	0.701	1.404	0.389	0.435	2.519
	SalGAN	0.701	1.398	0.377	0.483	1.544
360° models	ATSAL	0.777	**1.638**	**0.642**	**0.639**	**0.761**
	SalGAN360	**0.831**	1.598	0.639	0.611	0.798
	SaltiNet	0.702	1.057	0.536	0.541	1.098
Ours	VGG	0.760	1.548	0.538	0.569	0.922
	ResNet	0.769	1.601	0.584	0.591	0.849

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. 37], the KLD between S_p and S_{GT} is given by:

$$\mathcal{L}_{\text{KLD}}(S_p, S_{GT}) = \sum_{i=1}^{W \times H} \left(S_{GT_i} \times \log \left(\epsilon + \frac{S_{GT_i}}{\epsilon + S_{p_i}} \right) \right), \tag{6.4}$$

6.4 Experimental Setup

Training. We first train the encoder following the self-supervised procedure. In fact, contrastive learning requires a large amount of unlabelled data to be trained effectively. Due to the unavailability of large-scale 360° images datasets, we gathered a new one with 90K ODIs from multiple sources. The dataset comprises the following:

- PVS: HMEM [Xu et al. 286] contains 76 panoramic videos, and images were

sampled at a rate of 1 frame per second (fps).

- 360-Indoor [Chou et al. 52] contains 3024 complex indoor scenes containing common objects.

- VR-VQA [Xu et al. 284], a quality assessment dataset, comprises 48 ODVs from 8 classes: sport, movies, etc.

- Videos gathered from YouTube playlists[2] (1 fps).

Furthermore, the evaluation of these representations relies on the downstream task. Often, classification accuracy is used as a proxy for correlation between representations and class labels. Clearly, as the symmetries and invariances encoded in the representation are abstract, a more granular regression proxy task such as saliency prediction reveals more insights. Thus, we evaluate the unsupervised module's representational properties on three saliency datasets:

- Salient360! images [Rai et al. 229]: a small-scale dataset consisting of (80/23) 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.

- Due to the small amount of labeled static data (103 ODIs), we sampled at a rate of 1 fps from the large-scale video dataset VR-EyeTracking [Xu et al. 287]. The resulting set contains (4700/1300) 360° images.

- Sitzmann [Sitzmann et al. 250] containing a total of (14/11) training/validation ODIs; the authors captured and analyzed gaze and head orientation data of 169 users.

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

[2]Playlist 1, Playlist 2, Playlist 3

procedure and to prevent the encoder from adapting its parameters to saliency-specific requirements.

6.4.1 Results

Our approach is experimentally compared to five models: two state-of-the-art 2D static saliency models, UNISAL [Droste et al. 79] and SalGAN, and three 360° specific models: ATSAL [Dahou et al. 60], SalGAN360 [Chao et al. 43], and SaltiNet [Assens et al. 5]. 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 Metric (SIM), Linear Correlation Coefficient (CC), and AUC-Judd (AUC-J). Please refer to [Bylinskii et al. 37] for an extensive review of these metrics.

Technical details. Both the contrastive encoder and the supervised decoder were implemented in PyTorch and trained using two GPUs (RTX 3090 & RTX 2080). The contrastive encoder was optimized using SGD with a learning rate of 10^{-2} [Ruder 236]. The encoder was trained for 250 epochs using the maximum batch size of 80[3], with negative samples fixed at 16000, $\tau = 0.07$, and $\lambda = 0.7$. Choices regarding the total epochs and the number of negative samples are based on computation and time constraints. Training for more epochs or with a larger number of negative samples may provide a further boost in performance [Kolesnikov et al. 144, Misra and Maaten 198]. Adam, with a learning rate of 10^{-4}, was used to train the supervised decoder for 100 epochs.

Table 6.1 shows the comparative study with the aforementioned models according to the different saliency metrics on the Salient360! and (VR-EyeTracking+Sitzman) datasets of 25/1300 test 360° images. The proposed model is very competitive in the two datasets and exhibits the top score for all metrics on VR-EyeTracking+Sitzman. As expected, 2D state-of-the-art approaches fail to generalize on omnidirectional images, which questions the effectiveness of the direct transfer of visual attention

[3]the maximum batch size that could be achieved given our GPU memory size

features from 2D to 360° data.

VR-EyeTracking & Sitzman. The 1300 validation/test images were sampled from the 75 diverse test omnidirectional videos of the first dataset, mixed 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 $\downarrow = 7.110$).

Salient360! We used the 25 360° validation images for testing the model[4]. 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 for SalGAN360 and ATSAL, which were trained on this specific dataset, hence explaining their performance.

Figure 6.2 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 6.3). This demonstrates the effectiveness of the contrastive encoder in embedding the views as a superposition in the function weights and biases.

Computational load. As model efficiency is a key factor for real-world omnidirectional applications, Table 6.4 shows a GPU runtime comparison (processing time per 360° image) of the different competitors on the 4K Salient360! benchmark. Compared with other 360° specialized models, our model exhibits a remarkable improvement, being over 8× faster than ATSal [Dahou et al. 61], which is the fastest

[4]The reserved test set was unavailable due to COVID-19

Table 6.3: Comparative performance study on: Salient360! and VR-EyeTracking datasets. VGG_(i) following training setting (i), VGG_(ii) following training setting (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

Table 6.4: GPU inference time comparison of video saliency prediction methods (NVIDIA RTX 3090). All methods are reported based on the Salient360! benchmark [Xu et al. 287]. The best computational performance among dedicated 360° models is shown in bold.

Model	Runtime (s)
SalGAN360 [Chao et al. 43]	14.330
ATSAL [Dahou et al. 60]	0.230
SaltiNet [Assens et al. 5]	0.450
(*) SalEMA [Droste et al. 79]	0.020
(*) UNISAL [Pan et al. 212]	0.010
Ours (ResNet-based)	**0.025**

(*) 2D models.

Table 6.5: 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**

Figure 6.2: 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 6.6: The standard deviation of all predictions across the validation set.

	Salient360!				
	AUC-J ↑	NSS ↑	CC ↑	SIM ↑	KLD ↓
VGG	0.760±0.04	1.548±0.20	0.538±0.11	0.569±0.05	0.922±0.28
ResNet	0.769±0.03	1.601±0.18	0.584±0.08	0.591±0.03	0.849±0.16

model in this category.

6.4.2 Additional results

It can be seen in Figure 6.4 and Figure 6.5 that the VGG-based model overly biases the saliency towards the equator. Since the encoder's weights were fixed during the decoder's training on saliency, the learning relies on the latent representations learned from unsupervised training. We argue that the VGG encoder preserves the information of the pixels more so than learning higher-level features. Another interesting observation is the considerable change in the first layer for both encoders. We suspect this is related to the 360° input images; local filters change to adapt to the data representation. Table 6.5 analyzes the 25 latent representations obtained from both VGG and ResNet contrastive encoders. We measured the inter-map correlation: for a given representation $\Lambda \in \mathbb{R}^{512\times10\times20}$, we fix a target map $\Lambda_{\mathbf{i}} \in \mathbb{R}^{10\times20}$, and compute the average correlation coefficient among the maps, $j \in \{1,\ldots,512\}, j \neq i$. For VGG, the average $CC = 0.31$ while ResNet has $CC = 0.10$. Furthermore, training atop the VGG encoder appears to force the decoder to learn an average saliency distribution over the training set.

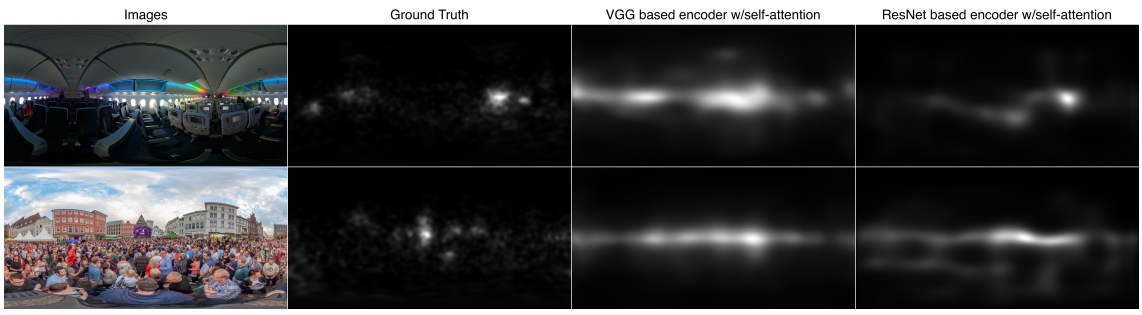


Figure 6.3: Predicted saliency maps from Salient360! samples.

Surprisingly, however, this still achieves reasonable results ($KLD = 0.922; NSS = 1.548$). An explanation could relate to the statistical bias present in HM/EM fixations. Viewers tend to gaze at regions near the equator (termed the equator bias [Sitzmann et al. 250, Lebreton and Raake 162]). In contrast, the decoder trained atop the ResNet-based encoder is able to learn a non-obvious saliency distribution, focusing on critical regions, further improving the model's accuracy. It can be observed that the ResNet-based model correlates well with the ground-truth distribution (see Figure 6.4, Figure 6.5). To further illustrate this, Table 6.8 measures the distance between an equator bias and the models' predictions; the VGG-based model has higher similarity with the equator bias (KLD = 1.867), whereas the ResNet-based model has lower similarity (KLD = 2.507).

Fine-tuning the whole model. As an additional experiment, we fine-tuned the ResNet-based model end-to-end on the saliency dataset using the Adam optimizer, adapting the unsupervised weights to the specific task at hand. The results are: (AUC-J: 0.790), (NSS: 1.616), (CC: 0.586), (SIM: 0.607), (KLD: 0.815). This outcome is not particularly surprising as the encoder weights are now more specific to the task and the in-distribution dataset.

Average pooling based encoder. We replaced the attention with average pooling on $f(x_t)$ (equivalent to equal attention). This induces a global attention effect. The results are (AUC-J: 0.728), (NSS: 1.544), (CC: 0.528), (SIM: 0.558), (KLD: 0.901). This demonstrates the importance of local attention specifically designed for downstream tasks.

Failure scenarios. To further analyze the model's expressive power, we have identified some outliers on the Salient360! validation set (i.e. the images with the highest KLD = 1.51; the lowest NSS = 1.09). Figure 6.6 illustrates two failure modes identified so far. In the top image, the model misses the object highlighted in the yellow box, which is salient to humans. This object, however, appears to have been blurred in capture due to camera motion, which likely affected the features typically used to detect an object as salient. However, this may be seen as a problem/artifact

Table 6.7: Statistics about the Salient360! 24 validation images latent representations $\Psi_{\mathbf{x}} \in \mathbb{R}^{512\times10\times20}$.

	CC
VGG contrastive encoder	0.295
ResNet contrastive encoder	0.107

Table 6.8: The probabilistic distance between the equator bias and the model's predictions on the Salient360! validation set.

	KLD ↓
VGG-based saliency model	1.867
ResNet-based saliency model	2.507

of the input data and not a problem due to model capacity. The second scenario is more challenging. The image was captured between two cabins in a train; the model is able to detect salient regions in the front cabin, but clearly, humans localized in the viewing sphere to generate the ground truth were focusing on people standing in the rear cabin. The resolution of this rear cabin is small in the ERP format. Hence, we suspect that detecting saliency at depth is a weakness of the model. A potential solution could be the use of multi-scale attention [Tao et al. 259], which has been shown to improve failure modes related to features that need to be detected at different spatial scales.

Model consistency. Averaging the metrics over the validation set biases the evaluation somewhat, as models can fit precisely some samples but give poor predictions for others. Table 6.6 shows the results on the Salient360! validation set, with confidence intervals. It can be seen that the ResNet-based model has less variance across all metrics, demonstrating the consistency of the predictor. The VGG-based model, on the other hand, has a wider confidence interval, which is explained by the fact that a substantial amount of the ground truth saliency has a high density at the equator.

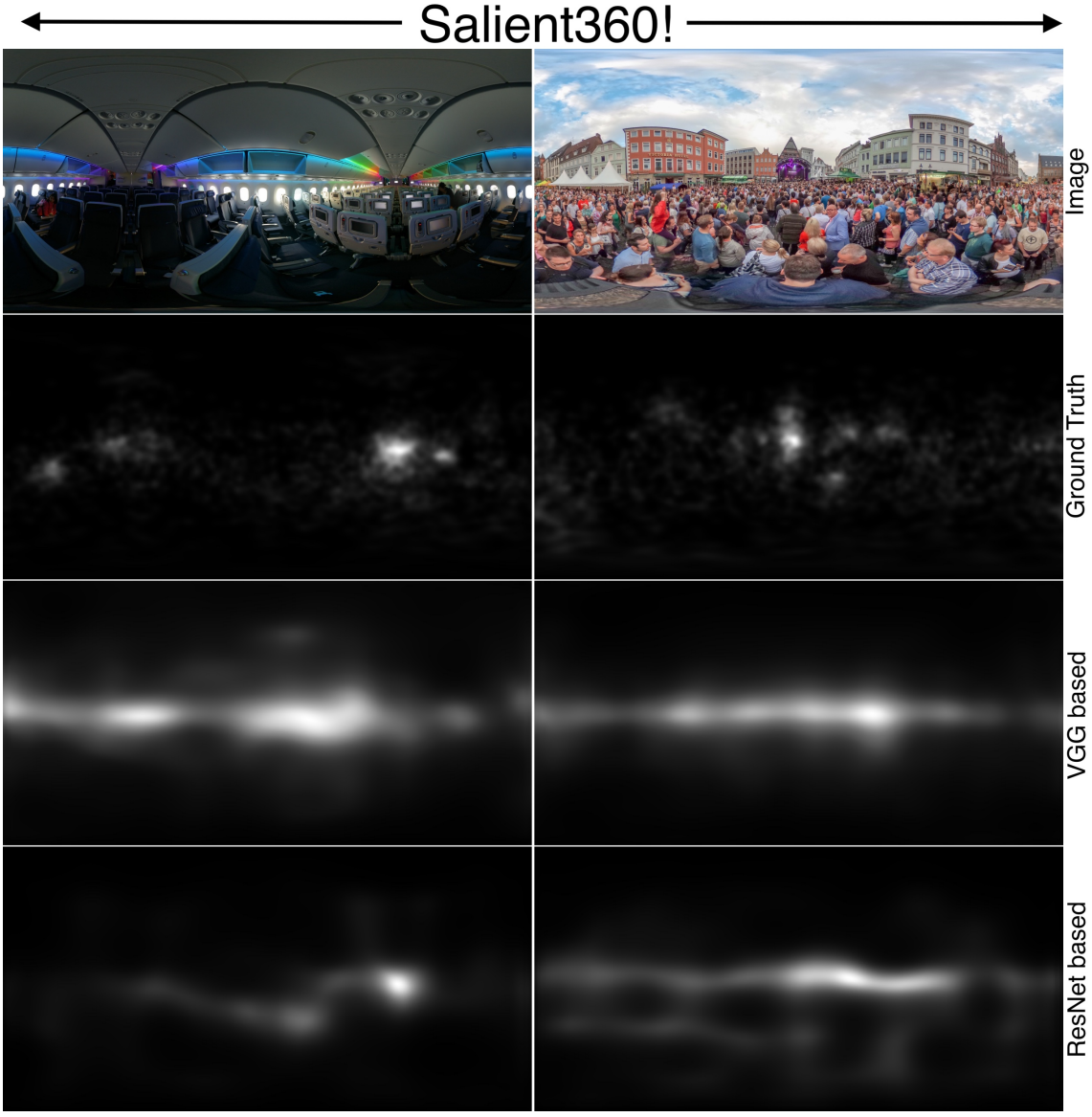


Figure 6.4: Qualitative results of model variants on sample images from Salient360! dataset.

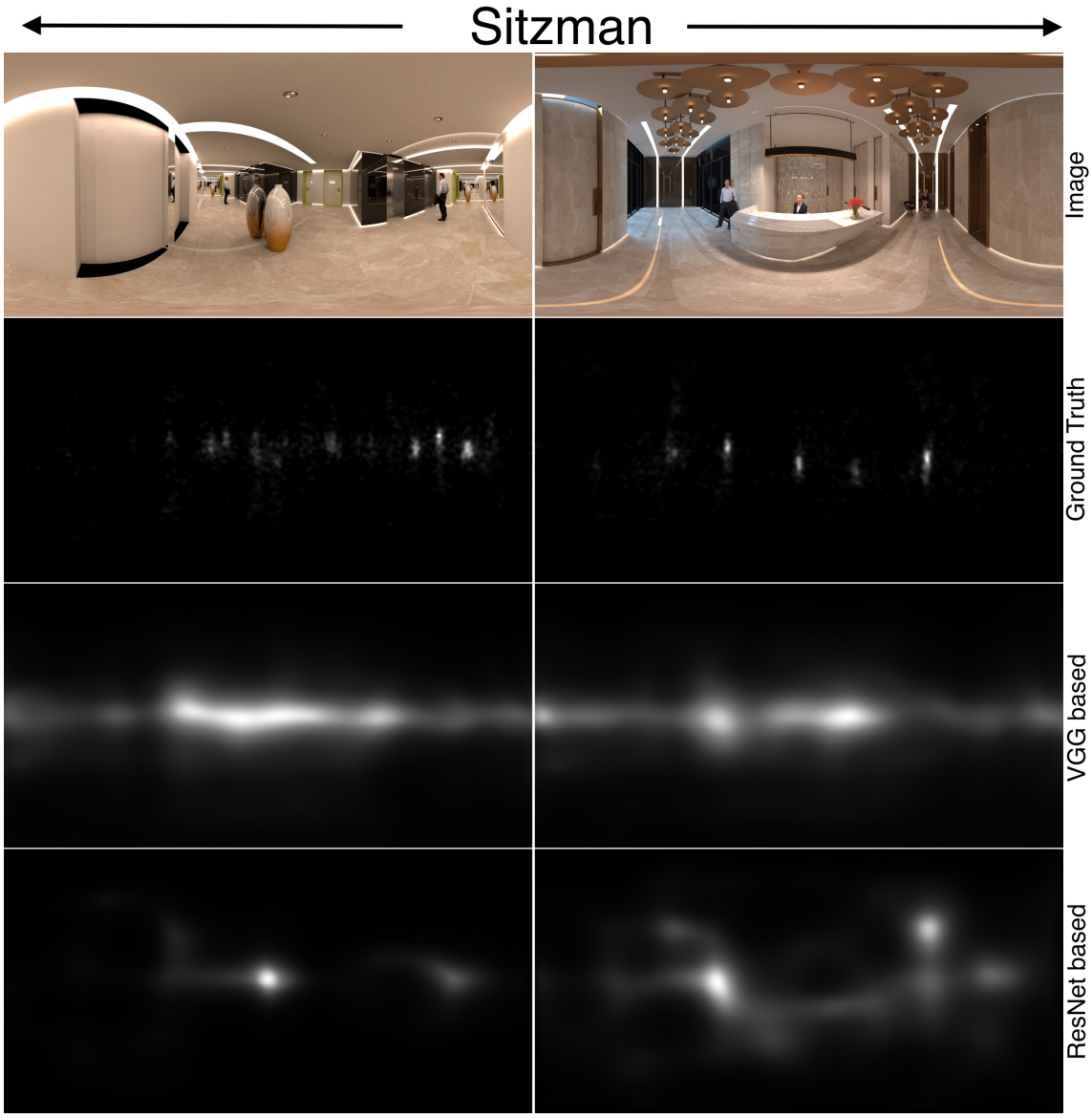


Figure 6.5: Qualitative results of model variants on sample synthetic images from Sitzman dataset.

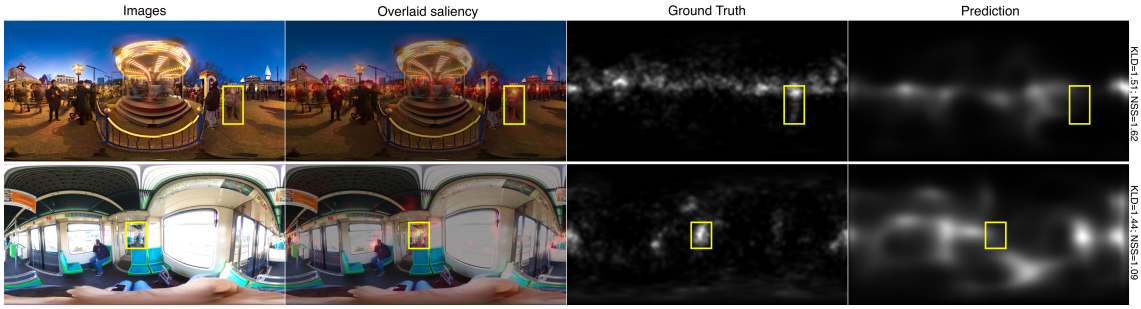


Figure 6.6: Failure modes of the model.

6.5 Ablation study.

Some variants of the proposed model were tested to validate the importance of the choices made.

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 6.3, we varied λ to 0.5, 0.7, and 0.9. As the results suggest, increasing λ improves the performance of the downstream task. Intuitively, if we look closely in Equation 6.3, giving more emphasis to $\mathcal{L}_{\mathrm{G}}$ biases the function to learn trivial solution as $\mathbf{m}_{\mathbf{x}}$ is moving average of $\mathbf{v}_{\mathbf{x}}$, 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 views of the same scene (object) and that too locally. This leads to better expressive power and generalization 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 this regime. Table 6.3 compares the VGG-based model trained with and without self-attention. A performance boost is observed irrespective of λ when using self-attention. Table 6.2 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 the 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 between 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).

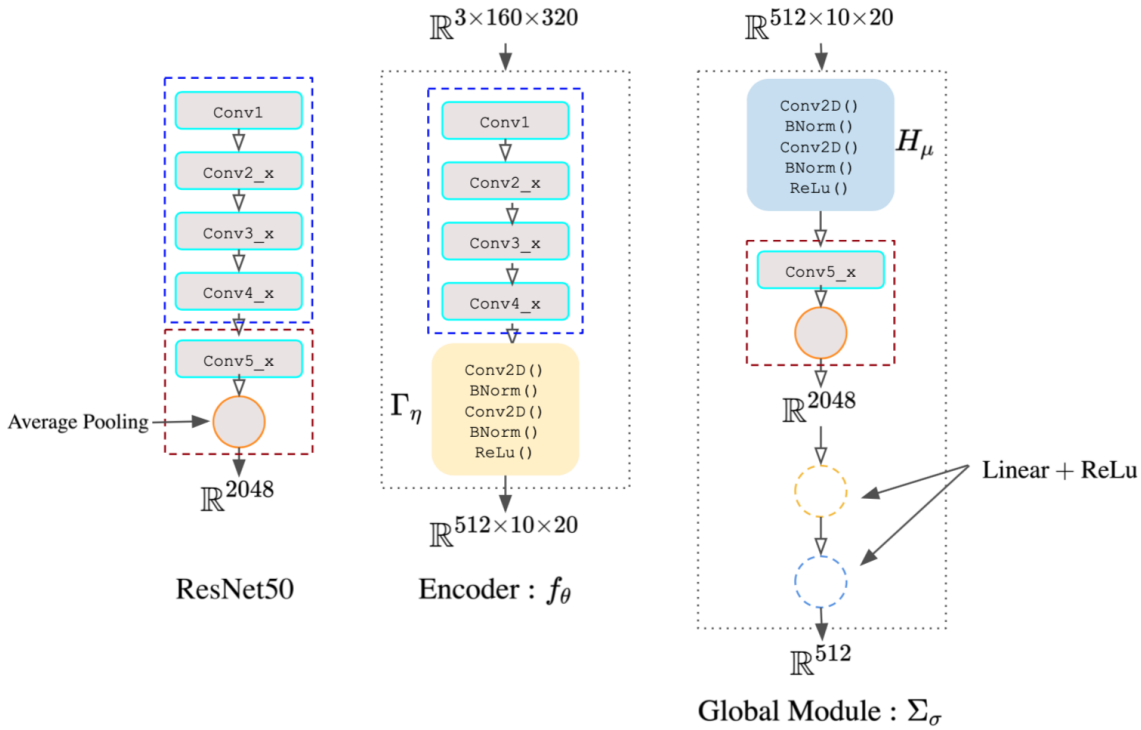


Figure 6.7: Detailed architecture for ResNet50 based encoder.

One could also argue training without augmentations and projections will lead us to NPID [Wu et al. 282] ($\lambda = 0$), and this is not our objective. Table 6.5 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. 76, Noroozi and Favaro 205], and not learning useful semantic 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. 196, Chen et al. 48]. This experiment further validates our hypotheses of exploiting the top/bottom to equator projections for contrastive learning.

6.6 Architecture Details

- **Encoder**: VGG16 encoder is derived from SalGAN [Pan et al. 212][5].
- **Global** module: It comprises of a convolution block, i.e., Conv2D → BatchNorm2D → ReLU → MaxPool2D] followed by two linear layers with rectified linear units to get a global representation.
- **Local** module: Consists of 2×[Conv2D → ReLU] followed by a BatchNorm2D layer to final local representation.

ResNet50

We give a detailed overview of the ResNet50-based encoder and its corresponding global/local module. The naming convention used is similar to [He et al. 111].

- **Encoder**: Figure 6.7 depicts the ResNet50 architecture used up to the average pooling layer, ignoring the last fully connected and consequently softmax layer. The encoder f_θ is composed of a ResNet50 up to `conv4_x` and an extra set of convolution layers denoted by Γ_η. This was done to keep the channel dimension fixed to 512 so as to be consistent with the VGG-based model.
- **Global** module Σ_σ is composed of H_μ and `conv5_x` up to the `average pooling layer` followed by two linear layers with rectified linear units. Again, H_μ was added to increase the channel dimensions to 1024 to make it compatible with the input channel dimension of `conv5_x` i.e., the original pipeline of ResNet (as we extend `conv4_x` to have 512 dimensions) to have same channel dimension of 1024.
- **Local** module is exactly same as VGG16 (local module).

$\boldsymbol{\mu}$ and $\boldsymbol{\eta}$ are parameters to sub-function H and Γ respectively.
Global module can also be considered as a projection layer as used in most of the self-supervised models, but in this case, it is applied in an asymmetric way.

[5]Code: `https://github.com/imatge-upc/salgan`

6.6.1 Self-Attention

Listing 1 depicts the PyTorch-styled implementation of the self-attention module. This takes as input 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 $512 \times 10 \times 20$. The objective of this module is to provide a mechanism to perform feature selection between the feature maps of two views. To achieve this, the latent representations are transformed into `Q` (query) and `K` (key) feature spaces through weight matrices $\mathbf{W}_{\mathrm{Q}}$ and $\mathbf{W}_{\mathrm{K}}$. Attention is calculated through dot product between the two representations, i.e. $\mathbf{W}_{\mathrm{Q}}(\Lambda_{\mathbf{x}})$ and $\mathbf{W}_{\mathrm{K}}(\Lambda_{\mathbf{x}_t})$. 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 ($\mathbf{W}_{\mathrm{Q}}(\Lambda_{\mathbf{x}})$ and $\mathbf{W}_{\mathrm{K}}(\Lambda_{\mathbf{x}_t})$), resulting in attention weights of size 200×50[6]. In Listing 1, `K` 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 (`Q`) with a patch (in `K`) because each pixel has a receptive field of 4 in K due to the max-pooling. Once we have these attention weights, we could now perform feature selection through `V` (value), i.e., $\mathbf{W}_{\mathrm{V}}(\Lambda_{\mathbf{x}_t})$. Further `O` 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 apply to the pool was made following [Zhang et al. 291][7].

```
in_channel = 512
K = 8
γ = 0 #trainable parameter

#64:=512/K,256:=512/2
W_Q = nn.Conv2d(512, 64, (1,1))
W_K = nn.Conv2d(512, 64, (1,1))
W_V = nn.Conv2d(512, 256, (1,1))
W_O = nn.conv2d(256, 512, (1,1))
```

[6] excluding batch-size for simplicity

[7] Implementation is taken from here [Zhang et al. 291] and adopted to PyTorch

```
def selfAttention(Λ_x,Λ_{x_t}):
    #Λ_x,Λ_{x_t} latent representations for x and x_t

    #Get Query
    Q = W_Q(Λ_x)#(bsz,64,10,20)
    Q = Q.view(bsz,64,-1).permute(0, 2, 1)#(bsz,200,64)

    #Get Key
    K = W_K(Λ_{x_t})#(bsz,64,10,20)
    K = F.MaxPool2D(K) #over (2,2) patches
    K = K.view(bsz, 64, -1)#Reshape (bsz,64,50)

    #Calculate importance weights
    attn = torch.bmm(Q, K) #(bsz, 200, 50)
    attn = F.softmax(attn, dim=-1) #Softmax
    attn = attn.permute(bsz,2,1)#(bsz, 50, 200)

    #Get Value
    V = W_V(Λ_{x_t})#(bsz,256,10,20)
    V = F.MaxPool2D(V) #over (2,2) patches
    V = V.view(bsz, 256, -1)#reshape (bsz,256,50)

    #Get importance vectors
    attn_val = torch.bmm(v, attn)#(bsz, 256, 200)
    attn_val = attn_val.reshape(bsz, 256, 10,20)

    #Apply another projection
    O = W_O(attn_val)

    return O*γ + Λ_{x_t}
```

Listing 6.1: PyTorch-based implementation

6.7 Conclusion

This chapter presents a method for modeling human visual attention through contrastive self-supervised learning. The proposed approach, which exploits the unique geometric properties of spherical data, demonstrates an improvement in the generalization and expressiveness of the model. By learning representations that maintain local consistency across different views, the method shows the potential to enhance the accuracy of saliency prediction. Here are a few points to consider:

Supervised vs Unsupervised learning. To the best of our knowledge, our work [Djilali et al. 73] was the first attempt to design an unsupervised learning approach for saliency prediction for 360° data. This work opened a new line of research exploring the subtle definition of gaze policy naturally embedded in the brain. [Zou et al. 304] designed a proxy task referred to as FindCMP, where the goal is to determine the position of a specific section of a Cubemap Projection (representing local detail) within the context of an Equirectangular Projection (representing the overall, global image). Subsequently, a cross-attention mechanism is employed to effectively merge and process both the detailed (local) and comprehensive (global) information, enabling them to benefit from and enhance each other's learning process. [Cokelek et al. 55] proposed an unsupervised loss function focused on consistency, serving as a regularizer to address and mitigate typical artifacts found in models that use projection techniques for 360° saliency.

In fact, early research into how the human visual system functions has produced many interesting results, demonstrating that visual attention can be influenced by regions that maximize a reward in a task-driven scenario [Sprague and Ballard 253], which are typically the most informative regions [Itti and Baldi 122, Bruce and Tsotsos 32]. 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 123, Borji and Itti 26]. Research in cognitive science (e.g., [Yan et al. 288, Li 171, Veale et al. 270]) indicates that low-level saliency in both humans and animals occurs 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. 285]. Visual attention can serve as a tool for compression approaches (e.g., saliency-aware adaptive coding [Zhu et al. 299, Luz et al. 182, Liu et al. 177]). Perceptual approaches for VQA require predicted saliency maps as weight maps [Ma and Zhang 183, Ma and Zhang 184, Li et al. 167]. Generally, the spherical characteristics of the input data mean that many of these tasks face the challenges addressed in this chapter. Thus, we suspect that the contrastive encoder can indeed be exploited for VQA and compression.

Furthermore, the empirical results from this chapter, both qualitative and quantitative, indicate that this approach is competitive with existing fully-supervised methods. This achievement is noteworthy as it represents one of the initial attempts to apply unsupervised learning techniques specifically to the task of saliency prediction in omnidirectional images. However, one issue we experienced over this and the previous chapter was benchmarking the different models, where simple baselines were nearly as effective as more sophisticated approaches. As we progress in this field, it becomes critical to address the challenges in evaluating models designed for predicting human fixations in 360° scenes. This is the focus of our next chapter, where we delve into the complexities of 360° saliency, a domain that necessitates rethinking the common 2D assumptions due to distinct factors like statistical bias variations (Equator vs Center bias).

Chapter 7

Benchmarking saliency prediction for 360° images/videos

7.1 Introduction

In Chapters 6 and 7, we focus on developing new approaches for solving the task of saliency in 360° data. Clearly, however, evaluating a model's capacity to predict human fixations in 360° scenes is a challenging task itself. Indeed, saliency on the spherical domain requires different assumptions compared to 2D as a result of the way the saliency maps are collected and pre-processed to account for the difference in statistical bias (Equator vs Center bias). However, the same classical metrics from the 2D saliency literature are typically used to evaluate 360° models. As discussed in Chapter 6, and unlike traditional 2D environments, 360° data scenes present unique challenges for saliency prediction. These challenges stem from the inherent nature of 360° environments - a spherical panorama that users can navigate freely. This freedom results in varied visual experiences influenced by individual viewing patterns and interactions with the environment [Ausin-Azofra et al. 6].

Precisely, the fundamental difference lies in the underlying geometrical representation. In 2D environments, data points are modeled in Euclidean space using

This chapter is based on our ICCV-W 2021 publication [Djilali et al. 75]

Cartesian coordinates (x, y) on a flat plane. This linear space permits the straightforward application of Euclidean metrics, which form the basis for metrics in saliency modeling. Conversely, 360° data is inherently represented in a spherical coordinate system, typically characterized by just two coordinates due to the fixed radial distance: polar angle (θ) and azimuthal angle (ϕ). This representation necessitates the use of non-Euclidean, spherical geometry for analysis and modeling. Additionally, the topology of a sphere introduces complexities absent in flat 2D spaces. For example, the concept of 'straight lines' in spherical geometry differs from that in Euclidean geometry, influencing the way visual paths and patterns are interpreted [Berger 17].

In practice, saliency is first projected onto 2D to sidestep the complexities associated with spherical geometry. This projection might introduce discrepancies arising from the unique ways subjects explore 360° scenes. Therefore, assessing a model's ability to predict human fixations in such a diverse and dynamic setting is inherently complex. This complexity is compounded by the need to adapt existing 2D saliency evaluation metrics to the 360° context. These adaptations often involve significant modifications to accommodate differences in scene composition and viewer behavior, ensuring that the metrics effectively measure model performance in a spherical environment.

Despite these challenges, the current standard practice in 360° saliency research often relies on classical 2D saliency metrics. These metrics, while robust in traditional 2D settings, may not accurately capture certain behaviors specific to 360° data. An illustrative example is the use of a simple constant predictor, such as the average map across specific training sets, which can surprisingly yield baseline results that are competitive with more specialized models. This phenomenon highlights potential flaws in the metrics when applied to 360° saliency and necessitates a reevaluation and possible redesign of these evaluation tools.

Given these limitations, this chapter proposes a new probabilistic metric specifically designed for 360° saliency evaluation. This metric is grounded in the independent Bernoulli assumption, aligning more closely with the nature of 360° scenes where each

point in the panorama could independently attract viewer attention. By adopting this approach, the metric aims to provide a more accurate and reliable assessment of models in predicting human fixations in 360° environments.

7.2 Works investigating saliency metrics

The work reported in [Riche et al. 232] performed experiments on Jian Li's human eye-tracking fixations dataset [Li et al. 170] and concluded that sAUC and KLD form a separate cluster from NSS, SIM, and CC. This is due to the fact that KLD assumes saliency maps to be strongly regularized, and sAUC does not account for the built-in center bias. The study of [Emami and Hoberock 84] randomly split the ground truth fixations into two sets to create the reference fixation map and the human fixation map. This human consistency test suggested that NSS and CC capture enough information, while KLD is the worst. Based on subjective studies, the authors of [Li et al. 169] argued that human perception is driven by the most salient and high-energy regions. Thus, NSS, CC, and SIM are the most consistent with human subjects. AUC reports unsatisfactory results, given that it relies solely on the ROC curve without considering the distribution of thresholds. This same work proposed a CNN-based metric using the users' data answers as labels. [Bylinskii et al. 37] conducted an extensive survey on all metrics and argued that properties of the inputs affect metrics differently: how the ground truth is defined, whether the prediction takes dataset statistical bias into account and whether the inputs are probabilistic. The author's main claim is that adapting the properties of metrics for the downstream task can guide metric selection for saliency model evaluation.

As discussed in Chapter 2, [Kümmerer et al. 153] attempted to address the issue of confounding factors in saliency model comparison by calculating the information gain, i.e., log-likelihood. The essence of their approach is to jointly optimize for scale, center bias, and spatial blurring as a pre-processing step for all saliency maps. This methodology led to a new consistent ranking of models. However, their approach assumes access to all models, which is not always practical. Furthermore,

the log-density method performs suboptimally on most metrics and can still produce inconsistent rankings [Kummerer et al. 154]. Inspired by Bayesian decision theory, [Kummerer et al. 154] proposed separating the saliency model from the saliency map, enabling the creation of specific saliency maps tailored to different metrics. Consequently, saliency models should be defined as metric-independent probability densities over possible fixations, allowing many different metric-dependent saliency maps to be derived from the same density for different error metrics. This approach allows saliency models to be meaningfully compared across all metrics at their original scale.

The consensus among the research community is that the metrics measure different things [Wilming et al. 281, Riche et al. 232, Bylinskii et al. 37] and that it is conceptually impossible to determine a best performing model independent of considering the different metrics. In this section, we highlight the lack of active research on this topic for 360° visual attention modeling and the different underlying assumptions for 360° data related to the visual attention mechanism [Xu et al. 285]. The main contributions of this chapter are:

- We investigate the behavior of the traditional metrics on 360° saliency datasets and show that a simple constant predictor (i.e., average saliency across the training set) can obtain high scores, calling into question the practical usefulness of these metrics under the 360° saliency setting.

- We propose an alternative formulation of the equirectangular saliency map as a probabilistic map with the Bernoulli assumption, where each pixel represents the probability of a fixation occurring.

- We adapt the probabilistic metrics (KLD and JSD) to this Bernoulli assumption and show that these new metrics are a better basis for ranking models.

7.3 Evaluation setup

The behavior of metrics and the properties of saliency maps are investigated using the following set of baselines. **The constant predictors** are the average saliency maps across the Salient360! and Sitzman dataset separately. For a dataset with N saliency maps, where each map is of size $H \times W$, the average saliency map S_avg is computed as:

$$S_{avg_{ij}} = \frac{1}{N} \sum_{k=1}^{N} S_{kij} \tag{7.1}$$

where $S_{avg_{ij}}$ represents the average saliency value at position (i, j), and S_{kij} is the saliency value at position (i, j) in the k-th saliency map.

The equator bias model is a symmetric Gaussian around the equator; it consists of a degenerate Gaussian with infinite variance in the x-direction and a variance to cover 20% of the equator in the y-direction. This choice is motivated by the statistics of the amount of fixation vs. non-fixation points of a saliency map:

$$G(y) = \frac{1}{\sigma_y \sqrt{2\pi}} \exp\left(-\frac{y^2}{2\sigma_y^2}\right) \tag{7.2}$$

where σ_y is the standard deviation in the y-direction, and y represents the latitude. The variance σ_y^2 is chosen such that the Gaussian covers 20% of the equator. This is calculated based on the range of latitude from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$. The final saliency map $S_equator$ is a function that is constant in x and varies as a Gaussian in y, reflecting a bias towards the equator in spherical images or VR environments.

The random model assigns a uniformly distributed saliency value to each pixel in the saliency map. This can be mathematically represented as follows:

For a saliency map with dimensions $H \times W$, each pixel (i, j) in the map is assigned a saliency value $S_{rand_{ij}}$ such that:

$$S_{rand_{ij}} \sim U(0, 1) \tag{7.3}$$

where $U(0, 1)$ represents a uniform distribution between the minimum possible

saliency value a and the maximum possible saliency value b. In this model, every pixel in the saliency map is equally likely to have any saliency value between 0 and 1, independently of the other pixels.

These three baselines are compared against deep saliency models in the literature using the five classical metrics (AUC-J, NSS, CC, SIM, KLD) as shown in Table 7.1. Both **the equator bias** and **the random model** are image-independent, dataset-independent models. The five metrics are significantly higher for the equator bias model, indicating the importance of capturing the dataset priors. However, the high number of zeros is detected by the KLD metric as it is sensitive to insufficient regularization. The random model performs poorly across all metrics, caused by the large amount of false positives/negatives. AUC-J is the least affected by low-valued false positives and thus reports a relatively higher score **(0.640)** compared to other metrics. We can say the metrics are somewhat robust to these two maps.

To a large extent, however, the constant predictors (image-dependent, dataset-dependent) succeed in fooling the metrics and producing scores comparable with deep learning models that were trained end-to-end on these datasets. This can be related to some potential explanations. First, the models may simply be learning to predict the average saliency across the training set. Second, it could be that the datasets are insufficiently varied, and there is a high correlation between scenes in the train and validation sets. The final interpretation relates to the representation of the saliency maps and the definition of the metrics in 360°domain. Furthermore, we study each of these hypothesize to uncover the reason behind this observation:

Issues with the models: We utilized some end-to-end supervised model to generate the saliency maps of a set of images from the validation sets (see Figure 7.1 and Figure 7.2). The models' predicted maps look more targeted and attend to specific regions of the input. Also, the correlation between the model's predictions and the constant predictor is low, suggesting that the models are not simply predicting the average training maps but are predicting more complicated mappings.

Dataset issue: According to [Gutiérrez et al. 103], the Salient360! stimuli

are influenced by various factors such as the shooting environment, the number of foreground objects, the distance to objects, and the presence of people. The eye-gaze data for this dataset was collected from 40 subjects per scene. Based on this comprehensive collection process, we hypothesize that the dataset is well-designed to reflect a diverse range of viewing conditions and viewer interactions. This challenges any explanation that the dataset's design could be a limiting factor in our analysis.

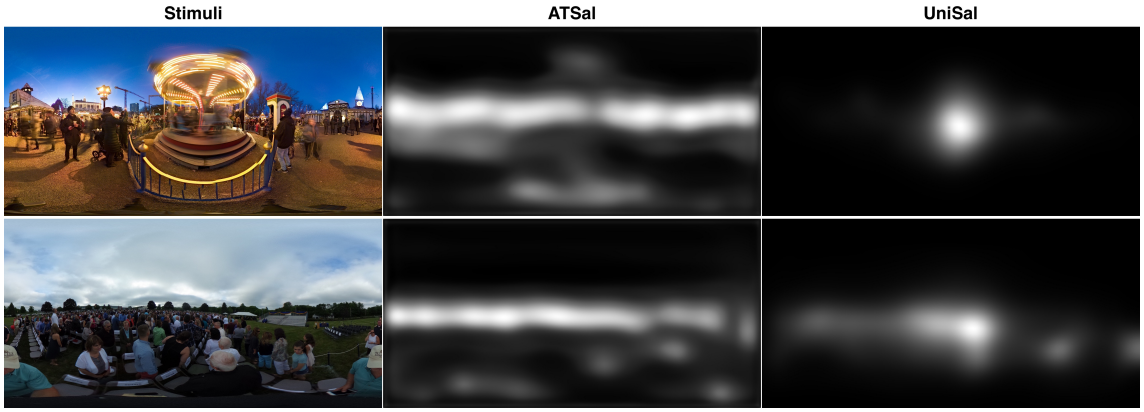


Figure 7.1: Predicted saliency maps from Salient360! dataset – samples of ATSal [Dahou et al. 60] and UniSal [Droste et al. 79].

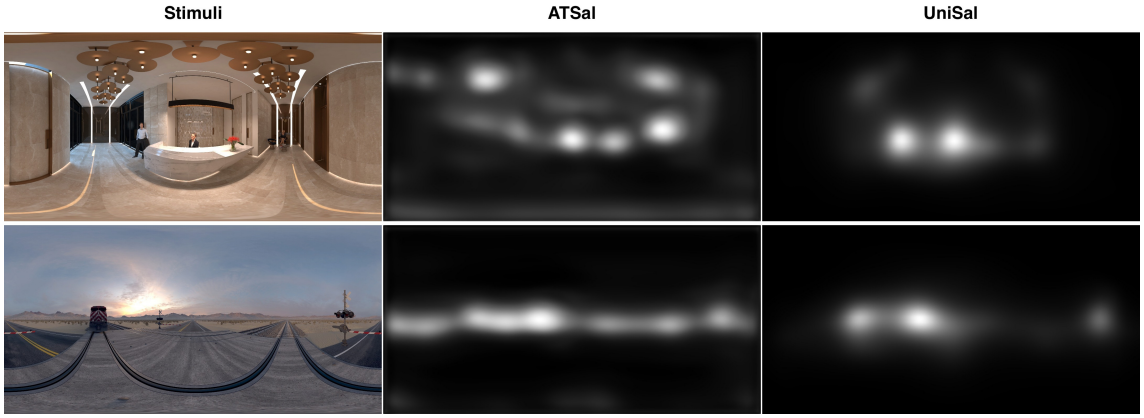


Figure 7.2: Predicted saliency maps from Sitzman dataset – samples of ATSal [Dahou et al. 60] and UniSal [Droste et al. 79].

Questioning the metrics/saliency formulation: Saliency is extracted from the coordinated motion data of the eyes and head to perform the attention task while moving in a panoramic scene. Measuring the quality of models that predict head and eye movements in 360° environments remains an open research question. The unique characteristics of 360° scenes, such as the panoramic field of view and the need to

consider a full spherical environment, could fundamentally alter the properties and behavior of traditional saliency metrics. We argue that directly applying metrics developed for 2D contexts to 360° data is problematic. This is because these metrics were not designed to accommodate the spherical distortions and the omnidirectional nature of the inputs, which may lead to diverging and inconsistent metric behaviors. This assertion is supported by observations that metrics often fail to account for the increased complexity and spatial variability inherent in 360° scenes.

7.3.1 Saliency formulation

Definition. As mentioned throughout the thesis, as a reminder, the ground truth saliency maps are computed by convolving each fixation or trajectory points (for all observers of one image) $\mathcal{M}_{\mathcal{GT}}$, with a Gaussian or Kent kernel. The resulting saliency map $S_{GT} \in [0,1]^{W \times H}$ 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. Probabilistic metrics induce a multinomial distribution on the predictions using a softmax; this assumes that viewers attend to one pixel for all time by dividing by the sum, concentrating the energy on one point.

Clearly, however, more than a single pixel may be attended, making it more appropriate to treat each predicted value as independent of the others. Thus, we believe the Bernoulli assumption makes more sense. Multivariate Bernoulli distributions are established using the concept of the Knonecker product from matrix calculus and provide an alternative to the traditional log-linear models for binary variables [Teugels 261].

f-divergences are used as general (entropic) distance-like functions to measure discrepancies between probability distributions. In this thesis, the Kullback-Leibler Divergence (KLD) measures an oriented statistical distance between two densities p and q, representing the ground truth and the model-predicted saliency maps, respectively. A common symmetrization of KLD is the Jensen-Shannon Divergence (JSD), also referred to as capacitory discrimination. JSD calculates the total KLD

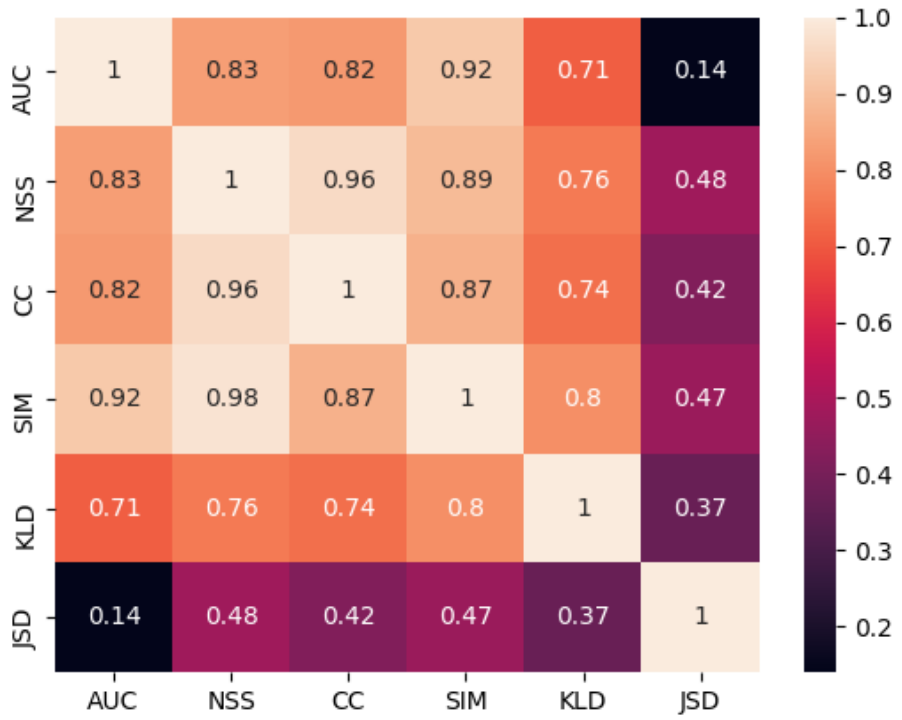


Figure 7.3: We sort the saliency models listed in Table 1 individually by each metric and compute the Spearman rank correlation between every pair of metrics. The first five metrics listed are highly correlated. JSD is most uncorrelated with other metrics due to their high sensitivity to zero-valued predictions at fixated locations.

to the average distribution $\frac{p+q}{2}$ and is particularly useful for densities with arbitrary support. Both KLD and JSD require the inputs to be valid probability distributions, typically ensured by applying a softmax function over saliency map pixels. This approach inherently assumes that there is always exactly one point fixated upon at any given time, reflecting a simplification in the modeling of viewer attention.

However, the spherical representation of the panoramic scenes allows viewers to explore many viewports (head movements) – the softmax over the ERP saliency map pixels puts most of the energy on the highly attended viewport by the average observer. We argue that treating saliency pixels as independent Bernoulli events better matches the subjects' behavior and is less harmful when evaluating models since it compares saliency maps pixel-wise and doesn't involve any prior regularization. We redefine KLD and JSD, as explained below.

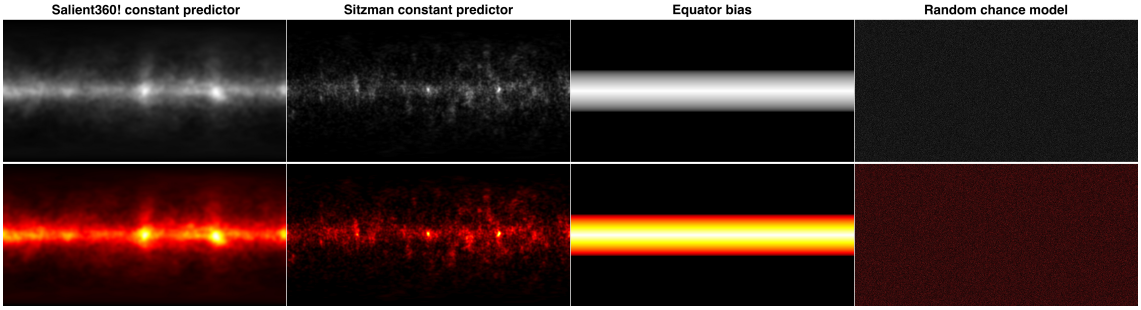


Figure 7.4: The Equi-rectangular saliency maps visualization of the four baselines. The upper row represents the density map, whereas the bottom row applies a color map to the saliency map for more perceptual figures.

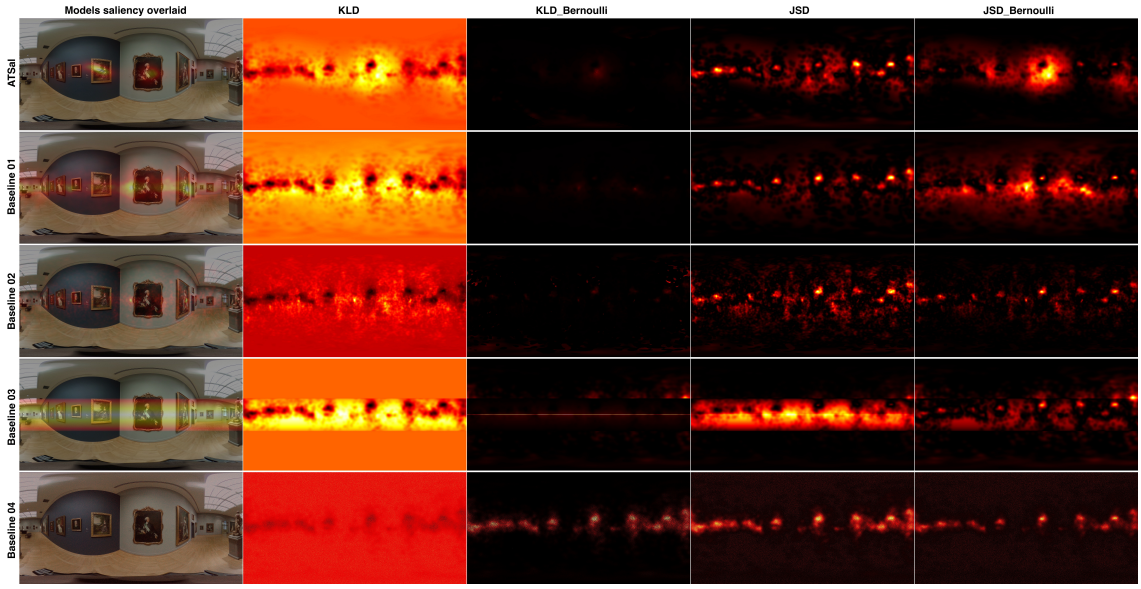


Figure 7.5: Visualizing metrics response can help us understand what behaviors of saliency models different evaluation metrics capture.

Bernoulli KL-divergence The formula for KL divergence that is usually used to evaluate saliency maps is based on the assumption that ground truth saliency q is a categorical distribution over the pixels in the image, i.e. $S_{GT} = (S_{GT_1}, \ldots, S_{GT_N}), S_{GT_i} \geq 0, S_{GT}^T \mathbf{1} = 1$. Given a softmax prediction $S_p = (S_{p_1}, \ldots, S_{p_N}), S_{p_i} \geq 0, S_p^T \mathbf{1} = 1$ the KL-divergence between the predicted distribution and the ground truth is given by:

$$\mathrm{KL}(S_{GT} \mid\mid p) = \sum_{i=1}^{N} S_{GT_i} \log \frac{S_{GT_i}}{S_{p_i}} \tag{7.4}$$

With appropriate regularization constants added for numerical stability. Here, we assume that the ground truth saliency is not normalized to sum to one but rather represents the probability that each pixel will be attended, independent of the others, i.e. $y_i \sim \mathrm{Bern}(S_{GT_i})$ where y_i is a binary RV indication that a fixation occurred at pixel i. The per-pixel KL divergence is then given by:

$$\mathrm{KL}(S_{GT_i} \mid\mid S_{p_i})_{\mathrm{Ber}} = S_{GT_i} \log \frac{S_{GT_i}}{S_{p_i}} + (1 - S_{GT_i}) \log \frac{1 - S_{GT_i}}{1 - S_{p_i}},$$

We propose to minimize the average KL-divergence between the ground truth and the predicted saliency map:

$$\mathrm{KL}(S_{GT} \mid\mid S_p)_{\mathrm{Ber}} = \frac{1}{N} \sum_{i=1}^{N} \left[S_{GT_i} \log \frac{S_{GT_i}}{S_{p_i}} + (1 - S_{GT_i}) \log \frac{1 - S_{GT_i}}{1 - S_{p_i}} \right] \tag{7.5}$$

Bernoulli Jensen-Shannon divergence The Jensen-Shannon (JS) divergence between two distributions S_{GT} and S_p defined on the same alphabet is given by:

$$\mathrm{JS}(S_{GT} \mid\mid S_p) = \frac{1}{2}\mathrm{KL}(S_p \mid\mid m) + \frac{1}{2}\mathrm{KL}(S_{GT} \mid\mid m),$$

Table 7.1: Comparative performance study on: Salient360! and Sitzman datasets. Baseline 01: the constant predictor from Salient360! dataset, Baseline 02: the constant predictor from the Sitzman dataset, Baseline 03 is the Equator bias model. Baseline 04 is the random chance model.

	Model	Salient360!									Sitzman						
		AUC-J	NSS	CC	SIM	KLD	JSD	KLD_B	JSD_B	AUC-J	NSS	CC	SIM ↑	KLD	KLD	KLD_B	JSD_B
Human		0.788	2.09	1.0	1.0	0.0	0.0	0.0	0.0	0.985	3.421	1.0	1.0	0.0	0.0	0.0	0.0
2D models	UNISAL [Droste et al. 79]	0.704	1.431	0.395	0.439	2.521	0.243	1.799	2.494	0.777	3.118	0.418	0.378	6.144	0.312	0.713	1.455
	SalGAN [Pan et al. 212]	0.690	1.335	0.287	0.439	1.723	0.242	1.902	2.396	0.654	2.505	0.109	0.190	9.717	0.470	1.510	2.320
360° models	ATSAL [Dahou et al. 60]	0.777	1.658	0.642	0.639	0.761	0.111	2.916	4.501	0.802	3.216	0.461	0.360	6.584	0.321	1.240	4.166
	SalGAN360 [Chao et al. 43]	0.774	1.621	0.613	0.617	0.881	0.125	1.143	1.295	0.830	3.136	0.428	0.387	6.176	0.302	1.184	1.617
Baselines	Baseline 01	0.768	1.635	0.593	0.603	1.022	0.132	2.327	2.920	0.777	2.978	0.340	0.295	7.827	0.375	1.729	5.196
	Baseline 02	0.725	1.530	0.491	0.550	1.744	0.170	2.414	1.978	0.735	2.948	0.321	0.301	7.939	0.373	1.5583	2.535
	Baseline 03	0.761	1.605	0.572	0.545	3.620	0.196	6.158	7.123	0.766	2.893	0.295	0.321	9.431	0.369	2.239	9.434
	Baseline 04	0.640	1.078	0.001	0.274	7.945	0.405	6.226	17.14	0.637	2.301	0.000	0.158	11.180	0.509	7.262	20.37

where $m = (S_p + S_{GT})/2$ is the midpoint distribution. Incorporating the Bernoulli KL-divergences of pixel i and simplifying gives:

$$\begin{aligned}\mathrm{JS}(S_{GT_i} \,||\, S_{p_i})_{\mathrm{Ber}} = \frac{1}{2}\Big[& S_{p_i} \log\left(\frac{2S_{p_i}}{S_{p_i} + S_{GT_i}}\right) + S_{GT_i} \log\left(\frac{2S_{p_i}}{S_{p_i} + S_{GT_i}}\right) \\ & - (S_{p_i} - 1) \log\left(\frac{2(S_{p_i} - 1)}{S_{p_i} + S_{GT_i} - 2}\right) \\ & - (S_{GT_i} - 1) \log\left(\frac{2(S_{GT_i} - 1)}{S_{p_i} + S_{GT_i} - 2}\right)\Big].\end{aligned}$$

again, we propose to minimize the average JS-divergence:

$$\mathrm{JS}(S_{GT} \,||\, S_p)_{\mathrm{Ber}} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{JS}(S_{GT_i} \,||\, S_{p_i})_{\mathrm{Ber}} \tag{7.6}$$

note that, unlike the KL-divergence, the JS-divergence is symmetric in p and S_{GT}, i.e. $\mathrm{JS}(S_p \,||\, S_{GT}) = \mathrm{JS}(S_{GT} \,||\, S_p)$.

7.4 Experiments and results

We use the Salient360! toolbox to calculate the metrics and include the implementation for the newly proposed ones. Table 7.1 represents the models' scores on distinct datasets, including 2D models, 360° specialized models and the four baselines (illustrated in Figure 7.4). The ranking of the models is highly inconsistent across metrics, and no model is optimal for all metrics. Furthermore, the constant predictors achieve comparable scores with deep learning models, except on the $\mathrm{KL}_{\mathrm{Ber}}$ and $\mathrm{JS}_{\mathrm{Ber}}$. This supports our main claim: by assuming saliency maps as multivariate Bernoulli

distributions in a principled way from model densities, deep learning models rank much better than the constant predictors.

Figure 7.3 illustrates the Spearman rank correlation across various saliency evaluation metrics. Spearman rank correlation is a non-parametric measure of rank correlation that assesses how well the relationship between two variables can be described using a monotonic function. In this study, the goal is to cluster the evaluation metrics based on their properties so that those that favor or penalize similar types of saliency behavior are grouped together. We sorted all models from Table 7.1 according to each metric and then computed the Spearman rank correlation between each pair of metrics to construct a covariance matrix, as shown in Figure 7.3. The metrics NSS (Normalized Scanpath Saliency) and CC (Correlation Coefficient) exhibit the highest correlation **(0.96)**. This high correlation can be explained by the fact that NSS is essentially a discrete version of CC, specifically adapted for saliency analysis. The pairwise correlations between AUC-J (Area Under the Curve), NSS, CC, and SIM (Similarity) range from [**0.82, 0.96**], indicating that they capture similar aspects of information and thus represent the same cluster.

KLD and JSD are sensitive to regularization and false negatives; this is beneficial if missing ground truth fixations should be severely penalized. Thus, pairwise correlation between KLD and the cluster ranges between 0.71 and 0.80. However, we demonstrated that just averaging the saliency maps on the training set can fool the metrics. For KLD, this can be explained as capturing the high energy and compact saliency region over the average scenes conditioned by the average fixation statistic; regularization to the unit sum will force the density to have one dominant mode and many smaller modes. Thus, the regularization constant ψ in the metric computation is of the same value range as the majority saliency map values. The KLD score is influenced by the distance between the most dominant modes. This may interpret the good scores of the constant predictors. Although probabilistic, the Bernoulli assumption differs from how $\mathrm{JS}_{\mathrm{Ber}}$ and $\mathrm{KL}_{\mathrm{Ber}}$ rank models because it modifies how saliency maps are defined: they measure the predictions locally pixel-wise. Despite

the fact the pixels aren't fully independent, the Bernoulli assumption is less harmful and permits the metrics to fairly score all baselines. Furthermore, the pairwise correlation between $\mathrm{JS}_{\mathrm{Ber}}$ and other metrics is low ([**0.14,0.48**]).

In Figure 7.5, we illustrate the metrics response per pixel-wise when comparing the ground truth saliency map to the models' saliency map. The red color indicates a worse score. In essence, it shows that the metric incorrectly accredited the model for incorrect predictions (false positives/negatives). Ideally, a black map means a perfect classification from the metric. For KLD, it is therefore fooled by the true positives of the constant predictors; other regions (top and bottom) are not penalized effectively, so the final score is mostly dominated by the intersection points. This may be due to the regularization constant ψ that is included in the metric computation and regularizing the input to be a valid probability distribution. Both $\mathrm{KL}_{\mathrm{Ber}}$ and $\mathrm{JS}_{\mathrm{Ber}}$ appear more stable, and better classify most pixels. Their local computation ensures these metrics do not favor the highest mode of the density from other modes. Each pixel is equally evaluated independently of the others. The fact that it expects the saliency value to be easily interpretable – as the probability that a fixation is expected to occur in that specific pixel – means that no softmax over pixels is required. This allows for a more stable computation of the metrics as fewer very small values occur on the map.

7.5 Recommendations for designing the metrics

Benchmarking saliency models remains an open research question due to inconsistent rankings when varying the metrics. We define the metrics as performance measures that assess saliency maps against ground truth fixations, and treat saliency maps as multivariate Bernoulli distributions. Our findings demonstrate that simple baselines can erroneously achieve high scores using classical metrics, highlighting the fragility of the current paradigm and the potential for misleading interpretations derived from benchmarking 360° head/eye movements prediction algorithms. However, when the same models are evaluated as multivariate Bernoulli distributions using probabilistic

metrics, they obtain lower scores, thus validating our approach. Although this chapter addresses issues related to simple baselines, the overarching research question remains unresolved. The ambiguity associated with benchmarks and metrics continues to impede algorithmic development in visual attention modeling for 360° scenes, which is crucial for various tasks. We advocate for a reevaluation of several aspects of benchmarking models. Building upon our initial findings [Djilali et al. 75]. The work of [Li et al. 168] conducted an analysis to understand why classical metrics tend to under-penalize the equator bias. Additionally, Goude et al. [Goudé et al. 98] further explored our results by comparing the impacts of single-subject fixations versus averaged fixations across all subjects.

Optimizing for the downstream task. Task-free saliency is the most common paradigm for most benchmarks. In practice, however, head/eye movement prediction is application dependant, such as in the case of visual quality assessment [Duan et al. 80], or compression and transmission for 360° video/images [Park et al. 215]. Each task holds specific assumptions on the expected saliency map. For region-based applications (e.g. compression and adaptive transmission), a location-based and local metric such as NSS would be more appropriate. However, if the task requires penalizing missing a fixation (e.g. object detection, surveillance, segmentation), metrics highly affected by false negatives (KLD and AUC) make more sense. This would require designing new task-specific datasets.

Decoupling the saliency model, map, and metric. Extending the work of [Kummerer et al. 154] to 360° saliency with careful domain considerations is interesting. Following the Bayesian decision theory, the saliency model is a posterior density over possible fixations, and the saliency metric is a utility function. Based on the posterior density and the utility function, a saliency map is then chosen to maximize the expected utility. Thus, for each metric, a specific saliency map is derived from the model density. This has proved to be effective in getting a single winner on the 2D saliency benchmark MIT300 [Judd et al. 135]. However, adapting the method to head/eye movement prediction requires rethinking the method developed

in [Kümmerer et al. 153] in the context of omnidirectional datasets.

Considering the spherical geometry in the metric computation. The ground truth saliency maps are represented in the equirectangular format [Gutiérrez et al. 103], which requires using helical sampling on the sphere. The metrics are calculated over the matching points at the polar positions. Thus, the scores are sensitive to the number of sampled points. Ideally, a measure should be invariant of the pre-processing steps. It is likely that representing the spherical specifications in the metric computation would be more effective.

7.6 Conclusion

This chapter points out a major flaw in the existing metrics to evaluate 360° saliency models. We have shown that just by using a constant saliency map; one can already achieve similar results as more sophisticated methods on the Salient360! and Sitzman dataset, a surprising finding. This means that deep learning-based saliency models are no better than the average training saliency maps according to these metrics. Therefore, the presented analysis suggests defining saliency maps as multivariate Bernoulli distributions as an alternative formulation. This is also supported by our understanding of how humans explore omnidirectional scenes. In practice, this means deriving new formulas for the probabilistic metrics. In this way, the baselines are detected to be poorer quality maps, whilst specialized models score better across these probabilistic metrics.

Chapter 8

Conclusion

The main contribution of this thesis is the integration of learning representations with deep learning in the context of saliency prediction, which is applied to both 2D and 360° visual data. The objective has been to provide a more robust approach to understanding and predicting visual attention. In chapters 3-4, we investigated how Vision Transformers (ViTs) align with human visual attention. Further, the "Saliency TRansformer (SalTR)" proves the adaptability of transformer-based models, highlighting the benefit of shifting from conventional continuous saliency maps to learning from discrete fixation maps. Then, we shifted to 360° data and proposed methods to address the geometrical challenges present in this modality.

Having introduced the main body of our work in Parts 1 and 2 of this thesis, we can now attempt to provide answers to our research questions posed in Chapter 1 and point towards interesting directions for future work. The first two research questions address learning on 2D data, covered in Part 1 of this thesis, whereas the other questions are concerned with learning on 360° data in Part 2 of this thesis.

Research Question 1: *How do the Vision Transformers (ViTs) align with human visual attention? Can self-attention maps from these transformers be translated into saliency maps?*

Our investigations in ([Djilali et al. 72], Chapter 3) have shown a remarkable alignment between the penultimate self-attention maps of Vision Transformers and conventional saliency maps used in human visual attention studies. This alignment

underscores the potential of ViTs in closely mimicking human visual processing patterns. The proposed approach, which combines tensor decomposition with a shallow convolutional neural network decoder, effectively translates self-attention maps into accurate saliency maps. This method connects ViTs with human visual attention.

Further, our comparative analysis between ViTs trained under supervised classification tasks and those in self-supervised regimes revealed a notable distinction. The self-supervised ViTs demonstrate a superior ability to generalize and adapt in saliency prediction tasks. In contrast, supervised models tend to focus collapse to the class labels; hence, they are not a good fit for saliency prediction.

Inspired by recent developments such as DINOv2 [Oquab et al. 209] and the CLIP [Radford et al. 225] model trained on LVD and LAION-2B [Schuhmann et al. 242]. An interesting open questions remain, future work could focus on these advanced self-supervised models, which might align even better with saliency information. Moreover, exploring weakly language-supervised models, like CLIP, could uncover new perspectives on how visual attention aligns with linguistic contexts, further enriching the capabilities of ViTs in modeling human-like attention.

Research Question 2: Can we optimize for discrete fixation maps to reflect actual human saccades? Can transformers be tailored to predict fixation points by learning directly from fixation maps?

The Saliency Transformer (SalTR) model, outlined in this ([Djilali et al. 74], Chapter 4), forecasts the saliency prediction as a set prediction problem. The model employs fixation queries that mimic viewer attention, focusing on image features and saccading to spatial positions that maximize the attention task. This approach not only improves the accuracy of saliency prediction but also offers a controllable and steerable property that is absent in existing saliency models. Such control is especially beneficial in applications like adaptive image and video compression, where the level of desired compression can dictate the number of fixation queries and the parameters of the Gaussian filter used for smoothing.

The use of the Hungarian algorithm for matching predicted fixation points with ground truth data and the incorporation of a normalized scanpath saliency (NSS) loss further refine the model's performance. These elements ensure that SalTR not only predicts fixation points accurately but also extends to the task of scan path prediction by handling ordered fixation predictions (i.e., scanpaths). Hence, we can answer Research Question 2 in the affirmative.

Research Question 3: Can we combine local and global features for learning better 360 video saliency?

We introduced the ATSal model in ([Dahou et al. 61], Chapter 5), a novel approach for enhancing 360° video saliency prediction by merging local and global features. ATSal features a dual-stream architecture, with one stream dedicated to extracting global saliency information and the other stream focused on local feature extraction through expert models trained for specific viewports. Then, an attention mechanism is used to select the information from both streams. This design enables ATSal to effectively capture the broad context of the entire scene while also extracting detailed local features. The fusion of these streams results in a comprehensive saliency map, demonstrating that the integration of local and global features improves the accuracy and robustness of 360° video saliency predictions. Thus, it provides an answer to Research Question 3.

Research Question 4: How can we learn a unified representation for 360 image saliency?

In ([Djilali et al. 73], Chapter 6), we presented a method that effectively incorporates the unique spherical nature of 360° data in the training procedure using contrastive self-supervised learning. Central to this approach is the sophisticated use of an encoder, trained to capture the distortion of the polar regions and exploit symmetries inherent in spherical data. The architecture integrates both global and local fine-grained spatial details. A self-attention mechanism further enriches this representation, allowing the model to focus adaptively on relevant parts of the scene from various viewpoints. The method employs a Noise-Contrastive Estimation-based

loss function augmented with a memory bank strategy for efficient learning. This setup distinguishes between positive and negative sample pairs, enabling the model to learn discriminative features effectively. A decoder module, trained to minimize the Kullback-Leibler Divergence between predicted and actual saliency maps, transforms the learned representations into accurate saliency predictions.

Overall, this chapter demonstrates that a unified representation for 360° image saliency is achievable through a carefully designed contrastive self-supervised learning framework. This method adeptly addresses the challenges posed by the spherical nature of the data, leading to robust and accurate saliency predictions. While this represents a significant advancement in the field, it also opens avenues for future research to further refine and extend these techniques.

Research Question 5: Are the classical saliency metrics suitable for the 360° data?

We carried out an in-depth examination of classical saliency metrics in the context of 360° data in ([Djilali et al. 75], Chapter 7); our findings present an understanding of their applicability and limitations. The exploration revealed that while traditional metrics such as AUC-J, NSS, CC, SIM, and KLD have been foundational in 2D saliency prediction, they exhibit inherent inadequacies when applied to measure saliency on spherical data. These metrics, initially designed for 2D environments, struggle to account for the unique visual attention dynamics and viewer behaviors present in 360° scenes, such as the varied viewing patterns and the equator versus center biases.

We then introduced and validated a novel probabilistic metric that fits the specifically of 360° saliency evaluation. This metric, based on the independent Bernoulli assumption, aligns more closely with the nature of 360° environments, where each point in the sphere could independently attract viewer attention.

Motivated by questions around how attention shapes the world of perception, what we see, and what we become consciously aware of versus what our brain registers

but we do not become aware of, this thesis was motivated by the idea of using deep learning approaches to model human visual attention. Addressing how sensory data reaching the human's eyes influences their attention, and considering the observation that humans focus on the same parts of an image on average (i.e., salient regions) when exploring a dataset of images, this work has taken us on a journey towards designing novel architectures for saliency prediction both on 2D and 360° visual data. The primary objective was to enhance the understanding and prediction of visual attention. The research investigates how Vision Transformers (ViTs) align with human visual attention, as detailed in Chapter 3. The introduction of the "Saliency TRansformer (SalTR)" model in Chapter 4 demonstrates the adaptability of transformer-based models in shifting from conventional continuous saliency maps to discrete fixation maps. Finally, we delve into proposing new models for the 360° image/video saliency prediction in Chapters 5 and 6.

In summary, the thesis provides deeper insights into the alignment of deep learning models with human visual attention but also opens up new possibilities for future research in this evolving field.

Bibliography

[1] H Abdi and LJ Williams. "Principal component analysis". In: *Wiley interdisciplinary reviews: computational statistics* 2.4 (2010), pp. 433–459.

[2] S Abnar and W Zuidema. "Quantifying attention flow in transformers". In: *arXiv preprint arXiv:2005.00928* (2020).

[3] S Adel Bargal, A Zunino, D Kim, J Zhang, V Murino, and S Sclaroff. "Excitation backprop for RNNs". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1440–1449.

[4] M Assens, X Giro-i Nieto, K McGuinness, and NE O'Connor. "PathGAN: Visual scanpath prediction with generative adversarial networks". In: *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 2018, pp. 0–0.

[5] M Assens, X Giro-i Nieto, K McGuinness, and NE O'Connor. "Scanpath and saliency prediction on 360 degree images". In: *Signal Processing: Image Communication* 69 (2018), pp. 8–14.

[6] JM Ausin-Azofra, E Bigne, C Ruiz, J Marín-Morales, J Guixeres, and M Alcañiz. "Do you see what i see? Effectiveness of 360-Degree vs. 2D video ads using a neuroscience approach". In: *Frontiers in Psychology* 12 (2021), p. 612717.

[7] S Bach, A Binder, G Montavon, F Klauschen, KR Müller, and W Samek. "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation". In: *PloS one* 10.7 (2015), e0130140.

[8] P Bachman, RD Hjelm, and W Buchwalter. "Learning representations by maximizing mutual information across views". In: *arXiv preprint arXiv:1906.00910* (2019).

[9] A Baevski, Y Zhou, A Mohamed, and M Auli. "wav2vec 2.0: A framework for self-supervised learning of speech representations". In: *Advances in neural information processing systems* 33 (2020), pp. 12449–12460.

[10] D Bahdanau, K Cho, and Y Bengio. "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473* (2014).

[11] C Bak, A Kocak, E Erdem, and A Erdem. "Spatio-temporal saliency networks for dynamic saliency prediction". In: *IEEE Transactions on Multimedia* 20.7 (2018), pp. 1688–1698.

[12] Y Bao, T Zhang, A Pande, H Wu, and X Liu. "Motion-prediction-based multicast for 360-degree video transmissions". In: *2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE. 2017, pp. 1–9.

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

[14] MA Bautista, A Sanakoyeu, E Sutter, and B Ommer. "Cliquecnn: Deep unsupervised exemplar learning". In: *arXiv preprint arXiv:1608.08792* (2016).

[15] R Bellman. "Dynamic programming". In: *Science* 153.3731 (1966), pp. 34–37.

[16] I Bello, B Zoph, A Vaswani, J Shlens, and QV Le. "Attention augmented convolutional networks". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 3286–3295.

[17] M Berger. *Geometry i*. Springer Science & Business Media, 2009.

[18] A Binder, G Montavon, S Lapuschkin, KR Müller, and W Samek. "Layer-wise relevance propagation for neural networks with local renormalization layers". In: *Artificial Neural Networks and Machine Learning–ICANN 2016: 25th International Conference on Artificial Neural Networks, Barcelona, Spain, September 6-9, 2016, Proceedings, Part II 25*. Springer. 2016, pp. 63–71.

[19] G Boccignone. "Nonparametric Bayesian attentive video analysis". In: *2008 19th International Conference on Pattern Recognition*. IEEE. 2008, pp. 1–4.

[20] G Boccignone and M Ferraro. "Modelling gaze shift as a constrained random walk". In: *Physica A: Statistical Mechanics and its Applications* 331.1-2 (2004), pp. 207–218.

[21] I Bogdanova, A Bur, and H Hugli. "Visual attention on the sphere". In: *IEEE Transactions on Image Processing* 17.11 (2008), pp. 2000–2014.

[22] I Bogdanova, A Bur, H Hügli, and PA Farine. "Dynamic visual attention on the sphere". In: *Computer Vision and Image Understanding* 114.1 (2010), pp. 100–110.

[23] A Borji. "Saliency prediction in the deep learning era: An empirical investigation". In: *arXiv preprint arXiv:1810.03716* (2018).

[24] A Borji. "Saliency prediction in the deep learning era: Successes and limitations". In: *IEEE transactions on pattern analysis and machine intelligence* 43.2 (2019), pp. 679–700.

[25] A Borji and L Itti. "Cat2000: A large scale fixation dataset for boosting saliency research". In: *arXiv preprint arXiv:1505.03581* (2015).

[26] A Borji and L Itti. "State-of-the-art in visual attention modeling". In: *IEEE transactions on pattern analysis and machine intelligence* 35.1 (2012), pp. 185–207.

[27] A Borji and L Itti. "State-of-the-art in visual attention modeling". In: *IEEE transactions on pattern analysis and machine intelligence* 35.1 (2013), pp. 185–207.

[28] A Borji, DN Sihite, and L Itti. “Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study”. In: *IEEE Transactions on Image Processing* 22.1 (2012), pp. 55–69.

[29] A Borji, HR Tavakoli, DN Sihite, and L Itti. “Analysis of scores, datasets, and models in visual saliency prediction”. In: *Proceedings of the IEEE international conference on computer vision*. 2013, pp. 921–928.

[30] T Brown, B Mann, N Ryder, M Subbiah, JD Kaplan, P Dhariwal, A Neelakantan, P Shyam, G Sastry, A Askell, et al. “Language models are few-shot learners”. In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[31] N Bruce and J Tsotsos. “Attention based on information maximization”. In: *Journal of Vision* 7.9 (2007), pp. 950–950.

[32] N Bruce and J Tsotsos. “Saliency based on information maximization”. In: *Advances in neural information processing systems*. 2006, pp. 155–162.

[33] NJ Butko and JR Movellan. “Optimal scanning for faster object detection”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2009, pp. 2751–2758.

[34] Z Bylinskii, T Judd, A Borji, L Itti, F Durand, A Oliva, and A Torralba. “Mit saliency benchmark”. In: (2015).

[35] Z Bylinskii, T Judd, A Borji, L Itti, F Durand, A Oliva, and A Torralba. “Mit saliency benchmark”. In: (2015).

[36] Z Bylinskii, T Judd, A Oliva, A Torralba, and F Durand. “What do different evaluation metrics tell us about saliency models?” In: *arXiv preprint arXiv:1604.03605* (2016).

[37] Z Bylinskii, T Judd, A Oliva, A Torralba, and F Durand. “What do different evaluation metrics tell us about saliency models?” In: *IEEE transactions on pattern analysis and machine intelligence* 41.3 (2018), pp. 740–757.

[38] GA Calvert. "Crossmodal processing in the human brain: insights from functional neuroimaging studies". In: *Cerebral cortex* 11.12 (2001), pp. 1110–1123.

[39] N Carion, F Massa, G Synnaeve, N Usunier, A Kirillov, and S Zagoruyko. "End-to-end object detection with transformers". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer. 2020, pp. 213–229.

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

[41] M Caron, I Misra, J Mairal, P Goyal, P Bojanowski, and A Joulin. "Unsupervised learning of visual features by contrasting cluster assignments". In: *arXiv preprint arXiv:2006.09882* (2020).

[42] M Caron, H Touvron, I Misra, H Jégou, J Mairal, P Bojanowski, and A Joulin. "Emerging properties in self-supervised vision transformers". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 9650–9660.

[43] FY Chao, L Zhang, W Hamidouche, and O Deforges. "Salgan360: Visual saliency prediction on 360 degree images with generative adversarial networks". In: *2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE. 2018, pp. 01–04.

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

[45] H Chefer, S Gur, and L Wolf. "Transformer interpretability beyond attention visualization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 782–791.

[46] LC Chen, G Papandreou, I Kokkinos, K Murphy, and AL Yuille. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs". In: *IEEE transactions on pattern analysis and machine intelligence* 40.4 (2017), pp. 834–848.

[47] LC Chen, Y Yang, J Wang, W Xu, and AL Yuille. "Attention to scale: Scale-aware semantic image segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3640–3649.

[48] T Chen, S Kornblith, M Norouzi, and G Hinton. "A simple framework for contrastive learning of visual representations". In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607.

[49] X Chen, H Fan, R Girshick, and K He. "Improved baselines with momentum contrastive learning". In: *arXiv preprint arXiv:2003.04297* (2020).

[50] HT Cheng, CH Chao, JD Dong, HK Wen, TL Liu, and M Sun. "Cube padding for weakly-supervised saliency prediction in 360 videos". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1420–1429.

[51] K Cho, B Van Merriënboer, D Bahdanau, and Y Bengio. "On the properties of neural machine translation: Encoder-decoder approaches". In: *arXiv preprint arXiv:1409.1259* (2014).

[52] SH Chou, C Sun, WY Chang, WT Hsu, M Sun, and J Fu. "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*. 2020, pp. 845–853.

[53] MM Chun and NB Turk-Browne. "Interactions between attention and memory". In: *Current opinion in neurobiology* 17.2 (2007), pp. 177–184.

[54] J Chung, C Gulcehre, K Cho, and Y Bengio. "Empirical evaluation of gated recurrent neural networks on sequence modeling". In: *arXiv preprint arXiv:1412.3555* (2014).

[55] M Cokelek, N Imamoglu, C Ozcinar, E Erdem, and A Erdem. "Spherical Vision Transformer for 360-degree Video Saliency Prediction". In: *arXiv preprint arXiv:2308.13004* (2023).

[56] JB Cordonnier, A Loukas, and M Jaggi. "On the relationship between self-attention and convolutional layers". In: *arXiv preprint arXiv:1911.03584* (2019).

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

[58] M Cornia, L Baraldi, G Serra, and R Cucchiara. "Predicting human eye fixations via an lstm-based saliency attentive model". In: *IEEE Transactions on Image Processing* 27.10 (2018), pp. 5142–5154.

[59] ED Cubuk, B Zoph, J Shlens, and QV Le. "Randaugment: Practical automated data augmentation with a reduced search space". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2020, pp. 702–703.

[60] Y Dahou, M Tliba, K McGuinness, and N O'Connor. "ATSal: An Attention Based Architecture for Saliency Prediction in 360 Videos". In: *Pattern Recognition. ICPR International Workshops and Challenges*. Ed. by A Del Bimbo, R Cucchiara, S Sclaroff, GM Farinella, T Mei, M Bertini, HJ Escalante, and R Vezzani. Cham: Springer International Publishing, 2021, pp. 305–320.

[61] Y Dahou, M Tliba, K McGuinness, and N O'Connor. "ATSal: An Attention Based Architecture for Saliency Prediction in 360◦ Videos". In: *International Conference on Pattern Recognition*. Springer. 2021, pp. 305–320.

[62] YA DAHOU DJILALI, M Sayah., K McGuinness., and NE O'Connor. "3DSAL: An Efficient 3D-CNN Architecture for Video Saliency Prediction". In: *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VIS-*

APP, INSTICC. SciTePress, 2020, pp. 27–36. ISBN: 978-989-758-402-2. DOI: 10.5220/0008875600270036.

[63] J Dai, H Qi, Y Xiong, Y Li, G Zhang, H Hu, and Y Wei. "Deformable convolutional networks". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 764–773.

[64] Z Dai, B Cai, Y Lin, and J Chen. "Up-detr: Unsupervised pre-training for object detection with transformers". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 1601–1610.

[65] EJ David, J Gutiérrez, A Coutrot, MP Da Silva, and PL Callet. "A dataset of head and eye movements for 360 videos". In: *Proceedings of the 9th ACM Multimedia Systems Conference*. 2018, pp. 432–437.

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

[67] D DeCarlo and A Santella. "Stylization and abstraction of photographs". In: *ACM transactions on graphics (TOG)* 21.3 (2002), pp. 769–776.

[68] J Deng, W Dong, R Socher, LJ Li, K Li, and L Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[69] J Devlin, MW Chang, K Lee, and K Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[70] R Dewhurst, M Nyström, H Jarodzka, T Foulsham, R Johansson, and K Holmqvist. "It depends on how you look at it: Scanpath comparison in multiple dimensions with MultiMatch, a vector-based approach". In: *Behavior research methods* 44 (2012), pp. 1079–1100.

[71] G Ding, N İmamoğlu, A Caglayan, M Murakawa, and R Nakamura. "SalFBNet: Learning pseudo-saliency distribution via feedback convolutional networks". In: *Image and Vision Computing* 120 (2022), p. 104395.

[72] YAD Djilali, K McGuinness, and NE O'Connor. *Vision Transformers are Inherently Saliency Learners*. 2023. URL: `https://papers.bmvc2023.org/0771.pdf` (visited on 2023).

[73] YAD Djilali, T Krishna, K McGuinness, and NE O'Connor. "Rethinking 360deg Image Visual Attention Modelling With Unsupervised Learning." In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 15414–15424.

[74] YAD Djilali, K McGuiness, and N O'Connor. "Learning Saliency From Fixations". In: *arXiv preprint arXiv:2311.14073* (2023).

[75] YAD Djilali, K McGuinness, and NE O'Connor. "Simple baselines can fool 360 saliency metrics". In: *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. IEEE. 2021, pp. 3743–3749.

[76] C Doersch, A Gupta, and AA Efros. "Unsupervised visual representation learning by context prediction". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1422–1430.

[77] P Domingos. "Every Model Learned by Gradient Descent Is Approximately a Kernel Machine". In: *arXiv preprint arXiv:2012.00152* (2020).

[78] A Dosovitskiy, L Beyer, A Kolesnikov, D Weissenborn, X Zhai, T Unterthiner, M Dehghani, M Minderer, G Heigold, S Gelly, et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[79] R Droste, J Jiao, and JA Noble. "Unified image and video saliency modeling". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*. Springer. 2020, pp. 419–435.

[80] H Duan, G Zhai, X Min, Y Zhu, Y Fang, and X Yang. "Perceptual quality assessment of omnidirectional images". In: *2018 IEEE international symposium on circuits and systems (ISCAS)*. IEEE. 2018, pp. 1–5.

[81] S d'Ascoli, H Touvron, ML Leavitt, AS Morcos, G Biroli, and L Sagun. "Convit: Improving vision transformers with soft convolutional inductive biases". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 2286–2296.

[82] KA Ehinger, B Hidalgo-Sotelo, A Torralba, and A Oliva. "Modelling search for people in 900 scenes: A combined source model of eye guidance". In: *Visual cognition* 17.6-7 (2009), pp. 945–978.

[83] L Elazary and L Itti. "Interesting objects are visually salient". In: *Journal of vision* 8.3 (2008), pp. 3–3.

[84] M Emami and LL Hoberock. "Selection of a best metric and evaluation of bottom-up visual saliency models". In: *Image and Vision Computing* 31.10 (2013), pp. 796–808.

[85] U Engelke and P Le Callet. "Perceived interest and overt visual attention in natural images". In: *Signal Processing: Image Communication* 39 (2015), pp. 386–404.

[86] U Engelke, H Liu, J Wang, P Le Callet, I Heynderickx, HJ Zepernick, and A Maeder. "Comparative study of fixation density maps". In: *IEEE Transactions on Image Processing* 22.3 (2012), pp. 1121–1133.

[87] S Fan, Z Shen, M Jiang, BL Koenig, J Xu, MS Kankanhalli, and Q Zhao. "Emotional attention: A study of image sentiment and visual attention". In: *Proceedings of the IEEE Conference on computer vision and pattern recognition*. 2018, pp. 7521–7531.

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

[89] NP Friedman and TW Robbins. “The role of prefrontal cortex in cognitive control and executive function”. In: *Neuropsychopharmacology* 47.1 (2022), pp. 72–89.

[90] D Gao and N Vasconcelos. “Discriminant saliency for visual recognition from cluttered scenes”. In: *Advances in neural information processing systems*. 2005, pp. 481–488.

[91] A Garcia-Diaz, XR Fdez-Vidal, XM Pardo, and R Dosil. “Decorrelation and distinctiveness provide with human-like saliency”. In: *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer. 2009, pp. 343–354.

[92] A Garcia-Diaz, XR Fdez-Vidal, XM Pardo, and R Dosil. “Saliency from hierarchical adaptation through decorrelation and variance normalization”. In: *Image and Vision Computing* 30.1 (2012), pp. 51–64.

[93] R Geirhos, K Meding, and FA Wichmann. “Beyond accuracy: quantifying trial-by-trial behaviour of CNNs and humans by measuring error consistency”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 13890–13902.

[94] S Gidaris, P Singh, and N Komodakis. “Unsupervised representation learning by predicting image rotations”. In: *arXiv preprint arXiv:1803.07728* (2018).

[95] S Goferman, L Zelnik-Manor, and A Tal. “Context-aware saliency detection”. In: *IEEE transactions on pattern analysis and machine intelligence* 34.10 (2011), pp. 1915–1926.

[96] S Goferman, L Zelnik-Manor, and A Tal. “Context-aware saliency detection”. In: *IEEE transactions on pattern analysis and machine intelligence* 34.10 (2012), pp. 1915–1926.

[97] I Goodfellow, J Pouget-Abadie, M Mirza, B Xu, D Warde-Farley, S Ozair, A Courville, and Y Bengio. “Generative adversarial nets”. In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[98] I Goudé, A Bruckert, AH Olivier, J Pettré, R Cozot, K Bouatouch, M Christie, and L Hoyet. "Real-time Multi-map Saliency-driven Gaze Behavior for Non-conversational Characters". In: *IEEE Transactions on Visualization and Computer Graphics* (2023).

[99] A Graves, G Wayne, and I Danihelka. "Neural turing machines". In: *arXiv preprint arXiv:1410.5401* (2014).

[100] A Gretton, O Bousquet, A Smola, and B Schölkopf. "Measuring statistical dependence with Hilbert-Schmidt norms". In: *International conference on algorithmic learning theory*. Springer. 2005, pp. 63–77.

[101] JB Grill, F Strub, F Altché, C Tallec, PH Richemond, E Buchatskaya, C Doersch, BA Pires, ZD Guo, MG Azar, et al. "Bootstrap your own latent: A new approach to self-supervised learning". In: *arXiv preprint arXiv:2006.07733* (2020).

[102] C Guo, Q Ma, and L Zhang. "Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2008, pp. 1–8.

[103] J Gutiérrez, E David, Y Rai, and P Le Callet. "Toolbox and dataset for the development of saliency and scanpath models for omnidirectional/360 still images". In: *Signal Processing: Image Communication* 69 (2018), pp. 35–42.

[104] J Gutiérrez, EJ David, A Coutrot, MP Da Silva, and P Le Callet. "Introducing un salient360! benchmark: A platform for evaluating visual attention models for 360 contents". In: *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE. 2018, pp. 1–3.

[105] M Gutmann and A Hyvärinen. "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings. 2010, pp. 297–304.

[106] MU Gutmann and A Hyvärinen. “Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics.” In: *Journal of Machine Learning Research* 13.2 (2012).

[107] J Harel, C Koch, and P Perona. “Graph-based visual saliency”. In: *Advances in neural information processing systems* 19 (2006).

[108] J Harel, C Koch, and P Perona. “Graph-based visual saliency”. In: (2007).

[109] M Hayhoe and D Ballard. “Eye movements in natural behavior”. In: *Trends in cognitive sciences* 9.4 (2005), pp. 188–194.

[110] K He, H Fan, Y Wu, S Xie, and R Girshick. “Momentum contrast for unsupervised visual representation learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 9729–9738.

[111] K He, X Zhang, S Ren, and J Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[112] D Hendrycks and K Gimpel. “Gaussian error linear units (gelus)”. In: *arXiv preprint arXiv:1606.08415* (2016).

[113] RD Hjelm, A Fedorov, S Lavoie-Marchildon, K Grewal, P Bachman, A Trischler, and Y Bengio. “Learning deep representations by mutual information estimation and maximization”. In: *arXiv preprint arXiv:1808.06670* (2018).

[114] S Hochreiter and J Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[115] H Hotelling. “The generalization of Student’s ratio”. In: *Breakthroughs in statistics: Foundations and basic theory*. Springer, 1992, pp. 54–65.

[116] X Hou, J Harel, and C Koch. “Image signature: Highlighting sparse salient regions”. In: *IEEE transactions on pattern analysis and machine intelligence* 34.1 (2011), pp. 194–201.

[117] X Hou and L Zhang. "Dynamic visual attention: Searching for coding length increments". In: *Advances in neural information processing systems* 21 (2008).

[118] HN Hu, YC Lin, MY Liu, HT Cheng, YJ Chang, and M Sun. "Deep 360 pilot: Learning a deep agent for piloting through 360 sports videos". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2017, pp. 1396–1405.

[119] G Huang, Z Liu, L Van Der Maaten, and KQ Weinberger. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[120] PJ Huang, CA Lu, and KW Chen. "Temporally-Aggregating Multiple-Discontinuous-Image Saliency Prediction with Transformer-Based Attention". In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 6571–6577.

[121] X Huang, C Shen, X Boix, and Q Zhao. "Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 262–270.

[122] L Itti and P Baldi. "Bayesian surprise attracts human attention". In: *Vision research* 49.10 (2009), pp. 1295–1306.

[123] L Itti and C Koch. "A saliency-based search mechanism for overt and covert shifts of visual attention". In: *Vision research* 40.10-12 (2000), pp. 1489–1506.

[124] L Itti, C Koch, and E Niebur. "A model of saliency-based visual attention for rapid scene analysis". In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 11 (1998), pp. 1254–1259.

[125] N Jacobson, YL Lee, V Mahadevan, N Vasconcelos, and TQ Nguyen. "A novel approach to FRUC using discriminant saliency and frame segmentation". In: *IEEE Transactions on Image Processing* 19.11 (2010), pp. 2924–2934.

[126] A Jaiswal, AR Babu, MZ Zadeh, D Banerjee, and F Makedon. "A survey on contrastive self-supervised learning". In: *Technologies* 9.1 (2021), p. 2.

[127] W James. *The principles of psychology*. Vol. 1. Cosimo, Inc., 1950.

[128] S Jia and ND Bruce. "Eml-net: An expandable multi-layer network for saliency prediction". In: *Image and Vision Computing* (2020), p. 103887.

[129] L Jiang, M Xu, and Z Wang. "Predicting video saliency with object-to-motion CNN and two-layer convolutional LSTM". In: *arXiv preprint arXiv:1709.06316* (2017).

[130] M Jiang, S Huang, J Duan, and Q Zhao. "SALICON: Saliency in Context". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015.

[131] L Jianxun, L Hanhe, D MARSHALL, et al. "TranSalNet: Towards perceptually relevant visual saliency prediction". In: *arXiv preprint arXiv:2110.03593* (2021).

[132] L Jing, P Vincent, Y LeCun, and Y Tian. "Understanding dimensional collapse in contrastive self-supervised learning". In: *arXiv preprint arXiv:2110.09348* (2021).

[133] L Jing and Y Tian. "Self-supervised visual feature learning with deep neural networks: A survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[134] T Jost, N Ouerhani, R Von Wartburg, R Müri, and H Hügli. "Assessing the contribution of color in visual attention". In: *Computer Vision and Image Understanding* 100.1-2 (2005), pp. 107–123.

[135] T Judd, F Durand, and A Torralba. "A Benchmark of Computational Models of Saliency to Predict Human Fixations". In: *MIT Technical Report*. 2012.

[136] T Judd, F Durand, and A Torralba. "A benchmark of computational models of saliency to predict human fixations". In: (2012).

[137] T Judd, K Ehinger, F Durand, and A Torralba. "Learning to predict where humans look". In: *2009 IEEE 12th international conference on computer vision*. IEEE. 2009, pp. 2106–2113.

[138] EM Kaya and M Elhilali. "Modelling auditory attention". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 372.1714 (2017), p. 20160101.

[139] S Khan, M Naseer, M Hayat, SW Zamir, FS Khan, and M Shah. "Transformers in vision: A survey". In: *ACM computing surveys (CSUR)* 54.10s (2022), pp. 1–41.

[140] W Kienzle, MO Franz, B Schölkopf, and FA Wichmann. "Center-surround patterns emerge as optimal predictors for human saccade targets". In: *Journal of vision* 9.5 (2009), pp. 7–7.

[141] DP Kingma and J Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[142] C Koch and S Ullman. "Shifts in selective visual attention: towards the underlying neural circuitry". In: *Matters of intelligence*. Springer, 1987, pp. 115–141.

[143] TG Kolda and BW Bader. "Tensor decompositions and applications". In: *SIAM review* 51.3 (2009), pp. 455–500.

[144] A Kolesnikov, X Zhai, and L Beyer. "Revisiting self-supervised visual representation learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1920–1929.

[145] G Kootstra, A Nederveen, and B De Boer. "Paying attention to symmetry". In: *British Machine Vision Conference (BMVC2008)*. The British Machine Vision Association and Society for Pattern Recognition. 2008, pp. 1115–1125.

[146] S Kornblith, M Norouzi, H Lee, and G Hinton. "Similarity of neural network representations revisited". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 3519–3529.

[147] I Kotseruba, C Wloka, A Rasouli, and JK Tsotsos. "Do Saliency Models Detect Odd-One-Out Targets? New Datasets and Evaluations". In: *arXiv preprint arXiv:2005.06583* (2020).

[148] A Krizhevsky, I Sutskever, and GE Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012), pp. 1097–1105.

[149] A Kroner, M Senden, K Driessens, and R Goebel. "Contextual encoder–decoder network for visual saliency prediction". In: *Neural Networks* 129 (2020), pp. 261–270.

[150] SS Kruthiventi, K Ayush, and RV Babu. "Deepfix: A fully convolutional neural network for predicting human eye fixations". In: *IEEE Transactions on Image Processing* 26.9 (2017), pp. 4446–4456.

[151] M Kümmerer, L Theis, and M Bethge. "Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet". In: *arXiv preprint arXiv:1411.1045* (2014).

[152] M Kümmerer, TSA Wallis, and M Bethge. "Saliency Benchmarking Made Easy: Separating Models, Maps and Metrics". In: *Computer Vision – ECCV 2018*. Ed. by V Ferrari, M Hebert, C Sminchisescu, and Y Weiss. Lecture Notes in Computer Science. Springer International Publishing, 2018, pp. 798–814.

[153] M Kümmerer, TS Wallis, and M Bethge. "Information-theoretic model comparison unifies saliency metrics". In: *Proceedings of the National Academy of Sciences* 112.52 (2015), pp. 16054–16059.

[154] M Kummerer, TS Wallis, and M Bethge. "Saliency benchmarking made easy: Separating models, maps and metrics". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 770–787.

[155] M Kummerer, TS Wallis, LA Gatys, and M Bethge. "Understanding low-and high-level contributions to fixation prediction". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 4789–4798.

[156] Q Lai, W Wang, H Sun, and J Shen. "Video saliency prediction using spatiotemporal residual attentive networks". In: *IEEE Transactions on Image Processing* 29 (2019), pp. 1113–1126.

[157] P Le Callet and E Niebur. “Visual attention and applications in multimedia technologies”. In: *Proceedings of the IEEE* 101.9 (2013), pp. 2058–2067.

[158] PH Le-Khac, G Healy, and AF Smeaton. “Contrastive representation learning: A framework and review”. In: *IEEE Access* (2020).

[159] O Le Meur, P Le Callet, and D Barba. “Predicting visual fixations on video based on low-level visual features”. In: *Vision research* 47.19 (2007), pp. 2483–2498.

[160] O Le Meur, P Le Callet, D Barba, and D Thoreau. “A coherent computational approach to model bottom-up visual attention”. In: *IEEE transactions on pattern analysis and machine intelligence* 28.5 (2006), pp. 802–817.

[161] P Lebreton, S Fremerey, and A Raake. “V-BMS360: A video extention to the BMS360 image saliency model”. In: *2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE. 2018, pp. 1–4.

[162] P Lebreton and A Raake. “GBVS360, BMS360, ProSal: Extending existing saliency prediction models from 2D to omnidirectional images”. In: *Signal Processing: Image Communication* 69 (2018), pp. 69–78.

[163] C Ledig, L Theis, F Huszar, J Caballero, AP Aitken, A Tejani, J Totz, Z Wang, and W Shi. “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network”. In: *CoRR* abs/1609.04802 (2016). arXiv: 1609.04802. URL: http://arxiv.org/abs/1609.04802.

[164] BK Lee, B Park, CW Kim, and YM Ro. “MoAI: Mixture of All Intelligence for Large Language and Vision Models”. In: *arXiv preprint arXiv:2403.07508* (2024).

[165] H Lee, J Huang, M Singh, and M Yang. “Unsupervised Representation Learning by Sorting Sequences”. In: *CoRR* abs/1708.01246 (2017). arXiv: 1708.01246. URL: http://arxiv.org/abs/1708.01246.

[166] TW Lee and TW Lee. *Independent component analysis*. Springer, 1998.

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

[168] G Li, X Sui, J Yan, and Y Fang. "Benchmarking 360° Saliency Models by General-Purpose Metrics". In: *2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE. 2022, pp. 1–6.

[169] J Li, C Xia, Y Song, S Fang, and X Chen. "A data-driven metric for comprehensive evaluation of saliency models". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 190–198.

[170] J Li, MD Levine, X An, X Xu, and H He. "Visual saliency based on scale-space analysis in the frequency domain". In: *IEEE transactions on pattern analysis and machine intelligence* 35.4 (2012), pp. 996–1010.

[171] Z Li. "A saliency map in primary visual cortex". In: *Trends in Cognitive Sciences* 6.1 (2002), pp. 9–16. ISSN: 1364-6613. DOI: https://doi.org/10.1016/S1364-6613(00)01817-9.

[172] A Linardos, M Kümmerer, O Press, and M Bethge. "DeepGaze IIE: Calibrated prediction in and out-of-domain for state-of-the-art saliency modeling". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 12919–12928.

[173] P Linardos, E Mohedano, JJ Nieto, NE O'Connor, X Giro-i Nieto, and K McGuinness. "Simple vs complex temporal recurrences for video saliency prediction". In: *arXiv preprint arXiv:1907.01869* (2019).

[174] GW Lindsay. "Attention in psychology, neuroscience, and machine learning". In: *Frontiers in computational neuroscience* 14 (2020), p. 516985.

[175] N Liu and J Han. "A deep spatial contextual long-term recurrent convolutional network for saliency detection". In: *IEEE Transactions on Image Processing* 27.7 (2018), pp. 3264–3274.

[176] N Liu, J Han, D Zhang, S Wen, and T Liu. "Predicting eye fixations using convolutional neural networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 362–370.

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

[178] Z Liu, Y Lin, Y Cao, H Hu, Y Wei, Z Zhang, S Lin, and B Guo. "Swin transformer: Hierarchical vision transformer using shifted windows". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 10012–10022.

[179] I Loshchilov and F Hutter. "Decoupled weight decay regularization". In: *arXiv preprint arXiv:1711.05101* (2017).

[180] J Lou, H Lin, D Marshall, D Saupe, and H Liu. "TranSalNet: Towards perceptually relevant visual saliency prediction". In: *Neurocomputing* 494 (2022), pp. 455–467.

[181] DG Lowe. "Distinctive image features from scale-invariant keypoints". In: *International journal of computer vision* 60.2 (2004), pp. 91–110.

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

[183] Q Ma and L Zhang. "Image quality assessment with visual attention". In: *2008 19th International Conference on Pattern Recognition*. IEEE. 2008, pp. 1–4.

[184] Q Ma and L Zhang. "Saliency-based image quality assessment criterion". In: *International Conference on Intelligent Computing*. Springer. 2008, pp. 1124–1133.

[185] Q Ma, L Zhang, and B Wang. "New strategy for image and video quality assessment". In: *Journal of Electronic Imaging* 19.1 (2010), p. 011019.

[186] WDK Ma, J Lewis, and WB Kleijn. "The HSIC bottleneck: Deep learning without back-propagation". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 04. 2020, pp. 5085–5092.

[187] M Mancas, VP Ferrera, N Riche, and JG Taylor. *From Human Attention to Computational Attention*. Vol. 2. Springer, 2016.

[188] S Marat, TH Phuoc, L Granjon, N Guyader, D Pellerin, and A Guérin-Dugué. "Modelling spatio-temporal saliency to predict gaze direction for short videos". In: *International journal of computer vision* 82.3 (2009), p. 231.

[189] A Maravita, C Spence, and J Driver. "Multisensory integration and the body schema: close to hand and within reach". In: *Current biology* 13.13 (2003), R531–R539.

[190] L Marchesotti, C Cifarelli, and G Csurka. "A framework for visual saliency detection with applications to image thumbnailing". In: *2009 IEEE 12th International Conference on Computer Vision*. IEEE. 2009, pp. 2232–2239.

[191] S Mathe and C Sminchisescu. "Actions in the eye: Dynamic gaze datasets and learnt saliency models for visual recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 37.7 (2014), pp. 1408–1424.

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

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

[194] R Milanese. "Detecting salient regions in an image: from biological evidence to computer implementation". In: *Ph. D Theses, the University of Geneva* (1993).

[195] K Min and JJ Corso. "TASED-net: Temporally-aggregating spatial encoder-decoder network for video saliency detection". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 2394–2403.

[196] M Minderer, O Bachem, N Houlsby, and M Tschannen. "Automatic shortcut removal for self-supervised representation learning". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 6927–6937.

[197] AK Mishra and Y Aloimonos. "Active segmentation". In: *International Journal of Humanoid Robotics* 6.03 (2009), pp. 361–386.

[198] I Misra and Lvd Maaten. "Self-supervised learning of pretext-invariant representations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 6707–6717.

[199] I Misra, CL Zitnick, and M Hebert. "Unsupervised Learning using Sequential Verification for Action Recognition". In: *CoRR* abs/1603.08561 (2016). arXiv: 1603.08561. URL: http://arxiv.org/abs/1603.08561.

[200] V Mnih, K Kavukcuoglu, D Silver, A Graves, I Antonoglou, D Wierstra, and M Riedmiller. "Playing atari with deep reinforcement learning". In: *arXiv preprint arXiv:1312.5602* (2013).

[201] G Montavon, S Lapuschkin, A Binder, W Samek, and KR Müller. "Explaining nonlinear classification decisions with deep taylor decomposition". In: *Pattern recognition* 65 (2017), pp. 211–222.

[202] N Murray, M Vanrell, X Otazu, and CA Parraga. "Saliency estimation using a non-parametric low-level vision model". In: *CVPR 2011*. IEEE. 2011, pp. 433–440.

[203] V Navalpakkam and L Itti. "An integrated model of top-down and bottom-up attention for optimizing detection speed". In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2. IEEE. 2006, pp. 2049–2056.

[204] A Nguyen, Z Yan, and K Nahrstedt. "Your attention is unique: Detecting 360-degree video saliency in head-mounted display for head movement prediction". In: *Proceedings of the 26th ACM international conference on Multimedia*. 2018, pp. 1190–1198.

[205] M Noroozi and P Favaro. “Unsupervised learning of visual representations by solving jigsaw puzzles”. In: *European conference on computer vision*. Springer. 2016, pp. 69–84.

[206] A Oliva and A Torralba. “The role of context in object recognition”. In: *Trends in cognitive sciences* 11.12 (2007), pp. 520–527.

[207] A Oliva, A Torralba, MS Castelhano, and JM Henderson. “Top-down control of visual attention in object detection”. In: *Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429)*. Vol. 1. IEEE. 2003, pp. I–253.

[208] Avd Oord, Y Li, and O Vinyals. “Representation learning with contrastive predictive coding”. In: *arXiv preprint arXiv:1807.03748* (2018).

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

[210] X Otazu, M Vanrell, and CA Párraga. “Multiresolution wavelet framework models brightness induction effects”. In: *Vision research* 48.5 (2008), pp. 733–751.

[211] N Ouerhani, H Hügli, R Müri, and R Von Wartburg. “Empirical validation of the saliency-based model of visual attention”. In: *Electronic letters on computer vision and image analysis*. 2003, pp. 13–23.

[212] J Pan, CC Ferrer, K McGuinness, NE O’Connor, J Torres, E Sayrol, and X Giro-i Nieto. “Salgan: Visual saliency prediction with generative adversarial networks”. In: *arXiv preprint arXiv:1701.01081* (2017).

[213] J Pan, E Sayrol, X Giro-i Nieto, K McGuinness, and NE O’Connor. “Shallow and deep convolutional networks for saliency prediction”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 598–606.

[214] L Panagiotis, M Eva, C Monica, G Cathal, and GiN Xavier. "Temporal Saliency Adaptation in Egocentric Videos". In: 2018.

[215] S Park, M Hoai, A Bhattacharya, and SR Das. "Adaptive streaming of 360-degree videos with reinforcement learning". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021, pp. 1839–1848.

[216] A Paszke, S Gross, F Massa, A Lerer, J Bradbury, G Chanan, T Killeen, Z Lin, N Gimelshein, L Antiga, A Desmaison, A Kopf, E Yang, Z DeVito, M Raison, A Tejani, S Chilamkurthy, B Steiner, L Fang, J Bai, and S Chintala. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *NeurIPS*. Ed. by H Wallach, H Larochelle, A Beygelzimer, F d'Alché-Buc, E Fox, and R Garnett. 2019.

[217] A Paszke, S Gross, F Massa, A Lerer, J Bradbury, G Chanan, T Killeen, Z Lin, N Gimelshein, L Antiga, et al. "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems* 32 (2019).

[218] D Pathak, P Krähenbühl, J Donahue, T Darrell, and AA Efros. "Context Encoders: Feature Learning by Inpainting". In: *CoRR* abs/1604.07379 (2016). arXiv: 1604.07379. URL: http://arxiv.org/abs/1604.07379.

[219] RS Pathak. *The wavelet transform*. Vol. 4. Springer Science & Business Media, 2009.

[220] A Patney, M Salvi, J Kim, A Kaplanyan, C Wyman, N Benty, D Luebke, and A Lefohn. "Towards foveated rendering for gaze-tracked virtual reality". In: *ACM Transactions on Graphics (TOG)* 35.6 (2016), pp. 1–12.

[221] RJ Peters, A Iyer, L Itti, and C Koch. "Components of bottom-up gaze allocation in natural images". In: *Vision research* 45.18 (2005), pp. 2397–2416.

[222] D Povey, A Ghoshal, G Boulianne, L Burget, O Glembek, N Goel, M Hannemann, P Motlicek, Y Qian, P Schwarz, et al. "The Kaldi speech recognition

toolkit". In: *IEEE 2011 workshop on automatic speech recognition and understanding*. CONF. IEEE Signal Processing Society. 2011.

[223] CM Privitera and LW Stark. "Algorithms for defining visual regions-of-interest: Comparison with eye fixations". In: *IEEE Transactions on pattern analysis and machine intelligence* 22.9 (2000), pp. 970–982.

[224] M Qiao, M Xu, Z Wang, and A Borji. "Viewport-dependent saliency prediction in 360° video". In: *IEEE Transactions on Multimedia* 23 (2020), pp. 748–760.

[225] A Radford, JW Kim, C Hallacy, A Ramesh, G Goh, S Agarwal, G Sastry, A Askell, P Mishkin, J Clark, et al. "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PMLR. 2021, pp. 8748–8763.

[226] A Radford, JW Kim, T Xu, G Brockman, C McLeavey, and I Sutskever. "Robust speech recognition via large-scale weak supervision". In: *arXiv preprint arXiv:2212.04356* (2022).

[227] M Raghu, T Unterthiner, S Kornblith, C Zhang, and A Dosovitskiy. "Do vision transformers see like convolutional neural networks?" In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 12116–12128.

[228] N Rahaman, S Wolf, A Goyal, R Remme, and Y Bengio. "Learning the Arrow of Time". In: *CoRR* abs/1907.01285 (2019). arXiv: `1907.01285`. URL: `http://arxiv.org/abs/1907.01285`.

[229] Y Rai, J Gutiérrez, and P Le Callet. "A dataset of head and eye movements for 360 degree images". In: *Proceedings of the 8th ACM on Multimedia Systems Conference*. 2017, pp. 205–210.

[230] Y Rai, P Le Callet, and P Guillotel. "Which saliency weighting for omni directional image quality assessment?" In: *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE. 2017, pp. 1–6.

[231] S Ren, K He, R Girshick, and J Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015).

[232] N Riche, M Duvinage, M Mancas, B Gosselin, and T Dutoit. "Saliency and human fixations: State-of-the-art and study of comparison metrics". In: *Proceedings of the IEEE international conference on computer vision*. 2013, pp. 1153–1160.

[233] N Riche, M Mancas, M Duvinage, M Mibulumukini, B Gosselin, and T Dutoit. "Rare2012: A multi-scale rarity-based saliency detection with its comparative statistical analysis". In: *Signal Processing: Image Communication* 28.6 (2013), pp. 642–658.

[234] RD Rimey and CM Brown. "Controlling eye movements with hidden Markov models". In: *International Journal of Computer Vision* 7.1 (1991), pp. 47–65.

[235] R Rosenholtz. "A simple saliency model predicts a number of motion popout phenomena". In: *Vision research* 39.19 (1999), pp. 3157–3163.

[236] S Ruder. "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747* (2016).

[237] D Rudoy, DB Goldman, E Shechtman, and L Zelnik-Manor. "Learning video saliency from human gaze using candidate selection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 1147–1154.

[238] DE Rumelhart, GE Hinton, RJ Williams, et al. *Learning internal representations by error propagation*. 1985.

[239] NG Sadaka and LJ Karam. "Efficient perceptual attentive super-resolution". In: *2009 16th IEEE International Conference on Image Processing (ICIP)*. IEEE. 2009, pp. 3113–3116.

[240] DD Salvucci. "An integrated model of eye movements and visual encoding". In: *Cognitive Systems Research* 1.4 (2001), pp. 201–220.

[241] B Schauerte and R Stiefelhagen. "Quaternion-based spectral saliency detection for eye fixation prediction". In: *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part II 12*. Springer. 2012, pp. 116–129.

[242] C Schuhmann, R Beaumont, R Vencu, C Gordon, R Wightman, M Cherti, T Coombes, A Katta, C Mullis, M Wortsman, et al. "Laion-5b: An open large-scale dataset for training next generation image-text models". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 25278–25294.

[243] J Schulman, F Wolski, P Dhariwal, A Radford, and O Klimov. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[244] M Schuster and KK Paliwal. "Bidirectional recurrent neural networks". In: *IEEE transactions on Signal Processing* 45.11 (1997), pp. 2673–2681.

[245] HJ Seo and P Milanfar. "Nonparametric bottom-up saliency detection by self-resemblance". In: *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE. 2009, pp. 45–52.

[246] HJ Seo and P Milanfar. "Static and space-time visual saliency detection by self-resemblance". In: *Journal of vision* 9.12 (2009), pp. 15–15.

[247] N Shazeer, A Mirhoseini, K Maziarz, A Davis, Q Le, G Hinton, and J Dean. "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer". In: *arXiv preprint arXiv:1701.06538* (2017).

[248] C Siagian and L Itti. "Biologically inspired mobile robot vision localization". In: *IEEE Transactions on Robotics* 25.4 (2009), pp. 861–873.

[249] K Simonyan and A Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[250] V Sitzmann, A Serrano, A Pavel, M Agrawala, D Gutierrez, B Masia, and G Wetzstein. "Saliency in VR: How do people explore virtual environments?" In: *IEEE transactions on visualization and computer graphics* 24.4 (2018), pp. 1633–1642.

[251] JP Snyder. *Map projections–A working manual*. Vol. 1395. US Government Printing Office, 1987.

[252] MM Sohlberg and CA Mateer. *Introduction to cognitive rehabilitation: Theory and practice.* Guilford Press, 1989.

[253] N Sprague and D Ballard. "Eye movements for reward maximization". In: *Advances in neural information processing systems*. 2004, pp. 1467–1474.

[254] LW Stark and YS Choi. "Experimental metaphysics: The scanpath as an epistemological mechanism". In: *Advances in psychology*. Vol. 116. Elsevier, 1996, pp. 3–69.

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

[256] R Strudel, R Garcia, I Laptev, and C Schmid. "Segmenter: Transformer for semantic segmentation". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 7262–7272.

[257] Z Sun, S Cao, Y Yang, and KM Kitani. "Rethinking transformer-based set prediction for object detection". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 3611–3620.

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

[259] A Tao, K Sapra, and B Catanzaro. "Hierarchical multi-scale attention for semantic segmentation". In: *arXiv preprint arXiv:2005.10821* (2020).

[260] BW Tatler. "The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions". In: *Journal of vision* 7.14 (2007), pp. 4–4.

[261] JL Teugels. "Some representations of the multivariate Bernoulli and binomial distributions". In: *Journal of multivariate analysis* 32.2 (1990), pp. 256–268.

[262] C Thomas. "Opensalicon: An open source implementation of the salicon saliency model". In: *arXiv preprint arXiv:1606.00110* (2016).

[263] Y Tian, D Krishnan, and P Isola. "Contrastive Multiview Coding". In: *CoRR* abs/1906.05849 (2019). arXiv: 1906.05849. URL: http://arxiv.org/abs/1906.05849.

[264] M Toneva and L Wehbe. "Interpreting and improving natural-language processing (in machines) with natural language-processing (in the brain)". In: *Advances in neural information processing systems* 32 (2019).

[265] A Torralba. "Modeling global scene factors in attention". In: *JOSA A* 20.7 (2003), pp. 1407–1418.

[266] AM Treisman and G Gelade. "A feature-integration theory of attention". In: *Cognitive psychology* 12.1 (1980), pp. 97–136.

[267] JK Tsotsos, SM Culhane, WYK Wai, Y Lai, N Davis, and F Nuflo. "Modeling visual attention via selective tuning". In: *Artificial intelligence* 78.1-2 (1995), pp. 507–545.

[268] S Tuli, I Dasgupta, E Grant, and TL Griffiths. "Are convolutional neural networks or transformers more like human vision?" In: *arXiv preprint arXiv:2105.07197* (2021).

[269] A Vaswani, N Shazeer, N Parmar, J Uszkoreit, L Jones, AN Gomez, Ł Kaiser, and I Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[270] R Veale, ZM Hafed, and M Yoshida. "How is visual salience computed in the brain? Insights from behaviour, neurobiology and modelling". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 372.1714 (2017), p. 20160113.

[271] E Vig, M Dorr, and D Cox. "Large-scale optimization of hierarchical features for saliency prediction in natural images". In: *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition. 2014, pp. 2798–2805.

[272] FTS VOCUS. "A visual attention system for object detection and goal directed search". PhD thesis. Ph. D. thesis, Bown: University of Bonn, 2005.

[273] E Voita, D Talbot, F Moiseev, R Sennrich, and I Titov. "Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned". In: *arXiv preprint arXiv:1905.09418* (2019).

[274] F Wang, M Jiang, C Qian, S Yang, C Li, H Zhang, X Wang, and X Tang. "Residual attention network for image classification". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 3156–3164.

[275] J Wang, DM Chandler, and P Le Callet. "Quantifying the relationship between visual salience and visual importance". In: *Human vision and electronic imaging XV*. Vol. 7527. SPIE. 2010, pp. 160–168.

[276] W Wang and J Shen. "Deep visual attention prediction". In: *IEEE Transactions on Image Processing* 27.5 (2017), pp. 2368–2378.

[277] W Wang, J Shen, F Guo, MM Cheng, and A Borji. "Revisiting video saliency: A large-scale benchmark and a new model". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4894–4903.

[278] X Wang, R Girshick, A Gupta, and K He. "Non-local neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7794–7803.

[279] Y Wang, X Zhang, T Yang, and J Sun. "Anchor DETR: Query Design for Transformer-Based Object Detection". In: *arXiv preprint arXiv:2109.07107* (2021).

[280] MJ Wieser and A Keil. “Temporal trade-off effects in sustained attention: dynamics in visual cortex predict the target detection performance during distraction”. In: *Journal of Neuroscience* 31.21 (2011), pp. 7784–7790.

[281] N Wilming, T Betz, TC Kietzmann, and P König. “Measures and limits of models of fixation selection”. In: *PloS one* 6.9 (2011), e24038.

[282] Z Wu, Y Xiong, SX Yu, and D Lin. “Unsupervised feature learning via non-parametric instance discrimination”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3733–3742.

[283] K Xu, J Ba, R Kiros, K Cho, A Courville, R Salakhudinov, R Zemel, and Y Bengio. “Show, attend and tell: Neural image caption generation with visual attention”. In: *International conference on machine learning*. PMLR. 2015, pp. 2048–2057.

[284] M Xu, C Li, Y Liu, X Deng, and J Lu. “A subjective visual quality assessment method of panoramic videos”. In: *2017 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE. 2017, pp. 517–522.

[285] M Xu, C Li, S Zhang, and P Le Callet. “State-of-the-art in 360 video/image processing: Perception, assessment and compression”. In: *IEEE Journal of Selected Topics in Signal Processing* 14.1 (2020), pp. 5–26.

[286] M Xu, Y Song, J Wang, M Qiao, L Huo, and Z Wang. “Predicting head movement in panoramic video: A deep reinforcement learning approach”. In: *IEEE transactions on pattern analysis and machine intelligence* 41.11 (2018), pp. 2693–2708.

[287] Y Xu, Y Dong, J Wu, Z Sun, Z Shi, J Yu, and S Gao. “Gaze prediction in dynamic 360 immersive videos”. In: *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5333–5342.

[288] Y Yan, L Zhaoping, and W Li. “Bottom-up saliency and top-down learning in the primary visual cortex of monkeys”. In: *Proceedings of the National*

Academy of Sciences 115.41 (2018), pp. 10499–10504. ISSN: 0027-8424. DOI: 10.1073/pnas.1803854115.

[289] Z Yang, X He, J Gao, L Deng, and A Smola. "Stacked attention networks for image question answering". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 21–29.

[290] L Yuan, Y Chen, T Wang, W Yu, Y Shi, ZH Jiang, FE Tay, J Feng, and S Yan. "Tokens-to-token vit: Training vision transformers from scratch on imagenet". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 558–567.

[291] H Zhang, I Goodfellow, D Metaxas, and A Odena. "Self-attention generative adversarial networks". In: *International conference on machine learning*. PMLR. 2019, pp. 7354–7363.

[292] J Zhang and S Sclaroff. "Saliency detection: A boolean map approach". In: *Proceedings of the IEEE international conference on computer vision*. 2013, pp. 153–160.

[293] K Zhang and Z Chen. "Video saliency prediction based on spatial-temporal two-stream network". In: *IEEE Transactions on Circuits and Systems for Video Technology* 29.12 (2018), pp. 3544–3557.

[294] L Zhang, MH Tong, TK Marks, H Shan, and GW Cottrell. "SUN: A Bayesian framework for saliency using natural statistics". In: *Journal of vision* 8.7 (2008), pp. 32–32.

[295] R Zhang, P Isola, and AA Efros. "Colorful Image Colorization". In: *CoRR* abs/1603.08511 (2016). arXiv: 1603.08511. URL: http://arxiv.org/abs/1603.08511.

[296] Z Zhang, Y Xu, J Yu, and S Gao. "Saliency detection in 360 videos". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 488–503.

[297] H Zhao, L Jiang, J Jia, PH Torr, and V Koltun. “Point transformer”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 16259–16268.

[298] Q Zhou, C Du, and H He. “Exploring the Brain-like Properties of Deep Neural Networks: A Neural Encoding Perspective”. In: *Machine Intelligence Research* 19.5 (2022), pp. 439–455.

[299] C Zhu, K Huang, and G Li. “An innovative saliency guided roi selection model for panoramic images compression”. In: *2018 Data Compression Conference*. IEEE. 2018, pp. 436–436.

[300] D Zhu, K Zhang, N Zhang, Q Zhou, X Min, G Zhai, and X Yang. “Unified Audio-visual Saliency Model for Omnidirectional Videos with Spatial Audio”. In: *IEEE Transactions on Multimedia* (2023).

[301] JY Zhu, T Park, P Isola, and AA Efros. “Unpaired image-to-image translation using cycle-consistent adversarial networks”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2223–2232.

[302] X Zhu, W Su, L Lu, B Li, X Wang, and J Dai. “Deformable detr: Deformable transformers for end-to-end object detection”. In: *arXiv preprint arXiv:2010.04159* (2020).

[303] B Zoph, V Vasudevan, J Shlens, and QV Le. “Learning transferable architectures for scalable image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8697–8710.

[304] Z Zou, M Ye, S Li, X Li, and F Dufaux. “360 Image Saliency Prediction by Embedding Self-Supervised Proxy Task”. In: *IEEE Transactions on Broadcasting* (2023).