TECHNISCHE UNIVERSITÄT BERLIN

Master's thesis

Gaze Prediction in Natural Videos with End-to-End Deep Learning

Yannic Spreen-Ledebur

Matriculation number: 451731

June 25, 2022

Supervisors

Prof. Dr. Klaus Obermayer
Neural Information Processing, Technical University of Berlin

Nicolas Roth
Neural Information Processing, Technical University of Berlin

Heiner Spieß
Neural Information Processing, Technical University of Berlin

Reviewers

Prof. Dr. Klaus Obermayer
Neural Information Processing, Technical University of Berlin

Prof. Dr.-Ing. Olaf Hellwich
Computer Vision & Remote Sensing, Technical University of Berlin

Eidesstattliche Erklärung
Statutory declaration

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

I hereby declare that I have written this thesis independently and on my own, and without unauthorised assistance and exclusively using the sources and aids listed.

Berlin, 25. Juni 2022 Yannic Spreen-Ledebur

Abstract

Human visual exploration influences most of our day-to-day actions as it defines what we can act on. This gaze exploration process consists of sequences of fixations with quick gaze relocations, saccades, in between. Previous research has gone into modeling these scanpaths in order to better understand human visual attention, with most works focusing on scanpaths in static visual stimuli. In related fields like saliency prediction and static scanpath prediction, deep learning models nowadays form the state-of-the-art. Due to the high dimensionality of the underlying problem, this is not the case in scanpath prediction in dynamic scenes.

This work attempts to overcome the problems of scanpath simulation in real-world dynamic scenes with an end-to-end deep learning approach. We use Feature Pyramid Networks for feature extraction from video frames while conserving spatial information and employ Recurrent Independent Mechanisms with several competing recurrent units in order to reach task specialization between different units and better generalization to new data. Opposing to most existing scanpath models, we use the same network for both fixation and saccade prediction.

We systematically train and evaluate our model on different partitions of the GazeCom natural video dataset, examining the model's capabilities to generalize to visual features and the influence of multiple observer ground truths.

We find that measured on Normalized Scanpath Saliency, our model outperforms random gaze prediction and a baseline that strictly predicts the middle of a video, representing the center bias observed in human scanpaths. We generally note that our model fails to recreate the distinct motion dynamics of fixation and saccade alternation when extending the model training to many videos or observers. We did not see an interpretable specialization in the recurrent units for fixation/saccade prediction, although this could be enforced in further iterations. In general, our regressive model loses the characteristic exploration dynamics when trained on many different observers.

Our results show that we likely need stronger inductive biases to generate human-like scanpaths. Due to the immense number of possible valid scanpaths on a visual stimuli, a purely generative model approach might be better suited to mirror the wide variance observed in human scanpaths.

Zusammenfassung

Die menschlische visuelle Exploration beeinflusst die meisten unserer alltäglichen Handlungen, da sie bestimmt, worauf wir reagieren können. Dieser Explorationsprozess besteht aus Sequenzen von Fixierungen mit schnellen Blickverlagerungen, Sakkaden, dazwischen. Die bisherige Forschung hat sich mit der Modellierung dieser Scanpfade beschäftigt, um menschliche visuelle Aufmerksamkeit besser zu verstehen, wobei sich die meisten Arbeiten auf Scanpfade bei statischen visuellen Stimuli konzentrieren. In verwandten Bereichen wie der Salienzvorhersage und der Scanpfad-Vorhersage auf Bildern bilden Deep-Learning-Modelle heute den aktuellen Stand der Technik. Aufgrund der hohen Dimensionalität des zugrunde liegenden Problems ist dies bei der Vorhersage von Scanpfaden in dynamischen Szenen bisher nicht der Fall.

In dieser Arbeit wird versucht, die Probleme der Scanpfadsimulation in realen dynamischen Szenen mit einem End-to-End Deep-Learning-Ansatz zu lösen. Wir verwenden Feature-Pyramiden Netzwerke für die Feature-Extraktion aus Videobildern unter Beibehaltung der räumlichen Informationen und setzen Recurrent Independent Mechanisms mit mehreren konkurrierenden rekurrenten Modulen ein, um eine Aufgabenspezialisierung zwischen verschiedenen Modulen und eine bessere Generalisierung auf neue Daten zu erreichen. Im Gegensatz zu den meisten existierenden Scanpfad -Modellen verwenden wir dasselbe Netzwerk sowohl für die Fixation- als auch für die Sakkadenvorhersage.

Wir trainieren und evaluieren unser Modell systematisch auf verschiedenen Partitionen des GazeCom-Datensatzes aus natürlichen Videos, um die Generalisierung über visuelle Features innerhalb des Modells und den Einfluss von Daten mehrerer Testpersonen zu untersuchen.

Wir stellen fest, dass unser Modell, gemessen an der Normalized Scanpath Saliency, bessere Ergebnisse erzielt als eine zufällige Blickvorhersage und eine Basislinie, die ausschließlich die Mitte eines Videos vorhersagt, welche das in menschlichen Scanpfaden auftretende Center Bias repräsentiert. Wir stellen allgemein fest, dass unser Modell nicht in der Lage ist, die unterschiedliche Bewegungsdynamik von Fixation- und Sakkadenwechsel nachzubilden, wenn das Modelltraining auf viele Videos oder Testpersonen ausgeweitet wird. Wir konnten weiterhin keine interpretierbare Spezialisierung in den rekurrenten Einheiten für die Fixation-/Sakkadenvorhersage feststellen, obwohl dies in weiteren Iterationen erzwungen

werden könnte. Im Allgemeinen verliert unser regressives Modell die charakteristische menschliche Explorationsdynamik, wenn es mit vielen verschiedenen Testpersonen trainiert wird.

Unsere Ergebnisse zeigen, dass wir wahrscheinlich stärkere induktive Annahmen benötigen, um menschenähnliche Scanpfade zu erzeugen. Aufgrund der immensen Anzahl möglicher gültiger Scanpfade auf einem visuellen Stimulus könnte ein rein generativer Modellansatz besser geeignet sein, die große Varianz zwischen menschlichen Scanpfaden abzubilden.

Acknowledgments

First and foremost, I want to thank my supervisors, Nicolas Roth and Heiner Spieß for their continuous support and patience. With your help and professional knowledge of the field, I have gained a valuable insight into research work over the course of this thesis and your confidence and last-minute feedback has helped me immensely to complete this thesis in its current form. I further want to thank Prof. Obermayer for accompanying this project and for his quick help whenever I needed it.

I want to thank Abdullah Emirhan Karagul and Mario Bonsembiante for giving me great advice for my thesis and repeatedly putting things into relation. To my parents, thank you for consistently trying everything in your power to understand me and enable to focus on this thesis. I also want to thank all my other friends who have always made me keep the motivation to finalize my education in this master's degree.

Lastly, I want to thank my flatmates for being very understanding during this time and enabling me a good work environment with opportunities to socialize during stressful months of writing this thesis. I am also grateful for our cat Efes, whose sole presence in my home office has enriched my days.

Contents

1 Introduction

We as humans rely on quickly gathering relevant information from visual input. As the area that can be foveated by the eye at a time is comparably small and the visual sensory input is overwhelmingly huge, our eyes have to decide on locations that are salient in information, often moving multiple times per second at up to $500°/s$ (Leigh and Zee 2015). This decision process based on visual attention might have enabled us to spot a predator in the past and nowadays lets us fall for magic tricks that rely on distraction. Research has been going into exploring this visual decision making, but due to the complexity of the task, it is still very difficult to accurately model human eye movements (Kümmerer and Bethge 2021). Observers can display totally different viewing behaviour and so for a given video sequence there exist a multitude of valid scanpaths. As of today, multiple models for scanpath prediction on image data have been developed (Assens, Giro-i-Nieto, et al. 2018; Böhme et al. 2006; Coutrot et al. 2017; Itti et al. 1998; Kümmerer, Bethge, and Wallis 2022), but virtually no models exist for predicting scanpaths on videos (Boccignone et al. 2020; Roth et al. 2021), which poses further challenges due to the high dimensionality of the problem and requirement for precise saccade timings. This is why, in this work, we introduce a new model for frame-wise scanpath prediction in videos.

As neural networks have been proven to learn non-trivial patterns, e.g. in computer vision or natural language, this work tries to explore the possibilities of modelling these causes through the implementation of an end-to-end deep learning approach. By using an end-to-end approach, the model can be freed from many, otherwise limiting, starting assumptions and is hoped to become flexible to discover own patterns instead. We combine a Feature Pyramid Network (Lin et al. 2016) to gain rich spatial features from a convolutional network with Recurrent Independent mechanisms (Goyal et al. 2019), a modular, attention-based reccurent network, to predict frame-wise gaze positions.

Visual attention can be guided by professional cuts and directing (Carmi and Itti 2006) and is influenced by tasks at hand (Yarbus 1967). To allow a general exploration of visual attention, we use gaze data from viewing natural videos of outdoor, real-world scenes collected by Dorr et al. (2010) in a free-viewing setup. During recording, the heads of observers were fixated in a chin rest to isolate eye movement.

Eye movements involved in the human visual exploration process are generally

separated into three different major phases. Fixations, which describe a resting state of gaze, while saccades describe the rapid gaze shift to a new location (Leigh and Zee 2015; Purves et al. 2000). The third phase type, Smooth pursuit, describes the focus on a moving object and is much slower moving than saccades. For static scenes only fixations and saccades can occur as no movement will occur within the scene. A sequence of eye movements is referred to as a scanpath.

Visual attention guiding saccades has been linked to the superior colliculus, a structure lying on the roof of the midbrain, and the frontal eye field, a region of the frontal lobe (Purves et al. 2000). Their neurons on each side of the brain contain a topographic map of the respective visual field. Neuronal activity correlates with external visual stimuli and with the probability of a saccade to a given position (Krauzlis et al. 2013). If stimulated artificially, saccades can be triggered to locations in the visual field as if there were salient locations present. While voluntary saccades have been linked mostly to the frontal eye field, involuntary and reflexive saccades are largely initiated in the superior colliculus (Purves et al. 2000).

In general, human gaze favors the center of the visual field, and often returns there, which is referred to as center bias (Parkhurst, Law, et al. 2002; Parkhurst and Niebur 2003). This is more so the case in the viewing of images and videos, where objects of interest are often located close to the center. This is also referred to as photographer bias (Tseng et al. 2009). Tatler (2007) however showed that center bias persists irrespective to the distribution of image features or viewing task. Tseng et al. (2009) names orbital reserve, motor bias, viewing strategyand center of screen biasas other potential causes of center bias besides photographer bias.

After foveating a location, humans tend to undergo an inhibitory period where the previously focused location is less likely to be attended to again (Bays and Husain 2012; Posner and Cohen 1984). This process is referred to as *inhibition of return* and has been specifically considered in some modeling approaches by down-weighting gaze probabilities for an area around a location that was foveated just before (Itti et al. 1998; Roth et al. 2021).

It has been shown that visual attention both employs bottom-up and top-down elements. According to the Feature Integration theory introduced by Treisman and Gelade (1980), visually salient locations that stick out from their surroundings through colour, edges or shape can attract our attention and for example make us quickly discover unusual objects in our environment. Conversely to these bottom-up processes, task motivations have been shown to influence our exploration behaviour as well (Yarbus 1967; Borji et al. 2013), but are much more difficult to account for. That is why most models developed to model visual attention are bottom-up models which take pixel values as input.

The most common way to model visual attention has been through the use of topographic saliency maps, which output an abstract attention value for each pixel in an input scene. Building upon a theoretical architecture proposed by Koch and Ullman (1985), Itti et al. (1998) used a combination of feature maps attending intensity and colour contrast as well as edge orientation to predict salient regions in images. Over the past decades, saliency modeling has started to integrate more high-level features like objects (Russell et al. 2013) or tasks at hand. Nowadays, as many applications in computer vision, most saliency models employ deep convolutional neural networks (Huang, Shen, et al. 2015; Kruthiventi et al. 2015; Kümmerer, Wallis, et al. 2016; Wang, Shen, Guo, et al. 2018a).

Saliency maps by themselves can only model probabilities for overall gaze attendance, assuming that gaze will focus on salient regions in an image. To model individual scanpaths, often fixation locations are simply sampled from saliency maps. This generally either happens in a probabilistic sampling process or through a winner-takes-it-all approach, where the most salient location is chosen (Itti et al. 1998; Kümmerer, Bethge, and Wallis 2022). As the history of past foveated locations also plays a role, this approach needs to incorporate findings from gaze heuristics like inhibition of return, which states that we are less likely to look at previously fixated objects. This can be employed by down-weighting saliency around recently attended locations.

As visual exploration is a process over time and can be seen as sequential decision making, the temporal domain has to be regarded as well for scanpath prediction. Static saliency maps are bound to the spatial dimension and don't consider development over time, like movements or changes in colour. And maybe most importantly for scanpath prediction, they don't consider the history of past focused locations. Most works therefore compute spatial saliency maps and then weight locations based on the last fixated locations (Roth et al. 2021; Schwetlick et al. 2021; Kümmerer, Bethge, and Wallis 2022). Others use recurrent networks to additionally track the development of image features over time and account for object movements and colour changes (Huang, Cai, et al. 2018).

In this work, we employ Recurrent Independent Mechanisms (RIMs; Goyal et al. 2019), a recently introduced, attention-based Bahdanau et al. (2014) and Vaswani et al. (2017) recurrent architecture, to process development over time. It consists of different recurrent units that compete via input attention with each other and means to enforce specialization between these units and therefore better generalization to unseen data. In a simulation of bouncing balls, Goyal et al. noted that individual RIM units specialized on the trajectory calculation of specific balls. We use the RIM architecture in the hope of achieving semantic specialisation between units, e.g. differentiation between visual exploration, object tracking and fixation on

a target, and stronger universality of the resulting model. Inside of the model, learned, state-dependent soft attention allows to only pass relevant input to a unit and regulates the information exchange between different units.

We test our model across different partitions of the used gaze dataset with regards to the modeling of differing visual data and differing viewing behaviours employed by observers.

The model implementation and data preprocessing source code of this thesis are publicly available on Github at https://github.com/h-spiess/thesis_gaze_prediction.

The remaining chapters of this thesis are structured as follows:

- Chapter 2 presents related work to methods of this thesis. We look at static as well as dynamic saliency modeling. Then, we discuss other models that predict scanpaths and their underlying approaches and shortly dive into gaze prediction on egocentric video data.
- Chapter 3 describes our experimental setting. First, we discuss the data used and pre-processing steps. Then, we present the model architecture used in this thesis, separated into a feature extraction module and a recurrent module as well as the loss function used for training. Finally, we present the application of Teacher Forcing (Williams and Zipser 1989) and the evaluation metrics.
- Chapter 4 presents our experimental results and discusses strengths and weaknesses of the proposed model.
- Chapter 5 discusses the insights gained from this thesis for the current state of scanpath prediction and recommends and motivates possible further modifications to our model.
- In chapter 6, the findings of this thesis are summarized and a general outlook is given for future works.

2 Related Work

2.1 Static Saliency Prediction

Static saliency prediction tries to predict saliency, that is information content, within an image. As an output generally a saliency map with pixel-wise saliency values is created. Following the idea proposed by Koch and Ullman (1985) that saliency corresponds to local differences in features like colour, intensity and local orientation, most saliency maps only process bottom-up features, but some works also incorporate tasks and context. Generally, a saliency map for each regarded feature is calculated and these are then combined into one final map. In newer works, this manual approach has mostly been replaced with deep learning architectures, which learn to extract abstract image features in deep convolutional neural networks (LeCun et al. 1998). These abstract features then are combined in further layers to generate a saliency map.

In their seminal contribution, Itti et al. (1998) combined feature maps attending to the 3 feature channels pixel intensity, colour contrast and edge orientation. Originally designed to predict fixation sequences, the model set the way for modern saliency modeling. Harel et al. (2006) used a graph-based network to find locations that diverge from their neighbours in a feature channel. Zhang et al. (2008) utilised a bayesian approach based on natural image statistics to predict salient locations. Moving away from manual feature extraction, Vig et al. (2014) used multilayer feature extractors to find fitting feature representations and used a linear SVM to predict saliency based on these features. Most processing pipelines for static saliency since then have employed convolutional neural networks to learn powerful feature maps at different scales (Huang, Shen, et al. 2015; Kruthiventi et al. 2015). In their *Deepgaze II* mode, Kümmerer, Wallis, et al. (2016) without further fine-tuning used a *VGG-19* (Simonyan and Zisserman 2014) backbone to set a new state-of-the-art in static saliency prediction on the *MIT* saliency benchmark (Bylinskii et al. 2016a; Judd et al. 2012; Kümmerer, Wallis, et al. 2018). This model was recently extended by combining the output of different backbone models Linardos, Kümmerer, et al. (2021).

A good overview over proposed saliency models is given in Borji (2018) and Borji et al. (2013). The most popular benchmarks for the comparison of static saliency models as of now are the *MIT/Tübingen* benchmark (Bylinskii et al. 2016a; Judd

et al. 2012; Kümmerer, Wallis, et al. 2018) and the *SALICON* benchmark (Jiang et al. 2015). While *MIT/Tübingen* evaluates results on fixation data across a relatively small, but diverse dataset, *SALICON* employs a large dataset, but uses click data as representation of attention.

2.2 Dynamic Saliency Prediction

Dynamic saliency prediction extends static saliency to the temporal domain and addresses saliency in videos. Human attention on video frames differs from that on images, as video frames are generally visible for a much shorter amount of time and can include movement and change of image features (Böhme et al. 2006; Itti 2005). Due to the additional temporal dimension, the computational efforts necessary are much higher compared to static saliency prediction and comparably fewer models exist for this subspace of saliency modeling. A popular approach for dynamic saliency prediction is to separately process image information in pre-trained convolutional neural networks and to process temporal developments with recurrent architectures to finally combine these two feature spaces for a saliency output over time (Bak et al. 2017; Fang et al. 2014). More recently, models employing direct 3D convolutions (Chang and Zhu 2021; Wang, Liu, et al. 2021) and 3D vision transformer architectures (Liu et al. 2021; Ma et al. 2022) have topped the video saliency benchmark *Videosal* (Wang, Shen, Guo, et al. 2018a; Wang, Shen, Xie, et al. 2019).

In Itti (2005), Itti et al. extended their approach from 1998 to dynamic scenes by also considering temporal flicker and orientated motion energies. Guo and Zhang (2010) calculated saliency maps for image and video compression by consideration of their Fourier phase spectrum and attended not only for colours and orientation, but also motion blur in a performant model. But with the advances in deep learning, newer models mainly employ CNNs for frame-wise spatial saliency calculation and recurrent networks to consider frame history. Bak et al. (2017) used a two-stream spatio-temporal network based on images and optical flow information to make frame-wise saliency predictions and reached better predictions with a convolutional than a frame-wise combination of the two streams. Wang, Shen, Guo, et al. (2018a) and Linardos, Mohedano, et al. (2019) used convLSTMs (Shi et al. 2015) to consider temporal information while conserving spatial information. Wang, Liu, et al. (2021) reached new state-of-the-art results using a Feature Pyramid structure (Lin et al. 2016) with 3D spatio-temporal convolutional layers, processing the spatial and temporal domain within the same convolutional module. Chang and Zhu (2021) employed a similar architecture, but additionally incorporated audio clues for the

saliency prediction. Recently, Ma et al. (2022) achieved promising results using a video transformer (Dosovitskiy et al. 2020; Liu et al. 2021; Vaswani et al. 2017) within a Feature Pyramid architecture.

2.3 Static scanpath prediction

Scanpath prediction is the prediction of discrete gaze locations over time. In static scanpath prediction, a sequence of consecutive fixation locations, and optionally fixation durations, is predicted for a static scene. Scanpath prediction is a challenging task, as scanpaths vary immensely between different observers, stimuli, tasks and observer states, and the sequence order of attended locations has to be regarded, different to saliency modeling.

Classic mechanical models generally generate internal saliency maps which are used to choose a next gaze location, often taking inspiration from neurologic research for sampling approaches. In Itti et al. (1998), inputs from the generated saliency map were used to excite neurons in a 2D "winner-take-all" network. When a neuron exceeded a threshold, a saccade to that location was executed and the area was reset to mimic inhibition of return. Böhme et al. (2006) implemented a linear model that output a saccade from candidates taken from a saliency map and past visited locations. Assens, McGuinness, et al. (2017) employed stacked convolutional layers to calculate spatio-temporal saliency volumes from which scanpaths were sampled. The model was trained on 360° images. Assens, Giro-i-Nieto, et al. (2018) used a generative approach by employing a generative adverserial network architecture (Goodfellow et al. 2014) and had the discriminator train to discriminate generated scanpaths from real human scanpaths. Coutrot et al. (2017) used a Hidden Markov model (Rabiner and Juang 1986) to model saccadic decisions. Schwetlick et al. (2021) used bayesian learning to generate observer- and task-dependant scanpaths based on ground truth fixation maps, building upon the SceneWalk model (Engbert et al. 2014; Schütt et al. 2016). Kümmerer, Bethge, and Wallis (2022) introduced a probabilistic generative model to predict fixations which created a spatial priority map from a saliency map and took gaze history into account in the form of a sequence of previous fixations.

A good overview of different modeling approaches is given in Kümmerer and Bethge (2021) .

2.4 Gaze prediction in egocentric vision

A special subclass of visual attention is built by egocentric videos, where videos are taken from the perspective of the observer. Opposed to classic scanpath prediction where the head of observers generally is fixated during recording of gaze data to focus only on eye movements in isolation, the visual scene in egocentric videos can additionally change quickly through the head movements of the observer. Most studies in this field examine task-dependant viewing and interaction with tools (Tavakoli et al. 2019). As the visual input will be different for each recorded observer, there is only one ground truth per stimuli in egocentric gaze predicition.

Although the data differs substantially from statically recorded videos, the approaches to modeling egocentric gaze are similar to general scanpath prediction. Yamada et al. (2011) used a bottom-up saliency map and egomotion information to predict gaze in egocentric real-world clips. To account for head movements, they calculated two attention maps, differentiating between rotation-based and translation-based movements. Huang, Cai, et al. (2018) processed spatial saliency and temporal saliency of stacked optical flow images in two separate CNNs and include information from previous frames through an LSTM unit. Tavakoli et al. (2019) used a CNN to extract features from individual video frames and uses a gated recurrent unit (GRU) to process these features over multiple time steps. They output a 20 x 20 probability grid and treat the task as a classification problem. In their study, they highlight the strong center bias due to re-focusing via head movements instead of eye movements and concluded that top-down features outperform bottom-up features due to the high influence of the scene understanding of an observer in egocentric vision.

2.5 Dynamic scanpath prediction

Dynamic scanpath prediction extends scanpath prediction to videos and sequences of images. Opposed to a sequence of fixations on a static visual input, a gaze location is predicted per frame or image within a sequence. Analogue to dynamic saliency prediction, dynamic scanpath modeling encounters the same increased computational complexity by working within the three-dimensional spatio-temporal space. Predicting the exact timing of saccades is a highly complex task and as of now, we know of only two models predicting scanpaths in dynamic scenes. A standardized comparison process for model has not yet been established for the field, to a part because many metrics used in static scanpath prediction are not trivially transferable

(see section 3.3), but mainly due to high variability and dimensionality involved in scanpaths.

Boccignone et al. (2020) proposed a model for scanpath prediction in conversational scenes based on multimodal inputs in form of frame and corresponding audio sequences. They calculate a visual and a spatial priority map at each time step and model gaze trajectory in a biased random Brownian walk. In their model, they predict salient patches and differentiate between within-patch exploitation and inter-patch exploration. Roth et al. (2021) introduced an object-based approach where attention for objects is accumulated in a drift-diffusion model (Ratcliff and McKoon 2008) until a saccade to the relevant object is executed. Attendable objects are humans, animals, vehicles and the frame background while attention to objects is quantified via the intersection of a saliency map and an object sensitivity mask calculated per frame. Center bias and inhibiton of return are implemented manually by addition of a gaussian to the saliency map and resetting of excitation levels after foveation.

The model proposed in this thesis employs an end-to-end deep learning process, meaning that the classification of relevant features is left to the training process and other factors like center bias and inhibition of return are not accounted for manually. This was chosen to allow freedom of assumptions and to reach more efficient computation.

3 Methods

In this chapter the methodology employed in this study will be described.

- In section 3.1, the dataset and preprocessing steps are presented
- In section 3.2, the model architecture and loss function are presented
- In section 3.3, we discuss the metrics used in this work to evaluate scanpath predictions
- In section 3.4, we present the training process and data splits involved in further experiments

3.1 Dataset

In this thesis, we work with the gaze dataset gathered in Dorr et al. (2010) within the GazeCom project, consisting of 18 natural videos of real-world, outdoor scenes with gaze data of 54 different observers. The videos are each around 20s long, shot from a static camera and captured at 1280 px x 720 px with 29.7 frames per second. The subjects, students between the ages of 18 and 34, watched the videos on a screen of size 40 cm x 22.5 cm at a distance of 45 cm. Gaze position was recorded in pixel coordinates at 250 Hz with an SR Research EyeLink II eye tracker (SR Research Ltd. 2022), with the coordinates $(0, 0)$ corresponding to a gaze location at the top left corner of the screen. This gaze data was then classified into fixations, saccades and smooth pursuit using velocity thresholding (Dorr et al. 2010). Blinks and other artefacts are furthermore classified as *unknown* and further uncharacteristic eye movements are labelled as *noise*, which can also include noise introduced by the eye tracker. In the following, *noise* and *unknown* are grouped together.

We also considered other datasets such as the DH1FK (Wang, Shen, Guo, et al. 2018b), Vidcom (Li et al. 2011), DIEM (Mital et al. 2011) and Hollywood-2 (Mathe and Sminchisescu 2015) datasets. Based on several factors, we decided to use the Gazecom natural video dataset for this work. First, natural videos and a free-viewing setup allow for the most general exploration of human visual attention free from cinematic cuts or task-specific attention. Second, the video and label format is consistent and of good quality and eye-movement classification data is annotated.

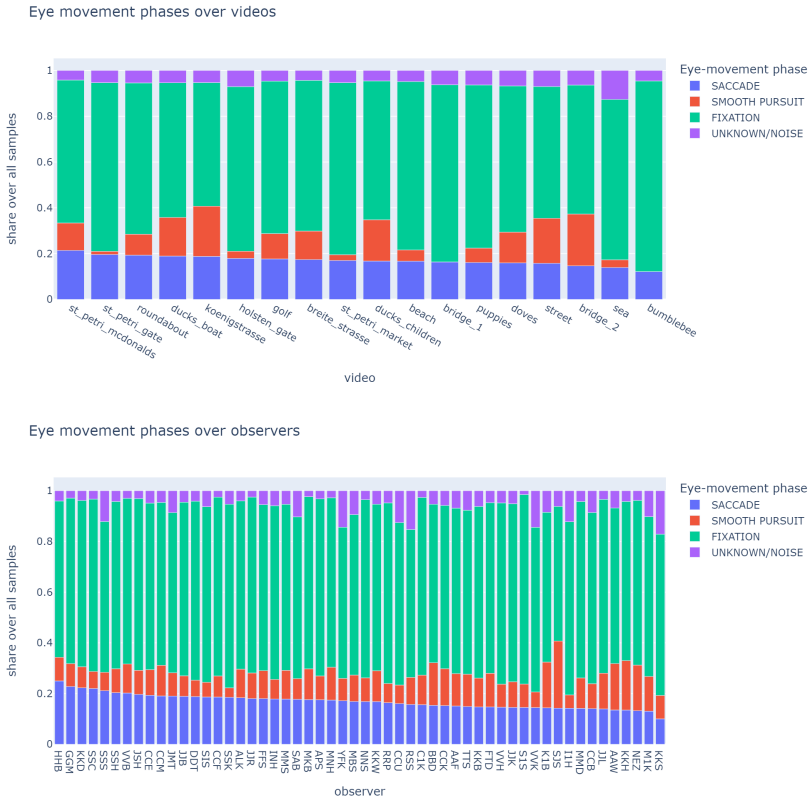

Figure 3.1: Eye-movement phase distribution in Gazecom dataset over videos and observers

Finally, it includes many observers recorded per video which which allows us to evaluate the prediction for multiple ground truths.

We excluded all data from the two videos *bumblebee* and *koenigsstrasse*, as they are shorter than the other videos and the recorded scanpaths have very low variance between observers. The remaining videos consist of 597 to 599 frames.

Viewing behaviour differs across observers and videos, as indicated by the distribution of eye movement phases (Figure 3.1). For videos, this seems to generally be linked to moving objects and object distribution. The deviations between observers highlight the differences in viewing strategies across multiple subjects.

Data preprocessing

For the following training process we down-sampled gaze data to the same temporal resolution as the video data. To gain one gaze label per frame, we took the mean

gaze pixel position and a majority vote for the eye-movement phase classification over all time steps j

$$j \in \left[\frac{(i-1) \cdot 250\,Hz}{29.7\,Hz}, \frac{i \cdot 250\,Hz}{29.7\,Hz} \right] \cap \mathbb{N} \tag{3.1}$$

for each frame i. Due to the short duration of saccades, we labelled a frame as an saccade if at least one recording was classified as a saccade within a time window as otherwise they might be omitted in the majority vote. We did not use the timestamps in the data as according to the initial paper these are not reliable due to I/O bottlenecks. As this is not clear from the data, we assumed here that the gaze recording starts simultaneously with the video presentation. This is supported by the fact that the down-sampled gaze data looks aligned with the video. The number of recordings often don't match the video duration, later recordings were therefore omitted. If an averaged gaze position would fall outside of the valid dimensions, the overflowing dimension would be set to the image boundary.

To keep the data dimension feasible for the training process, the data was resized to a resolution of 224 px x 224 px using *ffmpeg* (Bellard 2022).

For the training of the neural model architecture, the label data was normalized to the range $[-1, 1]$ by

$$y_{norm} = 2 \cdot \begin{pmatrix} \frac{y_x}{w} \\ \frac{y_y}{h} \end{pmatrix} - 1\,. \tag{3.2}$$

We use pre-trained models from the Python (Van Rossum and Drake 2009) deep learning library *Pytorch* (Paszke et al. 2019) for feature extraction and follow their normalization process for the video data. The channel data is divided by the maximum value 255 to reach the range $[0, 1]$, then it is normalized channel-specificly with the means $0.485, 0.456, 0.406$ and the standard deviations $0.229, 0.224, 0.225$ (Pytorch 2022).

$$i_{c,norm}(x, y) = \frac{i_c(x, y) - mean_c}{std_c} \tag{3.3}$$

In order to load the data, we modified a dataloader from the *pytorchvideo* library (Fan et al. 2021). The library already supports parallel video clip sampling, in a randomized or sequential order.

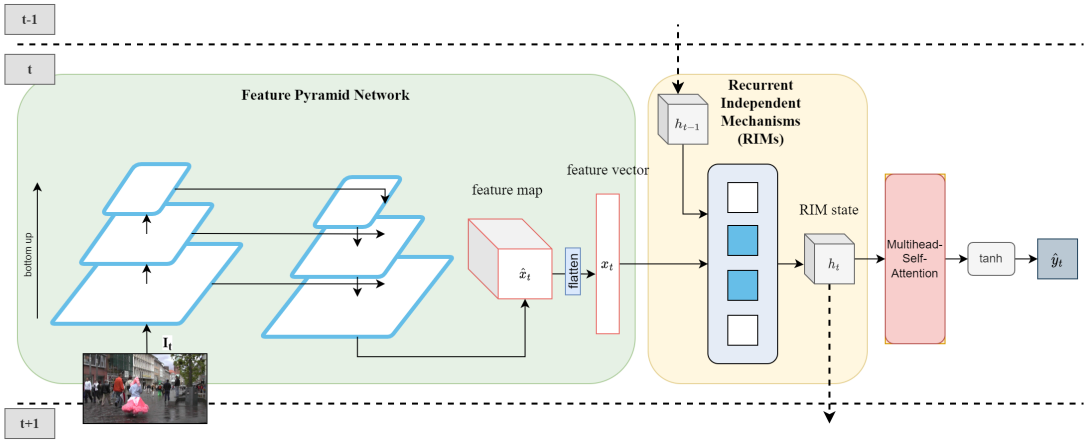

Figure 3.2: Our proposed model architecture. At each time step t the corresponding frame I_t is processed in a Feature Pyramid network (Lin et al. 2016), where image features are extracted in a bottom-up convolutional process which then are merged into an up-sampled spatial feature map in a top-down process. The resulting feature vector is then flattened and input into the recurrent unit which also receives its previous state h_{t-1}. After processing, the new hidden state h_t is fed through a multiheaded self-attention layer to aggregate the state into 2 scalars. After normalizing with a tanh-activation a pixel gaze position $\hat{y}_t$ is predicted.

3.2 Model

The model used in this work is separated into two main units, one for visual feature extraction and one to handle temporal development over a sequence of frames. The feature extraction unit employs a Feature Pyramid network to retrieve salient features from video frames. The recurrent unit uses the architecture of Recurrent Independent Mechanisms to learn independent interacting mechanisms of human visual exploration.

3.2.1 Visual Feature Extraction

The feature extraction unit extracts a spatial saliency map for each frame through the use of a Feature Pyramid Network (FPN) built on top of a Convolutional Neural Network (CNN) (see Figure 3.2). It operates frame-wise and does not incorporate any history from previous frames.

Throughout the last two decades, convolutional neural networks have established themselves as the state-of-the-art for feature processing in many complex computer

vision tasks (Krizhevsky et al. 2012; He et al. 2015; Tan and Le 2019). By breaking down the complex spatial input space in successive convolutional layers, they form condensed feature maps with abstract features learned for the training task.

The feature maps of the deeper layers generally don't have the same spatial resolution as the input, which is not limiting for tasks such as image classification. For many other tasks however, the conservation of finegrained spatial information is important. Feature Pyramid Networks were proposed to allow this (Lin et al. 2016). Inspired by feature pyramids (Adelson et al. 1983) in classical image processing, this architecture exploits the hierarchical structure of CNNs. In a top-down process, a FPN traverses the layers of the initial CNN in reversed order and at each step merges the gained abstract features of the deeper layer into the higher resolution spatial map of the next layer. Generally, only the respective deepest layers for each scale within the bottom-up process are considered in order to reduce computational complexity, as these should contain the most processed abstract features for a given scale.

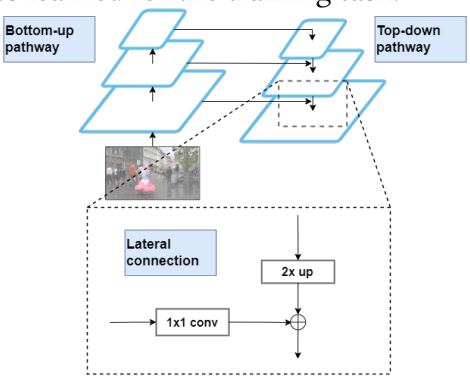

Figure 3.3: The lateral connection within a Feature Pyramid Network (Lin et al. 2016)

At each step in the top-down merging process, the feature map of the previous layer is upsampled to fit the feature dimensions of the current layer. The feature map of the current layer is put through a 1 x 1 convolutional layer to reduce channel dimensions and element-wise added to the upsampled previous feature map. This way the high-level semantic content of the deeper layers is merged with the high-resolution spatial information of the lower level.

Here, we use a pre-trained version of the *MobileNet v3 large* (Howard et al. 2019) architecture as CNN backbone, but also test predictive performance with *VGG-19* (Simonyan and Zisserman 2014) and *EfficientNet B7* (Tan and Le 2019) for the bottom-up process. All pre-trained models are taken from *Pytorch*. We select *MobileNet v3 large* for its efficient forward pass in order to keep inference and training times low and reduce memory demand. We generally set the number of output channels to $c_{out} = 8$. We calculate the top-down process generally down to the second down-sampling layer group, which for an input frame size of 224 px x 224 px generates a feature-vector of 51 x 51 x c_{out}.

We use the last resulting layer in the top-down process and flatten the feature-vector to be processed by the recurrent unit.

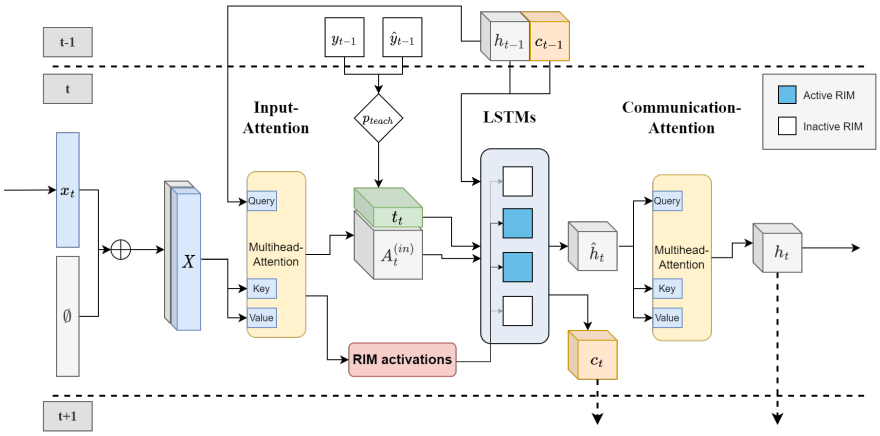

Figure 3.4: Recurrent Independent Mechanisms with Teacher Forcing. The flattened feature vector x_t is stacked with the null input $\emptyset$ and fed to the multiheaded input attention module, where the keys K and values V are calculated in linear transformations from the input. The queries Q_k depend on the hidden state $h_{k,t}$ of the respective RIM unit k and make the input attention RIM-specific. The best-scoring units are activated (blue) for the time step while the other units stay inactive (white). Teacher values t_t are then concatenated to the input vector to maintain a scanpath trajectory, randomly from the previous output $\hat{y}_{t-1}$ or from the previous ground truth label y_{t-1}. This concatenated input is processed in the active LSTM RIM units, which take into account the previous hidden state h_{t-1}. Finally, the RIM units can sparsely exchange information in the multiheaded communication self-attention.

Training: During training, only the top-down process is trained, we leave the parameters of the CNN-backbone untouched.

3.2.2 Recurrent Unit

The recurrent unit of the model (Figure 3.4) connects consecutive gaze predictions by incorporating prediction history as well as features extracted from the current frame. Here, we use Recurrent Independent Mechanisms (Goyal et al. 2019) for our recurrent architecture.

Recurrent Independent Mechanisms were proposed by Goyal et al. (2019) in order to reach higher generalization to unseen data than singular recurrent units through specialization between internal units, generally long short-term memory (LSTM) units (Hochreiter and Schmidhuber 1997) or gated recurrent units (GRUs) (Cho et al. 2014). At each time step, the units compete with each other through

an attention process with only the winners getting to attend to the input. This is aimed at enforcing that units specialize on specific input and tasks. The modular units can exchange information with each other through a sparse communication attention.

Here, we built upon the implementation of Didolkar (2022).

Multihead Scaled-Dot Attention

RIMs utilise content-based soft-attention (Bahdanau et al. 2014) to only attend to relevant information and to determine which units are most suited to attend to a given input. More specifically, Goyal et al. use the scaled-dot product attention proposed by Vaswani et al. (2017). This attention algorithm computes a weighted sum over elements in the input where the weighting is calculated from key- and query-vectors k and q, which are learned to filter only relevant information from the input. A dot product between the vectors is calculated, down-scaled by the square root of the key-dimension d_k and normalized through a softmax. Finally the value-vector v is multiplied with the calculated weight.

In order to process multiple queries at once, keys and values their vectors are stacked in matrix form to build the matrices Q, K and V of sizes $N_Q \; x \; d_k$, $N_k \; x \; d_k$ and $N_k \; x \; d_v$. The softmax function is then applied row-wise to the matrix resulting from the dot-product so that each row .

$$attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.4}$$

This attention process can be extended to multiple attention heads attending to different features. The individual attention results then have to be combined, Vaswani et al. (2017) use a linear layer for this purpose.

Input attention

The input attention mechanisms in RIMs filters which input features are relevant for each RIM unit dependent on the unit's cell state h_t and the input x_t and transform the input features of size d_i to the value dimension d_v.

At the same time, the attention scores are used as a relevance score to control which units are to be activated at each time step. The idea is to add a null-input, compare attention scores between the real input and null-input for each unit and activate the units where the null-input gets the lowest attention scores. The reasoning being that the chosen units have the highest relevant interest in the

presented input x_t. In order to achieve this, we first concatenate the null-input $\emptyset$ as a row of zeros to the input x_t:

$$X = \emptyset \oplus x_t \tag{3.5}$$

The keys $K = XW^e$ and values $V = XW^v$ are calculated in linear transformations from the input X while the queries $Q = h_{k,t}W_k^q$ depend on the hidden state $h_{k,t}$ of the respective RIM unit k and make the input attention RIM-specific. During training, the weight matrices W^e, W^v and W_k^q are learned in order to focus on relevant features for different RIM units. The input attention for a unit k is therefore defined as

$$A_k^{(in)}(X, h_{k,t}) = \text{softmax}\left(\frac{h_{k,t}W_k^q(XW^e)^T}{\sqrt{d_e}}\right)XW^v \tag{3.6}$$

The softmax-function assigns a respective probability for a RIM to attend to the null-input as well as to the input features. A dropout layer with $p_{dropout}^{(in)}$ is applied to these probabilities in order to reach better generalization during training and little dependence on single features. The top k_A RIMs which have the lowest attention probability for the null-input are chosen to be active for the respective time step. This is achieved via masking of the attention values.

For multihead-attention the mean of the attention scores across the attention heads is used for the activation process.

$$A_k^{(in)}(X, h_{k,t}) = \text{softmax}\left(\frac{1}{n_{heads}}\sum_{l=1}^{n_{heads}}\frac{h_{k,t}W_{l,k}^q(XW_l^e)^T}{\sqrt{d_e}}\right)\cdot\frac{1}{n_{heads}}\sum_{l=1}^{n_{heads}}XW_l^v \tag{3.7}$$

In order to parallelize the computation across the different attention heads, the linear transformations are processed across all heads at once by stacking their weight matrices. After the transformation, the results are disentangled again for the dot-product.

Teacher Forcing

In our model, we extend the RIM architecture with teacher forcing, a process named in Williams and Zipser (1989) which describes the feeding of ground truth

information into the training loop of a recurrent neural network (RNN). At each time step t, with a probability p_{teach}, the ground truth target for the last time step y_{t-1} is appended to the input x_t of the current step. This is supposed to prevent the RNN predictions to deviate far from the ground truth and to accelerate the training process. If the ground truth y_{t-1} is not passed, the previous output $\hat{y}_{t-1}$ of the network is passed along instead.

$$x'_t = x_t \oplus x_{teach} \tag{3.8}$$

$$\text{with } x_{teach} = \begin{cases} y_{t-1}, & \text{with } p = p_{teach} \\ \hat{y}_{t-1}, & \text{with } p = (1 - p_{teach}) \end{cases} \tag{3.9}$$

Additionally to the purpose of continuing a trajectory, we employ teacher forcing in our model to account for the existence of multiple ground truth gaze labels per visual sample. During the training process, the teacher values are aimed at guiding the predictions along the scanpath of the current observer. This way, the model can differentiate between different observers, despite being passed the same image data. We also test the influence of teacher forcing if left activated during inference. Note that not the output of the recurrent unit h_{t-1} is passed as a prediction, but the predicted gaze location $\hat{y}_{t-1}$.

In order to increase the probability that the teacher values are attended to and don't get drowned by the magnitude of the value dimension d_v, we append multiple teacher values to the input, each individually being sampled from the ground truth with probability p_{teach}. In our experiments, we set $p_{teach} = 0.3$ and $n_{teach} = 50$.

Recurrent RIM step

After teacher forcing has been applied, the augmented input gets passed to the RIMs for the actual recurrent step $D_k(A_k^{(in)}, h_{k,t})$ within the LSTM RIM units. Through the input attention step each RIM will receive a individually tailored transformation of the input x_t. While deactivated units still process their given input, their hidden state $\hat{h}_{k,t+1} = h_{k,t}$ does not change in this step due to the applied mask.

$$\hat{h}_{k,t+1}, c_{k,t+1} = D_k(A_k^{(in)}, h_{k,t}; \theta_k^{(D)}) = LSTM(A_k^{(in)}, h_{k,t}, c_{k,t}) \tag{3.10}$$

Where $h_{k,t}$ is the hidden state and $c_{k,t}$ the cell state of the LSTM unit k at time step t.

Communication attention

Once the input X has been processed in the RIMs and the hidden state $\hat{h}_k$ of each unit has been updated, the RIMs can exchange information through a communication attention process. During this, only activated RIMs can read the hidden state of all other RIMs to process contextual information. Goyal et al. reasons that the other RIMs are not involved in the current step and therefore should not change their internal state.

In the communication attention, the keys $\hat{K}$, values $\hat{V}$ and queries $\hat{Q}$ are all derived from the new hidden state $\hat{h}_{k,t+1}$ in the linear transformations $\hat{K} = \hat{h}_{k,t+1}\hat{W}_k^e$, $\hat{V} = \hat{h}_{k,t+1}\hat{W}_k^v$ and $\hat{Q} = \hat{h}_{k,t+1}\hat{W}_k^q$. The final new RIM state $h_{k,t+1}$ of each unit is determined through a residual connection to avoid vanishing or exploding gradients (Santoro et al. 2018).

$$h_{k,t+1} = \text{softmax}\left(\frac{\hat{h}_{k,t+1}\hat{W}_k^q(\hat{h}_{k,t+1}\hat{W}_k^e)^T}{\sqrt{d_e}}\right)\hat{h}_{k,t+1}\hat{W}_k^v + \hat{h}_{k,t+1} \tag{3.11}$$

Again, a dropout layer with $p_{dropout}^{(comm)}$ is applied to these probabilities in order to reach better generalization during training. As in the input attention, in practise multiple attention heads are used for the communication between RIM units. Here the attention for each head is calculated, concatenated and fed through a linear layer to combine the heads before adding the residual connection. In order to process the linear transformations of all heads together the weight matrices are again stacked and the results de-tangled for the dot-products.

3.2.3 Multihead-Attention for Aggregation

The final output of the model for each frame is a two-dimensional gaze location. In order to be receptive to variable features in the hidden state h and to let individual RIMs contribute variably, we again use multiheaded self-attention. The queries, keys and values are derived through the linear transformations $h_t\bar{W}^q$, $h_t\bar{W}^e$ and $h_t\bar{W}^v$, where the weight matrices $(\bar{W}^q, \bar{W}^e, \bar{W}^v)$ are learned in the training process.

$$A_h^{(out)}(h_t) = \text{softmax}\left(\frac{h_t\hat{W}_h^q(h_t\bar{W}_h^e)^T}{\sqrt{d_e}}\right)h_t\bar{W}_h^v \tag{3.12}$$

$$\text{with } h_t = h_{1,t} \oplus h_{2,t} \oplus \ldots \oplus h_{k_T,t} \tag{3.13}$$

The attention output $A^{(out)} = A_1^{(out)} \oplus A_2^{(out)} \oplus ... \oplus A_{n_{heads}}^{(out)}$ over the multiple attention heads is then concatenated and fed through a linear layer $a(A^{(out)})$ to generate a two-dimensional output. To limit this output to the range $[-1, 1]$ it is finally passed through a hyperbolic tangent $tanh$ function to calculate the predicted gaze position $\hat{y}_t$.

$$\hat{y}_t = tanh(a(A^{(out)}(h_t)) = \begin{pmatrix} \hat{y}_{t,x} \\ \hat{y}_{t,y} \end{pmatrix} \tag{3.14}$$

3.2.4 Loss function

The network is trained with a mean squared error (MSE) base loss function between the predicted gaze position $\hat{y}_t$ in a frame and the actual ground truth y_t.

$$l_{MSE}(\hat{y}, y) = \frac{1}{L} \cdot \sum_{i=1}^{L} \left(\begin{pmatrix} \hat{y}_{i,x} \\ \hat{y}_{i,y} \end{pmatrix} - \begin{pmatrix} y_{i,x} \\ y_{i,y} \end{pmatrix} \right)^2 \tag{3.15}$$

The MSE loss function punishes large deviations increasingly as it scales quadratically. The loss is only calculated on frames which are not annotated as noise as these are likely not behaviour that will help in the learning process.

Change Rate Regularization

During fixations, human gaze is incredibly stable and does generally not move at all. Conversely, saccades are a very abrupt and fast movement of the eyes, with a barely visible acceleration time (Leigh and Zee 2015). This stark non-linear behaviour may be difficult to learn for a neural network and rather lead to a smoothed middle ground between the two movement dynamics where fixations are not completely still while saccades accelerate slower and are generally slowed down. These assumptions correlate with dynamics that we noticed during earlier implementations of the model. To encourage the model to learn dynamics closer to the biologic example, we therefore introduce two regularization terms l_{fix} and l_{sacc}. $l_{fix}(\hat{y})$ is a MSE loss function which punishes gaze changes between two predictions $\hat{y}_t$ and $\hat{y}_{t-1}$ for fixation frames. This is aimed to reduce noise jitter during fixations. $l_{sacc}(\hat{y})$ is a L1 loss function which is aimed to reward stronger gaze changes during saccade frames. We use a L1 loss here instead of a MSE loss function we assume that the importance for bigger gaze changes does not correlate with the quadratic scaling of a MSE loss. Note that we can distinguish fixation and

saccade frames due to the eye movement phase annotations in our dataset (see section 3.1).

$$l(\hat{y}, y) = \frac{\left(\frac{1}{\lambda_{fix}} + l_{fix}(\hat{y})\right) \cdot l_{MSE}(\hat{y}, y)}{\frac{1}{\lambda_{sacc}} + l_{sacc}(\hat{y})} \tag{3.16}$$

$$\text{with} \quad l_{fix}(\hat{y}) = \frac{1}{|i \in fix|} \cdot \sum_{i \in fix} \left(\begin{pmatrix} \hat{y}_{i,x} \\ \hat{y}_{i,y} \end{pmatrix} - \begin{pmatrix} \hat{y}_{i-1,x} \\ \hat{y}_{i-1,y} \end{pmatrix} \right)^2 \tag{3.17}$$

$$\text{and} \quad l_{sacc}(\hat{y}) = \frac{1}{|i \in sacc|} \cdot \sum_{i \in sacc} \left| \begin{pmatrix} \hat{y}_{i,x} \\ \hat{y}_{i,y} \end{pmatrix} - \begin{pmatrix} \hat{y}_{i-1,x} \\ \hat{y}_{i-1,y} \end{pmatrix} \right| \tag{3.18}$$

We bind the two regularization terms as multiplicators to the base loss l_{MSE} as the regularization incentives will not be relevant anymore for $l_{MSE}(\hat{y}, y) \to 0$. With the hyperparameters λ_{fix} and λ_{sacc} the influence of the regularization can be tuned. For very large values, the regularization will have a more immediate impact on the loss as the will dominate the sum in the respective multiplicative term. For small values of λ_{fix} and λ_{sacc}, the inverse holds.

3.3 Scanpath evaluation metrics

To evaluate the results of the proposed predictive model, we want to quantify the validity of generated scanpaths. This, however, is not a trivial task and different approaches have been discussed in other works (Fahimi and Bruce 2020; Kümmerer and Bethge 2021; Bylinskii et al. 2016b). Given the same video twice, an observer will most likely not generate the same exact scanpath in their viewing process and the same goes for comparison with other observers. Therefore, given just a video sequence, it is impossible to deterministically predict the according scanpath. Still, all of these scanpaths are equally valid.

So far no unifying metric for dynamic scanpath prediction has been established. This hinders the comparison between models and will need to be addressed in future works. Transferring spatial scanpath metrics over to dynamic scanpath prediction can be problematic, often due to the much higher dimensionality and the sparseness of data points within the spatio-temporal space.

String-based similarity measures like the Levenshtein similarity (Privitera and Stark 2000) or ScanMatch (Cristino et al. 2010) assign image regions corresponding letters so that a sequence of fixations or gaze points can be transformed into a

string. They then assign a similarity score based on how many insertions, deletions and substitutions are needed in order to reach the ground truth. To handle the high dimensionality in video data, the spatio-temporal space has to be divided into bins or regions of interest (ROIs) (Boccignone et al. 2020). This is sensible for static visual data, but problematic in video data as a spatial region can change continously.

Area under the curve (AUC) (Wilming et al. 2011) is a metric based on a saliency map over pixels or ROIs and classifies regions as fixated with a decreasing threshhold. This is compared with the ground truth fixations so that a true positive, and false positive classification rate can be calculated. The AUC finally is the area under the curve in a plot of the two measures. *Shuffled AUC (sAUC)* (Tatler et al. 2005; Zhang et al. 2008) extends this approach by sampling its negative fixation locations from ground truth fixations over other images instead of all not-fixated pixels for the stimuli, in order to account for center bias. Both algorithms are however not naturally suited for the very sparse problem of dynamic scanpath prediction and do not account for distance offsets between predicted scanpath locations.

MultiMatch (Le Meur and Liu 2015) examines the similarity between saccades in the prediction and ground truth in terms of shape, length, direction, position, and duration after clustering the observed saccades into similar classes. Boccignone et al. (2020) used *MultiMatch* together with a 3D adaptation of *ScanMatch* for the evaluation of dynamic scanpath prediction. However, as described later in chapter 4, due to the non human-like gaze change dynamics observed in our model, a saccade classification was not sensible in this work.

The *Kullback Leibler* distance (KLD) (Rajashekar et al. 2004) and *log-likelihood* (LL) (also referred to as *information gain* (IG)) (Kümmerer, Wallis, et al. 2015) define distances between probability density functions and therefore require the transformation of a predicted scanpath into a probability distribution and don't always offer intuitively interpretable metric values.

In this work, we use the *Normalized Scanpath Saliency* (NSS) (Peters et al. 2005; Dorr et al. 2010), which employs a specially normalized version of a 3D spatio-temporal Gaussian density estimator generated from observer ground truths and is described in detail below. In theory, NSS is similar to KLD and IG, but does not require a probability density and computes easily interpretable values, with a score larger than $NSS = 0$ indicating a prediction better than by chance. NSS incorporates temporal order in a scanpath and handles temporal and spatial offset via the Gaussian spread around ground truth gaze points.

3.3.1 Normalized Scanpath Saliency

Normalized Scanpath Saliency was introduced by Peters et al. (2005) as a measure of scanpath validity. It builds a normalized 3D Gaussian density volume of recorded valid scanpaths within the spatio-temporal space and then compares a given scanpath with this volume. A scanpath is given as a sequence of attended image locations over time.

First, spatiotemporal Gaussians are added into a gaze map F for each gaze point $\vec{x}_i^j = (x, y, t)$ in given valid scanpaths:

$$F(\vec{x}) = \sum_{i \in O} \sum_{j=1}^{N_i} e^{-\frac{(\vec{x}-\vec{x}_i^j)^2}{2(\sigma_x^2+\sigma_y^2+\sigma_t^2)}} \tag{3.19}$$

where O are all observers, N_i is the number of gaze points in the scanpath of observer i and x_i^j is a gaze point within this scanpath. The calculated gaze volume is then normalized to zero mean and unit standard deviation to form a normalized scanpath saliency volume N:

$$N(\vec{x}) = \frac{F(\vec{x}) - \overline{F(\vec{x})}}{Std(F(\vec{x}))} \tag{3.20}$$

A NSS score of a sequence of attended locations then is their mean score within this map:

$$NSS(\vec{x}_k) = \frac{\sum_{j=1}^{N_k} N(\vec{x}_k^j)}{N_k}. \tag{3.21}$$

A score greater than zero suggests that the given scanpath has a higher similarity to the map than given by a randomly sampled scanpath. A score less than zero does conversely suggest a similarity less than if randomly selected.

As parameters for the gaussians, we chose the same values as in Dorr et al. (2010), with $\sigma_x = \sigma_y = 1.2°$ and $\sigma_t = 26.25$ ms.

3.4 Data partitions

In this section we will go through the training process and the different experiment setups that were examined, especially different partitions of the *GazeCom* dataset. These were chosen to test the predictive abilities of the proposed model in regards to specific sub-problems.

single clip: This data partition is a single 2 s clip out of the video *golf* (frames 115-174) with gaze data of the observer *AAW*. We use this very narrow train dataset as a general sanity check of the model and for hyperparameter tuning. The chosen clip was chosen because it contains clear, salient objects that the observer is following with very little noise in the gaze data. It is a small sequence that the model should be able to learn, including the fixation/saccade dynamics in the groundtruth gaze data, in order to proceed to more complex datasets. We use 2 s clips sampled from the video *ducks_children* with the observers *DDT, FFS* as test data. Because of the specificity of the train set however, we don't expect strong generalization to unseen data.

single video: This partition extends the *single clip* partition by sampling random 5 s clips from the video *golf* instead of only attending one clip. This can rule out that the model only "memorizes" the target sequence and disregards the input features. But again, we don't expect the trained model to generalize well to unseen videos or other observers.

all videos, single observer: We want to examine how well the model generalizes to different visual features, therefore we include all videos, but stay with a singular ground truth for each frame with gaze data from the observer *AAW*. Here, we will check if the model can learn a viewing behaviour of the observer and predict it for unseen visual data.

single video, all observers: As explained before, a large difficulty in predicting gaze data is that there is no single ground truth given visual input data. Here, we will test how well the model can learn to generate valid differing scanpaths, given the same input data. We again use the video *golf*, but all observer data.

all videos, all observers: Finally, we train the model on the whole dataset. We test if the model learns to generalize to new visual data and different observers in an end-to-end learning approach. We use the videos *breite_strasse, bridge_02* and *doves* as a test set and all other videos to train the model.

		training data	
partition	**clip length**	**videos**	**observers**
single clip	2s	*golf* (frames 115-174)	*AAW*
single video	5s	*golf*	*AAW*
single video all observers	5s	*golf*	all except test-observers
all videos single observer	5s	all except test-videos	*AAW*
all videos all observers	5s	all except test-videos	

	validation data	
partition	**videos**	**observers**
single clip	*ducks_children*	*DDT, FFS*
single video	*breite_strasse, bridge_02, doves*	*DDT, FFS*
single video all observers	golf	*SSS, TTS, VVB, VVH, YFK*
all videos single observer	*breite_strasse, bridge_02, doves*	*AAW*
all videos all observers	*breite_strasse, bridge_02, doves*	all

3.5 Experimental Setup

The model architecture was implemented in *Python* (Van Rossum and Drake 2009) with the deep learning libraries *Pytorch* (Paszke et al. 2019) and *Pytorch Lightning*

(Falcon et al. 2019). All models were trained on a 24 GB Nvidia Titan RTX, utilising 12 Intel Xeon E5-2630 cores and 64 GB RAM.

For momentum- and gradient-based optimization during the training process we use an *Adam* optimizer (Kingma and Ba 2014).

4 Results

In this chapter, we will present our experimental results and discuss them in the context of this work and the current state of scanpath prediction in videos.

First, we will go through the base setup of our model in terms of parameters. We will then evaluate the model's predictive performance on the in section 3.4 introduced dataset partitions and discuss possible strengths or weaknesses of the model. Next, we will examine the effect of our chosen regularization approach on the dynamics of predicted scanpaths. We will look at task specialization between RIM units for fixation / saccade prediction on the example of $k_T = 2$ and $k_A = 1$. Finally, we will test the influence of different model parameters like the general RIM architecture, hidden size d_h, and total (k_T) and activated (k_A) RIM units. We will also test performance with different CNN backbone networks.

parameter	value
$c_{out}^{(FPN)}$	8
k_T	6
k_A	4
d_h	400
$n_{heads}^{(in)}$	2
$d_k^{(in)}$	64
$d_v^{(in)}$	400
$p_{dropout}^{(in)}$	0
$n_{heads}^{(comm)}$	4
$d_k^{(comm)}$	32
$d_v^{(comm)}$	100
$p_{dropout}^{(comm)}$	0
$n_{heads}^{(out)}$	2

Table 4.1: Default model parameters

In order to ensure comparability of different models, the models in the following sections are trained based on the default parameters in Table 4.1 with single hyperparameters or model parts modified. Except mentioned differently, all models were trained with a learning rate of $lr = 10^{-5}$, a batch size of 16 and with *MobileNet v3 large* as CNN backbone for feature extraction.

While we will evaluate predictive performance on unseen data, our main objective is to examine if our architecture can model scanpath generation on the training data. For this reason, we generally overfit on the training set and deactivate the dropout layers $p_{dropout}^{(in)}$ and $p_{dropout}^{(comm)}$.

4.1 Predictive performance on different dataset partitions

To assess the predictive capabilities of the introduced model, we train the model on the in section 3.4 introduced partitions. For the complete dataset with all videos and all observers, we use a learning rate of $lr = 10^{-6}$ and for the partition *single clip*, we set $lr = 5 \cdot 10^{-5}$. The other three partitions were trained with the default learning rate of $lr = 10^{-5}$. As regularization parameters we use $l_{fix} = 1000$ and $l_{sacc} = 10$. The partition *all videos, all observers* was only trained for 47 epochs due to the very large training time per epoch. As the loss appears to stagnate at this stage, we think the results can however be compared to the training results over the other partitions.

As shown in Figure 4.1, the model achieves close to perfect train loss on a single 2 *s* clip with gaze data from one observer with $l_{min} = 2.7 \cdot 10^{-7}$ and $l_{MSE,min} = 4.31 \cdot 10^{-5}$ which corresponds to a mean deviation of $\overline{d}_{px} = \sqrt{l_{MSE,min}} \cdot 224\,\text{px} = 1.5\,\text{px}$ per frame from the ground truth location. This shows that the model can learn a short sequence including the corresponding gaze movement dynamics and serves as a first sanity check.

When moving to random 5 *s* training clips in one video for one observer, the model does not eliminate the train loss as well anymore with a mean distance of $\overline{d}_{px} = 10\,\text{px}$ to the ground truth and when we sample clips from all videos, this effect grows stronger with $\overline{d}_{px} = 36\,\text{px}$. It appears that the proposed model is not sufficient to model the vaster visual input space and corresponding scanpaths.

With the introduction of multiple observers in the partition *single video, all observers* and therefore a multitude of valid ground truths per frame, the average deviation of the trained model is even larger with $\overline{d}_{px} = 41\,\text{px}$. The results might indicate that the introduction of multiple observers has a slightly larger impact on the training process than the introduction of additional videos, although this is to be tested further as more observers were added than videos. When working with the whole dataset with all videos and all observer ground truth data, the mean deviation rises to $\overline{d}_{px} = 60\,\text{px}$, nearly one fifth of the diagonal of a frame. From these training results, it follows that the model struggles to fully represent correlations in even the training data if the data space grows larger in terms of visual input space or number of observer scanpaths.

Looking further at the validation loss on the respective unseen datasets, it is visible that for none of the tested partitions the validation loss decreases significantly, but instead rather increases for all partitions except for the partition *single video, all observers*. (Note that the absolute values are not intuitively comparable

Figure 4.1: Evaluation results for models trained on different data partitions; **top-left**: train loss over epochs; **top-right**: validation loss over epochs (absolute values differ due to different validation sets); loss values are smoothed with a rolling average with window size 5; **bottom**: NSS scores over 100 random samples in comparison to different baselines: **left** - to original scanpath; **right** - to always predicting the middle of the frame

as the validation sets differ between partitions) Our model evidently fails to learn general feature correlations that can be applied to new, previously unseen data. The contrast in the slightly decreasing loss for the partition *single video, all observers* can further be explained by similarity between observer scanpaths on the same visual stimuli. If the model for example learns to simply predict the mean gaze location between training observers for each frame, this is likely to be very roughly be applicable to other, yet unseen observers. This is not the same case for the other partitions, as the validation video data is not contained within the training dataset.

Normalized Scanpath Saliency

In order to assess the predictive performance in terms of the validity of attended scanpath locations, we further look at the Normalized Scanpath Saliency scores (NSS, see section 3.3) of over 100 randomly sampled clips from the respective training and validation datasets. The underlying gaussian density map is calculated per video from all available observers. We compare the NSS scores of each samples with two baselines, the NSS score of the original ground truth sample within the density map and the NSS score of a middle baseline in which a model would predict the middle of the video in each frame. Despite being an unrealistic scanpath in terms of variability and gaze change dynamics, the middle baseline serves as an informative baseline because of the center bias involved in gaze movement and has the lowest average distance to all pixels within a frame. As metric, we use the median over the taken samples as NSS scores are prone to strong outliers.

$$\overline{\Delta NSS}_{baseline} = median(NSS(\hat{\vec{y}}_i) - NSS(\vec{y}_{baseline,i})) \tag{4.1}$$

$$\text{with } i \in [1, N_{\text{samples}}] \tag{4.2}$$

We find that all trained models on average produce positive NSS scores over their respective training and validation dataset, therefore by definition produce results better than randomly selected locations. Still, all models generally score below the original baseline (see Figure 4.1), except for the partition *single clip* which nearly perfectly reproduces the original scanpath and therefore produces NSS scores nearly equal to the original with a median difference of $\overline{\Delta NSS}_{orig} = 0.01$. (Note that the original scanpaths have a bias to score higher in general as they are already contained in the gaussian density used to calculate the NSS scores, although this effect is small regarding the large number of observers) The partition *single clip* also heavily outscores the middle baseline with $\overline{\Delta NSS}_{mid} = 2.8$. A similar, but less pronounced pattern was recorded for the partition *single video, single observer* with

$\overline{\Delta NSS}_{orig} = -0.67$ and $\overline{\Delta NSS}_{mid} = 1.2$. When extending this to clips from many videos in the partition *all videos, single observer* however, the middle baseline is on average scoring better than the predicted scanpaths ($\overline{\Delta NSS}_{mid} = -0.29$). This indicates that with a larger visual input space to cover, the tested architecture is not sufficiently discovering salient locations, before even introducing multiple observer ground truths. We will examine in section 4.7, how this can be improved with a larger CNN backbone network.

Single video, all observers is the only data partition for which the trained model outperforms the middle baseline both on the training and the validation set with $\overline{\Delta NSS}_{mid}^{(train)} = 0.54$ and $\overline{\Delta NSS}_{mid}^{(val)} = 0.5$. This however can be linked to both sets containing the same visual data on which an averaged prediction between training observers will achieve a high NSS score. For the whole dataset in *all videos, all observers*, the observed performance on both sets scores on average slightly below the middle baseline with $\overline{\Delta NSS}_{mid}^{(train)} = -0.05$ and $\overline{\Delta NSS}_{mid}^{(val)} = -0.19$. The proposed model's predictions therefore are not closer to salient locations than the constant prediction of the middle.

As expected, the NSS scores on the validation show worse performance than on the training set throughout all data partitions. This effect is strongest the more specific the training dataset is, which indicates that the model does learn general prediction patterns. In qualitative visual tests, the predicted scanpaths don't always stay on clear objects but often move in middle positions between salient parts of the image. In further adjustments, regularization could be introduced to force the model to attend on one object instead of a middle ground.

In general, the validity of the visited locations measured by the NSS score for the model predictions is higher than by chance but is lower than that of the original scanpaths which is to be expected as it is an upper bound by the definition of the NSS score. The models trained on the partitions *single clip*, *single video, single observer* and *single video, all observers*, however, on average beat a model that always predicts the middle in a video while the models trained on *all videos, single observer* and *all videos, all observers* on average perform similarly to the middle baseline.

Gaze change dynamics

We have looked at frame-wise diversions from a ground truth in terms of a loss function and at frame-wise saliency scores of visited locations. By looking at the over-all distributions of length and direction of frame-wise gaze changes (see Figure 4.2), we can examine differences in gaze change dynamics between ground

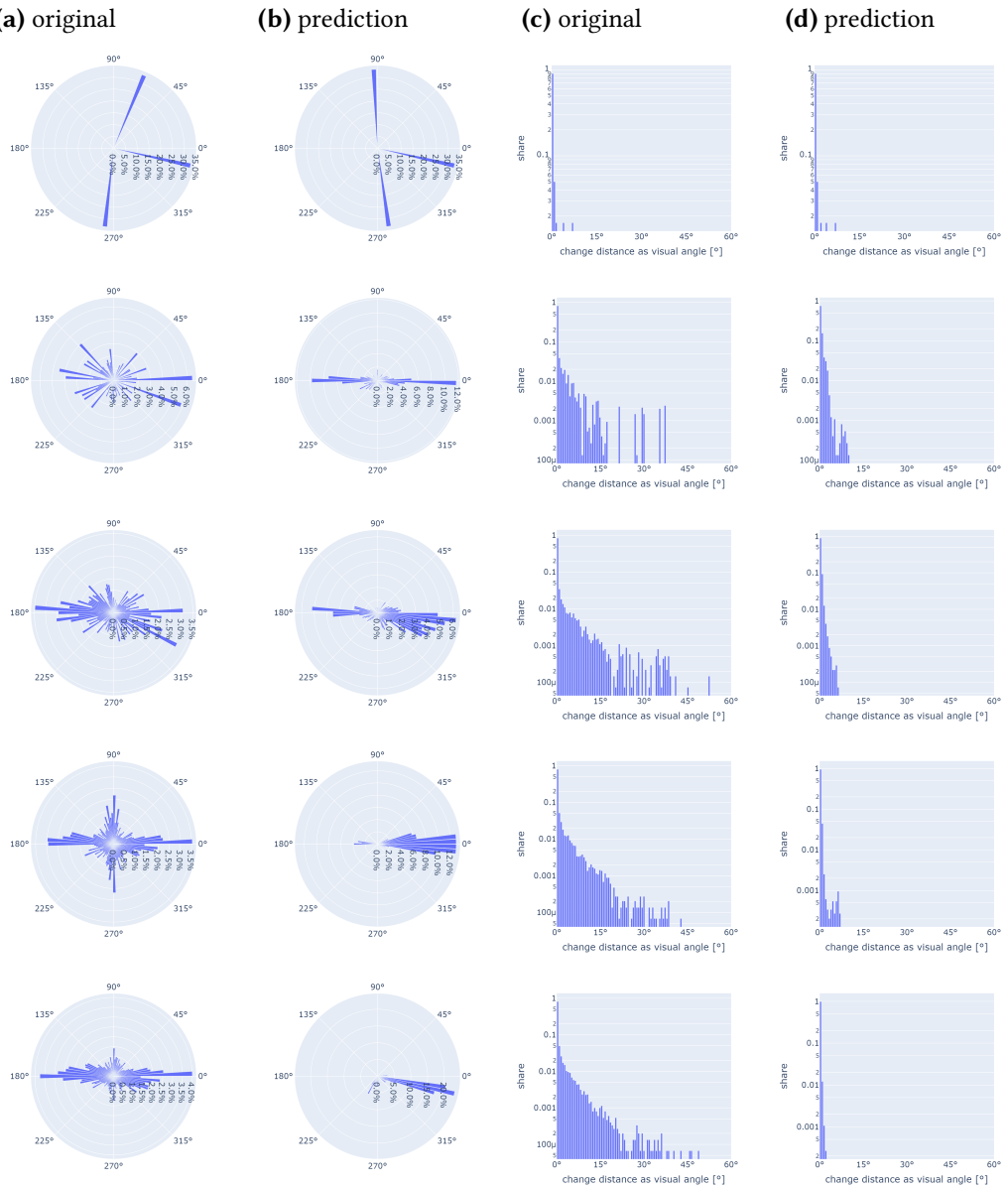

Figure 4.2: Frame-wise gaze change distribution in original and predicted clips over 100 randomly sampled clips from the respective used training set; **Left to right:** (a) gaze change direction in original / (b) gaze change direction in prediction / (c) gaze change distance in original / (d) gaze change distance in prediction (for the direction distribution changes over 1° of visual angle are considered; *Note that the y axis on the distance distributions uses logarithmic scaling*); **top to bottom:** *single clip* / *single video* / *all videos, single observer* / *single video, all observers* / *all videos, all observers*

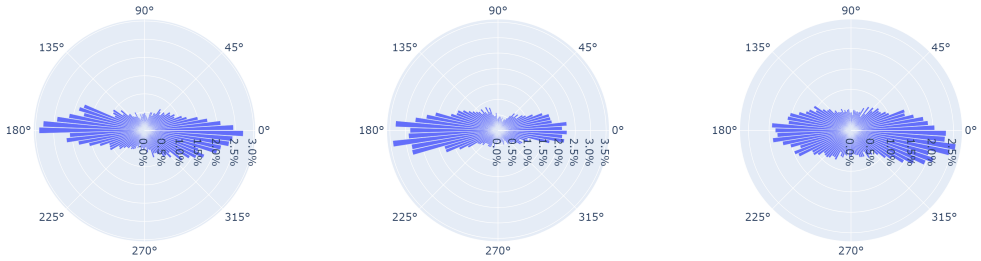

Figure 4.3: Gaze change direction in model predictions without filtering by gaze change distance; **Left to right:** *all videos, single observer* / *single video, all observers* / *all videos, all observers*; taken over 100 randomly sampled clips from the respective used training set

truth scanpaths and predicted scanpaths. We use a logarithmic scaling to visualize the distribution of gaze change distances, as the vast majority of changes is very small. During our evaluation of the trained models, we observed a much more smoothed movement, with little total stillness or saccadic jumps, which both are characteristics visible in human scanpaths. This observed pattern is also visible in the distribution of frame-wise gaze change distances with original scanpaths containing frame-wise jumps (Figure 4.2) of 45° of visual angle and more, while the model distances hardly contain gaze changes with over 10°. While the distance distributions are still almost identical for the partition *single clip*, fewer large jumps occur the larger the training dataset is, with the partition *all videos, all observers* containing no changes covering more than 4° of visual angle.

Our model seemingly does not sufficiently predict fixation and saccade intervals and instead predicts a middle ground without either movement extreme. In another experiment, we tested the capabilities of the model to predict fixation, saccade and smooth pursuit labels in a classification task additional to a gaze location but did not reach any positive results, which supports this hypothesis. The classification task was implemented by one-hot encoding the labels and adding a binary cross-entropy loss for the classification task.

To predict gaze movements which better mimic the movement observed in the ground truth, we tested the effect of our loss regularization terms in detail (see section 4.2).

Besides gaze change distances, we also looked at the distribution of gaze change directions. We consider all changes with a change greater than 1° of visual angle in order to not include noise in the ground truth in the form of gaze changes over one or two pixels, which register as directions of multiples of 90° and to a lesser extent 45° due to the discrete pixel raster. We noticed that gaze labels in many videos have

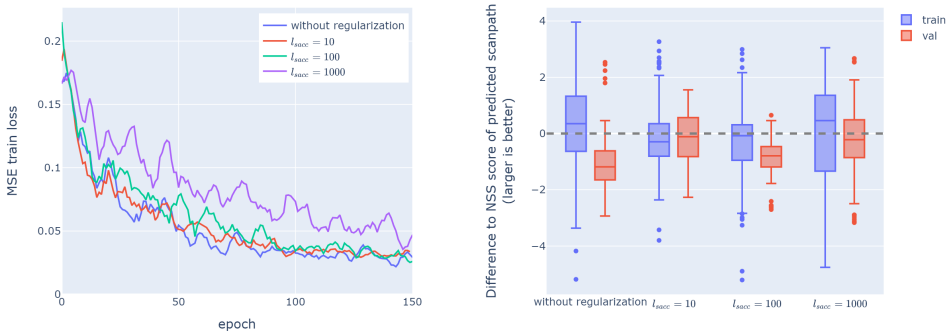

Figure 4.4: Train MSE loss / NSS scores compared to middle baseline when using different regularization strengths; trained on the partition *all videos / single observer* with $d_h = 400$, $k_t = 6$, $k_A = 4$

a bias towards horizontal movements, which most likely corresponds to saccadic movements along objects within the horizontal plane and is fittingly not visible in videos where no horizontal horizon or ground is visible such as *bumblebee* and *doves*. We see this bias to some degree mirrored in the predictions, the models trained on the partitions *single video, all observers* and *all videos, all observers* however show a large bias for movements towards the right side of the video. When regarding gaze change direction over all gaze change distances, this bias to the right is not visible anymore and just a horizontal bias to both sides persists (Figure 4.3). It appears that the model generally learns the orientation of gaze changes better than the distance of gaze changes, however this is not true when accounting for gaze change distance.

4.2 Influence of regularization term

We have established that the model's predictions fail to display the characteristic gaze movements seen in fixations and saccades. To better achieve the characteristic jumps during saccade frames and stillness during fixation frames, we introduced the two regularization losses l_{sacc} and l_{fix}. In this section, we examine the effectiveness of these regularization losses to achieve this goal, especially of the saccade regularization loss. For this purpose, we trained our model on the partition *all videos / single observer* with different regularization strengths. We used a learning rate of $lr = 10^{-5}$ and the default parameters ($d_h = 400$, $k_t = 6$, $k_A = 4$). We examine predic-

tions without regularization, with $l_{fix} = 100/l_{sacc} = 10$, with $l_{fix} = 100/l_{sacc} = 100$ and $l_{fix} = 100/l_{sacc} = 1000$.

First, we test if the training loss still converges with increased regularization strength and if attended locations are valid. We note that while closeness to the ground truth in terms of the mean squared error loss decreases slower with increased regularization, it still is minimized to similar levels (see Figure 4.4). With no regularization, the training process reaches a minimal MSE loss of $l_{MSE,min} = 0.013$ and a corresponding mean distance between prediction and ground truth of $\overline{d}_{px} = 26\,\text{px}$, while with a strong regularization of $l_{fix} = 100$ and $l_{sacc} = 1000$ we still reach $l_{MSE,min} = 0.022$ and $\overline{d}_{px} = 33\,\text{px}$. We can therefore say that not too much gaze location accuracy was lost in order to enforce stronger movement accuracy.

Secondly, we use NSS scores to assess if the validity of the attended locations is affected by regularization strength. We did not see a clear correlation between regularization strength and NSS scores, however the model trained without regularization achieved by far the lowest NSS scores on its validation set, on average scoring worse than a random baseline with a median NSS score of $\overline{NSS}_{pred} = -0.04$.

Looking at the gaze change distributions (see Figure 4.5), we see a clear increase in frame-wise covered distances with regularization ($l_{sacc} = 1000$, $\Delta\hat{y}_{max} = 9.3°$, $\overline{\Delta\hat{y}} = 0.43°$) compared to the model trained without regularization ($\Delta\hat{y}_{max} = 5.3°$, $\overline{\Delta\hat{y}} = 0.76°$). Despite a large increase in gaze change distances, the induced change under a strong regularization still fails to reach movement dynamics similar to the observed ground truth. The used regularization term might therefore not be sufficient to reach the desired saccadic jumps visible in human scanpaths and other options should be explored. It might make sense to predict saccades first and then employ separate modules for the prediction of fixation and smooth pursuit gaze locations and for the prediction of saccade gaze locations. This would allow different modules to specialize on the respective gaze movement type.

4.3 Recurrent Independent Mechanisms vs. single LSTM cell

In our model architecture, we use Recurrent Independent Mechanisms (RIMs) with separate, ideally specializing units, to generalize well to new data. In this section we will try to assess the effect of RIMs by comparing a model utilising RIMs with one utilising a single LSTM cell in their place. We again train our models on the data partition *all videos / single observer* with a learning rate of $lr = 10^{-5}$, a hidden size of $d_h = 400$ and $k_t = 6$ total/$k_A = 4$ activated units for the RIM model.

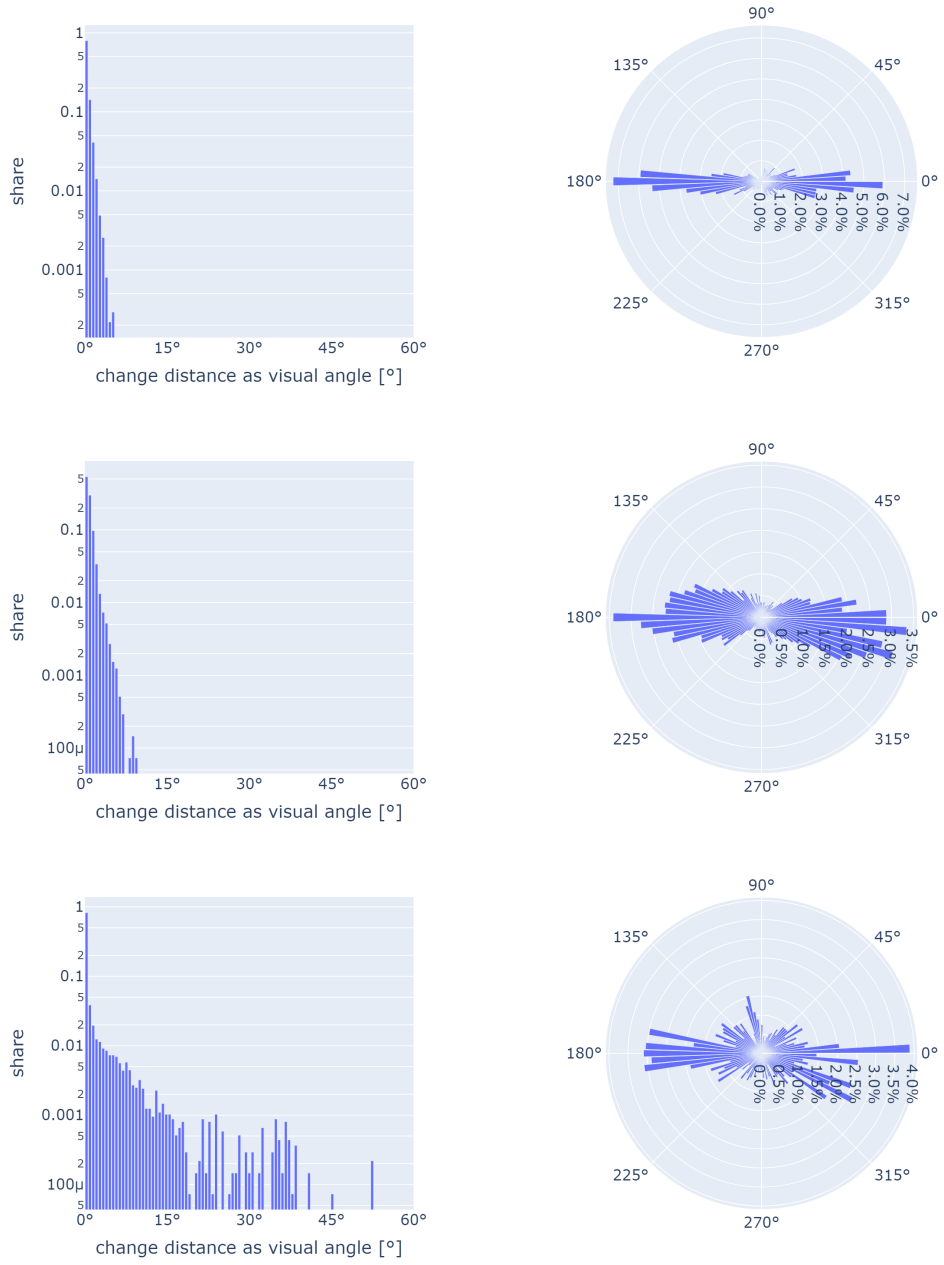

Figure 4.5: Frame-wise gaze change distance (*Note that the y axis uses logarithmic scaling*) and orientation (changes over 1° of visual angle are considered) distribution over 100 randomly sampled clips from the respective used training set; **top**: no regularization; **middle**: $l_{fix} = 100$, $l_{sacc} = 1000$; **bottom**: ground truth; the models were trained on the partition *all videos / single observer*

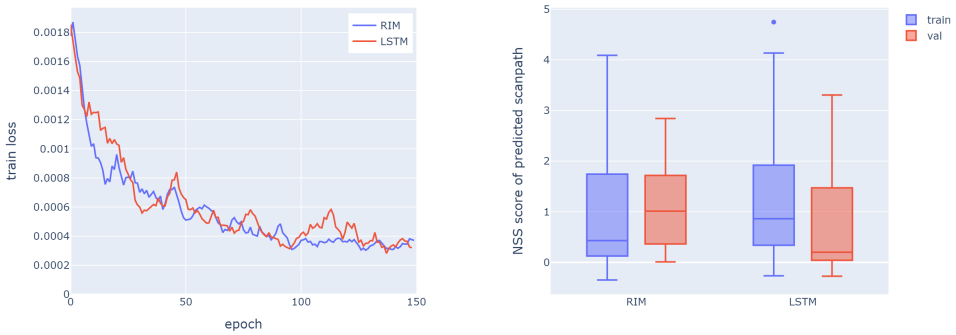

Figure 4.6: Train loss / NSS scores when using RIMs vs. a single LSTM cell; trained on the partition *all videos / single observer* with $d_h = 400$, $k_t = 6$, $k_A = 4$

We saw no significant difference in the training process (see Figure 4.6) between the two models, with both minimizing the training loss to similar levels. NSS scores on the training set on average were slightly higher (see Figure 4.6) for the model employing the single LSTM cell with a median NSS score of $\overline{NSS}^{(train)}_{pred,LSTM} = 0.9 \pm 1.2$ compared to $\overline{NSS}^{(train)}_{pred,RIM} = 0.4 \pm 1.1$ for the RIM model. On the validation set the inverse applies with $\overline{NSS}^{(train)}_{pred,LSTM} = 0.2 \pm 0.9$ and $\overline{NSS}^{(train)}_{pred,RIM} = 1.1 \pm 0.8$. This would be in line with our expectations, as the LSTM is not limited by the bottleneck of communication attention and unit activation for the training process, but would be worse for unseen data than the more general, specialized RIM units. The comparably large variance in the taken metrics however does not allow us to deduce a definite winner and would have to be examined on a larger sample size.

4.3.1 Specialization in RIMs on saccade prediction

An intuitive specialization between RIM units would be the specialization on saccade and fixation or smooth pursuit prediction, given these eye movement types differ drastically in their movement dynamics. To test this in a controlled environment, we train a model with $k_T = 2$ RIM units and $k_A = 1$ on the data partition *single clip*. We used this partition as it is the only one on which a trained model is showing clear saccades in the predicted scanpaths. A learning rate of $lr = 10^{-5}$ and a hidden size of $d_h = 400$ were used in the training process.

In Figure 4.7 (a), the RIM activations of the two units as well as the annotated eye movement type for each frame are shown. The second RIM unit is only active

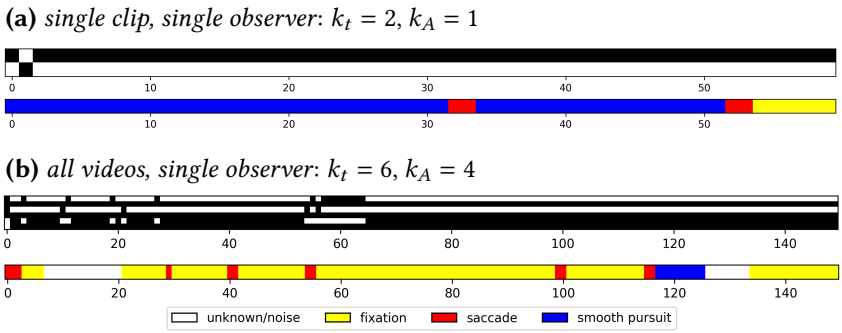

Figure 4.7: RIM unit activations compared with eye movement type; black colouring means a unit was activated at a given timestep, while white symbolises inactivity; (a) trained on the partition *single clip* with $d_h = 400$, $k_t = 2$, $k_A = 1$; (b) trained on the partition *all videos, single observer* with $d_h = 400$, $k_t = 6$, $k_A = 4$

for the second time step, in all other time steps the first unit is activate. During the two saccades visible in the clip, we don't see a change of activation and generally no correlation between eye movement types and RIM activations. In this model and clip, we can therefore not see any intuitive specialization between RIM units. We observed the same on other samples, as can be seen for a random clip from the training set for a model trained on the data partition *all videos, single observer* with $d_h = 400$, $k_t = 6$, $k_A = 4$ (see Figure 4.7 (b)). With a different surrounding architecture allowing for saccadic movements for bigger datasets, this could however be examined again.

4.4 Influence of Teacher Forcing on training process

We introduced teacher forcing (subsection 3.2.2) as a measure to improve convergence during the training process and in order to keep predicted trajectories consistent in themselves and to the underlying ground truth. To examine its effect we train a model without the use of teacher forcing and one with $p_{teacher} = 0.3$ and $n_{teacher} = 50$ on the data partition *all videos / single observer* ($lr = 10^{-5}$, $d_h = 400$, $k_t = 6$, $k_A = 4$).

We note that our training loss converges slightly faster with teacher forcing as expected, but does not lower the final loss (Figure 4.8). As the gaze position on the previous frame for most frames approximates the gaze position on the current frame, it is expected that this additional information would aid the predictive process.

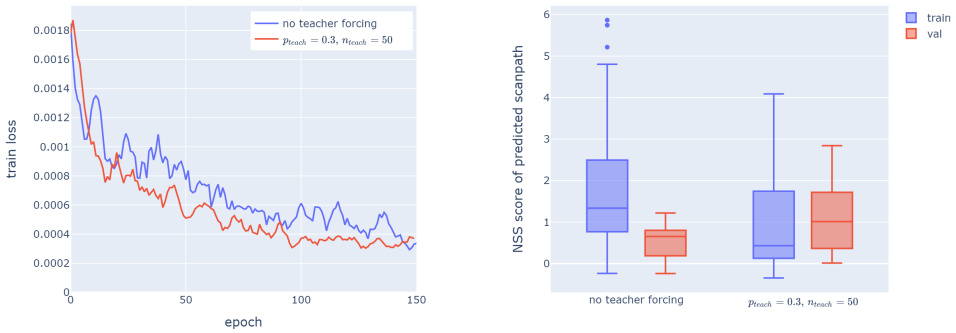

Figure 4.8: Train loss / NSS scores with and without teacher forcing; trained on the partition *all videos / single observer* with $d_h = 400$, $k_t = 6$, $k_A = 4$

In terms of normalized scanpath saliency, we record a decrease of the average score on the training set and an increase on the validation set. As the model learns during training with teacher values for the training set, the forward pass might rely on them and therefore perform worse on the training set in inference. As the achieved clip NSS scores however contain a strong variance, these differences are not conclusive.

4.4.1 Teacher Forcing on inference

Teacher forcing is generally aimed at aiding the training process and therefore does not include ground truth labels on inference. We test leaving teacher forcing activated during inference and thereby changing the model problem to a softened auto-regressive task where the model at each time step t may have information about the gaze label of the previous time step $t-1$. We sample 100 random clips from the models trained on the different data partitions in section 4.1 with teacher forcing on inference and compare it to the results without to assess the effect.

We find that while leaving teacher forcing active during inference has little effect on the highly overfitted models for the data partitions *single clip* and *single video*, we reach a clear improvement in NSS scores for the train and validation set of the other three partitions. This is especially pronounced for the partition *single video, all observers* with an improvement of the median NSS score by $\Delta\overline{NSS}_{pred}^{(train)} = 1.8$ on the training data and by $\Delta\overline{NSS}_{pred}^{(val)} = 2.0$ on the training data, which indicates that the model learns to separate different observer scanpaths by the added ground truth history. The results show that the appended teacher values are used for the

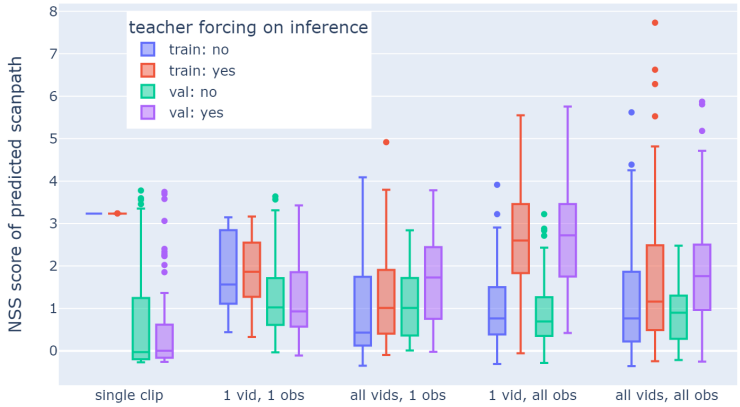

Figure 4.9: NSS scores across data partitions with and without teacher forcing *on inference*; trained with $d_h = 400$, $k_t = 6$, $k_A = 4$

generation of predictions and that a ground truth label evidently helps achieve more accurate predictions and differentiate between observer scanpaths and different visual stimuli.

It can be argued that passing ground truth labels of the previous time step on inference is too large of a concession and that it would make a model impractical to be used outside of practical experimentation. However, the prediction could be computed based on the current frame, the hidden state (h_0, c_0) and the previous gaze location. This would be similar to other works like Kümmerer, Bethge, and Wallis (2022), where the previous 4 fixations are taken as input for a scanpath prediction. As our model is frame-based and not fixation-based on the temporal scale, the much smaller distance between time steps might mean that application is impractical in practice however and limit a model's usefulness severely. It could be used however to give a model a starting trajectory, similarly to Kümmerer, Bethge, and Wallis (2022). The rest of the ground truth values would have to be masked for this.

4.5 Training with observer-specific initial RIM state

As our proposed model only receives visual input, it cannot distinguish between different observers. Different observers might however display very different

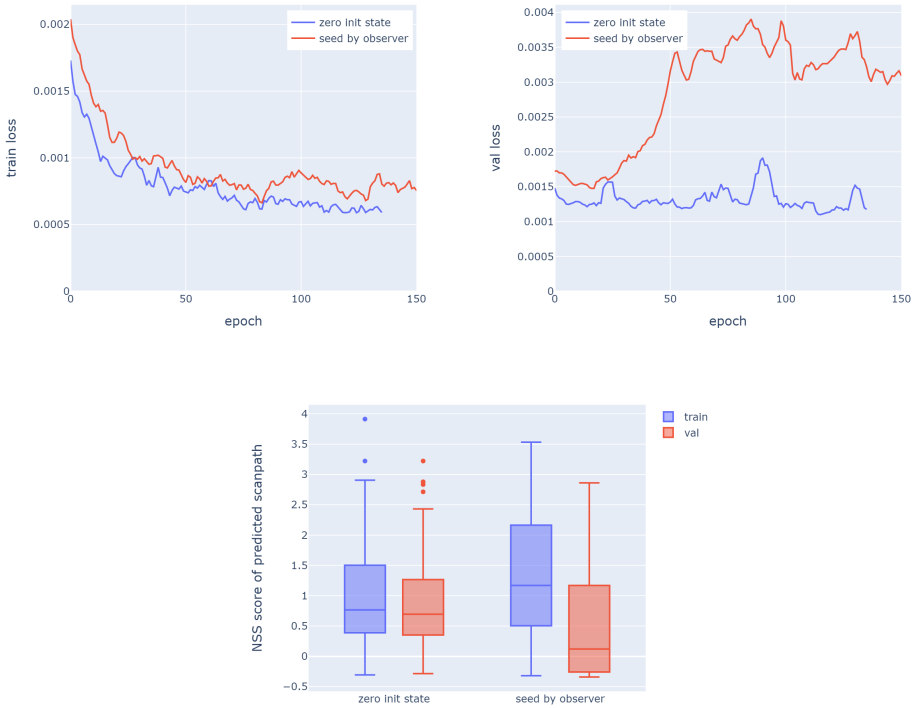

Figure 4.10: Train loss / validation loss / NSS scores with $h_0 = c_0 = \vec{0}$ and with a random, observer-specific initial hidden (h_0) and cell state (c_0); trained on the partition *single video / all observers* with $d_h = 400$, $k_t = 6$, $k_A = 4$

viewing patterns that could be adjusted to when predicting a scanpath. We therefore generate an observer-specific initial state $(h_0(s),\ c_0(s))$ through a pseudo-random number generator with a seed $s(obs)$ that depends on the observer obs. We test the model performance with and without this observer-specific initial state. We train our models on the data partition *single video / all observers* in order to include multiple observers between which the model can learn to differentiate. We again use a learning rate of $lr = 10^{-5}$, a hidden size of $d_h = 400$ and $k_t = 6$ total and $k_A = 4$ activated RIM units.

With the implementation of an observer-specific init state, the model converges at a slower pace and has a noisier convergence trajectory (see Figure 4.10), possibly because the introduction of a variable, pseudo-random init state complicates the loss landscape further. In terms of NSS scores, we see that the modified model performs slightly better on the training set with a median NSS score of $\overline{NSS}_{pred}^{(train)} = 1.2$ than on the original model ($\overline{NSS}_{pred}^{(train)} = 0.8$). It appears that the model learns to specialize to predict observer-specific scanpaths, although the improvement is clouded by a strong variance in scores. On the other hand, the observer-specific model performs worse on the validation set. This is also visible in the validation loss which ends up being much higher compared to training with a zero init state. This makes sense however as the model operates with an init state that did not occur in training as the observers in the validation set were not included in the training set. In future works, it could be examined if observer-specific training could help for predictions on previously unseen by learning viewing patterns of different observers. Due to time constraints and the large training time on the dataset partition *all videos, all observers*, we were not able to train a dataset for this comparison in the scope of this work.

4.6 Predictive performance with different clip durations

Sequence length is a significant parameter within any recurrent neural network. For long sequences, gradients might vanish in training while a model might not be able to build a meaningful internal state without being passed previous historic data for short sequences. In this section, we will test the influence of different clip durations l_{clip} for the predictive performance of the model. For this purpose we train three models with the clip durations $l_{clip} = 2\,\text{s}$, $l_{clip} = 5\,\text{s}$ and $l_{clip} = 10\,\text{s}$. We train our models as before on the data partition *all videos / single observer* with a learning rate of $lr = 10^{-5}$, and default parameters $d_h = 400$, $k_t = 6$ and $k_A = 4$.

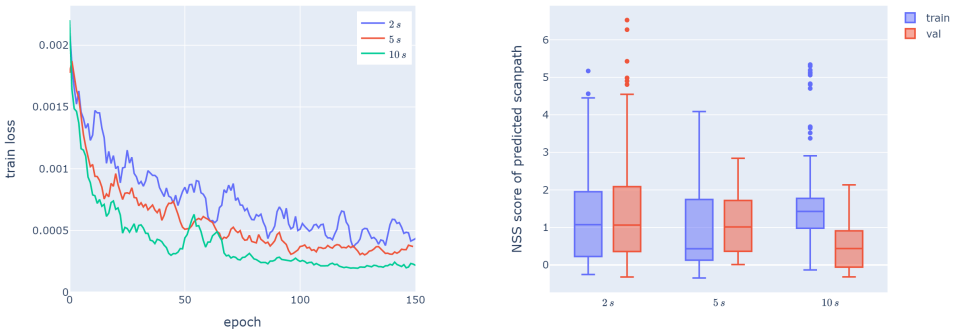

Figure 4.11: Train loss / NSS scores for different clip lengths; trained on the partition *all videos / single observer* with $d_h = 400$, $k_t = 6$, $k_A = 4$

We first note that the training loss declines faster with a longer clip duration (see Figure 4.11). As the RIM units are initialized with a zero hidden state of $h_0 = \vec{0}$, the network might need a number of steps to build a meaningful internal state and would predict less valid gaze points without knowledge of a previous trajectory. For shorter clip durations, the effect would be bigger and therefore lead to worse mean losses.

In terms of NSS scores (Figure 4.11), we record the highest tested clip duration $l_{clip} = 10$ s with the highest average NSS score of $\overline{NSS}_{pred}^{(train)} = 1.4$ on the training set while achieving the lowest average score on the validation set with $\overline{NSS}_{pred,RIM}^{(val)} = 0.4$. The longer clip length might allow the network to specialize further on the training data, therefore overfitting and losing generalisation for the validation data.

4.7 Evaluation of different backbones

In order to attend to meaningful visual features within the model, these features need to be provided by the Feature Pyramid network and the underlying CNN backbone. As different convolutional networks develop differingly suited feature representations , we will examine the predictive properties of our proposed model with 3 different networks for feature extraction: *MobileNet-v3 large* (Howard et al. 2019) with 5.5 million parameters, *EfficientNet B7* (Tan and Le 2019) with 66 million parameters and *VGG-19* (Simonyan and Zisserman 2014) with 144 million parameters. To train the model with *MobileNet-v3 large* as backbone, we use a batch size of $b = 16$ and a learning rate of $lr = 10^{-5}$. For *EfficientNet B7* we use $b = 5$ and

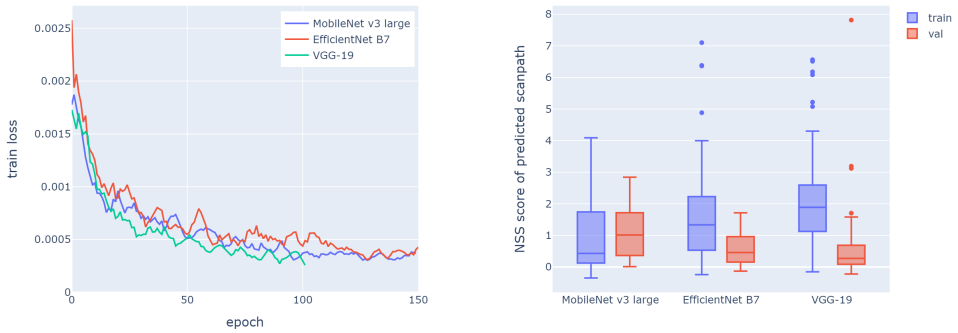

Figure 4.12: Train loss / NSS scores with the CNN backbones *MobileNet v3 large*, *EfficientNet b7* and *VGG-19*; trained on the partition *all videos / single observer* with $d_h = 400$, $k_t = 6$, $k_A = 4$

$lr = 5 \cdot 10^{-6}$ and for *VGG-19* we set $b = 3$ and $lr = 5 \cdot 10^{-6}$, so that the forward pass can fit our available memory. We train all models on the data partition *all videos / single observer* with default parameters $d_h = 400$, $k_t = 6$ and $k_A = 4$.

Although training times are much higher for *EfficientNet B7* and *VGG-19* due to the small batch size and the much larger model, the training loss trajectories of the 3 trained models are very similar (Figure 4.12), with the model with *VGG-19* converging slightly faster than the others. As the model trained with *MobileNet-v3* utilizes a training rate twice of the other models it technically learns slower per epoch. This is somewhat equalized by a lower training time per epoch of $\overline{\Delta t}_{MobileNet} = 27$ s compared to $\overline{\Delta t}_{VGG} = 33$ s and $\overline{\Delta t}_{EfficientNet} = 40$ s.

When looking at the achieved NSS scores, we note that the models employing the larger networks *EfficientNet B7* ($\overline{NSS}^{(train)}_{pred} = 1.3$) and *VGG-19* ($\overline{NSS}^{(train)}_{pred} = 1.9$) score higher on the training set samples than the model employing *MobileNet-v3* ($\overline{NSS}^{(train)}_{pred} = 0.4$) while scoring slightly lower on the validation set (*EfficientNet B7*: $\overline{NSS}^{(val)}_{pred} = 0.5$; *VGG-19*: $\overline{NSS}^{(val)}_{pred} = 0.3$; *MobileNet-v3*: $\overline{NSS}^{(val)}_{pred} = 1.0$). For 34 % of samples on the training set and 47 % on the validation set the model employing *MobileNet-v3* beats the middle baseline, while for *VGG-19* this is the case for 77 % of samples of the training set and only for 19 % out of the validation set. Due to the high variation in the scores these comparisons contain possible volatility, but it appears that the larger convolutional backbones allow the model to specialize further to the training data while learning less general applicable patterns. This

effect could most likely be mitigated by applying a dropout layer onto the visual input features, in order to avoid overfitting.

5 Discussion

The objective of this thesis was to further explore modeling approaches for the so far sparsely explored area of dynamic scanpath prediction and to test the validity of Recurrent Independent Mechanisms (RIMs, Goyal et al. 2019) as a tool of reaching improved generalization to unseen data compared to existing recurrent models like LSTMs (Hochreiter and Schmidhuber 1997) or GRUs (Cho et al. 2014). We proposed and evaluated an end-to-end deep learning approach extracting visual features from individual frames via a Feature Pyramid network (Lin et al. 2016) and processing temporal development within RIMs. We employ teacher forcing (Williams and Zipser 1989) in order to predict consistent scanpath trajectories. The model was trained on natural videos in a free-viewing setup (Dorr et al. 2010). Due to the high dimensionality in video data and the immense and complex variance between scanpaths given a visual stimuli, dynamic scanpath prediction poses a challenging task. Compared to other fields of visual attention, like saliency prediction (Borji 2018; Borji et al. 2013) or static scanpath prediction on images (Kümmerer and Bethge 2021), few computational models (Boccignone et al. 2020; Roth et al. 2021) have been proposed in the field of dynamic scanpath prediction. The development of better models in the field bears importance, as it can lead researchers to develop a deeper understanding of the human visual exploration process by evaluating which approaches produce promising results and which modelling approaches struggle to capture correlations.

We saw that we can successfully overfit the proposed architecture on small clips with a singular ground truth and that the model generally learns to attend salient locations. This indicates that the model in principle is able to learn patterns translating visual stimuli to valid gaze locations. We however also found that the model fails to generate human-like eye movements when moving to bigger datasets in terms of more video data or further observer ground truths. Even with the help of a regularization term in the loss function to incentivise bigger jumps during frames in which an observer executed a saccade, the observed predicted gaze movements seemed smoothed out and far less abrupt than human eye movements (section 4.2). We believe that the observed dynamics were caused by the model not being able to predict saccade timings and therefore predicting an averaged, smoothed scanpath trajectory. We base this on a failed attempt to predict eye movement phase data for each frame, which failed even when supplementing ground truth gaze locations

via teacher forcing. Trained on the whole dataset, the model outperforms random uniform prediction as a lower baseline and performs similarly well to a constant center gaze, representing the center bias observed in human scanpaths (Parkhurst, Law, et al. 2002; Parkhurst and Niebur 2003; Tatler et al. 2005; Tseng et al. 2009). This was measured by spatiotemporal proximity of predicted gaze locations to observer gaze locations, expressed in the Normalized Scanpath saliency (Dorr et al. 2010; Peters et al. 2005). We furthermore did not see a visible specialization among the RIM units (subsection 4.3.1) or a significantly increased generality to new, previously unseen data (section 4.3).

Because of the absence of saccadic jumps in the observed gaze change dynamics of our model's gaze predictions (section 4.1), a classification into fixation, smooth pursuit and saccade frames was not a feasible option in this work. This makes the application of typical evaluation mechanisms between scanpaths impossible, such as the distribution of foveation duration, saccade amplitudes and return times to previously fixated objects, as many depend on eye movement type classification in the first place. These metrics should however be regarded when improvements through further works are made. In the future, the generation of explicit gaze events could be forced through the implementation of saccade trigger mechanisms and a separation between the prediction of slow and abrupt gaze movements.

Even though the prediction results differ starkly from human scanpaths in their movement dynamics and we did not observe specialization between RIM units, the validity of RIMs for this purpose can not be ruled out, as the poor movement characteristics most likely are a consequence of modeling all eye movement types within the same unit and using a regression objective. Human eye movements move along highly non-linear trajectories, with abrupt changes, and might be difficult to model within the same neural unit. Abrupt saccadic jumps might better be predicted in an independent unit that is triggered by a saccade generation mechanism. Saccade generation could be done by accumulating attention for locations until a threshhold is reached as seen in Engbert et al. (2014), Roth et al. (2021), and Schütt et al. (2016). The different predictive units could then manually be triggered and specialize on the prediction of slow and rapid eye movements without having to combine both together. Such a modeling approach would be supported by evidence which points to saccade and fixation eye movements being processed within partially separate regions of the human brain (Purves et al. 2000).

Generative modeling

When training on a dataset containing multiple different observer ground truths, we would expect a model to mirror this variance between scanpaths over the same

visual stimuli in its predictions. In our evaluation, we did not see an equivalent variance in the predicted scanpaths. Most variance was minor and random, with the best results achieved when activating teacher forcing during inference (subsection 4.4.1). We believe that a regression model might be ill-suited to generate new valid scanpaths compared to a naturally generative modeling approach, unless given much initial information, for example in the form of teacher forcing on inference. Due to the probably near infinite number of possible scanpaths, it is arguable if a dataset for a regressive model can feasibly represent all relevant feature combinations. A fully generative model might be better suited to reach predictions not included within the initial dataset.

In our current model, variance in generation can be reached by activating the dropout layers within the RIM units during inference, through modification of the initial RIM state (h_0, c_0) or by using teacher forcing during inference. Dropout generally is a method used to avoid overfitting during training, but when activated during inference can be used to introduce generative randomness to a model. In our experiments, variation via dropout however appeared to simply introduce jitter from a mean path during scanpath generation and failed to generate scanpath variance as seen between human observers. Similar effects were introduced when starting with a random initial RIM state $(h_0, c_0) \sim \mathcal{N}(0, 1)$ in order to introduce random variance between predictions. A better variance in scanpaths was reached when generating differing init states per observer $(h_0, c_0) = f(obs)$, assuming a different viewing behaviour between observers. This however requires the observer as an additional input to the model and can introduce undefined behaviour for observers which were not included in the training set. We observed this reflected in poor performance in *NSS* scores on the validation set and a high validation loss in section 4.5. Finally, the strongest variance in predicted scanpaths between observers is reached when using teacher forcing during inference, but also requires the most additional input information in the form of previous ground truth gaze location labels. It shows that teacher forcing can be used to initiate or guide scanpath prediction and could be used to pass the beginning or parts of a scanpath to the model and let the model complete it. Additionally, by randomly sampling each teacher value from the ground truth label y_{t-1} and the previous model prediction $\hat{y}_{t-1}$, some random variance is further introduced to model predictions.

In order to extend the proposed model to a more generative one, a simple approach could be to change the output to an internal probability map for the next gaze position and randomly sample from it. This would produce randomly varying scanpaths, however, most likely additional measures like resetting local probability after an fixation would have to be implemented in order to prevent the model to always attend to the same high-probability locations and to reach meaningful vari-

ance between generated scanpaths. On the other hand, the model also should not constantly jump between these high-probability locations and should probably take inspiration for saccade frequency from observed experimental results and not jump to new locations during every frame. For the generation of an internal probability map, Gaussian Mixture models (GMMs) (Mclachlan and Peel 2000) could be used which model distribution maps through the superposition of multiple Gaussians. GMMs might be suited since they don't introduce much more complexity to the model and can model spread-out saliency of different salient objects independently. Such a model would however be limited by the number of Gaussians considered, unless the parameter is flexible within the model, and also by the circular spread of Gaussians which needs to be compensated by combining multiple Gaussians for more complex distributions.

For more complex generative modelling, concepts like Generative Adversarial networks (GANs; Goodfellow et al. 2014), variational Auto-Encoders (VAEs; Kingma and Welling 2013) or diffusion models (Sohl-Dickstein et al. 2015) could be utilised. GANs are deep learning models trained employing adversarial training where a generative model competes with a discriminatory model so that it outputs are not recognized as artificially generated by the discriminator. They have been proven powerful for the generation of images and other synthetic data and were used in Assens, Giro-i-Nieto, et al. (2018) to generate scanpaths on static visual stimuli. VAEs on the other hand are auto-encoders where the learned latent representation is regularized so that random points in the latent space translate to new valid data when input into the decoder. Diffusion models are a model class inspired by non-equilibrium statistical physics. During training, inside a Markov chain, Gaussian noise is added iteratively to the input until it is not recognizable anymore, then the process is inverted to recreate the input. By passing a noise input to the reverse process, new data can be generated. Diffusion models have become popular throughout the past few years for the generation of synthetic data (Ramesh et al. 2021) and have been shown to outperform GANs on tasks like image synthesis (Dhariwal and Nichol 2021). An implementation building upon any of these concepts could flexibly generate new scanpaths and could be a next step towards more human-like scanpath generation for dynamic real world scenes.

Preserving spatial relations

As we used unmodified LSTMs within our RIM architecture, we needed to flatten the feature map extracted in the Feature Pyramid network before passing it to the recurrent unit. During the flattening process, the underlying spatial structure of the extracted features is discarded, which could influence prediction results.

Positional encoding is a possible approach to this problematic, where each element in the extracted feature matrix is encoded depending on its corresponding position within the video frame. Positional encoding can be learned or fixed (Gehring et al. 2017; Vaswani et al. 2017). We experimented in our model with a learned additive positional encoding which was randomly initialized, but did not observe a visible learned structure and did not see improvements in predictive performance.

A different approach would be to use convLSTMs (Shi et al. 2015) and replace the linear transformations inside of the LSTM units with convolutions. This would eliminate the need to flatten the feature vector and preserve spatial information. A downside of this approach is the increased computational complexity with introduced convolutions and the combined difficulty to parallelize the computation across RIM units.

Finally, as recently employed in dynamic saliency prediction (Chang and Zhu 2021; Liu et al. 2021; Ma et al. 2022), spatial and temporal patterns could be processed together in 3D spatio-temporal convolutions (Tran et al. 2017; Xie et al. 2017) or spatio-temporal video transformers (Dosovitskiy et al. 2020; Liu et al. 2021; Vaswani et al. 2017). A downside of these methods can be that they tend to look ahead which, depending on the use case, might need to be accounted for.

6 Conclusion

In this thesis we present an end-to-end deep learning model for dynamic scanpath prediction and train and evaluate it on free-viewing gaze data of natural videos. We employ Recurrent Independent Mechanisms (Goyal et al. 2019) to process recurrency between frames and to generalize well to new data through specialization between competing recurrent units. Our proposed model uses a Feature Pyramid network (Lin et al. 2016) to extract visual features from individual frames while preserving spatial information. In order to predict consistent trajectories and to reduce training times, we apply teacher forcing and and append ground truth labels of previous steps to the model input during training.

Our work contributes to the current state of research in three ways. First, we introduce a new model for frame-wise dynamic scanpath prediction, an area where only few models exist as of now, compared to related fields like saliency modeling and static scanpath prediction on images. Second, we examine the performance of Recurrent Independent Mechanisms on a highly complex task and compare it with the implementation of an LSTM. Third, we test and discuss different methods to work with multiple observers on gaze prediction in natural videos.

We trained and evaluated our model on different partitions of the GazeCom natural video dataset (Dorr et al. 2010) to test the influence of multiple observer ground truths and a larger visual input space. We find that our model can learn accurate gaze prediction on small clips with one observer, but that predictive performance on the training set diminishes with more introduced videos or observers. The model generally learns to attend salient locations which is supported by Normalized Scanpath saliency (Peters et al. 2005) scores, but despite introduced regularization does not learn to predict human-like scanpaths with characteristic saccadic jumps in gaze location. With teacher forcing on inference, we see a significant improvement in prediction results and that the model differentiates better between observers. Upon testing specialization between RIM units, we could not find an intuitive pattern correlating with eye movement phases in the RIM activations and did not see a significantly improved generality to unseen data compared to performance with a LSTM.

For future works, we suggest the implementation of a more generative model sampling from Gaussian mixtures or employing a state-of-the-art generative archi-

tecture. We also believe that modeling a saccade trigger and separating fixation and saccade gaze prediction can help to reach human-like gaze predictions.

Bibliography

Adelson, Edward, Charles Anderson, James Bergen, Peter Burt, and Joan Ogden (Nov. 1983). **Pyramid Methods in Image Processing**. *RCA Eng.* 29 (see page 15).

Assens, Marc, Xavier Giro-i-Nieto, Kevin McGuinness, and Noel E. O'Connor (2018). *PathGAN: Visual Scanpath Prediction with Generative Adversarial Networks*. DOI: 10.48550/ARXIV.1809.00567. URL: https://arxiv.org/abs/1809.00567 (see pages 1, 7, 52).

Assens, Marc, Kevin McGuinness, Xavier Giro-i-Nieto, and Noel E. O'Connor (2017). *SaltiNet: Scan-path Prediction on 360 Degree Images using Saliency Volumes*. DOI: 10.48550/ARXIV.1707.03123. URL: https://arxiv.org/abs/1707.03123 (see page 7).

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). *Neural Machine Translation by Jointly Learning to Align and Translate*. DOI: 10.48550/ARXIV.1409.0473. URL: https://arxiv.org/abs/1409.0473 (see pages 3, 17).

Bak, Cagdas, Aysun Kocak, Erkut Erdem, and Aykut Erdem (2017). **Spatio-Temporal Saliency Networks for Dynamic Saliency Prediction**. arXiv: 1607.04730 [cs.CV] (see page 6).

Bays, Paul and Masud Husain (Aug. 2012). **Active inhibition and memory promote exploration and search of natural scenes**. *Journal of vision* 12. DOI: 10.1167/12.8.8 (see page 2).

Bellard, Fabrice (2022). *FFmpeg: A complete, cross-platform solution to record, convert and stream audio and video*. URL: https://ffmpeg.org/ (visited on 05/29/2022) (see page 13).

Boccignone, Giuseppe, Vittorio Cuculo, Alessandro D'Amelio, Giuliano Grossi, and Raffaella Lanzarotti (2020). **On Gaze Deployment to Audio-Visual Cues of Social Interactions**. *IEEE Access* 8, 161630–161654. DOI: 10.1109/ACCESS.2020.3021211 (see pages 1, 9, 23, 49).

Böhme, Martin, Michael Dorr, Christopher Krause, Thomas Martinetz, and Erhardt Barth (2006). **Eye movement predictions on natural videos**. *Neurocomputing* 69:16. Brain Inspired Cognitive Systems, 1996–2004. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2005.11.019. URL: https://www.sciencedirect.com/science/article/pii/S092523120600172X (see pages 1, 6, 7).

Borji, Ali (2018). *Saliency Prediction in the Deep Learning Era: Successes, Limitations, and Future Challenges*. DOI: 10.48550/ARXIV.1810.03716. URL: https://arxiv.org/abs/1810.03716 (see pages 5, 49).

Borji, Ali, Dicky N. Sihite, and Laurent Itti (2013). **Quantitative Analysis of Human-Model Agreement in Visual Saliency Modeling: A Comparative Study**. *IEEE Transactions on Image Processing* 22, 55–69 (see pages 2, 5, 49).

Bylinskii, Zoya, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand (2016a). **What do different evaluation metrics tell us about saliency models?** *arXiv preprint arXiv:1604.03605* (see page 5).

– (2016b). *What do different evaluation metrics tell us about saliency models?* DOI: 10.48550/ARXIV.1604.03605. URL: https://arxiv.org/abs/1604.03605 (see page 22).

Carmi, Ran and Laurent Itti (2006). **Visual causes versus correlates of attentional selection in dynamic scenes**. *Vision Research* 46:26, 4333–4345. ISSN: 0042-6989. DOI: https://doi.org/10.1016/j.visres.2006.08.019. URL: https://www.sciencedirect.com/science/article/pii/S0042698906003816 (see page 1).

Chang, Qinyao and Shiping Zhu (2021). *Temporal-Spatial Feature Pyramid for Video Saliency Detection*. DOI: 10.48550/ARXIV.2105.04213. URL: https://arxiv.org/abs/2105.04213 (see pages 6, 53).

Cho, Kyunghyun, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio (2014). *On the Properties of Neural Machine Translation: Encoder-Decoder Approaches*. DOI: 10.48550/ARXIV.1409.1259. URL: https://arxiv.org/abs/1409.1259 (see pages 16, 49).

Coutrot, Antoine, Janet Hsiao, and Antoni Chan (Apr. 2017). **Scanpath modeling and classification with hidden Markov models**. *Behavior Research Methods* 50, 1–18. DOI: 10.3758/s13428-017-0876-8 (see pages 1, 7).

Cristino, Filipe, Sebastiaan Mathôt, Jan Theeuwes, and Iain Gilchrist (Aug. 2010). **ScanMatch: A novel method for comparing fixation sequences**. *Behavior research methods* 42, 692–700. DOI: 10.3758/BRM.42.3.692 (see page 22).

Dhariwal, Prafulla and Alex Nichol (2021). *Diffusion Models Beat GANs on Image Synthesis*. DOI: 10.48550/ARXIV.2105.05233. URL: https://arxiv.org/abs/2105.05233 (see page 52).

Didolkar, Aniket (2022). *An implementation of Recurrent Independent Mechanisms in PyTorch*. URL: https://github.com/dido1998/Recurrent-Independent-Mechanisms (visited on 06/12/2022) (see page 17).

Dorr, Michael, Thomas Martinetz, Karl Gegenfurtner, and Erhardt Barth (Aug. 2010). **Variability of eye movements when viewing dynamic natural scenes**. *Journal of vision* 10, 28. DOI: 10.1167/10.10.28 (see pages 1, 11, 23, 24, 49, 50, 55).

Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby (2020). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. DOI: 10.48550/ARXIV.2010.11929. URL: https://arxiv.org/abs/2010.11929 (see pages 7, 53).

Engbert, Ralf, Hans Trukenbrod, Simon Barthelmé, and Felix Wichmann (Apr. 2014). **Spatial statistics and attentional dynamics in scene viewing**. *Journal of vision* 15. DOI: 10.1167/15.1.14 (see pages 7, 50).

Fahimi, Ramin and Neil Bruce (Aug. 2020). **On metrics for measuring scanpath similarity**. *Behavior Research Methods* 53. DOI: 10.3758/s13428-020-01441-0 (see page 22).

Falcon et al., William (Mar. 2019). **PyTorch Lightning**. URL: https://www.pytorchlightning.ai (visited on 06/23/2022) (see page 27).

Fan, Haoqi, Tullie Murrell, Heng Wang, Kalyan Vasudev Alwala, Yanghao Li, Yilei Li, Bo Xiong, Nikhila Ravi, Meng Li, Haichuan Yang, Jitendra Malik, Ross Girshick, Matt Feiszli, Aaron Adcock, Wan-Yen Lo, and Christoph Feichtenhofer (2021). **PyTorchVideo: A Deep Learning Library for Video Understanding**. In: *Proceedings of the 29th ACM International Conference on Multimedia*. https://pytorchvideo.org/ (see page 13).

Fang, Yuming, Zhou Wang, Weisi Lin, and Zhijun Fang (Sept. 2014). **Video saliency incorporating spatiotemporal cues and uncertainty weighting**. English. *IEEE Transactions on Image Processing* 23:9, 3910–3921. ISSN: 1057-7149. DOI: 10.1109/TIP.2014.2336549 (see page 6).

Gehring, Jonas, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin (2017). *Convolutional Sequence to Sequence Learning*. DOI: 10.48550/ARXIV.1705.03122. URL: https://arxiv.org/abs/1705.03122 (see page 53).

Goodfellow, Ian J., Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). *Generative Adversarial Networks*. DOI: 10.48550/ARXIV.1406.2661. URL: https://arxiv.org/abs/1406.2661 (see pages 7, 52).

Goyal, Anirudh, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf (2019). *Recurrent Independent Mechanisms*. DOI: 10.48550/ARXIV.1909.10893. URL: https://arxiv.org/abs/1909.10893 (see pages 1, 3, 16, 17, 20, 49, 55).

Guo, Chenlei and Liming Zhang (Feb. 2010). **A Novel Multiresolution Spatiotemporal Saliency Detection Model and Its Applications in Image and Video Compression**. *Image Processing, IEEE Transactions on* 19, 185–198. DOI: 10.1109/TIP.2009.2030969 (see page 6).

Harel, Jonathan, Christof Koch, and Pietro Perona (Jan. 2006). **Graph-Based Visual Saliency**. In: vol. 19, 545–552 (see page 5).

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015). **Deep Residual Learning for Image Recognition**. DOI: 10.48550/ARXIV.1512.03385. URL: https://arxiv.org/abs/1512.03385 (see page 15).

Hochreiter, Sepp and Jürgen Schmidhuber (Dec. 1997). **Long Short-term Memory**. *Neural computation* 9, 1735–80. DOI: 10.1162/neco.1997.9.8.1735 (see pages 16, 49).

Howard, Andrew, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam (2019). *Searching for MobileNetV3*. DOI: 10.48550/ARXIV.1905.02244. URL: https://arxiv.org/abs/1905.02244 (see pages 15, 45).

Huang, Xun, Chengyao Shen, Xavier Boix, and Qi Zhao (Dec. 2015). **SALICON: Reducing the Semantic Gap in Saliency Prediction by Adapting Deep Neural Networks**. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (see pages 3, 5).

Huang, Yifei, Minjie Cai, Zhenqiang Li, and Yoichi Sato (2018). **Predicting Gaze in Egocentric Video by Learning Task-dependent Attention Transition**. DOI: 10.48550/ARXIV.1803.09125. URL: https://arxiv.org/abs/1803.09125 (see pages 3, 8).

Itti, L., C. Koch, and E. Niebur (1998). **A model of saliency-based visual attention for rapid scene analysis**. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20:11, 1254–1259. DOI: 10.1109/34.730558 (see pages 1–3, 5, 7).

Itti, Laurent (Aug. 2005). **Quantifying the contribution of low-level saliency to human eye movements in dynamic scenes**. *Visual Cognition* 12, 1093–1123. DOI: 10.1080/13506280444000661 (see page 6).

Jiang, Ming, Shengsheng Huang, Juanyong Duan, and Qi Zhao (June 2015). **SALICON: Saliency in Context**. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (see page 6).

Judd, Tilke, Frédo Durand, and Antonio Torralba (2012). **A Benchmark of Computational Models of Saliency to Predict Human Fixations**. In: *MIT Technical Report* (see page 5).

Kingma, Diederik P and Max Welling (2013). *Auto-Encoding Variational Bayes*. DOI: 10.48550/ARXIV.1312.6114. URL: https://arxiv.org/abs/1312.6114 (see page 52).

Kingma, Diederik P. and Jimmy Ba (2014). *Adam: A Method for Stochastic Optimization*. DOI: 10.48550/ARXIV.1412.6980. URL: https://arxiv.org/abs/1412.6980 (see page 27).

Koch, Christof and Shimon Ullman (1985). **Shifts in selective visual attention: towards the underlying neural circuitry.** *Human neurobiology* 4 4, 219–27 (see pages 3, 5).

Krauzlis, Richard, Lee Lovejoy, and Alexandre Zenon (May 2013). **Superior Colliculus and Visual Spatial Attention**. *Annual review of neuroscience* 36. DOI: 10.1146/annurev-neuro-062012-170249 (see page 2).

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey Hinton (Jan. 2012). **ImageNet Classification with Deep Convolutional Neural Networks**. *Neural Information Processing Systems* 25. DOI: 10.1145/3065386 (see page 15).

Kruthiventi, Srinivas S. S., Kumar Ayush, and R. Venkatesh Babu (2015). **DeepFix: A Fully Convolutional Neural Network for predicting Human Eye Fixations**. In: arXiv. DOI: 10.48550/ARXIV.1510.02927. URL: https://arxiv.org/abs/1510.02927 (see pages 3, 5).

Kümmerer, Matthias and Matthias Bethge (2021). *State-of-the-Art in Human Scanpath Prediction*. DOI: 10.48550/ARXIV.2102.12239. URL: https://arxiv.org/abs/2102.12239 (see pages 1, 7, 22, 49).

Kümmerer, Matthias, Matthias Bethge, and Thomas S. A. Wallis (Apr. 2022). **DeepGaze III: Modeling free-viewing human scanpaths with deep learning**. *Journal of Vision* 22:5, 7–7. ISSN: 1534-7362. DOI: 10.1167/jov.22.5.7. eprint: https://arvojournals.org/arvo/content_public/journal/jov/938587/i1534-7362-22-5-7_1650885429.74577.pdf. URL: https://doi.org/10.1167/jov.22.5.7 (see pages 1, 3, 7, 42).

Kümmerer, Matthias, Thomas Wallis, and Matthias Bethge (Dec. 2015). **Information-theoretic model comparison unifies saliency metrics**. *Proceedings of the National Academy of Sciences* 112, 201510393. DOI: 10.1073/pnas.1510393112 (see page 23).

Kümmerer, Matthias, Thomas S. A. Wallis, and Matthias Bethge (2016). *DeepGaze II: Reading fixations from deep features trained on object recognition*. DOI: 10.48550/ARXIV.1610.01563. URL: https://arxiv.org/abs/1610.01563 (see pages 3, 5).

– (2018). **Saliency Benchmarking Made Easy: Separating Models, Maps and Metrics**. In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss. Lecture Notes in Computer Science. Springer International Publishing, 798–814 (see pages 5, 6).

Le Meur, Olivier and Zhi Liu (Feb. 2015). **Saccadic model of eye movements for free-viewing condition**. *Vision research* 116. DOI: 10.1016/j.visres.2014.12.026 (see page 23).

LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner (1998). **Gradient-based learning applied to document recognition**. *Proc. IEEE* 86, 2278–2324 (see page 5).

Leigh, R.J. and D.S. Zee (2015). **The Neurology of Eye Movements**. Contemporary neurology series. Oxford University Press. ISBN: 9780199969289. URL: https://books.google.de/books?id=v2s0BwAAQBAJ (see pages 1, 2, 21).

Li, Zhicheng, Shiyin Qin, and Laurent Itti (Jan. 2011). **Visual Attention Guided Bit Allocation in Video Compression**. *Image Vision Comput.* 29:1, 1–14. ISSN: 0262-8856. DOI: 10.1016/j.imavis.2010.07.001. URL: https://doi.org/10.1016/j.imavis.2010.07.001 (see page 11).

Lin, Tsung-Yi, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie (2016). **Feature Pyramid Networks for Object Detection**. *CoRR* abs/1612.03144. arXiv: 1612.03144. URL: http://arxiv.org/abs/1612.03144 (see pages 1, 6, 14, 15, 49, 55).

Linardos, Akis, Matthias Kümmerer, Ori Press, and Matthias Bethge (Oct. 2021). **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 (ICCV)*, 12919–12928 (see page 5).

Linardos, Panagiotis, Eva Mohedano, Juan Jose Nieto, Noel E. O'Connor, Xavier Giro-i-Nieto, and Kevin McGuinness (2019). **Simple vs complex temporal recurrences for video saliency prediction**. arXiv: 1907.01869 [cs.CV] (see page 6).

Liu, Ze, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu (2021). *Video Swin Transformer*. DOI: 10.48550/ARXIV.2106.13230. URL: https://arxiv.org/abs/2106.13230 (see pages 6, 7, 53).

Ma, Cheng, Haowen Sun, Yongming Rao, Jie Zhou, and Jiwen Lu (2022). **Video Saliency Forecasting Transformer**. *IEEE Transactions on Circuits and Systems for Video Technology*, 1–1. DOI: 10.1109/TCSVT.2022.3172971 (see pages 6, 7, 53).

Mathe, Stefan and Cristian Sminchisescu (2015). **Actions in the Eye: Dynamic Gaze Datasets and Learnt Saliency Models for Visual Recognition**. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37:7, 1408–1424. DOI: 10.1109/TPAMI.2014.2366154 (see page 11).

Mclachlan, G. and David Peel (Jan. 2000). **Finite Mixture Model**. *Finite Mixture Models* 44. DOI: 10.1002/0471721182 (see page 52).

Mital, Parag, Tim Smith, Robin Hill, and John Henderson (Mar. 2011). **Clustering of Gaze During Dynamic Scene Viewing is Predicted by Motion**. *Cognitive Computation* 3, 5–24. DOI: 10.1007/s12559-010-9074-z (see page 11).

Parkhurst, Derrick, Klinton Law, and Ernst Niebur (Feb. 2002). **Parkhurst D, Law K, Niebur E. Modeling the role of salience in the allocation of overt visual attention. Vision Res 42: 107-123**. *Vision research* 42, 107–23. DOI: 10.1016/S0042-6989(01)00250-4 (see pages 2, 50).

Parkhurst, Derrick J. and Ernst Niebur (2003). **Scene content selected by active vision.** *Spatial vision* 16 2, 125–54 (see pages 2, 50).

Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library." In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf (see pages 13, 26).

Peters, Robert, Asha Iyer, Laurent Itti, and Christof Koch (Aug. 2005). **Components of bottom-up gaze allocation in natural images**. *Vision research* 45, 2397–416. DOI: 10.1016/j.visres.2005.03.019 (see pages 23, 24, 50, 55).

Posner, Michael and Yoav Cohen (Jan. 1984). **Components of visual orienting**. *Attention and performance X: Control of language processes* 32, 531– (see page 2).

Privitera, C.M. and L.W. Stark (2000). **Algorithms for defining visual regions-of-interest: comparison with eye fixations**. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22:9, 970–982. DOI: 10.1109/34.877520 (see page 22).

Purves, D., D. Fitzpatrick, L.C. Katz, A.S. Lamantia, J.O. McNamara, S.M. Williams, and G.J. Augustine (2000). **Neuroscience**. Sinauer Associates. ISBN: 9780878937431. URL: https://books.google.de/books?id=F4pTPwAACAAJ (see pages 2, 50).

Pytorch (2022). *Pytorch documentation: Models and pre-trained weights*. URL: https://pytorch.org/vision/stable/models.html (visited on 05/29/2022) (see page 13).

Rabiner, L. and B. Juang (1986). **An introduction to hidden Markov models**. *IEEE ASSP Magazine* 3:1, 4–16. DOI: 10.1109/MASSP.1986.1165342 (see page 7).

Rajashekar, Umesh, Lawrence Cormack, and Alan Bovik (Feb. 2004). **Point of Gaze Analysis Reveals Visual Search Strategies**. *Proceedings of SPIE - The International Society for Optical Engineering* 5292. DOI: 10.1117/12.537118 (see page 23).

Ramesh, Aditya, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever (2021). *Zero-Shot Text-to-Image Generation*. DOI: 10.48550/ARXIV.2102.12092. URL: https://arxiv.org/abs/2102.12092 (see page 52).

Ratcliff, Roger and Gail McKoon (2008). **The Diffusion Decision Model: Theory and Data for Two-Choice Decision Tasks**. *Neural Computation* 20, 873–922 (see page 9).

Roth, Nicolas, Pia Bideau, Olaf Hellwich, Martin Rolfs, and Klaus Obermayer (2021). *A modular framework for object-based saccadic decisions in dynamic scenes*. arXiv: 2106.06073 [cs.CV] (see pages 1–3, 9, 49, 50).

Russell, Alexander, Stefan Mihalas, Rudiger von der Heydt, Ernst Niebur, and Ralph Etienne-Cummings (Oct. 2013). **A model of proto-object based saliency**. *Vision research* 94. DOI: 10.1016/j.visres.2013.10.005 (see page 3).

Santoro, Adam, Ryan Faulkner, David Raposo, Jack Rae, Mike Chrzanowski, Theophane Weber, Daan Wierstra, Oriol Vinyals, Razvan Pascanu, and Timothy Lillicrap (2018). *Relational recurrent neural networks*. DOI: 10.48550/ARXIV.1806.01822. URL: https://arxiv.org/abs/1806.01822 (see page 20).

Schütt, Heiko H., Lars Rothkegel, Hans A. Trukenbrod, Sebastian Reich, Felix A. Wichmann, and Ralf Engbert (2016). *Likelihood-based Parameter Estimation and Comparison of Dynamical Cognitive Models*. DOI: 10.48550/ARXIV.1606.07309. URL: https://arxiv.org/abs/1606.07309 (see pages 7, 50).

Schwetlick, Lisa, Daniel Backhaus, and Ralf Engbert (2021). **A dynamical scan path model for task-dependence during scene viewing**. DOI: 10.48550/ARXIV.2112.11067. URL: https://arxiv.org/abs/2112.11067 (see pages 3, 7).

Shi, Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo (2015). *Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting*. DOI: 10.48550/ARXIV.1506.04214. URL: https://arxiv.org/abs/1506.04214 (see pages 6, 53).

Simonyan, Karen and Andrew Zisserman (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. DOI: 10.48550/ARXIV.1409.1556. URL: https://arxiv.org/abs/1409.1556 (see pages 5, 15, 45).

Sohl-Dickstein, Jascha, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli (2015). *Deep Unsupervised Learning using Nonequilibrium Thermodynamics*. DOI: 10.48550/ARXIV.1503.03585. URL: https://arxiv.org/abs/1503.03585 (see page 52).

SR Research Ltd. (2022). *EyeLink II*. URL: https://www.sr-research.com/eyelink-ii/ (visited on 06/23/2022) (see page 11).

Tan, Mingxing and Quoc V. Le (2019). **EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks**. DOI: 10.48550/ARXIV.1905.11946. URL: https://arxiv.org/abs/1905.11946 (see pages 15, 45).

Tatler, Benjamin, Roland Baddeley, and Iain Gilchrist (Apr. 2005). **Visual correlates of fixation selection: Effects of scale and time**. *Vision research* 45, 643–59. DOI: 10.1016/j.visres.2004.09.017 (see pages 23, 50).

Tatler, Benjamin W. (Nov. 2007). **The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions**. *Journal of Vision* 7:14, 4–4. ISSN: 1534-7362. DOI: 10.1167/7.14.4. eprint: https://arvojournals.org/arvo/content_public/journal/jov/932846/jov-7-14-4.pdf. URL: https://doi.org/10.1167/7.14.4 (see page 2).

Tavakoli, Hamed Rezazadegan, Esa Rahtu, Juho Kannala, and Ali Borji (Jan. 2019). **Digging Deeper Into Egocentric Gaze Prediction**. In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. DOI: 10.1109/wacv.2019.00035. URL: https://doi.org/10.1109%2Fwacv.2019.00035 (see page 8).

Tran, Du, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri (2017). *A Closer Look at Spatiotemporal Convolutions for Action Recognition*. DOI: 10.48550/ARXIV.1711.11248. URL: https://arxiv.org/abs/1711.11248 (see page 53).

Treisman, Anne M. and Garry Gelade (1980). **A feature-integration theory of attention**. *Cognitive Psychology* 12:1, 97–136. ISSN: 0010-0285. DOI: https://doi.org/10.1016/0010-0285(80)90005-5. URL: https://www.sciencedirect.com/science/article/pii/0010028580900055 (see page 2).

Tseng, Po-He, Ran Carmi, Ian Cameron, Douglas Munoz, and Laurent Itti (July 2009). **Quantifying center bias of observers in free viewing of dynamic natural scenes**. *Journal of vision* 9, 4. DOI: 10.1167/9.7.4 (see pages 2, 50).

Van Rossum, Guido and Fred L. Drake (2009). **Python 3 Reference Manual**. Scotts Valley, CA: CreateSpace. ISBN: 1441412697 (see pages 13, 26).

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). *Attention Is All You Need*. DOI: 10.48550/ARXIV.1706.03762. URL: https://arxiv.org/abs/1706.03762 (see pages 3, 7, 17, 53).

Vig, Eleonora, Michael Dorr, and David Cox (June 2014). **Large-Scale Optimization of Hierarchical Features for Saliency Prediction in Natural Images**. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2798–2805. DOI: 10.1109/CVPR.2014.358 (see page 5).

Wang, Wenguan, Jianbing Shen, Fang Guo, Ming-Ming Cheng, and Ali Borji (2018a). **Revisiting Video Saliency: A Large-scale Benchmark and a New Model**. DOI: 10.48550/ARXIV.1801.07424. URL: https://arxiv.org/abs/1801.07424 (see pages 3, 6).

– (2018b). **Revisiting Video Saliency: A Large-scale Benchmark and a New Model**. DOI: 10.48550/ARXIV.1801.07424. URL: https://arxiv.org/abs/1801.07424 (see page 11).

Wang, Wenguan, Jianbing Shen, Jianwen Xie, Ming-Ming Cheng, Haibing Ling, and Ali Borji (2019). **Revisiting Video Saliency Prediction in the Deep Learning Era**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–1. DOI: 10.1109/TPAMI.2019.2924417 (see page 6).

Wang, Ziqiang, Zhi Liu, Gongyang Li, Yang Wang, Tianhong Zhang, Lihua Xu, and Jijun Wang (2021). **Spatio-Temporal Self-Attention Network for Video Saliency Prediction**. *IEEE Transactions on Multimedia*, 1–1. DOI: 10.1109/tmm.2021.3139743. URL: https://doi.org/10.1109%2Ftmm.2021.3139743 (see page 6).

Williams, Ronald J. and David Zipser (1989). **A Learning Algorithm for Continually Running Fully Recurrent Neural Networks**. *Neural Computation* 1:2, 270–280. DOI: 10.1162/neco.1989.1.2.270 (see pages 4, 18, 49).

Wilming, Niklas, Torsten Betz, and TC Kietzmann (Jan. 2011). **Measures and Limits of Models of Fixation Selection**. *PloS one* 6. DOI: 10.1371/Citation (see page 23).

Xie, Saining, Chen Sun, Jonathan Huang, Z. Tu, and Kevin Murphy (Dec. 2017). **Rethinking Spatiotemporal Feature Learning For Video Understanding** (see page 53).

Yamada, Kentaro, Yusuke Sugano, Takahiro Okabe, Yoichi Sato, Akihiro Sugimoto, and Kazuo Hiraki (Nov. 2011). **Attention Prediction in Egocentric Video Using Motion**

and Visual Saliency. In: vol. 7087, 277–288. DOI: 10.1007/978-3-642-25367-6_25 (see page 8).
Yarbus, A. L. (1967). **Eye Movements and Vision**. Plenum. New York. (see pages 1, 2).
Zhang, Lingyun, Matthew H. Tong, Tim K. Marks, Honghao Shan, and Garrison W. Cottrell (Dec. 2008). **SUN: A Bayesian framework for saliency using natural statistics**. *Journal of Vision* 8:7, 32–32. ISSN: 1534-7362. DOI: 10.1167/8.7.32. eprint: https://arvojournals.org/arvo/content_public/journal/jov/933536/jov-8-7-32.pdf. URL: https://doi.org/10.1167/8.7.32 (see pages 5, 23).

Appendix: Examination of RIM hyperparameters

Figure 6.1: Train loss / NSS scores for different number k_T of RIM units; trained on the partition *all videos / single observer* with $d_h = 400$, $k_A = 4$

Figure 6.2: Train loss / NSS scores for different number of active k_A of RIM units; trained on the partition *all videos / single observer* with $d_h = 400$, $k_T = 6$

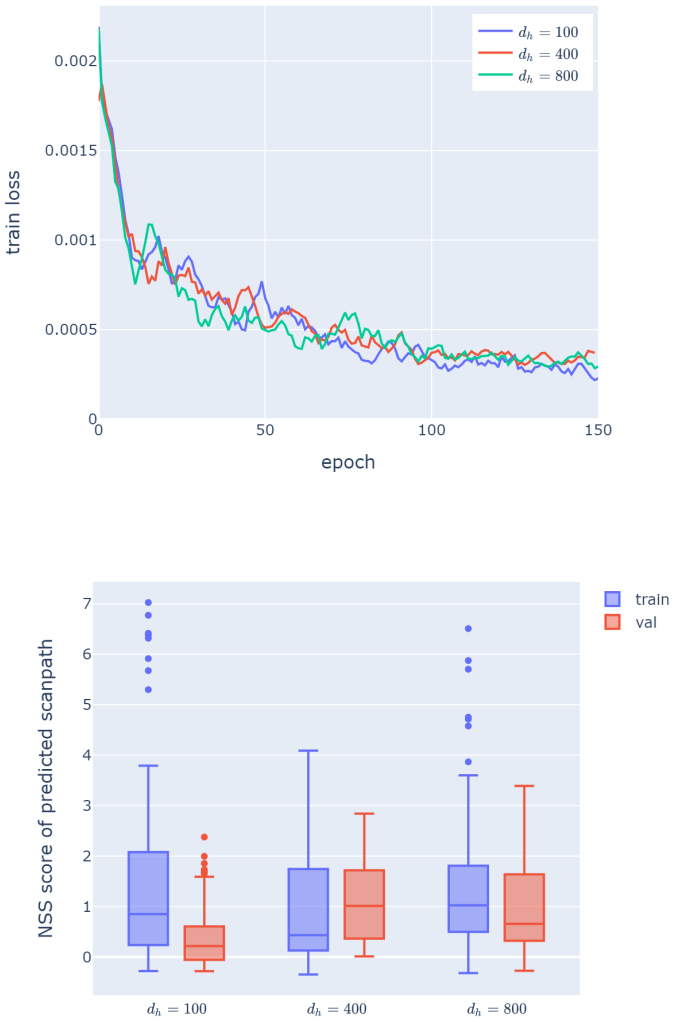

Figure 6.3: Train loss / NSS scores for different hidden size d_h in RIM units; trained on the partition *all videos / single observer* with $d_h = 400$, $k_T = 6$, $k_A = 4$