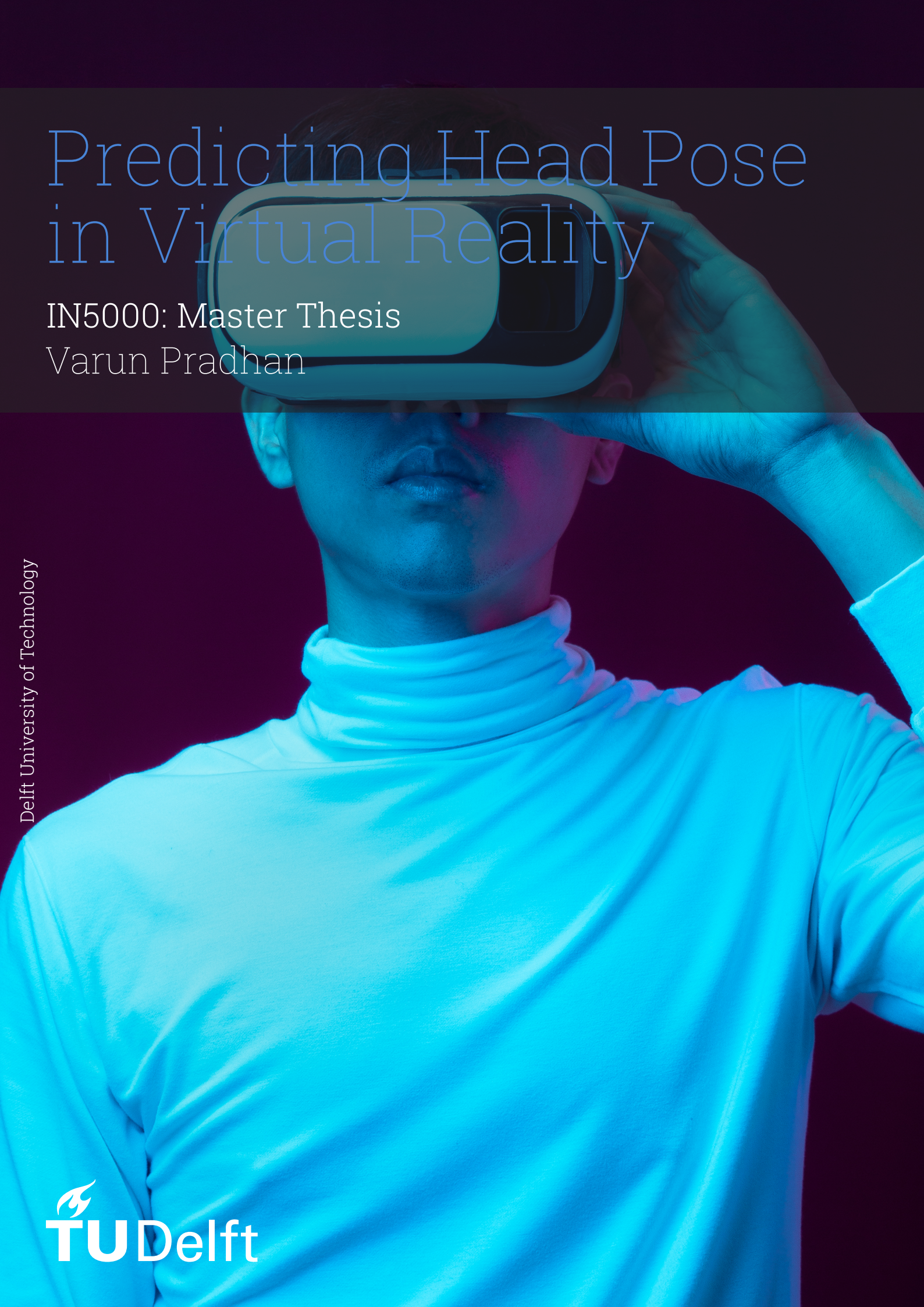
# Predicting Head Pose in Virtual Reality

IN5000: Master Thesis

Varun Pradhan



Delft University of Technology

TU Delft

# Predicting Head Pose in Virtual Reality

by

Varun Pradhan

| Student Name | Student Number |
|---|---|
| V. Pradhan | 5755263 |

| | |
|---|---|
| Professor: | Prof. Dr. Pablo Cesar |
| Daily Supervisor: | Dr. Silvia Rossi |
| Project Duration: | April, 2024 - December, 2024 |
| Faculty: | Faculty of Computer Science, Delft |

| | | |
|---|---|---|
| Thesis Committee: | Prof. Dr. Pablo Cesar | TU Delft, CWI, supervisor |
| | Dr. Petr Kellnhofer | TU Delft |
| | Dr. Silvia Rossi | CWI, external expert |

Cover: Goggles Stock photo, Vecteezy (https://shorturl.at/xOJgK)
An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

CWI
Centrum Wiskunde & Informatica

# Preface

With this thesis, I am completing the final stage of my Master of Science degree in Computer Science at TU Delft. This project marks the culmination of an exciting journey through the master's program. I am grateful to have had the opportunity to carry out this research at a national research institute, where I was able to delve into the challenges of head pose prediction for virtual reality.

I conducted my thesis work at Centrum Wiskunde & Informatica in Amsterdam, under the supervision of Silvia Rossi and Pablo Cesar. I would like to express my sincere gratitude to them for their guidance, insightful feedback, and support throughout this project. Their expertise and encouragement were instrumental in shaping the direction and outcomes of my research.

I hope that this work contributes meaningfully to our understanding of human behavior in virtual reality, and the ongoing efforts at the Distributed & Interactive Systems Group. I also want to extend my appreciation to the team for creating a welcoming and collaborative environment, making my time there both productive and enjoyable.

*Varun Pradhan*
*Delft, December 2024*

# Summary

In recent years, immersive media, particularly Virtual Reality (VR) technology, has seen significant growth. VR technology immerses users in fully digital environments, offering interactive experiences beyond traditional media. However, delivering high-quality 360° videos over the internet poses challenges such as bandwidth constraints and latency issues. Latency, in particular, can disrupt the sense of presence by causing unrealistic interactions that break the illusion of being in a virtual environment. One promising solution is the prediction of a user's head pose trajectories to preemptively adapt the content delivery and minimize delays.

Head pose prediction enables adaptive streaming systems to prioritize and deliver only the relevant portions of 360° videos, significantly reducing bandwidth requirements while ensuring a smooth user experience. Despite advances in predictive modeling, existing approaches often struggle with accuracy when user behavior is unpredictable, influenced by content characteristics and individual differences. To address these challenges, this thesis investigates the potential of leveraging entropy metrics, such as Actual Entropy (AE) and Instantaneous Entropy (IE), as measures of user predictability to improve head pose prediction.

Through an exploratory analysis of 360° video datasets and existing state-of-the-art prediction models, we identify a linear correlation between prediction errors and entropy metrics, highlighting the potential of entropy-driven approaches. We develop two adaptive attention-based models: an LSTM-based model with entropy-modulated attention and a multi-head adaptive attention model. In addition, we explore entropy-augmented baseline approaches. While adaptive models achieve mixed results, a baseline model combining head pose and instantaneous entropy was found to be more stable, demonstrating the utility of even straightforward entropy integration.

Although the entropy-based models did not consistently outperform state-of-the-art methods, our findings demonstrate that entropy augmentation offers a promising avenue for improving the stability and robustness of head pose prediction in specific scenarios. This thesis highlights that understanding dataset characteristics and how entropy is incorporated into model architectures is crucial for optimizing performance. These insights suggest that future work should focus on adapting model designs to better account for user predictability, which could lead to more adaptive and responsive VR systems.

# Contents

**Preface** i

**Summary** ii

**1 Introduction** 1
- 1.1 Research Domain . . . 1
- 1.2 Research Questions . . . 5
- 1.3 Outline of the Thesis . . . 5
- 1.4 Main Contributions . . . 7

**2 Background and Related Work** 8
- 2.1 360° videos and Virtual Reality . . . 8
- 2.2 Streaming of VR content . . . 9
- 2.3 Behavioral Analysis . . . 12
- 2.4 Head Pose Prediction . . . 13
  - 2.4.1 Deep Learning for Head Pose Trajectory Forecasting . . . 14
  - 2.4.2 Existing Deep Learning Models for Head Pose Prediction . . . 15
  - 2.4.3 Attention for Head Pose Prediction . . . 17
- 2.5 Datasets . . . 19

**3 Methodology** 22
- 3.1 Key Terms . . . 22
- 3.2 Key Definitions and Problem Formulation . . . 23
- 3.3 Content and User Metrics . . . 25
- 3.4 Methodology . . . 27
  - 3.4.1 Exploratory Analysis of Datasets and State-of-the-Art-Models . . . 28
  - 3.4.2 Integrating Findings into New Head Pose Prediction Models . . . 30
  - 3.4.3 Evaluation of Entropy-Based Models . . . 30

**4 Exploratory Analysis of Datasets and State-of-the-Art-Models** 31
- 4.1 Dataset Analysis . . . 31
  - 4.1.1 Train-test split . . . 34
- 4.2 Comparing Existing models . . . 36

**5 Adaptive Attention for Head Pose Prediction** 40
- 5.1 Entropy-based Adaptive Attention LSTM (E-AALSTM) . . . 40
  - 5.1.1 LSTM Layer . . . 41
  - 5.1.2 Adaptive Attention Layer: . . . 41
- 5.2 Multi-head Adaptive Attention (AMH) . . . 43
  - 5.2.1 Input Embedding . . . 43
  - 5.2.2 Positional Embedding . . . 43
  - 5.2.3 Multi-Head Adaptive Attention Layer . . . 44

5.2.4 Position-wise Feed-Forward Network . . . . . 45

**6 Evaluation and Results** **46**
6.1 Experimental Setup . . . . . 46
6.1.1 Entropy Enhanced Baselines . . . . . 46
6.1.2 Benchmarks and Comparison Models . . . . . 47
6.1.3 Hyperparameter Settings . . . . . 48
6.1.4 Train-test Split . . . . . 48
6.1.5 Evaluation Metric . . . . . 48
6.2 Results . . . . . 49
6.2.1 Model Evaluation: Entropy vs No-Entropy Models . . . . . 49
6.2.2 Comparison to the State-of-The-Art . . . . . 55

**7 Discussion and Conclusion** **60**
7.1 Use of Entropy Metrics in Prediction . . . . . 60
7.2 Limitations . . . . . 61
7.3 Future Work . . . . . 62
7.4 Conclusion . . . . . 63

**References** **65**

# List of Figures


1.1 Navigation of user in immersive content: a) 3-DoF head movements; b) navigation system on viewing sphere at an arbitrary time instant; c) navigation trajectory over time. [57] . . . . . 2

2.1 360° video streaming pipeline (A simplified version of the pipeline presented by Rossi et al. [57]). . . . . 9
2.2 (a) Viewport and head pose in case of low motion-to-photon latency, (b) Viewport and head pose in case of high motion-to-photon latency (expected viewport in red). Inspired by [74] . . . . . 11
2.3 360° video streaming pipeline with head pose prediction [57]. . . . . 12
2.4 Standard seq2seq encoder-decoder architecture. Inspired by [56, 69]. . 14
2.5 VPT360 architecture [9] (a) and scaled dot-product attention [72] (b) . . 18

3.1 Azimuth angle and elevation for describing head Pose[15] . . . . . 24
3.2 Head pose prediction: Head pose from time-stamp t to t+H is predicted using head pose and video data from time-stamp t-M to t . . . . . 24
3.3 SI/TI processing pipeline [32] . . . . . 25
3.4 Methodology . . . . . 28

4.1 Mean Temporal Information vs Mean Spatial Information plots with a heatmap for mean SE for videos in (a) NOSSDAV17, (b) MM22, (c) PAMI18, (d) MMSys18 . . . . . 32
4.2 Mean SE vs Mean AE for videos in (a) NOSSDAV17, (b) MM22, (c) PAMI18, (d) MMSys18, (e) All four datasets . . . . . 33
4.3 Representative scenes from *VRBasketball* (a) and *Mario* (c). Along with the average X coordinate of user trajectories over the course of *VRBasketball* (b) and *Mario* (d). . . . . 34
4.4 Distribution of training and testing videos in (a) NOSSDAV17, (b) MM22, (c) PAMI18, and (d) MMSys18 . . . . . 35
4.5 Performance of position only baseline, TRACK, DVMS (with K=2 and K=5) and VPT360 on (a) NOSSDAV17, (b) MM22, (c) PAMI18, and (d) MMSys18 datasets . . . . . 36
4.6 (a) Performance of DVMS K=5 on each video in the MM22 dataset, and (b) Mean SE and AE of test videos in MM22 dataset, the size of the marker represents the mean IE . . . . . 37
4.7 Performance of DVMS K=5 [24] for a prediction horizon of 5 seconds for *Skydive*, *Chariot* and *Mario* compared with the entropy of saliency maps (a,c,e) and the mean instantaneous entropy of users (b,d,f) . . . . . 38

5.1 Seq2Seq architecture with attention layer . . . . . 41
5.2 Entropy-adaptive scaled dot-product attention . . . . . 42

5.3 Architecture of proposed AMH model . . . . . 44

6.1 Performance comparison between E-AALSTM and ALSTM on (a) NOSSDAV17, (b) MM22, (c) PAMI18, and (d) MMSys18 datasets. . . . . 49
6.2 Performance comparison between AMH and VPT360 on (a) NOSSDAV17, (b) MM22, (c) PAMI18, and (d) MMSys18 datasets. . . . . 50
6.3 Performance comparison between pos-only, pos-augmented, and pos-weighted on (a) NOSSDAV17, (b) MM22, (c) PAMI18, and (d) MMSys18 datasets. . . . . 51
6.4 Average prediction loss at each timestep for pos-only, and pos-augmented on (a) NOSSDAV17, (b) MM22, and (c) MMSys18 datasets, based on the predicted head pose 5 seconds after the input window. . . . . 52
6.5 Violin plots of Orthodromic distance for pos-only and pos-augmented on (a) NOSSDAV17, (b) MM22, (c) PAMI18, and (d) MMSys18 datasets 53
6.6 Performance of pos-augmented, E-AALSTM, and AMH on (a) NOSSDAV17, (b) MM22, (c) PAMI18, and (d) MMSys18 datasets . . . . . 54
6.7 Performance of pos-only, VPT360, TRACK, DVMS-5, and pos-augmented (a) NOSSDAV17, (b) MM22, (c) PAMI18, and (d) MMSys18 datasets . . 56
6.8 Violin plots of Orthodromic distance for pos-only and pos-augmented on (a) NOSSDAV17, (b) MM22, (c) PAMI18, and (d) MMSys18 datasets 57
6.9 Orthodromic loss for the pos_augmented and pos_only models plotted along with the average instantaneous entropy of viewers across three videos from the MM22 dataset with varying levels of Actual Entropy (High AE: video_27, Medium AE: video_23, Low AE: video_13). . . . . 58

# List of Tables

2.1 Features of the datasets discussed. . . . . 20

3.1 Definitions of the variables used in the thesis. . . . . 22
3.2 Notable Head Pose Prediction Models . . . . . 29

4.1 AE and SE thresholds for different datasets. . . . . 36
4.2 Correlation and p-values for IE and SE with loss for different videos. . 39

6.1 Average Loss at 2.6 Seconds and 5 Seconds for Each Model and Dataset 57

# 1

# Introduction

## 1.1. Research Domain

Over the past decade, there has been a surge in interest in immersive media that provides a more realistic experience than traditional media. Virtual Reality (VR) stands out as a prime example of immersive technology. It is a rapidly growing market projected to grow from an estimated $35.7 Billion in 2023 to $111.8 Billion by 2030 [22]. The main novelty of VR technology is that it **immerses** users into a fully digital environment, replacing the real world. This allows individuals to experience a completely new reality that they can interact with, instead of just viewing content on a screen [57]. By doing so, VR technology makes the viewer feel like they are actually in this virtual environment instead of their real, physical environment. This feeling is referred to as **presence** [65]. **Interactivity** is the ability for an individual to change their virtual environment with their movements [62]. VR enhances the feelings of presence and immersion by allowing users to interact with the virtual world through their actions [43], much like they would in a real environment. A key challenge with VR technology is ensuring a sufficient level of immersion, presence, and interactivity [51, 43]. These three factors are crucial to guarantee a high level of user satisfaction when using a VR system [43]. In particular, unrealistic interactions with the environment can break the illusion of “being there" and can cause discomfort [51]. To ensure a sufficient level of immersion, presence, and interactivity, novel types of content are required.

To provide an immersive and interactive experience, 360-degree videos have emerged as a novel media format. Unlike traditional videos, 360° videos shift the user’s role from a passive viewer of traditional videos to an interactive and active viewer of VR content [12]. When viewing 360° videos, users are provided with a VR device, called a Head-Mounted Display (HMD), which allows them to look around and navigate the virtual environment by rotating their head. Specifically, participants can interact with the virtual environment over 3-Degrees of Freedom (DoF) by looking up or down (pitch), left or right (yaw), and by tilting their head from side to side (roll)[57] as shown in Figure 1.1 (a). This interaction is enabled by virtually placing the viewer at the center of the virtual space in a 360° video, called the viewing sphere, as shown in Figure 1.1 (b). The viewing sphere represents the entire virtual environment surrounding

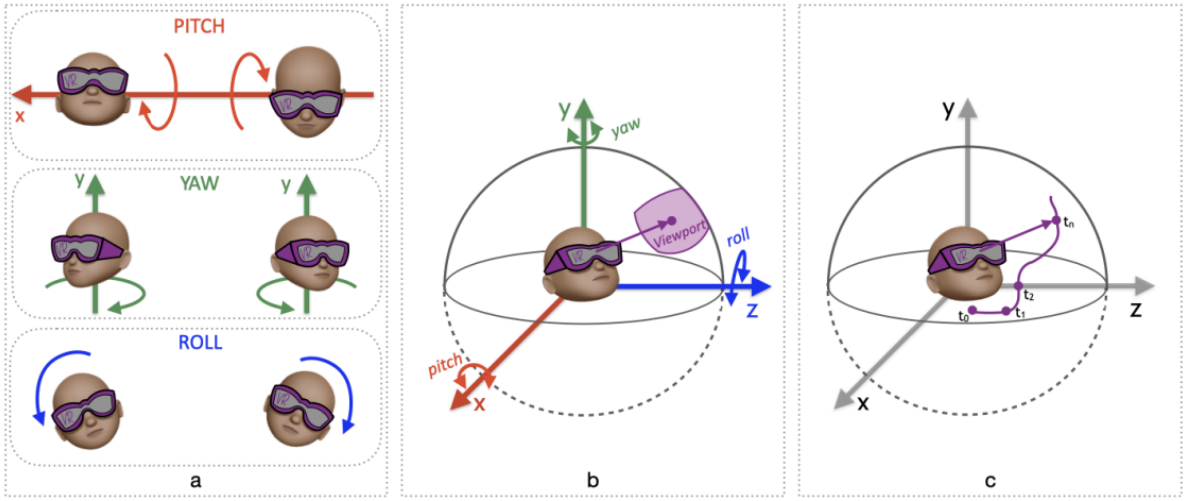



**Figure 1.1:** Navigation of user in immersive content: a) 3-DoF head movements; b) navigation system on viewing sphere at an arbitrary time instant; c) navigation trajectory over time. [57]

the user. The HMD mimics the viewer's Field of View (FoV) by only displaying a portion of the virtual environment called the viewport, as shown in Figure 1.1 (b). The displayed viewport is selected based on the direction and orientation of the user's head within the virtual space, known as their head pose. The sequence of the user's head pose, and therefore the viewing direction, over time can be approximated by the centre of the viewport projected on the viewing sphere (Figure 1.1 (c)). This sequence is called the user's navigation trajectory [57, 56].

In order to consume this type of immersive 360° content, it must be transmitted and streamed efficiently. To guarantee an immersive experience while streaming 360° videos, it is crucial to ensure a high Quality of Experience (QoE), which depends on a few key factors [51]. Two main factors for ensuring a high QoE are a high resolution and a low latency [57]. The resolution requirement is exacerbated by the close proximity of the screen to the eyes in VR [56, 28, 55]. A high switching delay when the user's viewport changes can result in discomfort along with a degraded experience. Ideally, a high QoE can be ensured by sending the entire content at a high quality to the user, and exporting the desired viewport during rendering. Initial advances in streaming 360° videos aimed at improving overall system performance in terms of bandwidth, storage cost, and networking reliability when sending the entire 360° video content [57]. Unfortunately, these solutions still suffered from high bandwidth requirements due to the high resolution of 360° videos (8K resolution) [48]. These high-resolution videos had data rates up to two orders of magnitude higher than those of regular videos [48].

Since the user only views a small portion (roughly 15% [26]) of the 360° video content at a given time, some researchers have focused on more personalized systems by processing the streaming content based on the viewer's head pose [74]. This is known as adaptive streaming [57]. These systems include segmenting the video spatially into tiles and dynamically adjusting the quality of the tiles based on their proximity to the viewer's viewport [53, 76, 20]. Another approach involves using projections of the spherical content, prioritizing the highest quality for the most important areas

[19, 27]. Additionally, adaptive streaming systems also take into account network conditions, adjusting the quality of the video tiles depending on factors such as available bandwidth and latency [57]. The main goal of adaptive video streaming systems is to optimize viewer QoE while overcoming streaming limitations of 360° videos. Through adaptive streaming, we can significantly reduce the amount of data being streamed, thereby reducing the bandwidth requirements [57]. However, due to the time taken to transmit user viewport data to the server and then transmit the content information based on this viewport back to the client, there can be a mismatch between the actual head pose of the user and the pose for which the frame is rendered [82]. To overcome this limitation, Qian et al. developed the first user-centric design that predicts future user viewports in advance [53]. By predicting the user's head pose trajectory, adaptive streaming can tailor content information based on the user viewport in the near future instead of the current user viewport. Using three simple logistic regression models with a moving window of 1 second, they successfully anticipated viewer behavior in a short-term window and showed that giving higher fetching priority to the tiles most likely to be displayed could reduce bandwidth usage by up to 80% [57, 53].

Even though Qian et al.'s user-centric design demonstrated the feasibility of overcoming viewport mismatches by predicting future user head poses in advance [53], it only focused on single-user trajectories and struggled to provide accurate predictions for windows longer than one second. To build on this, researchers have tried to better understand how users interact with 360° content. Recent behavioral studies have shown that users tend to navigate 360° videos in similar ways [60, 47, 35]. For example, Rossi et al. [60] showed that viewers have highly similar trajectories when viewing content with a dominant focus of attention. These findings have been the motivation for the development of cross-user models. Content-agnostic cross-user models rely solely on user trajectories to predict a user's head pose. Cross-user models predominantly use LSTM-based architectures, such as bidirectional LSTMs [23] and data-fusion LSTMs [28], to forecast user movements. However, being content-agnostic, they only utilize user trajectories and do not incorporate additional information such as content features or individual differences in user behavior, resulting in degraded predictions for windows larger than ~4 seconds [56].

Behavioral studies have also shown that the type of content also affects user trajectories [60, 47, 35]. Ozcinar et al. showcased that there is a high correlation between user fixation and video complexity measured through **Spatial Information (SI) and Temporal Information (TI)** [32]. As a result, various content-aware designs that augmented user trajectories with some form of content information (usually in the form of saliency maps) have also been proposed [56, 17, 50]. To analyze various content-aware prediction models, Rondón et al. proposed a "Unified Evaluation Framework" [55], which homogenizes datasets and provides common metrics for evaluation. Through their analysis, they found that the main challenge with incorporating content information is that an effective design should attenuate the impact of content information in the short term and give it more importance in the long term. Following these findings, the current state-of-the-art approaches attenuate content information by passing it through dedicated recurrent units before merging it with head pose data [56] and

by incorporating additional content information such as actor information in social videos [1], motion maps [50] and the flow of objects over the course of the video [49]. However, these models do not incorporate any content information that may affect the predictability of user movements. Sitzmann et al. [64] demonstrated that viewers tend to explore the virtual environment more slowly when viewing content that lacks a dominant focus of attention. They discussed how **the entropy of saliency maps**, a measure of the distribution of salient regions in a given scene, can be used to quantify how the predictability of user trajectories may be affected by the content being viewed. This unpredictability usually stems from two key factors: the lack of a dominant focus of attention in the displayed 360° content [64, 60] or variations in user disposition [59]. Notably, the entropy of saliency maps is an aggregate measure, as the saliency maps are derived from the trajectories of all users viewing a given video, reflecting how the overall content affects user behavior predictability. The findings of Rossi et al. also found a consistent correlation between the **actual entropy** of user trajectories and their prediction error [59].

In this context, the main focus of this thesis is advancing head pose prediction models. Rossi et al. primarily use actual entropy to assess the unpredictability of user trajectories in 360° video environments [59], but the inclusion of additional entropy metrics, such as instantaneous entropy and the entropy of saliency maps, presents a compelling opportunity to enhance predictive models. While actual entropy measures the degree of predictability of a viewer's entire trajectory throughout a session. Instantaneous entropy provides a real-time measure of trajectory predictability, which could be used to enable models to adjust predictions based on current user behaviors, resulting in a more responsive model. The entropy of saliency maps, on the other hand, quantifies the distribution of salient regions in a given scene and provides more information for content-aware designs, possibly enhancing predictive accuracy.

Despite significant research in head pose prediction, user metrics that summarize or analyze individual user behavior, such as entropy, have not been extensively incorporated. Jin et al. found that, while viewers tend to focus on more salient regions in the content, there are notable individual differences in user behavior patterns [35]. There are existing meta-learning approaches that factor in these differences in user disposition [39, 42], but there have not been any deep learning approaches that explicitly use features that describe user disposition. This gap presents an opportunity for enhancing head pose prediction models by explicitly factoring in individual user behavior, creating more robust and accurate systems.

This thesis seeks to fill this gap by exploring the use of **entropy of user trajectories** to capture and incorporate unpredictability into head pose prediction models. One approach involves using adaptive attention mechanisms based on instantaneous entropy to dynamically adjust predictions, while another leverages a straightforward strategy that augments head pose data with entropy information to improve stability and accuracy. Our goal is to develop more accurate and responsive models, which will ultimately enhance the user experience in 360° video content.

## 1.2. Research Questions

With the goal to **accurately predict head pose trajectories**, we aim to answer the following questions.

Q1. Can aggregated user information, such as the entropy of saliency maps, provide insights into the predictability of head pose trajectories?

Q2. How do we account for user predictability in a head pose prediction model? Is incorporating the entropy of user trajectories into prediction models a valuable addition?

By addressing these questions, we aim to advance our understanding of head pose prediction and develop a more accurate and responsive prediction model using deep learning techniques. Effectively analyzing the impact of entropy of saliency maps (Q1) will help enhance our understanding of head pose predictions, while integrating individual behavioral characteristics like user entropy (Q2) will allow for more robust and responsive models.

## 1.3. Outline of the Thesis

This thesis is structured as follows:

In the **Background and Related Works** chapter we begin by outlining the VR streaming pipeline, emphasizing the need for accurate head pose prediction due to the importance of minimizing motion-to-photon (MTP) latency for an immersive experience. Following this, we review existing behavioral analyses of users in VR, focusing on research that examines head pose trajectories as a means of understanding user behavior and improving prediction accuracy. We then continue with an overview of head pose prediction techniques, particularly deep learning approaches, and explore the potential of attention-based models for enhancing prediction performance. Finally, the chapter discusses the datasets commonly used for head pose prediction tasks, highlighting the challenges involved in creating a heterogeneous dataset.

In the **Methodology** chapter, we summarize all the key terms used in this thesis, define key concepts related to the head pose prediction task, and outline the formulae used for various content and user metrics. We then provide a detailed overview of the methodology, which includes the preliminary data analysis, a comparison of existing state-of-the-art models, the integration of our findings into a new head pose prediction model, and the subsequent testing process.

The **Exploratory Analysis of Datasets and State-of-the-Art-Models** chapter analyses the diversity of 360° video datasets in the context of Spatial Information, Temporal Information, Entropy of Saliency Maps (SE), and Actual Entropy (AE). Our analysis reveals a negative correlation between SE and AE, where videos with dynamic centers of focus tend to elicit higher trajectory randomness, as measured by AE, while those with static focus exhibit slower, more predictable user movements. Following our analyses, we evaluate the current state-of-the-art models and find that model performance deteriorates for high-AE videos, highlighting the challenge of predicting head movements in highly dynamic or exploratory viewing scenarios. Additionally,

we find a correlation between model loss and entropy metrics: videos where users exhibit higher trajectory entropy (AE) tend to be associated with higher prediction loss, while videos with higher saliency map entropy (SE) show a negative relationship with prediction loss. These insights suggest that entropy metrics may offer valuable insights for improving head pose prediction models.

In the **Adaptive Attention for Head Pose Prediction** chapter, we introduce two models for head pose prediction, each incorporating entropy information to enhance the model's attention mechanism. These models differ in how they utilize entropy to improve prediction accuracy, with the goal of better capturing user behavior dynamics. The first model, **Entropy-based Adaptive Attention LSTM (E-AALSTM)**, uses an LSTM-based architecture with a dynamic, entropy-modulated attention mechanism. This model adjusts its attention at each time step based on the predictability of the user's trajectory, measured by instantaneous entropy. The second model, **Multi-head Adaptive Attention (AMH)**, takes inspiration from the transformer architecture [9, 72] and incorporates multi-head attention with entropy modulation across attention heads. While E-AALSTM leverages entropy dynamically using a sequential prediction process, AMH predicts the entire trajectory in the output window simultaneously, using only entropy and head pose values from the input window. AMH offers a more computationally efficient alternative to E-AALSTM. The goal of both attention-based models is to focus more on predictable parts of the trajectory, avoiding the randomness introduced by highly unpredictable parts of user trajectories.

In the **Evaluation and Results** chapter, we first provide a detailed overview of the experimental setup used to assess the performance of the proposed models. This includes a discussion of two enriched baselines, i.e., modified versions of the position-only seq2seq encoder-decoder model [56] that incorporate entropy information. These models are the **Position-only baseline augmented with entropy information (pos-augmented)**, where instantaneous entropy is appended to the input at each time step, and the **Position-only baseline with entropy-weighted loss (pos-weighted)**, where the loss function is adjusted based on the trajectory's entropy. These enriched baselines serve as more straightforward methods of incorporating entropy into the model, allowing for a better understanding of how different strategies of incorporating entropy information affect prediction accuracy. We then discuss baseline benchmarks, ablated versions, and state-of-the-art (SOTA) models selected for comparison. We also outline the hyperparameters, datasets, and evaluation metrics used to ensure consistency in our testing process. Following this, we present the results of our evaluation. We first compare our proposed models against ablated versions and baseline models to assess the impact of incorporating entropy information. Following this, we evaluate the best-performing entropy-based model in relation to current state-of-the-art methods. This final comparison highlights the strengths and weaknesses of our approach, placing our work within the context of existing research and demonstrating its effectiveness.

Finally, in the **Discussion and Conclusion** chapter, we discuss the results in the context of our research questions, along with the limitations of our thesis and future research avenues. Lastly, we present the conclusions of our work.

This structured approach aims to contribute to the field by deepening our understanding

of the relationship between the entropy of saliency maps, user trajectory entropy, and the predictability of head pose trajectories. Building on this, we propose a novel head pose prediction model that leverages user trajectory entropy with the intention of improving accuracy and potentially advancing performance over existing models.

## 1.4. Main Contributions

The contributions of this work can be summarized as follows:

- We conduct an analysis of state-of-the-art head pose prediction models, investigating their performance across different types of video content and varying levels of user predictability, measured using instantaneous entropy. Through this analysis, we discover significant correlations between model performance and both the entropy of user trajectories and the entropy of saliency maps.
- We introduce and evaluate entropy-based adaptive attention based models for head pose prediction, specifically E-AALSTM and AMH, which aim to incorporate the entropy of user trajectories. While these models showed limited improvements in performance and struggled to leverage entropy information effectively, they offer insights into the challenges of properly leveraging entropy-based features in deep learning models.
- We implement a "pos-augmented" model that incorporates entropy values into a seq2seq encoder-decoder framework. This model showed modest improvements in prediction stability, especially for datasets where user trajectories tend to be more unpredictable, as measured by their entropy. While it did not outperform all state-of-the-art models, it provided more stable and reliable predictions than other methods, suggesting that entropy augmentation may still have valuable applications in certain contexts.

# 2

# Background and Related Work

In this chapter, we provide a brief overview of 360° videos and virtual reality, followed by a discussion of the 360° video streaming pipeline. Within this context, we highlight the challenge of motion-to-photon latency, a key factor affecting user experience, and examine how researchers have addressed this challenge by introducing additional steps to the streaming pipeline. Next, we review established methods for analyzing user head pose, including their proposed metrics. We then discuss state-of-the-art approaches for head pose prediction, starting with the widely used seq2seq architecture. This is followed by an exploration of specific state-of-the-art models and a focus on the attention-based architecture, which forms the foundation of our proposed model. Lastly, we review existing datasets for head pose prediction.

## 2.1. 360° videos and Virtual Reality

The concept of VR dates back to the 1960s with Ivan Sutherland's first HMD prototype, designed to immerse users in a simulated environment [71]. However, VR technology remained niche until the 2010s when it gained traction with the introduction of consumer-friendly headsets like the Oculus Rift [25]. The acquisition of Oculus by Facebook in 2014 and the release of other devices like the HTC Vive [73] and Samsung VR [70] made VR more accessible to everyday consumers. While VR was initially marketed towards gaming, the increased accessibility helped expand its appeal to diverse applications in the fields of education, training, therapy, and many others [25]. Among VR content formats, 360-degree videos have emerged as one of the most common forms of content for VR experiences, allowing users to explore virtual environments from all angles. This format, coupled with the use of Head-Mounted Displays (HMDs), enhances the sense of immersion by providing a fully navigable viewing space. HMDs serve as the primary tool for interacting with these virtual environments through head movements. HMDs are equipped with sensors such as gyroscopes, motion sensors, and cameras to capture user information. The combined data from these sensors informs the HMD about the viewer's orientation and location information, enabling the system to render different viewing angles based on the viewer's head pose trajectory [74]. Real-time adjustment of the displayed viewport based on the user's head movements creates a sense of presence and immersion, which

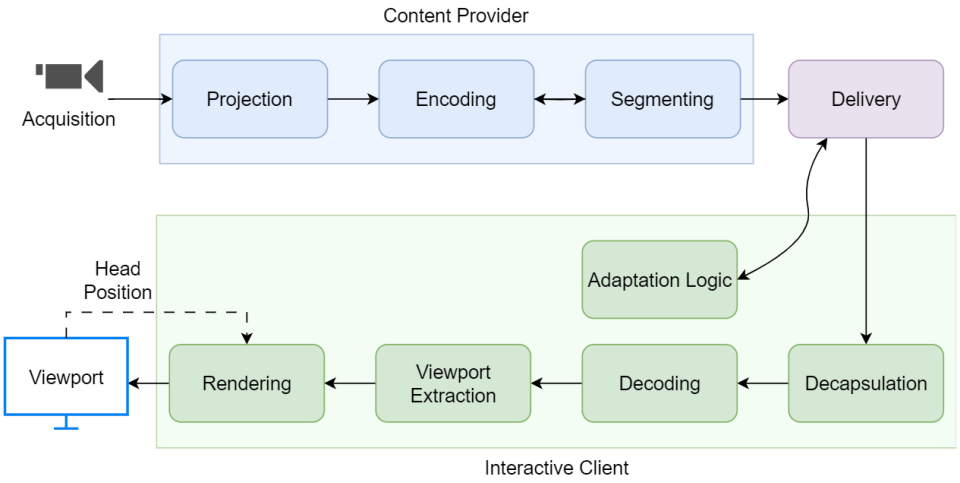



**Figure 2.1:** 360° video streaming pipeline (A simplified version of the pipeline presented by Rossi et al. [57]).

are key to ensuring a good VR experience. However, this requires minimizing latency between user movements and viewport updates to avoid discomfort. High resolution is also essential to maintain a sense of immersion. To deliver this type of high-resolution 360-degree content, efficient streaming is necessary. The streaming pipeline, discussed in the following section, aims to tackle the ongoing challenge of adapting to fluctuating network conditions and user preferences, striving to balance resolution, bandwidth usage, and responsiveness to ensure the best possible viewing experience.

## 2.2. Streaming of VR content

In this section, we go over the 360° video streaming pipeline in the context of MPEG-DASH (Dynamic Adaptive Streaming over HTTP) [66]. The primary objective of this pipeline is to adjust the quality of the content in real-time, based on different conditions such as head movement and network bandwidth, to ensure a smooth playback. As shown in Figure 2.1, the pipeline involves several key steps from the acquisition to the rendering of VR content, which are described below [57]:

1. **Acquisition**: 360° video cameras capture a 360-degree field of view using multiple sensors. The resulting images from these sensors are then stitched together into a spherical format.
2. **Projection**: The spherical content is then projected into a 2D planar representation called *Panorama*. The most common sphere-to-plane projections are the equirectangular projection (ERP) and the cubemap projection (CMP) [21]. ERP is the simpler and more popular projection but suffers from distortion near the poles [5] while CMP offers better overall quality with some distortion due to a higher distribution of pixels at the corners of the cube.
3. **Encoding**: The planar representation is then encoded to reduce the amount of data being transmitted, for example, by using the state-of-the-art codec High-Efficiency Video Coding (HEVC/H.265) [68]. In DASH, multiple resolutions and quality levels called *representations* are created. Allowing each client to

dynamically select the best quality based on their requirements.

4. **Segmenting**: The encoded video is divided into temporal chunks and stored on the server.
5. **Adaptation Logic**: The client dynamically decides the best representations to request for upcoming chunks based on their network conditions, buffer status, and device capabilities.
6. **Delivery**: The client then selects the most appropriate chunk representation and an HTTP origin server processes their requests to deliver these chunks over the internet.
7. **Decapsulation and decoding**: The client processes the received chunks to extract the HEVC bitstream, decodes it, and stores it in a playout buffer.
8. **Viewport extraction**: The user's viewing direction is used to determine the content of interest i.e. the viewport.
9. **Rendering**: The decoded planar representation is back-projected into a spherical format and the content of interest is displayed to the user.

Advances in the streaming pipeline generally aim at improving overall system performance in terms of consumed bandwidth, storage cost, and networking reliability metrics. In terms of adaptation logic, there are different approaches to improve the overall system performance. For example, *Viewport-independent* solutions approach the problem like traditional 2D video streaming and treat the entire panorama equally. This means that each representation available at the server side is encoded with a uniform quality and resolution across the entire panorama. As a result, the client needs to download the entire panorama at high quality. This approach, while ensuring low switching latency, has extremely high costs in terms of storage and bandwidth usage [2]. To minimize bandwidth usage without sacrificing switching latency, viewer interactivity needs to be taken into account, leading to *viewport-dependent* streaming strategies that also adapt to the user.

*Viewport-dependent* strategies rely on the assumption that the entire panorama is not equally important as the viewers focus on some areas more than others. Alface et al. were the first to show that perfect knowledge of the user interaction with 360° videos could save bandwidth by only transmitting the viewport [3]. Viewport-dependent streaming can be implemented in the projection or the encoding step of the streaming pipeline. The former, also named the projection-based approach, adapts the bitrate allocation during projection in such a way that areas most likely to be viewed by the user are the least distorted [80]. This can be accomplished through techniques like pyramid projections [37] and offset cubemap projections [83]. A viewport-dependent strategy in the encoding step, also known as tile-based encoding, was first proposed by Corbillon et al. [13] and then improved by standardised tiled streaming using the HEVC MCTS [81]. The key novelty of a tile-based system is that the panorama is divided into multiple regions called tiles and the tiles are encoded at different bitrates and resolutions. This results in per-tile representations that can be fetched independently by the client at the desired quality level. This allows for a high degree of flexibility as clients can fetch higher quality tiles for the viewport and lower quality

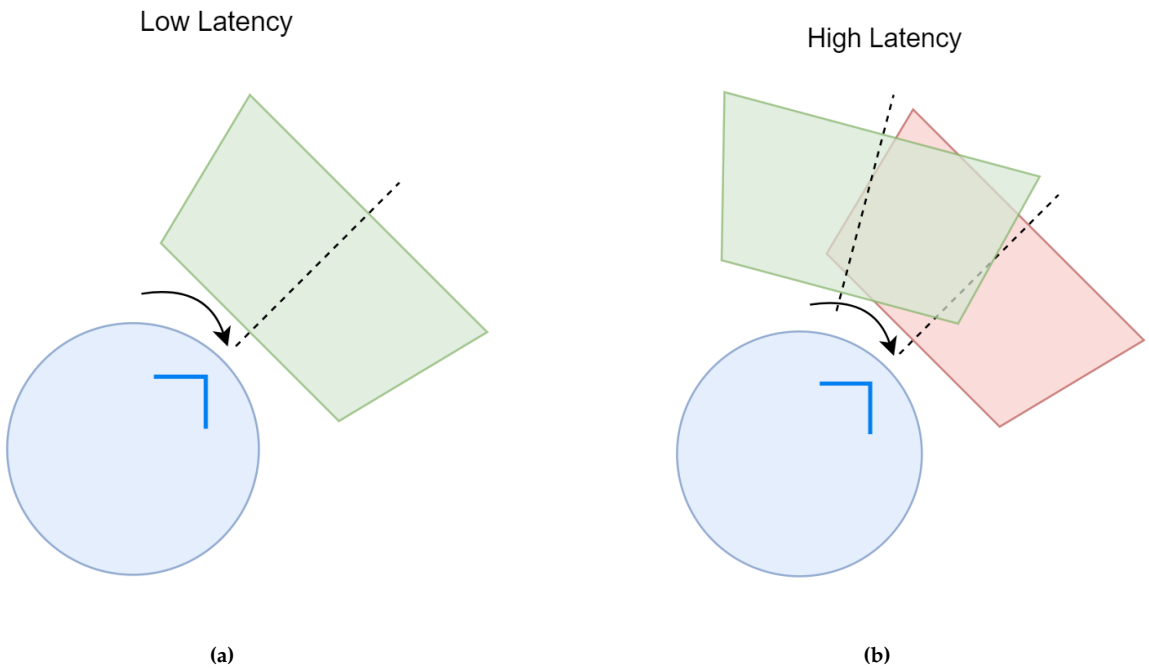



**Figure 2.2:** (a) Viewport and head pose in case of low motion-to-photon latency, (b) Viewport and head pose in case of high motion-to-photon latency (expected viewport in red). Inspired by [74]

tiles for peripheral areas.

However, a key challenge with viewport-dependent streaming is that the head pose information, used to determine the viewport, is often outdated by the time it reaches the server. This delay between the user's head movements and the server's corresponding updates to the viewport is known as Motion-to-Photon (MTP) latency. High MTP latency can lead to a noticeable mismatch between a user's physical movements and the visual feedback they receive, reducing immersion. The discrepancy can also cause dizziness, nausea, and discomfort, ultimately degrading the user's experience [51]. Figure 2.2 illustrates the effect of motion-to-photon latency: Figure 2.2a shows the desired scenario where the rendered viewport (green box) corresponds to the expected viewport, while Figure 2.2b highlights how high latency can result in a viewport (green box) that is not aligned with the expected viewport (red box); contributing to a lower degree of immersion. Qualitative studies have shown that users can perceive latencies as small as 1ms when a 0-latency reference is provided [44]. In studies where no reference was given, users were able to adapt to latencies of up to 100ms without noticeable discomfort, showing some flexibility of the human visual system when no direct comparison is available [4].

Since it is not feasible to eliminate latency entirely, researchers focus on anticipating a user's head movements and predicting their future head positions. This allows the server to deliver content that is more aligned with the user's expected view, thus reducing the negative effects of latency. Qian et al. showed that predicting the viewer's future viewports can help address the motion-to-photon latency [53]. To achieve more accurate viewport predictions, researchers have introduced three additional steps

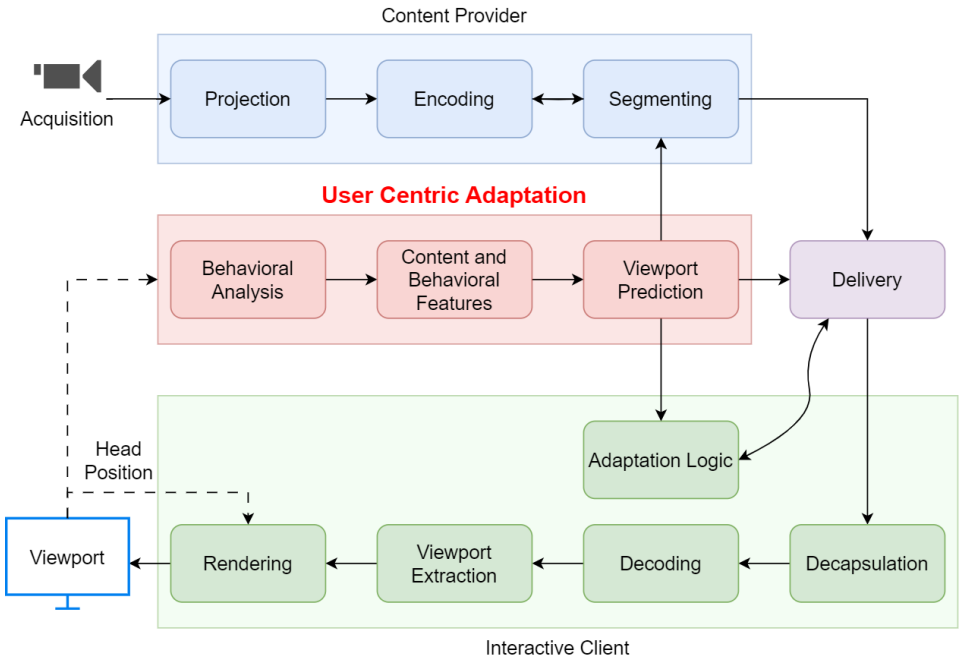



**Figure 2.3:** 360° video streaming pipeline with head pose prediction [57].

to the 360° video streaming pipeline: **Behavioral analysis, content and behavioral features,** and **head pose prediction**, as highlighted in red in Figure 2.3 [57]. These steps, by analyzing user behavior and content features, aim to refine viewport predictions, ensuring that the system better anticipates the user's movements and delivers content with reduced latency. This thesis builds upon these methods, focusing on improving the accuracy of behavioral analysis and viewport prediction, with particular emphasis on the predictability of individual users. The next section delves deeper into the methods used for behavioral analysis and the techniques that researchers have applied to better predict user behavior, which form the foundation of these steps.

## 2.3. Behavioral Analysis

As seen in the previous section, head pose prediction is crucial to mitigate the impact of motion-to-photon latency and ultimately improve the quality of experience when streaming VR content. To accurately predict a user's future head pose, we must first have an understanding of how users behave when viewing 360° video content. We review the outcomes of behavioral studies on how user head pose trajectories are not only affected by the 360° video content, but also by individual differences between users.

**User behavior changes based on the type of VR content being viewed** [60, 64, 47]. Ozcinar et al. demonstrated a direct correlation between the distribution of user fixation points and the video complexity, characterized by **Spatial Information (SI) and Temporal Information (TI)** [47, 32]. SI measures the spatial information in a given frame by applying a Sobel filter and TI measures the temporal information between two frames as the difference between them. Ozcinar et al. found that content with low TI and high SI tends to have the most fixations while content with low SI

and high TI receives a low number of average fixations [47]. Similarly, Rossi et al. observed that when viewing content with no clear focus of attention (as in the case of documentaries), users exhibit highly exploratory trajectories [60]. This behavior is particularly noticeable when we measure the similarity or “affinity" of user trajectories. Users tend to have highly similar navigation patterns when viewing content with a main focus of attention, while their navigation patterns vary more with content that lacks a main focus of attention.

**User behavior also varies due to individual differences**. Jin et al. found that while users, on average, tend to focus on the salient region, there are still significant differences in behavior patterns that should be considered [35]. They confirmed previous findings that future head movements can be predicted using content saliency information but in the short term, the inertia of the user’s movement has a larger effect [56]. Rossi et al. identified a consistent correlation between the **entropy of user trajectories** and prediction errors [59]. They used the entropy metric presented by Song et al. called **actual entropy** (AE), which “captures the full spatio-temporal order present in a person’s mobility pattern" [67]. As the true probabilities are difficult to obtain in real-world scenarios, Rossi et al. make use of the Lempel-Ziv compression algorithm [84] to compute an estimate of the actual entropy [59]. Since actual entropy compares the behavior of a single user over time, it serves as an *intra-user* metric. Rossi et al’s findings highlight the potential of **entropy of user trajectories** as a more explicit measure of the user predictability [58, 8]. Showing clearly that trajectories of individuals with a highly regular style of navigating tend to have a low entropy, while users with a high value of entropy have less predictable movements. Baumann et al. used a variation of actual entropy called instantaneous entropy (IE) to measure the predictability of a user *mobility* in the real world at each moment [8]. Similar to actual entropy, instantaneous entropy may also be extended to viewer head pose trajectories in VR. While AE and IE focus on individual trajectories, Sitzmann et al. proposed the **entropy of saliency maps** as an *inter-user* metric [64]. Instead of analyzing individual trajectories, this method compares the head poses of all users at each moment in the same video, generating saliency maps based on user averages. A high entropy results from a large number of similarly salient objects distributed throughout the scene, while low entropy typically corresponds to a scene with a clear focus [64]. Their analysis found that users tend to explore the scene faster when viewing static scenes with low entropy i.e. scenes with a well-defined center of focus [64]. Previous findings also show that user gaze, on average, remains within 13.85° of the head orientation [64, 16].

In addition to these behavioral differences, Shi et al. [63] also discovered that the accuracy of camera-based gaze estimation methods may be affected by HMD shifts over the course of a VR experience and introduced a method to compensate for these shifts. It is possible that these adjustments may also affect head pose estimation.

## 2.4. Head Pose Prediction

In this section, we review methodologies employed for predicting head pose trajectories. We first discuss how the time-series forecasting framework applies to head pose prediction. We briefly discuss how deep learning approaches can be used to solve the head pose prediction problem, with a focus on the LSTM-based seq2seq architecture

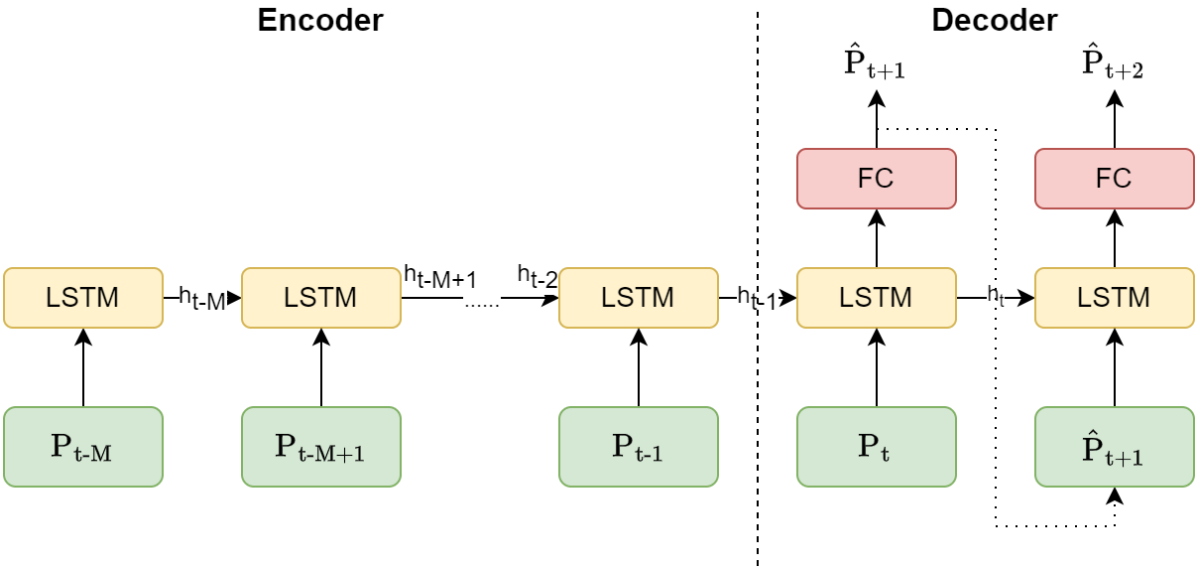



**Figure 2.4:** Standard seq2seq encoder-decoder architecture. Inspired by [56, 69].

utilized by most deep learning head pose prediction techniques. We then explore existing solutions for head pose prediction and conclude with a discussion of attention mechanisms and their potential for the task of head pose prediction.

### 2.4.1. Deep Learning for Head Pose Trajectory Forecasting

Since head movements can be represented as a sequence of data over time, the task of head pose trajectory prediction can be presented as a time-series forecasting problem [56]. Time-series forecasting relies on utilizing past trajectories to predict future movements. This approach naturally aligns with deep learning approaches, specifically through the use of convolutional neural networks (CNN), which are effective at extracting visual dependencies, and recurrent neural networks (RNN), that can model temporal dependencies in sequential data [56, 9]. Among the state-of-the-art, RNN approaches typically use a sequence-to-sequence (seq2seq) architecture [69] using Long Short Term Memory (LSTM) units as the recurrent blocks [56, 45, 79] because of its effectiveness at capturing complex dependencies from input sequences and generated realistic output sequences.

Seq2seq architectures make use of LSTMs' ability to remember relevant information over long sequences, addressing the vanishing gradient problem found in traditional RNNs. LSTMs use two key components: the *hidden state*($h_t$) and the *cell state* ($c_t$), both of which enhance the model's ability to track dependencies across time. The hidden state, $h_t$, holds information about the sequence at a specific point in time, while the cell state, $c_t$, acts as a long-term memory to store information throughout the sequence.

A basic seq2seq model, depicted in Figure 2.4, takes an input sequence of length $M$, denoted as $P = [P_{t-M}, \dots P_t]$ and returns an output sequence of variable length, that in our case is equal to 2, $\hat{P} = [\hat{P}_{t+1}, \hat{P}_{t+2}]$ by passing the input through an encoder-decoder architecture as follows:

- **Encoder:** The encoder *encodes* the input sequence from $P_t - M$ to $P_t - 1$ by passing

each input through LSTM layers and returns a fixed length vector representation, consisting of both $h_{t-1}$ and $c_{t-1}$. These representations capture the temporal dependencies in the input sequence [11].

- **Decoder:** The decoder combines the fixed length vector representation with the latest position in the sequence, $P_t$, and *decodes* it to generate the next position in the sequence, $\hat{P}_{t+1}$. This process can be extended to predict future steps by using the output $\hat{P}_{t+1}$ and the updated vector representation $h_t$ generated by the LSTM at timestep $t$.

Here, the sequence $P$ can represent any form of sequential data, such as head pose, motion maps, or other relevant features.

In the next subsection, we discuss notable existing head pose prediction models, some of which utilize machine learning techniques beyond deep learning.

### 2.4.2. Existing Deep Learning Models for Head Pose Prediction

Based on the insights obtained from user behavioral studies discussed in Section 2.3, several head pose prediction models have been proposed. The majority of these state-of-the-art models incorporate deep learning in some capacity. We broadly categorize these models into two types: Content-aware approaches and User-centric approaches.

**Content-aware approaches** are the ones that, along with the past trajectory of the user, also use information extracted from the 360° video content that is being viewed. This information can include saliency maps and motion maps [56, 78, 79], semantic information [49] and even representations produced by deep learning layers [50]. In this context, Rondón et al. showed that previous attempts [17, 78, 79, 45] failed to adequately incorporate saliency data and performed worse than a position-only baseline [56]. This was exacerbated for short prediction horizons due to the models giving too much importance to saliency features. Their findings showed that, in the short term, the inertia of the user's movement is much more important than content information. As a result, an effective model should be able to attenuate the effect of content information in the first few prediction steps and give it more importance in the later steps. To accomplish this, TRACK [56] processes visual features such as content saliency through an RNN before combining them with the user's positional features. This also aligns with the models evaluated by Park et al. that utilized a 3D CNN model to encode spatio-temporal features from videos through motion maps and saliency maps [50]. Capturing these spatio-temporal features before combining them with the user's movement information results in better predictions compared to simply passing the motion and saliency maps through an RNN. Abawi et al. successfully incorporated a variety of features unique to social videos such as facial expression, actor gaze, and actor face locations to improve their head-pose prediction model [1]. Park et al. proposed SEAWARE, a semantic aware prediction system that exploits video semantic information by encoding the flow of objects over the course of a 360° video [49]. SEAWARE showed highly competitive prediction accuracy and efficiency. While content-aware approaches extract additional information from the viewed content, they do not consider the user's individual characteristics beyond the past trajectory. As a result, researchers also explored user-centric approaches that try to extract additional

information from user behaviors.

The first few **user-centric approaches** were simple logistic regression models that utilized the past and current trajectory of a single user [53]. These models were able to successfully anticipate user behavior in short time windows. These findings were able to showcase the potential of head pose prediction. However follow-up works that explored single-user models [46, 52] found that they suffered from poor prediction accuracy in the long term mainly due to the lack of other users and content information [57].

Behavioral studies, like the ones discussed in Section 2.3 showed a strong consistency and similarity in the way users navigate 360° video content. This opened the gate to cross-user models that utilized information from multiple users. Ban et al. [7] proposed an improvement over previous linear regression models by incorporating K-Nearest-Neighbors (K-NN) clusters. Xie et al. [77] used the predictions of the LR model to form a K-NN set of users with the closest viewport centers and the set was used to compute the viewing probability per tile. While this approach was accurate in a time window of 3 seconds and longer, the clusters neglected the actual spherical geometry and were instead based on Euclidean distance [57]. These initial models paved the way for deep learning frameworks trained on datasets of collected trajectories [28, 39]. State-of-the-art architectures like LSTMs [56] and transformers [9] are shown to be highly effective at predicting trajectories in both short and long-term windows. Beyond using state-of-the-art architectures, researchers have also augmented deep learning approaches by introducing uncertainty [24], using novel representations of head pose information [28] and adding distance constraints to the loss term [38]. Guimard et al. [24] incorporated a degree of uncertainty in the head trajectories by introducing a latent variable, $z$ that results in multiple predictions based on the dimensions of $z$. During training, the model utilizes a best-of-many-samples loss to use the best prediction at a given timestep. Along with improved predictions, the model also provides likelihood estimates for multiple predicted trajectories allowing for more flexibility in streaming optimization. Illahi et al. [28] found that a 3-point representation of pose that represents the back of the head, the left eye, and the right eye in Euclidean space was better for complex models like LSTMs as opposed to quaternion representations. They also noted that even if the only goal is to predict head pose, a model that takes both the position of the HMD in space (head position) and head pose as input outperforms a model that only takes head pose input. Lastly, Lan et al. [38] found that adding a distance constraint term to the loss function reduces the possibility of abrupt changes in predicted head pose, resulting in a more stable model. While these models exploit behavioral information from multiple users, very few explicitly try to consider individual differences between users. Meta-learning approaches that try to account for these individual differences have shown promising results [39, 42]. Li et al. [39] utilized a model agnostic meta-learning (MAML) [18] to adjust the weights of a Bidirectional LSTM (BiLSTM) [23]. Lu et al. [42], on the other hand, used a model-based meta-learning framework where the learning step for both the meta parameters and the inner parameters is wrapped up in a single model in a feed-forward manner. Both approaches showed improvements over standard LSTM-based approaches. Chen et al. [10] also highlighted the potential of using

user-aware metrics by designing a white-box and explainable algorithm for FOV prediction.

Aside from general meta-learning approaches, there have not been any deep learning approaches that explicitly use features that describe individual differences between users. Few techniques have been designed to describe user differences such as clustering [61] and entropy [58, 8]. Clustering of user trajectories also gives insight into how much of the viewed content is shared between users [61]. Some users may inherently be more unpredictable than others, resulting in worse head pose predictions [59]. Incorporating the entropy metrics discussed in Section 2.3 into prediction models may help models learn better representations for these unpredictable users. The entropy of their trajectories provides a quantitative measure of user predictability [58, 8]. Specifically, actual entropy acts as a single measure for the entire trajectory of a user, based on both the visiting rate and the temporal order of the visited areas, and instantaneous entropy [8], an estimate for the momentary predictability of user mobility.

### 2.4.3. Attention for Head Pose Prediction

While a large number of state-of-the-art head pose prediction models utilize recurrent networks, there are two notable problems with recurrent models typically used for time-series forecasting. Firstly, they process sequential data in order as they must generate a sequence of hidden states, $h_t$ as a function of the previous hidden state $h_{t-1}$ and the input for position $t$ (as discussed in Section 2.4.1). This sequential nature makes it impossible to parallelize computations [72]. Secondly, in an encoder-decoder mode, the neural network needs to be able to encode all the necessary information of the past into a single fixed-length memory [6].

In order to address these issues, attention mechanisms have been widely adopted in fields such as language modeling and machine translation [6, 36, 72]. Attention mechanisms work by maintaining a set of hidden representations that scale with the size of the source [36]. By doing so, the model can perform an internal inference step to perform a soft-search over these representations, effectively maintaining a variable-length memory [6]. Before the attention mechanism processes an input sequence, each element of the sequence, $P$, is passed through an embedding layer:

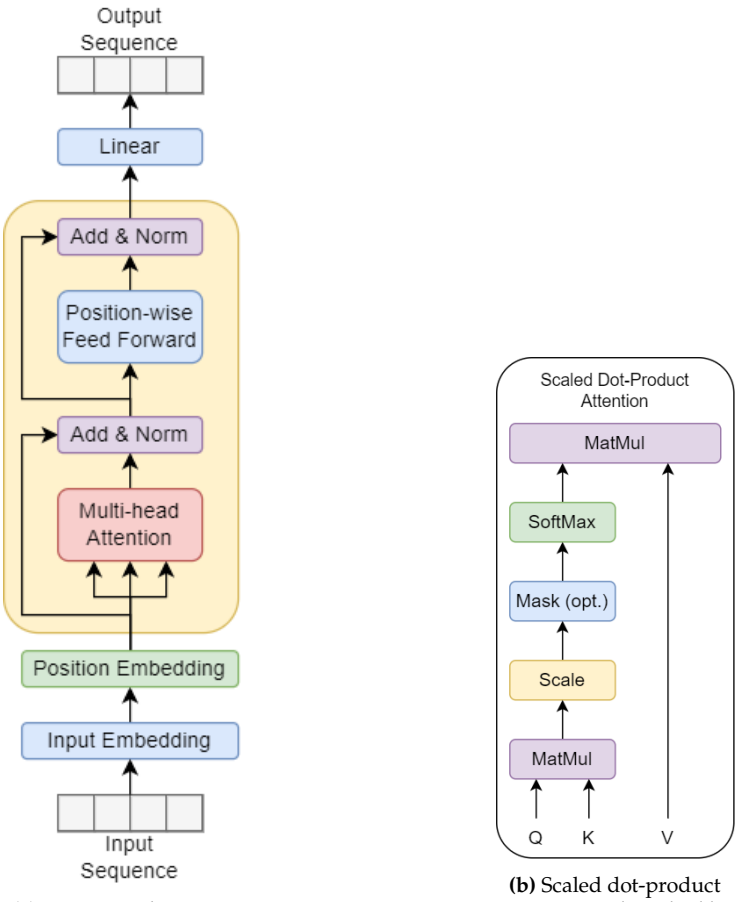
$$E = XW_E \tag{2.1}$$

where $W_E$ are the weights for the embedding layer. Following this, the attention mechanism computes the **Queries (Q), Keys (K),** and **Values (V)** using learned linear transformations, defined as:

$$Q = EW_Q, \quad K = EW_K, \quad V = EW_V \tag{2.2}$$

where $W_Q, W_K, W_V$ are the weight matrices for the queries, keys, and values, respectively. Finally, attention weights are calculated using the scaled dot-product approach as shown in figure 2.5b [72]:

$$Attn(Q, K, V) = softmax(\frac{\mathbf{QK}^T}{\sqrt{d_k}}) \tag{2.3}$$




**(a)** VPT360 architecture [9]

**(b)** Scaled dot-product attention as described by [72]

**Figure 2.5:** VPT360 architecture [9] (a) and scaled dot-product attention [72] (b)

where $d_k$ is the dimension of the keys. This enables the model to generate a context vector by taking the weighted sum of value vectors, where the weights are the attention weights. The resulting context vector is based on past sequences that are most closely associated with the current position. Instead of relying on a fixed past, the model can predict future positions based on these context vectors generated by the attention mechanism [6]. Vaswani et al. proposed an architecture, called a Transformer that completely abandons recurrence and relies entirely on attention mechanisms to draw global dependencies between input and output, allowing for significantly more parallelization [72]. While discarding recurrence lowers the effective resolution, Vaswani et al. counteract this effect through multi-head attention [72]. Their experiments showed that not only do Transformers take significantly less time to train, but they also outperform models using both recurrence and attention on machine translation tasks.

Inspired by these results, Chao et al. proposed VPT360, a transformer model that addresses viewport prediction as a time series forecasting problem [9]. VPT360 forgoes recurrent layers and uses a transformer-based architecture, as shown in figure 2.5a. By leveraging the power of multi-headed attention, VPT360 achieved superior prediction accuracy to the state-of-the-art methods at the time even without utilizing any content

information.

To build on these findings, our work aims to explore the potential of including user-specific entropy metrics in head pose prediction models. Inspired by the work presented by Jiang et al. which enhances few-shot image classification through adaptive attention [34]. By integrating a weight generator that refines the context vector using entropy metrics as support features, we believe our model can offer more nuanced query representations and improve prediction accuracy. In the next section, we examine the datasets used in previous works to contextualize our exploration of user behavior and its impact on model performance.

## 2.5. Datasets

360° navigation datasets collect user navigation data during immersive VR experiences. The navigation trajectory of each viewer is collected for a variety of 360° videos. Specifically, each dataset consists of sequences of spatio-temporal points representing the user's viewing direction over the course of various 360° videos. Each sequence is called a **trace**. Along with this, some datasets also capture the eye movements of the users within the FoV, which can be used to get more details about user attention and salient regions.

During data collection, the first step involves selecting the video content to be used. The length of the videos can vary significantly across datasets. For instance, some datasets use short video clips, such as the 10-second segments employed by Ozcinar et al. [47], while others include full-length videos ranging from 164 seconds to 655 seconds, as demonstrated by Wu et al. [75]. Although full-length videos provide a more comprehensive representation of user behavior, they may cause discomfort for users during extended viewing. Consequently, most datasets opt for shorter, representative segments of approximately 60 seconds [35].

In addition to length, video selection often considers content genre and camera dynamics. Datasets may include a diverse range of genres, such as movies, documentaries, and gaming content [60, 78], as well as videos with varying camera movements, like fixed or moving cameras [35, 17]. This diversity helps capture a broad spectrum of user behaviors. The number of videos in a dataset also plays a crucial role; while a larger dataset ensures heterogeneity, it introduces challenges in data collection. For example, in the CVPR18 dataset [79], only 31 out of 45 participants viewed each video, limiting user coverage per video.

After navigation data is collected, it must be processed and stored in the dataset. User rotational movements in 3-DoF are typically represented in one of three conventions: *Euler angles* [17], *spherical coordinates* [78, 79, 45], or *quaternions* [35]. These representations are discussed in Section 3.2. For deeper analysis, datasets may also include additional information, such as metadata about the selected videos [35, 79, 78] or visual features like saliency and motion maps [17].

These datasets allow us to perform reproducible analysis and experiments for fair comparisons between different head pose prediction methods [35]. In the context of this thesis, we decided to focus on datasets presented in [55] by Rondón et al., which is

| Dataset Name | Head Pose Format | Saliency Maps | Raw Video | Content Details | | |
|---|---|---|---|---|---|---|
| | | | | Length (in s) | No. of videos | No. of users |
| NOSSDAV17 [17] | Yaw, Pitch, and Roll | Yes | No | 60 | 10 | 30 |
| PAMI18 [78] | Longitude and Latitude | No | Yes | 10-80 | 76 | 58 |
| CVPR18 [79] | Longitude and Latitude | No | Yes | 20-60 | 208 | at least 31 per video |
| MM18 [45] | 3D Position in unit sphere | Yes | No | 20-45 | 9 | 48 |
| MMSys18 [14] | Longitude and Latitude | Yes | Yes | 20 | 19 | 57 |
| MM22 [35] | Quaternion | No | Yes | 60 | 27 | 100 |

**Table 2.1:** Features of the datasets discussed.

a collection of existing datasets along with a “Unified Evaluation Framework". This framework provides methods to post-process and standardize the datasets so that they can easily be used to train and evaluate head pose prediction methods. It also includes additional tools to generate saliency maps either by using user attention as described by Rondón et al. [55] or through the content using the PanoSalNet described by Nguyen et al. [45]. The framework allows for consistent comparison of head pose prediction models across diverse datasets.

Some notable datasets are described below:

- **NOSSDAV17 [17]** consists of traces collected on three types of video content: *Fast pace, computer generated*, *Fast pace, natural image*, and *Slow pace, natural image*. Data is collected from 30 users over 10 videos. Each video is 60 seconds long. Head pose is represented with the Yaw, Pitch, and Roll data in the range [-180, 180]. This dataset also contains saliency maps for each video.
- **PAMI18 [78]** consists of traces collected over a diverse collection of videos from YouTube and VRCun, including computer animation, acting, sports, driving, scenery, and so forth. Data is collected from 58 users over 76 videos. The videos range from 10 to 80 seconds long. Head pose is represented as latitude and longitude in the Geographic Coordinate System (GCS), ranging from [-180, 180] with the center of the video as the origin.
- **CVPR18 [79]** consists of traces collected over a diverse collection of YouTube videos including computer animation, acting, sports, and driving. Data consists of at least 31 traces for each video out of a collection of 208 videos. The videos range from 20 to 60 seconds long. Head pose is represented as latitude and longitude in the Geographic Coordinate System (GCS), ranging from [0, 1] with the top-left corner as the origin.
- **MM18 [45]** data is collected for 48 users over 9 videos. The videos range from 20 to 45 seconds long. Head pose is represented as the 3D position of the center of the viewport in the unit sphere, (x, y, z), in the range [0, 1]. The dataset also contains saliency maps for each video.
- **MMSys18 [14]** consists of traces collected on videos in various indoor and outdoor settings in rural and urban environments. Data is collected from 57 users over 19 videos. Each video is 20 seconds long. Head pose is represented as

latitude and longitude data in GCS, ranging from [0, 1] with the top-left corner as origin. The dataset also contains saliency maps for each video.

- **MM22 [35]** consists of traces collected over a diverse collection based on video quality, amount of camera movement, and dispersion of regions of interest. Data is collected from 100 users over 27 videos. Each video is 20 seconds long. Head pose is available in multiple representations, Euler angles, quaternions, and 3D position in the unit sphere.

Table 2.1 summarizes the main features of the datasets we will consider in this work.

# 3

# Methodology

This chapter first summarizes all the variables mentioned in this thesis with a table. It then presents the key definitions and the Problem Formulation related to head pose forecasting framework. We also describe the equations related to content and user metrics discussed in Section 2.3. Lastly, we briefly discuss the steps involved in carrying out this project.

## 3.1. Key Terms

Table 3.1 summarizes all the key terms used in this thesis.

**Table 3.1:** Definitions of the variables used in the thesis.

| Variable | Definition |
|---|---|
| $P_t$ | User's head pose at time step $t$ |
| $\theta_t$ | Azimuthal angle at time step $t$ |
| $\phi_t$ | Elevation angle at time step $t$ |
| $r$ | Length of vector in angular representation |
| $x_t, y_t, z_t$ | Unit vector in the direction of the equirectangular projection at time $t$ |
| $qx, qy, qz, qw$ | Quaternion representation of head pose at time step $t$ |
| $V$ | Viewed video content |
| $T$ | Length of video $V$ |
| $P = [P_0, P_1..., P_T]$ | Trajectory of a user over the course of video $V$ |
| $I_t$ | All the visual information at time $t$ of a generic video $V$ |
| $H$ | Prediction horizon |
| $M$ | Length of input window $V$ |
| $p \in [0, H]$ | Each time step within the prediction horizon |
| $\mathcal{M}$ | Head pose prediction model |
| $O(.), L_{orth}$ | Orthodromic distance/loss |
| $F_t$ | Pixel representation of the frame at time $t$ |
| $F_t^{PQ}$ | Luminance values of frame $F_t$ in the perceptual quantizer |

| Variable | Definition |
|---|---|
| | domain. |
| $SI_t$ | Spatial Information for the frame at time $t$ |
| $TI_t$ | Temporal Information for the frame at time $t$ |
| $s$ | Ground truth saliency of a given frame |
| $SE$ | Entropy of Saliency Maps |
| $AE$ | Actual entropy |
| $C$ | Quantized sequence of user trajectory $P$ |
| $Pr(.)$ | Probability of finding a subsequence in the past trajectory |
| $\lambda_t$ | Length of a subsequence of $C$ |
| $IE$ | Instantaneous entropy |
| $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ | Query, Key and Value matrices for Attention |
| $\mathbf{W}, \mathbf{U}$ | Weight matrices for a given neural layer |
| $\mathbf{b}$ | Bias vector for a given neural layer |
| $M_t$ | Entropy-based Modulating Factor for timestep t |

## 3.2. Key Definitions and Problem Formulation

To formulate the problem of head pose prediction, we need to first define some notations.

Considering that the user is watching a 360° video $V$ of length $T$, the trajectory of the user over the course of the video can then be described as $P = [P_0, P_1 ... P_{T-1}, P_T]$. Here, $P_t$ denotes the user's head pose at a given time $t$. This allows us to represent viewport information with a single value, i.e. the center of the viewport on the viewing sphere. $P_t$ is generally represented using one of three notations:

- Angular representation: $P_t = [\theta_t, \phi_t]$ with $0 \leq \theta \leq 2\pi$ as the azimuthal angle, $0 \leq \phi \leq \pi$ as the elevation angle, and $r = 1$ as the radius of the sphere, as shown in figure 3.1.
- Unit Vector representation: $P_t = [x_t, y_t, z_t]$ which represents a unit vector in the direction of the equirectangular projection.
- Quaternion representation: $P_t = [qx, qy, qz, qw]$ represents the axis/angle representation for the rotation.

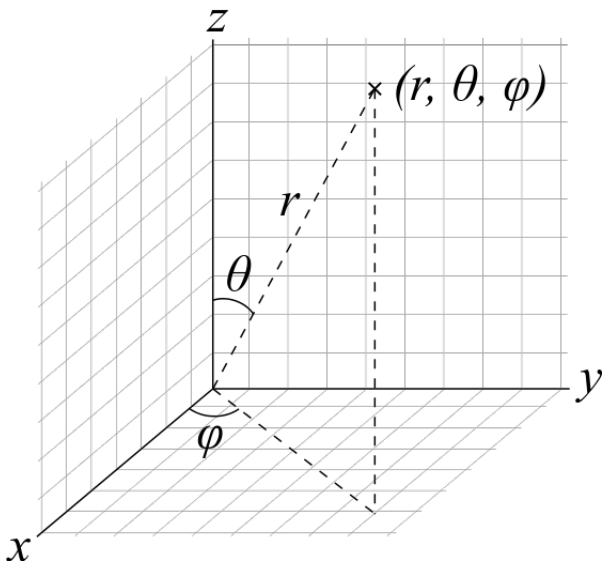



**Figure 3.1:** Azimuth angle and elevation for describing head Pose[15]

We define $I_t$ as the given visual information $I$ at time $t$ of a generic video $V$, this can include raw RGB data, pre-computed saliency information obtained from a saliency extractor, and even optical flow information through motion mapping. Furthermore, $H$ is the prediction horizon and $M$ is the input window. Each time step for which the prediction is computed is referred to as a prediction step, denoted as $p$, where $p \in [0, H]$.

Given the above notation, we now define the head pose prediction task as a model $\mathcal{M}$ that can predict the user trajectory, $P = [P_t, ..., P_{t+H}]$ for all prediction steps, $p$ starting at any given timestamp $t \in [T_{init}, T]$, using the information from $t - M$ to $t$ as shown by Figure 3.2. To allow for a complete input window, we define $T_{init} = M$

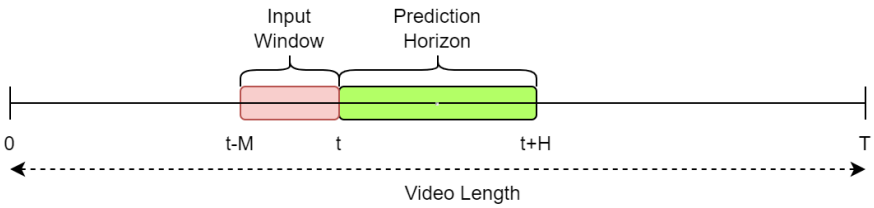



**Figure 3.2:** Head pose prediction: Head pose from time-stamp t to t+H is predicted using head pose and video data from time-stamp t-M to t

We formulate the task of optimizing the model $\mathcal{M}$ as finding the best model $\mathcal{M}^*$ verifying:

$$\mathcal{M}^* = \arg\min_{\mathcal{M}} \mathbb{E}_t[O([P_{t+1}, ..., P_{t+H}], \mathcal{M}(P_{t-M:t}, I_{t+1:t+H}))],$$

where $O(.)$ is the orthodromic distance [55] between the ground-truth series of future positions, and the predicted series of future positions. Orthodromic distance represents the shortest distance between two points on the surface of a sphere, measured along the surface of the sphere [56]. The Orthodromic distance between two unit vectors $\vec{n}_1$

and $\vec{n_2}$ is calculated as:

$$O(\vec{n_1}, \vec{n_2}) = \arccos \vec{n_1} \cdot \vec{n_2} \tag{3.1}$$

## 3.3. Content and User Metrics

Having previously discussed the definitions of these metrics in Section 2.3, we now present the formulae for the **content metrics**, i.e., Spatial Information (SI) and Temporal Information (TI), and the **user metrics**, i.e., Actual Entropy (AE), Instantaneous Entropy (IE) and Entropy of Saliency Maps (SE).

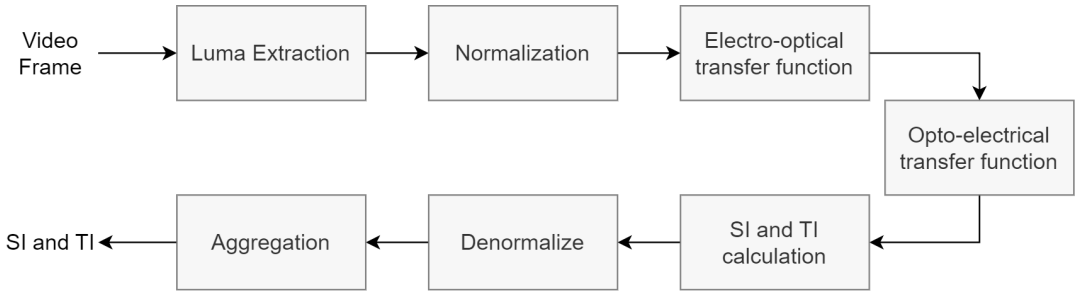



**Figure 3.3:** SI/TI processing pipeline [32]

- **Spatial Information (SI) and Temporal Information (TI):** SI measures the spatial information in a given frame. It is usually higher for more spatially complex scenes. TI measures the number of temporal changes between two consecutive frames, it is usually higher for high-motion sequences. SI and TI are not meant to be a measure of entropy, but instead provide subjective measures to check if videos in datasets span a wide range of spatiotemporal complexity [54]. SI and TI are calculated using the processing pipeline outlined in Figure 3.3 [32]:
  - **Luma extraction:** We first pre-process the frames of a video, $F$, from RGB to luma values in, $F_l$. For a video, the luma represents the *perceived* brightness of a frame. The luma value of a pixel, Y is calculated following ITU-R BT.709 [30]:

    $$Y = 0.2126R + 0.7152G + 0.0722B, \tag{3.2}$$

    where R, G, and B denote the Red, Blue, and Green components of the pixel.
  - **Normalization:** $F_l$ values are then normalized to be in the range [0,1], resulting in $F'_l$. For an 8-bit representation, the normalize function is as follows:

    $$\text{Normalize}(x) = \frac{x}{2^8 - 1} \tag{3.3}$$

    $$F'_l = \text{Normalize}(F_l) \tag{3.4}$$

  - **Electro-optical transfer function (EOTF):** The EOTF function is applied to $F'_l$ to convert the luma values into luminance values in the physical domain, $F_{le}$. We use a simplified version of the EOTF function described in ITU-R BT.1886 [31], provided by VQEG [54]:

    $$F_{le} = (l_w - l_b)F'^{\gamma}_l + l_w, \tag{3.5}$$

where $\gamma = 2.4$, and $l_w$ and $l_b$ denote the screen luminance for white and black respectively. For a display with the standard dynamic range, $l_w = 0.1\,\mathrm{cd/m^2}$ and $l_b = 300\,\mathrm{cd/m^2}$ [54].

– **Opto-electronic transfer function (OETF):** The luminance values, $F_{le}$ are then converted to the Perceptual Quantizer (PQ) domain, $F^{PQ}$, which can represent luminance levels up to $10,000\,\mathrm{cd/m^2}$. We use the OETF function defined in ITU-R BT.2100-2 [29], provided by VQEG [54]:

$$l_{m1} = 10000^n, \tag{3.6a}$$

$$l_{m1} = F_{le}^n, \tag{3.6b}$$

$$F^{PQ} = \frac{c_1 l_{m1} + c_2 l_{m2}}{l_{m1} + c_3 l_{m2}}^m, \tag{3.6c}$$

where $n$, $m$, $c_1$, $c_2$, and $c_3$ are constants with values defined as follows: $n = 0.1593$, $m = 78.8437$, $c_1 = 0.8359$, $c_2 = 18.8515$, and $c_3 = 18.6875$.

– **SI Calculation:** The SI at time $t$ is calculated by applying a Sobel filter to the $t$th frame, $F_t^{PQ}$ and taking the standard deviation (SD) over the resulting pixels:

$$SI_t = \sigma[Sobel(F_t^{PQ})] \tag{3.7}$$

– **TI Calculation:** TI is based on the motion difference features, $M_t(i,j)$, which is the difference between pixel values at the same spatial location across successive frames:

$$M_t(i,j) = F_t^{PQ}(i,j) - F_{t-1}^{PQ}(i,j) \tag{3.8}$$

where $F_t^{PQ}(i,j)$ is the pixel at the $i$th row and $j$th column of the $t$th frame. The TI is then computed as the SD over space of $M_t(i,j)$ over the entire frame:

$$TI_t = std([M_t(i,j)]) \tag{3.9}$$

Note that since the calculation of TI requires 2 frames, it starts at $t = 2$

– **Denormalization:** SI and TI values are denormalized using the inverse of the normalization function:

$$\text{Denormalize}(x) = x * (2^8 - 1) \tag{3.10}$$

– **Aggregation:** SI and TI values are aggregated by taking the mean over all frames, following ITU-T P.910 [32]:

$$SI = \frac{1}{T}\sum_{1}^{T} SI_t, \tag{3.11a}$$

$$TI = \frac{1}{T-1}\sum_{2}^{T} TI_t, \tag{3.11b}$$

$$\tag{3.11c}$$

where T is the total length of the video.

- **Actual Entropy (AE):** AE is a measure of the predictability of a user's trajectory over the course of a video. It is defined as:

$$AE_i = -\sum_{P_i^{'} \subset P_i} Pr(P_i^{'}) \log[Pr(P_i^{'})] \tag{3.12}$$

where $Pr(P_i^{'})$ is the probability of finding the subsequence $P_i^{'}$ in the past trajectory $P_i$ of the user. However, since the true probabilities cannot be obtained in real-world scenarios, researchers utilize a Lempel-Ziv compression algorithm [84] to compute an estimate of the AE [59]. By discretizing the spherical space of the 360° video, the trajectory $P$ can be quantized into a sequence, $C$, of the blocks to which $P$ belongs to at time $t$. Now we can define $L_t = [c_t, c_{t+1}, ..., c_{t-1+\lambda_t}]$ as a sub-sequence of $C$ of length $\lambda_t$ time-slots. The Lempel-ziv approximation is then applied as:

$$AE(C) \approx (\frac{1}{T}\sum_{t=1}^{T} \lambda_t)^{-1} \log_2(T) \tag{3.13}$$

where $\lambda_t$ is the shortest subsequence in $C$ starting at, and not appearing before, time $t$.

- **Instantaneous Entropy (IE):** IE is a variation of AE that measures the predictability of a user's movements at any given instant. IE at a given instant $k$ is calculated similarly by calculating the AE of the trajectory up to the timestep $k$. The quantized sequence can be represented as $C^k$, therefore the Lempel-ziv approximation can be applied as:

$$IE(C^k) \approx (\frac{1}{k}\sum_{t=1}^{k} \lambda_t)^{-1} \log_2(k) \tag{3.14}$$

- **Entropy of Saliency Maps (SE):** SE is an inter-user metric that quantifies how much the head poses of different users align with one another for a given frame by utilizing saliency maps. The SE of a single frame is given by the Shannon entropy:

$$\sum_{i=1}^{N} s_i^2 log(s_i^2) \tag{3.15}$$

where $s$ represents the ground truth saliency and $N$ is the number of pixels.

## 3.4. Methodology

In this section, we discuss the methodology used for each step of the project. We began by analyzing the datasets used, followed by an evaluation of state-of-the-art models. After analyzing the existing data and models for head pose prediction, we integrated our findings into a head pose prediction model. Finally, we conducted tests and ablation studies to evaluate the impact of incorporating entropy metrics into head pose prediction models. These steps are summarized in Figure 3.4.


**Figure 3.4:** Methodology

### 3.4.1. Exploratory Analysis of Datasets and State-of-the-Art-Models

This step involved exploring the datasets listed in Table 2.1 to assess the diversity of video content and analyze the relationship between entropy metrics, specifically the Entropy of Saliency Maps (SE) and the Entropy of User Trajectories (AE). Following this exploratory analysis, we evaluated current state-of-the-art models, analyzing their performance in relation to entropy metrics throughout the duration of a video. This evaluation revealed valuable insights into the relationship between entropy metrics and prediction accuracy, motivating the integration of these metrics into head pose prediction models.

We began by exploring each dataset listed in Table 2.1 to determine the diversity of the video content based on Spatial Information (SI), Temporal Information (TI) [32], and the entropy of saliency maps (SE) [64]. These datasets are commonly used in head pose prediction studies and are included in the Unified Evaluation Framework [55], making them relevant for comparing our work with prior research, and future research that utilizes the framework. Additionally, they cover a variety of scenarios in terms of video length, number of participants, and video content, providing a solid foundation for contextualizing our approach within the existing literature. Two datasets, CVPR18 [79] and MM18 [45], were excluded from the analysis and testing due to incomplete trajectories and missing video content, respectively. User metrics, specifically SE and actual entropy (AE) of user trajectories [58], were compared for all videos to check for any trends. We also took a closer look at a few 360° videos in the datasets to gain a better understanding of these trends.

After analyzing the datasets and exploring the relationships between entropy metrics, we performed the train-test split of the videos in each dataset to ensure that the splits were representative of the entire dataset. The Unified Evaluation Framework [55] performs the train-test split at the video level by selecting a fraction of videos and training models on all user trajectories for the selected videos. However, this approach can lead to data contamination, as user trajectories from the same users might appear in both the training and test sets, albeit on different videos. This overlap could cause information leakage and affect performance metrics. To address this, we selected 60% of the videos and 60% of the users for the training set, leaving the remaining user-video pairs for testing. This approach ensured that the models were evaluated on previously unseen user-video pairs, allowing for a more accurate assessment when evaluating

state-of-the-art (SOTA) models.

Following the dataset analysis and train-test split, we evaluated several state-of-the-art models using the Unified Evaluation Framework [55]. The models were chosen based on their performance, uniqueness of approach, and availability of code. The models evaluated, summarized in Table 3.2, include:

- **Pos-only [56]:** This is a baseline LSTM based model that uses the Seq2Seq encoder-decoder architecture discussed in Section 2.4.1. As input, it only takes the user's head pose over the input window.
- **TRACK [56]:** TRACK is a variant of the seq2seq encoder-decoder model that processes content saliency maps with an RNN before merging it with head pose data. This approach ensures that the effect of content information is attenuated in the short term while still considering its impact in later steps through the recurrent network's representation of the content information.
- **DVMS [24]:** DVMS also uses a seq2seq decoder architecture, but it utilizes a latent variable, $z$, to generate multiple predictions based on the dimensions of $z$. This introduces a degree of uncertainty in the head pose trajectories and allows us to train the model based on the best prediction for each timestep.
- **VPT360 [9]:** VPT360 uniquely forgoes the use of recurrence, employing a transformer-based architecture that uses multi-head attention to predict the user's trajectory over the output horizon all at once.

| Reference | Approach | Notes | Content-Aware |
|---|---|---|---|
| Pos-only [56] | LSTM trained only on pitch and yaw | Baseline model | No |
| TRACK [56] | RNN to process saliency before merging with pose data. | Baseline for information beyond head pose | Yes |
| DVMS [24] | Multiple trajectory prediction | Incorporation of uncertainty of user trajectories | No |
| VPT360 [9] | Multi-head attention based model that forgoes recurrence | Forgoes recurrence altogether | No |

**Table 3.2:** Notable Head Pose Prediction Models

We compared the accuracies of these models on the datasets explored during the dataset analysis. The accuracy metric used was the orthodromic distance between the predicted and actual center of the viewport [55]. Prediction accuracy was evaluated over the course of a prediction window of *5 seconds*, in line with previous studies on head pose prediction [55, 9, 39], which allowed for the evaluation of both short-term and long-term predictions.

Along with that, the models were assessed based on how much their performance degraded when predicting trajectories for videos where users tend to exhibit high entropies compared to those where users exhibit low entropies [8, 59]. This was done by comparing the average accuracy over the prediction horizon for each video in a given dataset, and by selecting specific videos to compare the prediction loss at each timestep with the entropy of saliency maps and user trajectories.

By performing this analysis, we identified key findings that motivated the integration of entropy metrics into our proposed models.

### 3.4.2. Integrating Findings into New Head Pose Prediction Models

Building on the insights from our exploratory analysis, we proposed and evaluated multiple head pose prediction models. First, we developed two adaptive attention-based models: an LSTM-based model with an entropy-based adaptive attention layer and a multi-head attention model incorporating entropy modulation. These models were designed to dynamically adjust attention scores based on the entropy of user trajectories. Specifically, attention scores were modulated based on the Instantaneous Entropy (IE), being lowered when IE was high and raised when IE was low, to focus more on predictable, stable sections of the trajectories while placing less emphasis on unpredictable sections. The LSTM-based model recalculates IE for the predicted head poses and uses those values as input again, allowing the model to adapt as the trajectory evolved. In contrast, the multi-head attention model only uses the input window to modulate attention, without recalculating IE for future poses. The integration of entropy features into these models aimed to improve prediction stability and accuracy by adjusting the model's focus towards more stable behaviors.

### 3.4.3. Evaluation of Entropy-Based Models

Building on the adaptive attention models proposed in the previous chapter, we also introduced two additional models that enrich the position-only seq2seq encoder-decoder with entropy information. One model augments the input with instantaneous entropy, while the other modifies the loss function to incorporate entropy. These variations allowed us to explore different methods of integrating entropy into the prediction process. We evaluated the proposed models and the entropy-enriched baselines by comparing them against their baseline versions, which did not incorporate entropy-based features. Specifically, we assessed the performance of the proposed adaptive attention-based models against their non-adaptive counterparts, and the performance of the augmented models against the baseline position-only model [56]. This comparison allowed us to identify the best-performing entropy-based model based on accuracy and robustness in handling unpredictable user trajectories. The evaluation followed the methodology outlined in Subsection 3.4.1 and helped highlight the impact of entropy features on model performance.

After selecting the best-performing model from this initial evaluation, we conducted a comprehensive comparison against state-of-the-art models. This allowed us to assess how well our best proposed model performs in terms of accuracy and robustness compared to leading existing methods. Additionally, we selected a few videos to evaluate the performance of our model throughout each video, comparing its stability to that of another state-of-the-art model. This comparison provided a clear understanding of how our model performs relative to existing methods and highlighted the value of incorporating entropy metrics into head pose prediction.

# 4

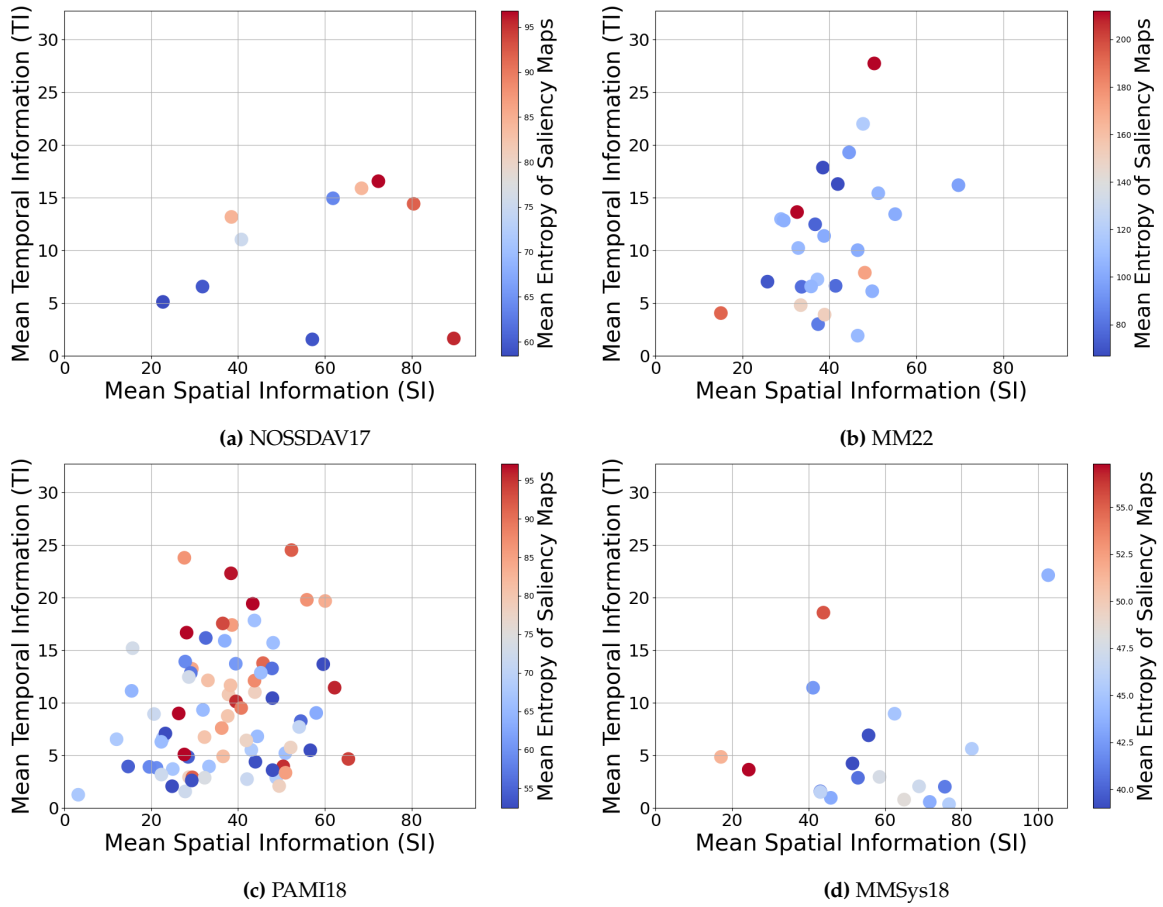
# Exploratory Analysis of Datasets and State-of-the-Art-Models

In this chapter, we analyze the datasets used in this study, focusing on the diversity of 360° video content and exploring the relationship between the studied entropy metrics. This analysis confirms the diversity of the selected datasets and provides insights into the connection between entropy of saliency maps (SE) and actual entropy (AE), enhancing our understanding of user behavior in 360° video environments. Before going into the evaluation of existing state-of-the-art models, We discuss our train-test splitting approach, ensuring no overlap between users in the training and testing sets while maintaining the representativeness of the overall dataset. Finally, we evaluate the performance of existing state-of-the-art models and explore the correlations between the entropy metrics (SE and Instantaneous Entropy) and the prediction error.

## 4.1. Dataset Analysis

To ensure that the selected datasets cover a diverse range of 360° video content, we analyze the distribution of aggregated spatial information (SI), temporal information (SI) and Entropy of Saliency Map (SE) values per video in the dataset. Figure 4.1 shows the mean SI and TI for the four selected datasets, NOSSDAV17 [17], MM22 [35], PAMI18 [78] and MMSys18 [14]. The color of each dot represents the mean entropy saliency maps where blue represents low entropy and red represents high entropy. As we can see from Figure 4.1, the selected datasets have a wide range of SI and TI values, with SI predominantly in the [0,50] range and TI in the [0,25] range. The SE values also cover a broad spectrum, ranging from 55 to 90 for NOSSDAV17 [17] and PAMI18 [78]. MM22 shows an even wider range of entropies from 80 to 200. Saliency maps for videos in MMSys18 show lower entropy values, ranging from 40 to 55. This variability ensures that the video content is heterogeneous and enables the models to generalize across a wide range of 360°-video content. We observe a linear correlation between SI and TI in the NOSSDAV17 dataset and MM22 dataset, but this correlation is not as prominent in the PAMI18 and MMSys18 datasets.

In Figure 4.2, we compare the SE and Actual Entropy (AE) of viewers of the videos



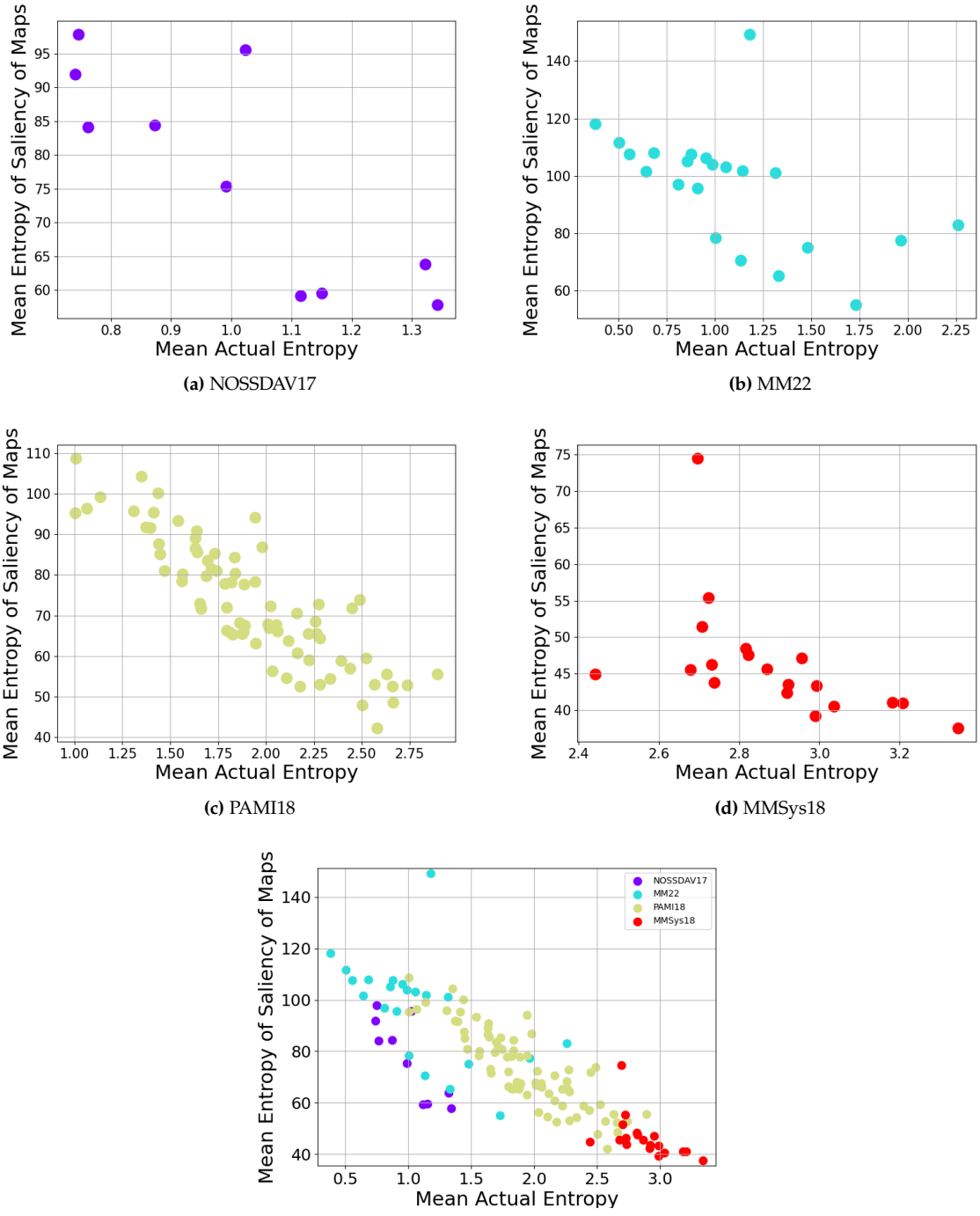

**(a)** NOSSDAV17 **(b)** MM22

**(c)** PAMI18 **(d)** MMSys18

**Figure 4.1:** Mean Temporal Information vs Mean Spatial Information plots with a heatmap for mean SE for videos in (a) NOSSDAV17, (b) MM22, (c) PAMI18, (d) MMSys18

in these selected datasets to examine any correlations. Notably, the MM22 dataset contains 2 outlier videos with really high SE (>150), which have been excluded from the plots to provide a clearer visualization of the overall trend in the dataset. As described in Section 2.3, SE is an inter-user metric that compares the head poses of each user for a given frame, and a low SE represents a frame with a dominant center of attention. Actual Entropy, on the other hand, is an intra-user metric based on the trajectory of each individual user, and a high AE indicates a highly random trajectory. We observe a negative correlation between the entropy of saliency maps and actual entropy in Figure 4.2. This suggests that as the average SE of a video decreases, user trajectories become more unpredictable, potentially due to heightened engagement levels where viewers exhibit faster movements that involve following a strong center of focus. There is also considerable variation in both AE and SE across the datasets, reflecting the diversity of user behaviors and video content. Furthermore, Figure 4.2e confirms that this trend holds true across all videos over all four datasets. This shows that this negative correlation is not isolated within the collected data but represents a common trend in 360° videos.

For a better understanding of AE and SE, we take a look at the video *VRBasketball* in the PAMI18 dataset [78] which has a high mean AE, equal to 2.89, and a relatively low mean SE, equal to 55.56. The content of *VRBasketball* is dominated by two static




**(a)** NOSSDAV17

**(b)** MM22

**(c)** PAMI18

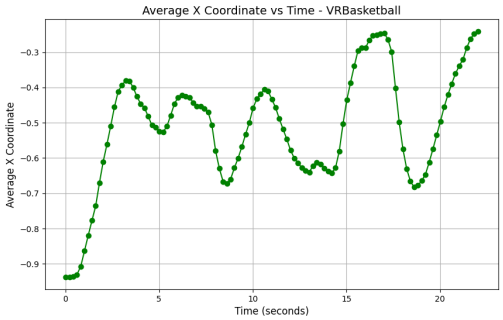
**(d)** MMSys18

**(e)** All datasets

**Figure 4.2:** Mean SE vs Mean AE for videos in (a) NOSSDAV17, (b) MM22, (c) PAMI18, (d) MMSys18, (e) All four datasets

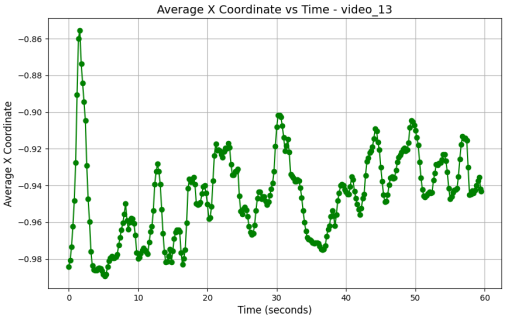
speakers, and an engaging and dynamic center of focus, a basketball. The presence of this dynamic object explains why the mean SE is relatively low. The high AE is likely due to the unpredictable, rapid movements of the basketball around the scene. This results in rapid and varied head movements as viewers track the ball, leading to unpredictability in their trajectories. Since most head movements occur in the

**(a)** Scene from *VRBasketball*




**(b)** Average X coordinate for *VRBasketball*

**(c)** Scene from *Mario*



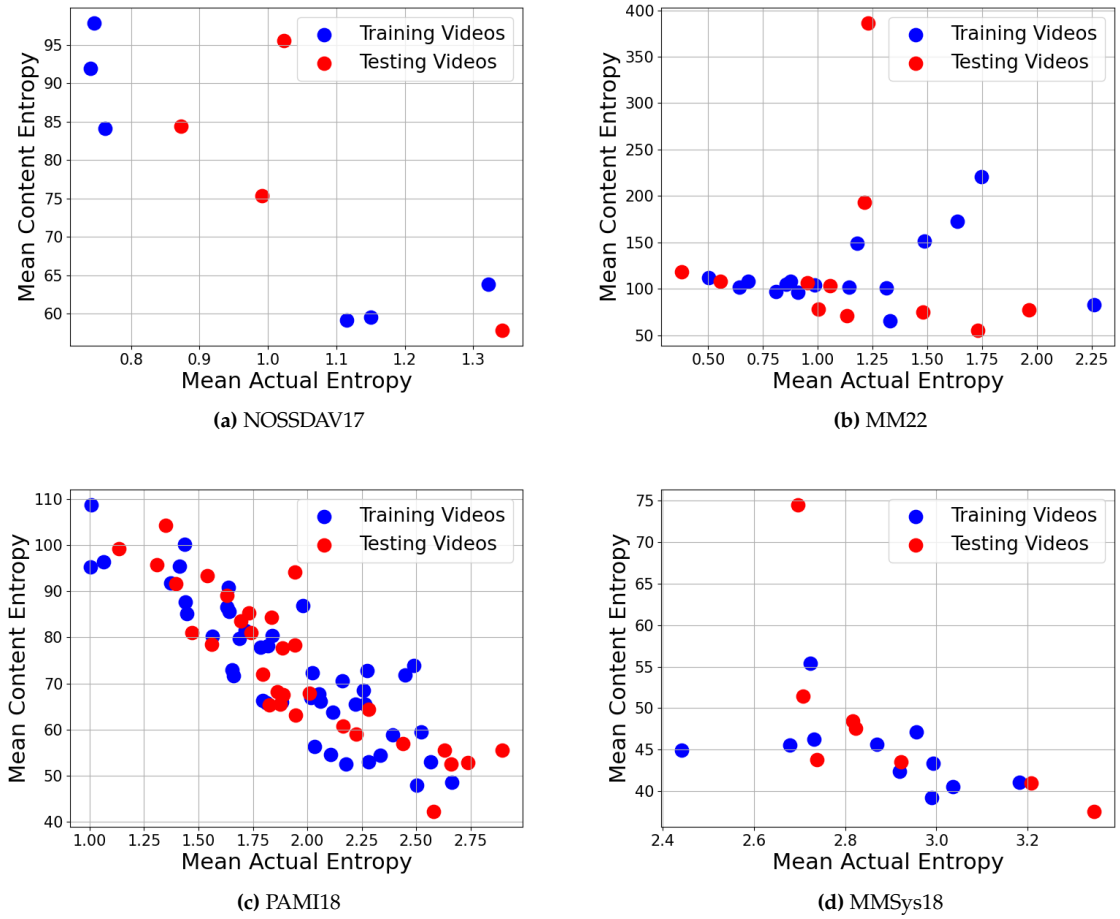

**(d)** Average X coordinate for *Mario*

**Figure 4.3:** Representative scenes from *VRBasketball* (a) and *Mario* (c). Along with the average X coordinate of user trajectories over the course of *VRBasketball* (b) and *Mario* (d).

horizontal plane, we plot the average X-coordinate of all the users' head poses over the course of *VRBasketball* in Figure 4.3b. As shown by the plot, users show rapid movements when the speakers pass the ball to each other or perform some tricks. Specifically, the ball is passed from one speaker to another at timestamps 3 seconds, 6 seconds, and 16 seconds. At 10 seconds, one of the speakers performs some dribbling tricks while moving around with the ball. This suggests that a dynamic center of focus with unpredictable movements results in unpredictable head pose trajectories. In contrast, we also examine video *mario* in the MM22 dataset [35] which has the lowest mean AE, equal to 0.37, and a high mean SE, equal to 118.18. *Mario's* content has a static center of focus along the center of the frame. The average X-coordinate over time, shown in Figure 4.3d, reveals that users show a very stable trajectory with the average X coordinate staying relatively unchanged. This would explain the low AE, as users exhibit slower movements while viewing the video. The high SE can be attributed to users' slow exploration of the scene, driven by the fact that while the video has a stable center of focus, it is not particularly interesting. This comparison suggests that videos with dynamic centers of attention, like *VRBasketball*, result in more unpredictable head movements as viewers track moving objects, while videos with static centers of focus, like *Mario*, lead to slower, more predictable head movements.

### 4.1.1. Train-test split

Given the diversity of the datasets, we ensure a balanced and representative train-test split by partitioning videos into non-overlapping groups and selecting a fraction (in




**(a)** NOSSDAV17

**(b)** MM22

**(c)** PAMI18

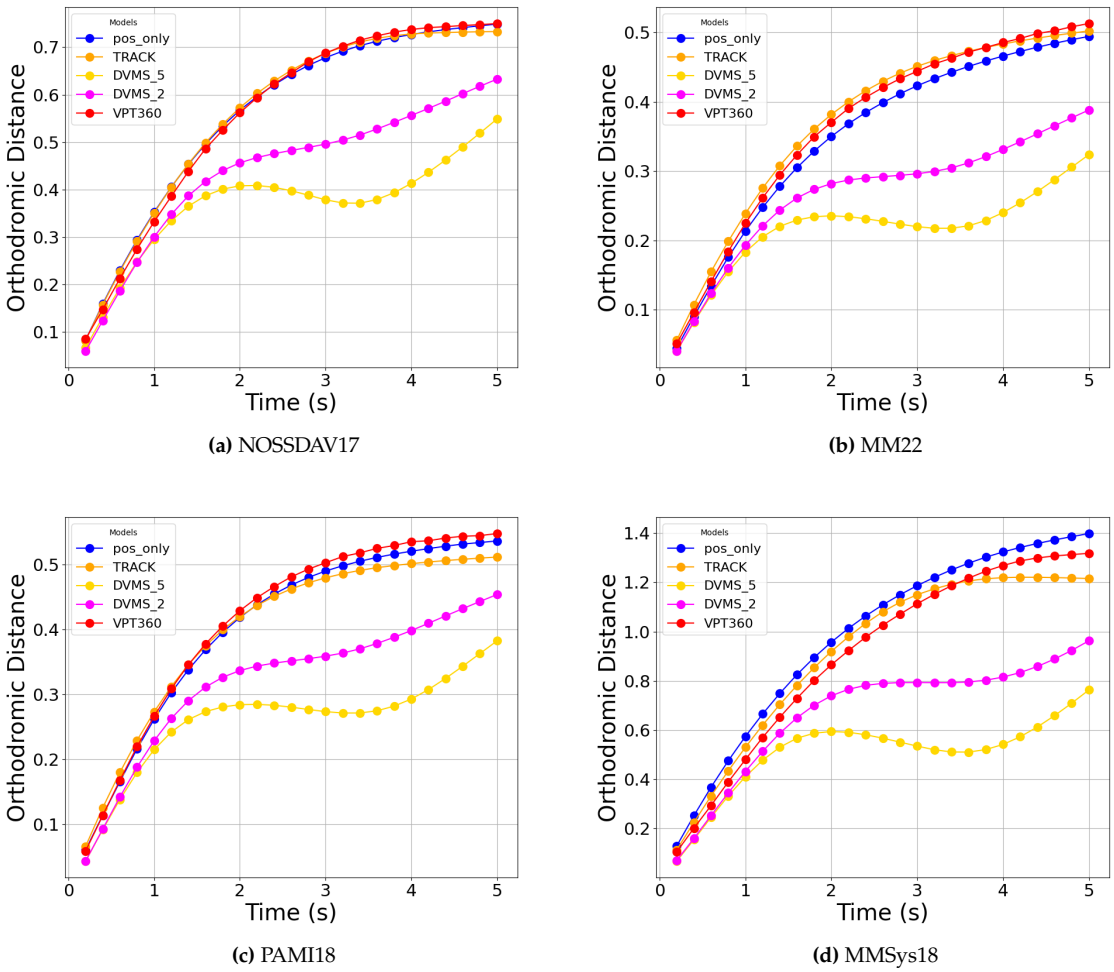
**(d)** MMSys18

**Figure 4.4:** Distribution of training and testing videos in (a) NOSSDAV17, (b) MM22, (c) PAMI18, and (d) MMSys18

our case, 0.6) of the videos in each group for the test set. This approach, known as stratified, uses a stratification key that categorizes the videos in each dataset into 4 groups based on the AE and SE of user trajectories: *Low SE* and *Low AE*, *Low SE* and *High AE*, *High SE* and *Low AE*, and *High SE* and *High AE*. The thresholds, determined by taking the median SE and AE for all videos in the dataset, are listed in Table 4.1 along with the mean AE and SE for the videos in the train and test set. Videos below the threshold are classified as low SE or low AE, while those above the threshold are categorized as high SE or high AE, respectively. This stratification ensures that both the train and test sets are representative of videos where users exhibit both low and high entropies, avoiding bias toward a particular type of video content. Additionally, we ensure that users in the training set do not overlap with those in the test set by using only 60% of users for training. Some earlier research, such as the Unified Evaluation Framework [55], repeated users across training and testing for different videos, but our approach ensures a stricter separation. The mean SE and mean AE plots of the training videos and testing videos can be seen in Figure 4.4.

| Dataset Name | mean AE Threshold | mean SE Threshold | Train Set | | Test Set | |
|---|---|---|---|---|---|---|
| | | | mean AE | mean SE | mean AE | mean SE |
| NOSSDAV17 | 1.01 | 79.74 | 0.97 | 76.08 | 1.01 | 76.97 |
| MM22 | 1.13 | 103.96 | 1.14 | 117.22 | 1.15 | 120.25 |
| PAMI18 | 1.88 | 71.93 | 1.92 | 74 | 1.92 | 73.44 |
| MMSys18 | 2.87 | 44.93 | 2.86 | 44.67 | 2.88 | 46.27 |

**Table 4.1:** AE and SE thresholds for different datasets.




**(a)** NOSSDAV17

**(b)** MM22

**(c)** PAMI18

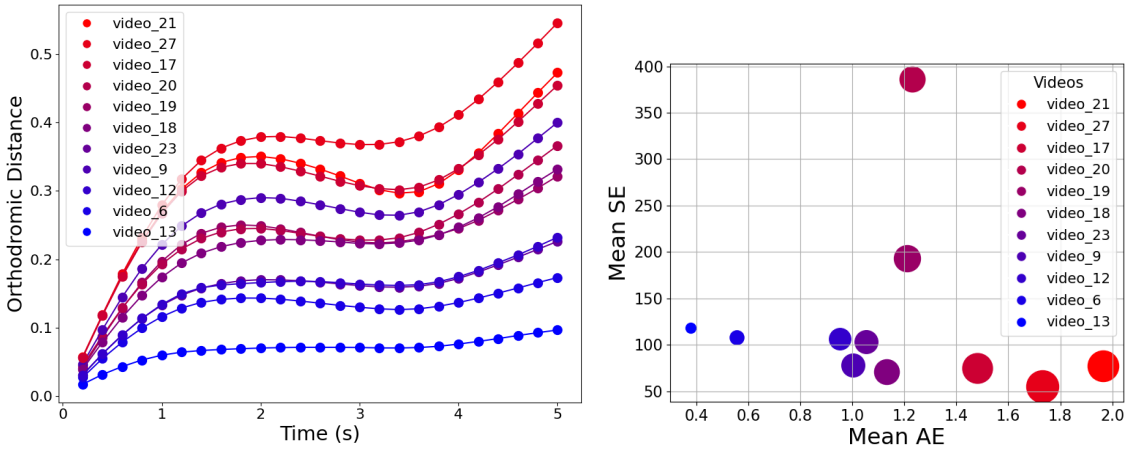
**(d)** MMSys18

**Figure 4.5:** Performance of position only baseline, TRACK, DVMS (with K=2 and K=5) and VPT360 on (a) NOSSDAV17, (b) MM22, (c) PAMI18, and (d) MMSys18 datasets

## 4.2. Comparing Existing models

We compare the performance of the following models (which have been discussed in more detail in Section 2.4.2):

- **Position-only baseline [55] :** LSTM-based model that takes the pitch and yaw as input.
- **TRACK [56] :** LSTM-based model that uses the unit vector representation of head




**(a)** Orthodromic Distance Loss per video

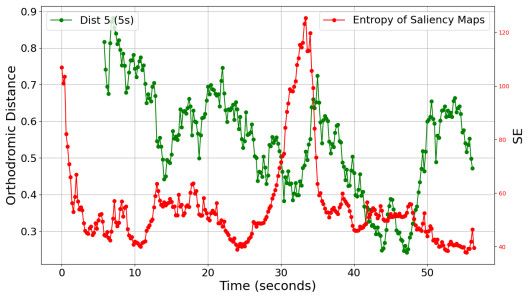
**(b)** Distribution of test videos

**Figure 4.6:** (a) Performance of DVMS K=5 on each video in the MM22 dataset, and (b) Mean SE and AE of test videos in MM22 dataset, the size of the marker represents the mean IE

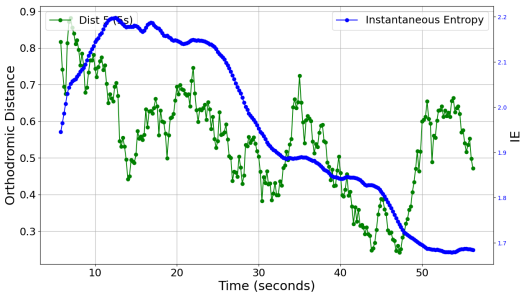
pose and processes content saliency information using an RNN before merging it with head pose data.

- **DVMS [24] :** LSTM-based model that introduces a latent variable, $z$ resulting in multiple predictions based on the dimensions of $z$, represented by K. We train two versions of DVMS, one trained on 2 predictions for each input (K=2), DVMS_2 and one trained on 5 predictions for each input (K=5), DVMS_5. DVMS_2 is chosen as it has the lowest training time while still utilizing multiple predictions, DVMS_5 is chosen as the model does not show significant improvements for more predictions.
- **VPT360 :** Multi-head attention-based model with no recurrent units.

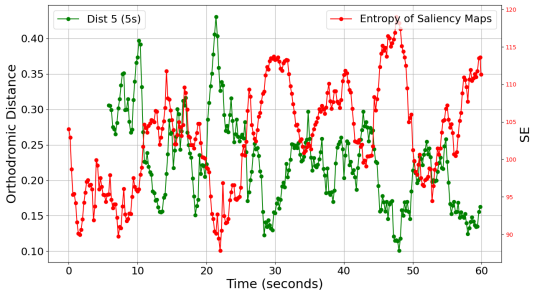
Each model is trained with an input window of 5 frames and an output window of 25 frames for 500 epochs, stopping early if no improvement is observed after 25 epochs. To evaluate model performance, we plot the average orthodromic distance between the true head pose and the predicted head pose across the entire 25-frame prediction horizon, as shown in Figure 4.5. Based on these results, we observe that DVMS outperforms TRACK and the position-only baseline across all three datasets. Notably, DVMS displays significantly better predictions for longer prediction horizons, highlighting the effectiveness of incorporating uncertainty into the forecasting model. Interestingly, DVMS shows similar loss values for predictions ranging from 2 to 4 seconds in the future, indicating that the model maintains consistency in its performance over these timeframes. We also note that all models tend to perform worse on the MMSys18 dataset, as the losses exceed 0.6 shortly after the first second. This reduced accuracy, combined with the higher mean entropy of user trajectories observed in the MMSys18 dataset (as shown in Table 4.1), can likely be attributed to the nature of the content being viewed. MMSys18 consists of shorter 20-second videos, where users typically display highly erratic trajectories at the beginning of the video as they familiarize themselves with the scene. This exploratory behavior increases the overall trajectory entropy, leading to greater unpredictability and consequently higher prediction errors.



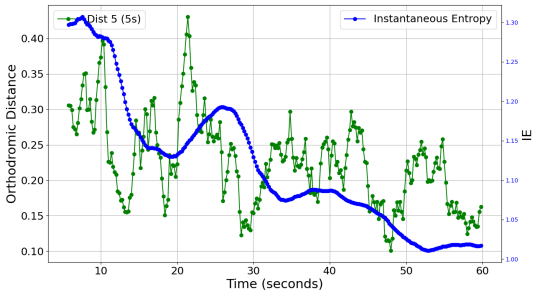

**(a)** Orthodromic distance vs SE for video_27 (Skydive)



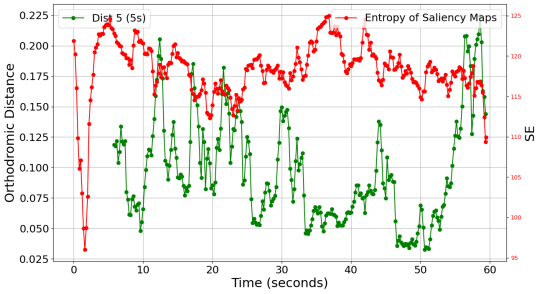

**(b)** Orthodromic distance vs IE for video_27 (Skydive)



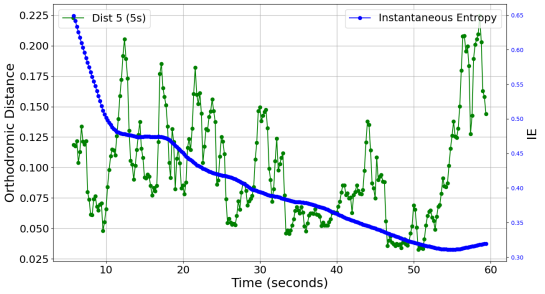

**(c)** Orthodromic distance vs SE for video_23 (chariot)




**(d)** Orthodromic distance vs IE for video_23 (chariot)




**(e)** Orthodromic distance vs SE for video_13 (mario)




**(f)** Orthodromic distance vs IE for video_13 (mario)

**Figure 4.7:** Performance of DVMS K=5 [24] for a prediction horizon of 5 seconds for *Skydive*, *Chariot* and *Mario* compared with the entropy of saliency maps (a,c,e) and the mean instantaneous entropy of users (b,d,f)

To obtain deeper insights into how entropy metrics affect the predictability of user trajectories, we take a closer look at the performance of DVMS, K=5, on each video of the MM22 dataset. Figure 4.6a shows the orthodromic loss for each video across the 25-frame prediction window, the videos are ordered in the legend from the highest mean AE at the top, to the lowest at the bottom. Figure 4.6b plots the videos based on SE and AE, and the size of each marker is based on the mean IE over the course of each video for all users. By comparing Figures 4.6a and 4.6b we can see that the model's accuracy tends to decrease for videos where users exhibit high actual entropy (AE). Specifically, we see that the videos with the highest mean AE such as *video_21*, *video_27*, *video_17* also exhibit higher loss values. This aligns with findings from Rossi et al. [59], which indicate that users exhibiting higher entropies are more challenging to predict.

To gain a better understanding of each video, we select 3 videos based on the mean

| Video | IE | | SE | |
|---|---|---|---|---|
| | **Correlation** | **p-value** | **Correlation** | **p-value** |
| *skydive* | 0.377 | $2.5 \times 10^{-15}$ | -0.22 | $3.3 \times 10^{-4}$ |
| *chariot* | 0.37 | $2.1 \times 10^{-18}$ | -0.74 | $2.7 \times 10^{-47}$ |
| *mario* | 0.16 | 0.01 | -0.373 | $1.38 \times 10^{-13}$ |

**Table 4.2:** Correlation and p-values for IE and SE with loss for different videos.

actual entropy: High AE (video_27, "skydive"), Medium AE (video_23, "chariot"), and Low AE (video_13, "mario"). We compare the average loss for DVMS (K=5) over the course of the entire video to the entropy of saliency maps and the instantaneous entropies, as shown in figure 4.7. While the relationship between IE and the loss is not too clear, we can observe that whenever the mean SE of the video increases, the loss decreases as seen by the peaks in SE corresponding to the valleys in the loss. This trend is most evident for *chariot*, as shown by Figure 4.7c.

Table 4.2 summarizes the correlations between the loss and the two entropy metrics (IE and SE) for the selected videos. For each video, we observe a positive correlation between the loss and IE, and a negative correlation between the loss and SE.

Based on our exploratory analysis, we note that the selected datasets provide a diverse range of 360° video content, offering a broad spectrum of user behaviors and exhibited entropies. This diversity is crucial for ensuring that the models generalize well across various types of content. Among the models evaluated, DVMS stands out as the most effective, consistently providing superior performance compared to others. On the other hand, TRACK and VPT360 perform comparably to the position-only baseline. Additionally, the analysis of entropy correlations offers valuable insights, revealing a negative correlation between AE and SE. The most notable finding is the positive correlation between prediction error and IE, and the negative correlation between prediction error and SE, suggesting that incorporating entropy metrics into head pose prediction models may be beneficial for improving their performance.

# 5

# Adaptive Attention for Head Pose Prediction

In this chapter, we propose two adaptive attention-based models, an LSTM-based architecture with an Entropy-based Adaptive Attention layer, and a multi-head attention-based architecture that uses adaptive attention. The goal of these adaptive attention-based models is to enhance head pose prediction by dynamically adjusting attention scores, focusing on the more predictable parts of the user's trajectory, as indicated by the entropy of user behavior. We also enrich 2 baseline models with entropy information for a complete comparison.

## 5.1. Entropy-based Adaptive Attention LSTM (E-AALSTM)

In the previous chapter, our exploratory analysis revealed a correlation between the entropy of user trajectories and their predictability. Building on this observation, we propose two attention-based models that incorporate an entropy-based modulating factor that dynamically adjusts attention scores based on the user's behavioral dynamics. By modulating the attention scores, these models dynamically allocate higher attention to more predictable and stable segments of the user's trajectory, prioritizing input regions that contribute most reliably to accurate predictions. Specifically, we propose an LSTM-based adaptive attention architecture, **E-AALSTM**, designed to enhance the attention mechanism's ability to handle varying levels of predictability, as measured by the instantaneous entropy. E-AALSTM modifies the standard sequence-to-sequence (seq2seq) encoder-decoder architecture by adding an adaptive attention layer, illustrated in Figure 5.1. This attention mechanism incorporates an entropy-based modulating factor aimed at selectively focusing on input segments that exhibit greater predictability, corresponding to lower values of instantaneous entropy. The LSTM-based architecture iteratively generates predictions for head pose across each time step in the output window. At each step, it calculates instantaneous entropy for the current prediction and uses this adjusted output to inform the head pose prediction at the following time step. By continuously adjusting attention based on these entropy measurements, E-AALSTM seeks to enhance predictions by focusing on areas of the trajectory where predictability is higher. In the following subsections, we formally define each part of

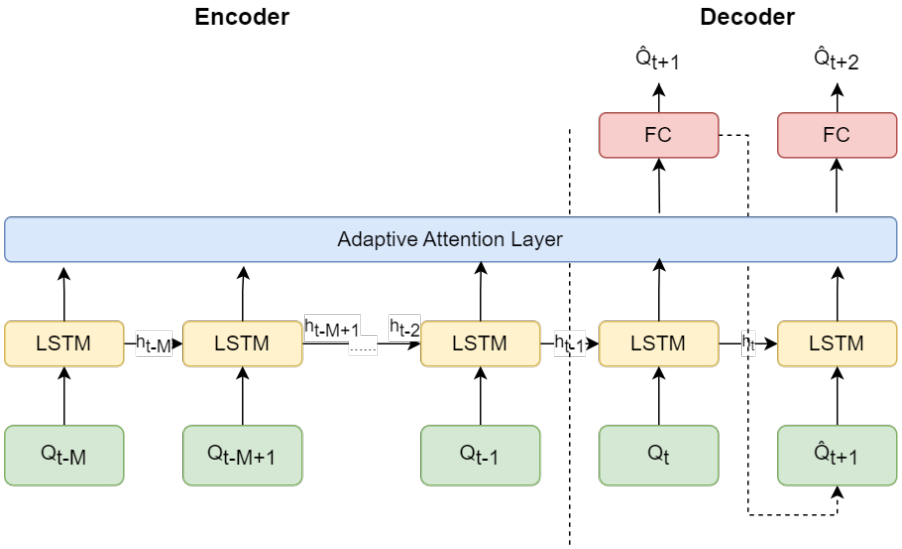



**Figure 5.1:** Seq2Seq architecture with attention layer

the model.

### 5.1.1. LSTM Layer

The **LSTM layer** processes the input sequence and generates hidden states $\mathbf{h}_t$ and cell states $\mathbf{c}_t$ at each time step $t$, as described by the following equations:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i x_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i) \tag{5.1a}$$
$$\mathbf{f}_t = \sigma(\mathbf{W}_f x_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f) \tag{5.1b}$$
$$\mathbf{o}_t = \sigma(\mathbf{W}_o x_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o) \tag{5.1c}$$
$$\mathbf{g}_t = \tanh(\mathbf{W}_g x_t + \mathbf{U}_g \mathbf{h}_{t-1} + \mathbf{b}_g) \tag{5.1d}$$
$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t \tag{5.1e}$$
$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \tag{5.1f}$$

Here, the input, $X = [x_1, x_2, \ldots, x_M]$ represents a subsequence of the trajectory, where $M$ is the input window size. $\sigma$ denotes the sigmoid function, and $\odot$ denotes element-wise multiplication. The hidden states serve as compressed representations of the input sequence and are used in subsequent attention calculations.

### 5.1.2. Adaptive Attention Layer:

The adaptive attention layer modifies the standard attention mechanism to incorporate the entropy values. This mechanism guides the model to focus on more predictable portions of the input while lowering the contribution of frames where user behavior is more erratic. We use the hidden states from the LSTM to generate the **Query, Key and Value Matrices**:

$$\mathbf{Q} = \mathbf{W}_q \mathbf{h}_t + \mathbf{b_q} \tag{5.2a}$$
$$\mathbf{K} = \mathbf{W}_k \mathbf{H} + \mathbf{b_k} \tag{5.2b}$$
$$\mathbf{V} = \mathbf{W}_v \mathbf{H} + \mathbf{b_v} \tag{5.2c}$$

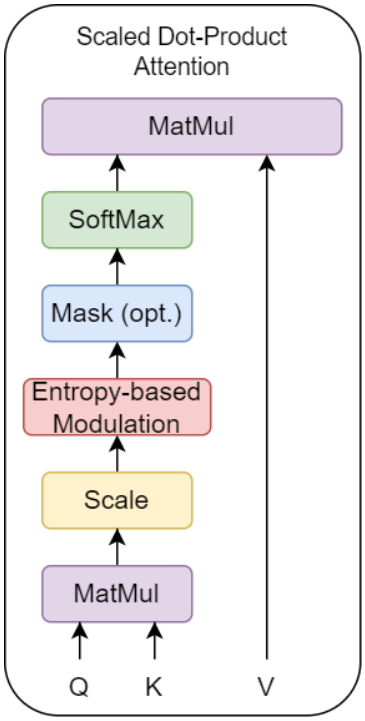



**Figure 5.2:** Entropy-adaptive scaled dot-product attention

where $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_M]$ represents all hidden states in the input window, and $h_t$ is the hidden state at the current timestep, $t$.

To obtain the modulated attention scores, we implement an adaptive version of scaled dot-product attention, as shown in Figure 5.2. The attention scores are first calculated using the scaled dot-product between the Query and Key matrices:

$$scores = \frac{\mathbf{QK}^T}{\sqrt{d_k}} \tag{5.3}$$

where $d_k$ is the dimensionality of the Key vectors. These scores represent the relevance of each timestep in the input sequence for predicting the output.

**Entropy-based Modulation:** To improve the model's ability to handle unpredictable user trajectories, we introduce an *entropy-based modulating factor*. This factor dynamically adjusts the attention scores based on the entropy of the user's trajectory at each timestep. For each timestep $t$, we calculate the instantaneous entropy, $\mathrm{IE}_t$. We use this entropy to obtain a modulating factor:

$$\mathbf{M}_t = \exp(-\mathbf{W_s} \cdot \mathrm{IE}_t) \tag{5.4}$$

where $W_s$ are learnable weights that adjust the influence of the entropy on the attention mechanism. The attention scores are then modulated by this factor and the adaptive attention weights are computed by using a softmax function:

$$scores_{mod} = \mathbf{M}_t \cdot (scores) \tag{5.5a}$$

$$AW_{adpt} = softmax(scores_{mod}) \tag{5.5b}$$

Finally, we compute the context vector using these adaptive attention weights:

$$c_t^{\text{adaptive}} = AW_{adpt} \cdot \mathbf{V} \tag{5.6}$$

This context vector is then passed as an input to the linear layers to generate the final head pose predictions.

**Saliency-based Modulation:** As an alternative, to minimize computations during runtime, we propose an alternative modulating factor that utilizes the entropy of saliency maps instead of instantaneous entropy. Since we noted a negative correlation between the entropy of saliency maps and trajectory entropy, we use tanh-based scaling instead of exponential decay.

$$\mathbf{M}_t = \tanh(\mathbf{W_s} \cdot \mathrm{SE}_t) \tag{5.7}$$

Here, $SE_t$ represents the entropy of the saliency map of the frame at timestep $t$.

## 5.2. Multi-head Adaptive Attention (AMH)

We also propose a variation of the multi-head attention architecture inspired by the VPT360 model proposed by Chao et al. [9]. In this model, we incorporate the entropy-based modulation factor (Equation 5.4) into each head's attention scores. The key difference between the incorporation of entropy in this approach and its usage in E-AALSTM lies in how the entropy values are leveraged. The transformer-based model only uses entropy values from the input window, constraining the model to pre-existing information. In contrast, the seq2seq model continually measures IE using the predicted head pose at each timestep to dynamically adapt and improve future predictions in the output window. Thus, this transformer-based approach tests a fixed entropy-informed modulation, while the seq2seq model enables iterative adaptation for more dynamic prediction refinement. Additionally, since AMH predicts all timesteps in the output window simultaneously, it requires fewer computations compared to the sequential processing of E-AALSTM, making it more computationally efficient. The architecture of AMH, shown in Figure 5.3, closely follows the transformer-based VPT360 architecture. In the following subsections, we describe the AMH architecture in more detail:

### 5.2.1. Input Embedding

We generate input embeddings by passing the input sequence through a linear layer to create vector representations of the input:

$$\mathbf{emb_X} = \mathbf{W_1 X} + \mathbf{b_1} \tag{5.8}$$

Here, the input, $X = [x_1, x_2, \ldots, x_M]$ represents a subsequence of the trajectory where $M$ is the input window size.

### 5.2.2. Positional Embedding

Since the AMH model does not use a recurrent unit to capture temporal features, we use positional embedding to incorporate relative position information of the elements

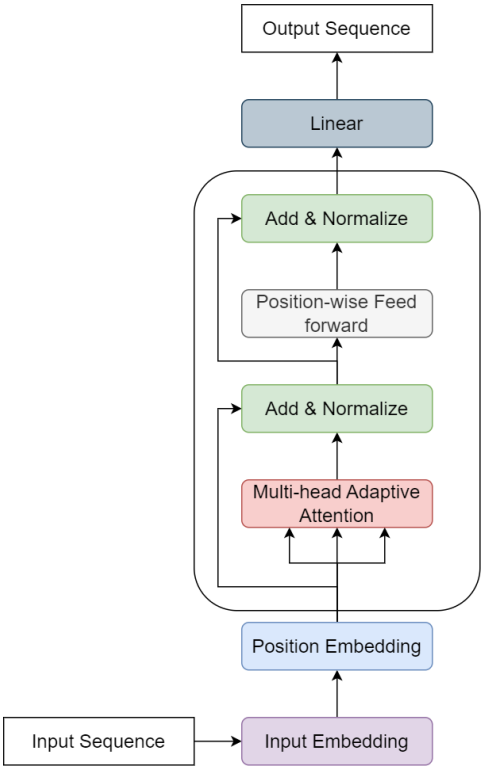



**Figure 5.3:** Architecture of proposed AMH model

of the input sequence. We implement the positional embedding used in [72] and use the summation of the input sequence with sine and cosine functions of different frequencies to obtain our final embeddings.

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{model}}\right) \tag{5.9a}$$

$$PE_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d_{model}}\right) \tag{5.9b}$$

$$\mathbf{e_X} = \mathbf{emb_X} + PE_{pos} \tag{5.9c}$$

where $pos$ is the position, $i$ is the dimension and $d_model$ is size of the hidden layer.

### 5.2.3. Multi-Head Adaptive Attention Layer

We compute the Query, Key, and Value matrices using the embeddings:

$$\mathbf{Q} = \mathbf{W}_q\mathbf{e_X} + \mathbf{b_q} \tag{5.10a}$$

$$\mathbf{K} = \mathbf{W}_k\mathbf{e_X} + \mathbf{b_k} \tag{5.10b}$$

$$\mathbf{V} = \mathbf{W}_v\mathbf{e_X} + \mathbf{b_v} \tag{5.10c}$$

After transformation, the $Q$, $K$, and $V$ matrices are split into multiple heads, enabling the model to learn from different aspects of the embedding space.

**Attention calculation:** For each head, we calculate the modulated attention score following the entropy-adaptive scaled dot-product described in Equations 5.3, 5.4, 5.5, and 5.6. The final output of the multi-head attention, $\mathbf{c}^{\text{adaptive}}$, is produced by concatenating the output of each head. This output is then added back to the

embeddings and normalized to form an enriched representation:

$$\mathbf{e}'_{\mathbf{X}} = \mathbf{e}_{\mathbf{X}} + \mathbf{c}^{\text{adaptive}} \tag{5.11}$$

This combined embedding now incorporates entropy-adaptive attention, which encourages the model's ability to adjust attention weights based on the predictability of head movements in the input sequence, as measured by the instantaneous entropy.

### 5.2.4. Position-wise Feed-Forward Network

After the attention weights are calculated and added to the embeddings, they are passed through a fully connected feed-forward network (FFN) to consider interactions between dimensions:

$$\text{FFN}(\mathbf{e}'_{\mathbf{X}}) = \text{ReLU}(\mathbf{e}'_{\mathbf{X}} W_1 + b_1) W_2 + b_2 \tag{5.12}$$

The output of the position-wise FFN is added to $e'_x$ and normalized. The predicted head pose sequence is then produced by passing this output through a final linear layer.

# 6

# Evaluation and Results

In this chapter, we discuss the experimental setup including the models being compared, the hyperparameters, and the train-test split. We also go over the criteria used to evaluate the proposed solutions. Finally, we present the results of our experiments.

## 6.1. Experimental Setup

In this section, we describe the experimental setup used to evaluate the performance of the proposed models. We begin by discussing two enhanced baselines that incorporate entropy information, followed by the baseline benchmarks, ablated versions of each model, and the state-of-the-art models used for comparison. We then outline the hyperparameter configurations, train-test splitting of the selected datasets, and the evaluation metric used to compare the models. This setup is designed to provide consistent and comparable conditions across all models, isolating the effect of entropy-based modulation and other architectural variations.

### 6.1.1. Entropy Enhanced Baselines

For a complete comparison, we enrich the position only seq2seq encoder-decoder model [56], described in Section 2.4.2. To do so, we consider 2 approaches:

- **Position-only baseline augmented with entropy information (pos-augmented)**: In this approach, we augment the head pose information by appending the user's instantaneous entropy at that instant to the head pose information, and pass them as an input to the position-only baseline [56]. This approach will act as our baseline model for a straightforward incorporation of entropy.
- **Position-only baseline with entropy-weighted loss (pos-weighted)**: In this approach, we use the position-only baseline architecture [56], but we modify the loss function to more heavily penalize incorrect predictions for frames where the user's trajectory exhibits a higher entropy. Specifically, we use a weighted loss function. Let $L_{orth}$ be the orthodromic distance loss, $\alpha$ be a positive coefficient and $AE_t$ be IE at time t. We can now represent the loss as:

$$L_t = IE_t(1 + \alpha) \times L_{orth} \tag{6.1}$$

These enhanced baselines help us investigate how different methods of adding entropy, either as an input feature (pos-augmented) or through a weighted loss function (pos-weighted), influence the model's performance.

### 6.1.2. Benchmarks and Comparison Models

To evaluate the performance of our proposed models, we use a diverse set of models for comparison, including baseline benchmarks, ablated versions, and state-of-the-art (SOTA) methods from previous works. This comparison helps us understand the impact of entropy modulation and evaluate the proposed models in the context of existing approaches.

**Models used for ablation (No entropy information):** We first evaluate the entropy-based models and compare them to equivalent models that do not incorporate entropy information. The equivalent models for E-AALSTM, AMH, and the two enriched baselines are listed as follows:

- **Attention-based LSTM (ALSTM):** This is an ablated version of the proposed E-AALSTM model, where the entropy-based modulation layer is removed. In this case, the attention mechanism operates without the dynamic adjustment based on entropy, allowing us to see the performance difference when the entropy metrics are not considered.
- **VPT360 [9]:** Since AMH adds the entropy-based modulation factor to the multi-head attention architecture of VPT360, VPT360 is effectively the ablated version of AMH. Comparing AMH and VPT360 shows us how the model performs when entropy-based modulation is disabled.
- **Position-only Baseline [56]:** This baseline model predicts head pose based solely on positional data using a seq2seq encoder-decoder architecture. It serves as a basic point of comparison to measure the effectiveness of incorporating more complex mechanisms like attention and entropy modulation. It also acts as an ablated version of pos-augmented and pos-weighted.

**State-of-the-art Models used for comparison:** After evaluating the ablated models, we compare the best-performing entropy-based model against a range of state-of-the-art (SOTA) methods. These models, representing current best practices in head pose prediction, offer a valuable context for understanding the relative effectiveness of incorporating entropy information:

- **Position-only Baseline [56]:** Since many state-of-the-art models use modified versions of the seq2seq encoder-decoder architecture, we use the position-only baseline as a benchmark.
- **VPT360 [9]:** We include the VPT360 model in the comparison to evaluate the stability of the predictions of our proposed models to the existing state-of-the-art.
- **TRACK [56]:** A state-of-the-art trajectory prediction model that leverages saliency maps to predict head-pose using a modified seq2seq encoder-decoder architecture.
- **DVMS [24]:** An LSTM-based seq2seq model that introduces a latent variable, $z$,

allowing it to generate multiple predictions based on the dimensions of $z$, $K$. We compare two versions of DVMS:

- **DVMS_2:** A version trained with 2 predictions for each input ($K = 2$), chosen due to its lower training time while still utilizing multiple predictions.
- **DVMS_5:** A version trained with 5 predictions for each input ($K = 5$, included as a reference for a higher number of predictions, and because increasing K further does not show any significant improvements.

### 6.1.3. Hyperparameter Settings

To ensure consistency, we standardized hyperparameter configurations across all models where possible. The following hyperparameters were selected based on training times and previous literature:

- **Learning Rate:** 0.0001 for all models, chosen for stability across architectures.
- **Optimizer:** AdamW [41] optimizer with a weight decay of 0.01
- **Epochs:** A maximum of 500 epochs with early stopping if the loss does not improve for 25 consecutive epochs.
- **Batch Size:** 128, to balance computational efficiency and training stability.
- **Model-specific hyperparameters:**
  - **Dropout Rate:** 0.1 applied in attention and LSTM layers
  - **Hidden Layer Size:** 512 for all LSTM and Transformer layers to maintain consistent model complexity.
  - **Entropy Modulation Factor:** Weights for entropy modulation are initialized randomly from a normal distribution, $\mathcal{N}(0, 0.1^2)$.

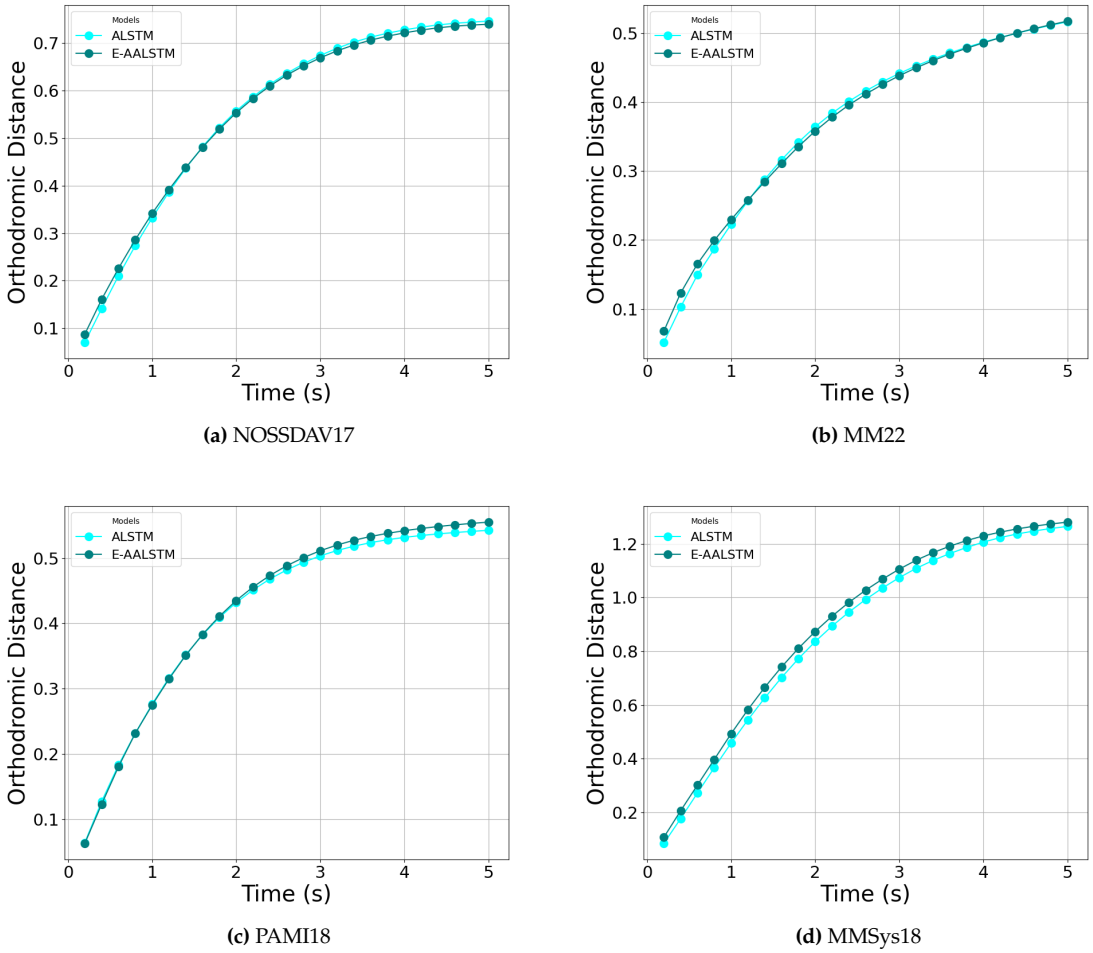
These settings were applied consistently across benchmarks, ablated models, and state-of-the-art models to ensure that differences in performance are solely due to architectural differences.

### 6.1.4. Train-test Split

We evaluate our models on the four datasets discussed in Chapter 4, namely: **NOSS-DAV17** [17], **MM22** [35], **PAMI18** [78] and **MMSys18** [14]. We follow the same train-test splitting approach utilized for our initial evaluation of the state-of-the-art models (described in Section 4.1.1). Specifically, we perform a stratified split based on the mean entropy of saliency maps and the mean actual entropy of user trajectories, to ensure both training and testing data are representative of the overall dataset. We perform a 60-40 split for the videos and a 60-40 split over the users ensuring that the models are evaluated on data (both users and videos) that was not included in the training process.

### 6.1.5. Evaluation Metric

To assess the performance of the models, we utilize the Orthodromic distance between the predicted head pose, $\hat{P}_t$, and the viewer's true head pose, $P_t$. Orthodromic distance




**(a)** NOSSDAV17

**(b)** MM22

**(c)** PAMI18

**(d)** MMSys18

**Figure 6.1:** Performance comparison between E-AALSTM and ALSTM on (a) NOSSDAV17, (b) MM22, (c) PAMI18, and (d) MMSys18 datasets.

is calculated using equation 3.1:

$$O(\hat{P}_t, P_t) = \arccos \hat{P}_t \cdot P_t$$

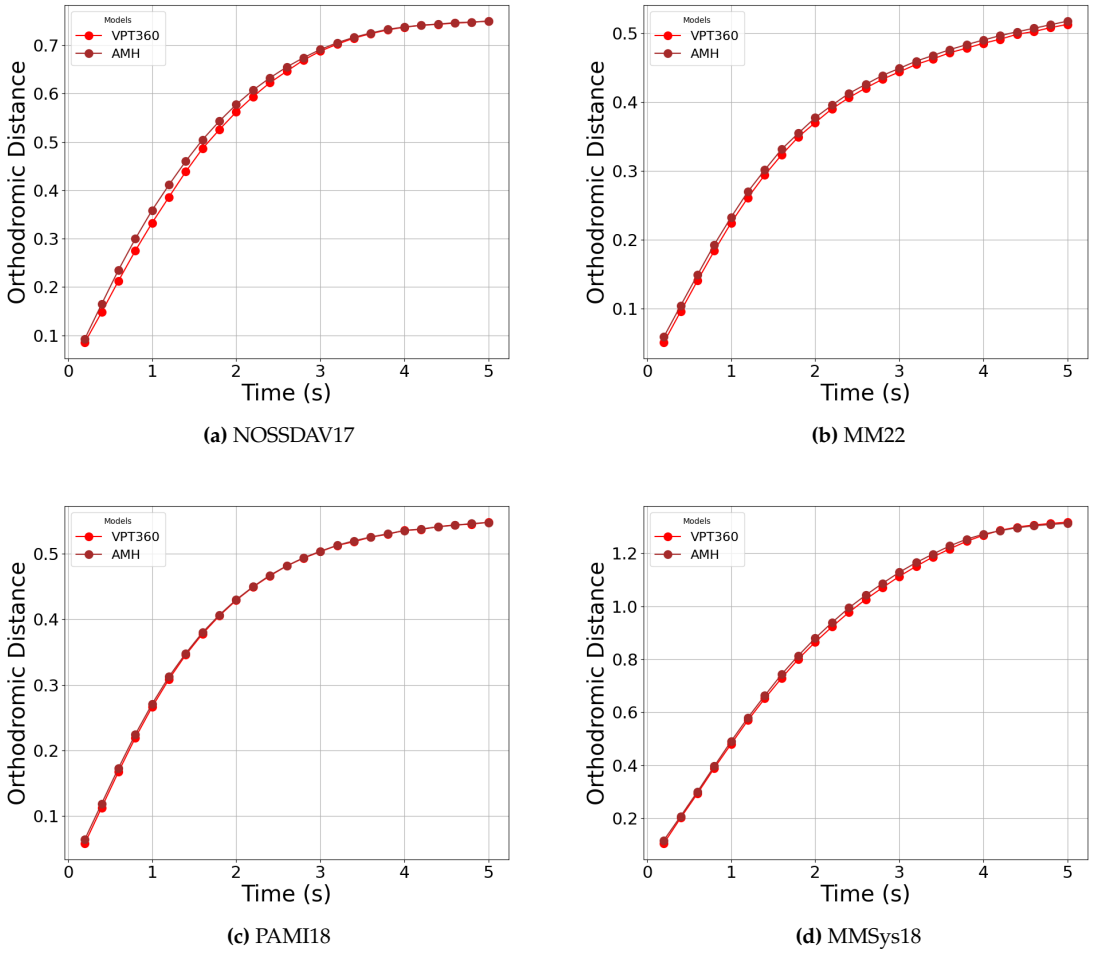
We compare the Orthodromic distance for each time step to get a better idea of how each model performs over a 5-second prediction horizon.

## 6.2. Results

In this section, we compare the proposed models with their respective no-entropy benchmarks and then compare the best proposed model with the current state-of-the-art.

### 6.2.1. Model Evaluation: Entropy vs No-Entropy Models

By comparing each proposed model alongside a baseline that excludes entropy, we can assess the direct effect of incorporating entropy on performance over the prediction horizon. This analysis also helps streamline the final comparison with the state-of-the-art, as we can highlight the best entropy-based model without repeating evaluations




**(a)** NOSSDAV17

**(b)** MM22

**(c)** PAMI18

**(d)** MMSys18

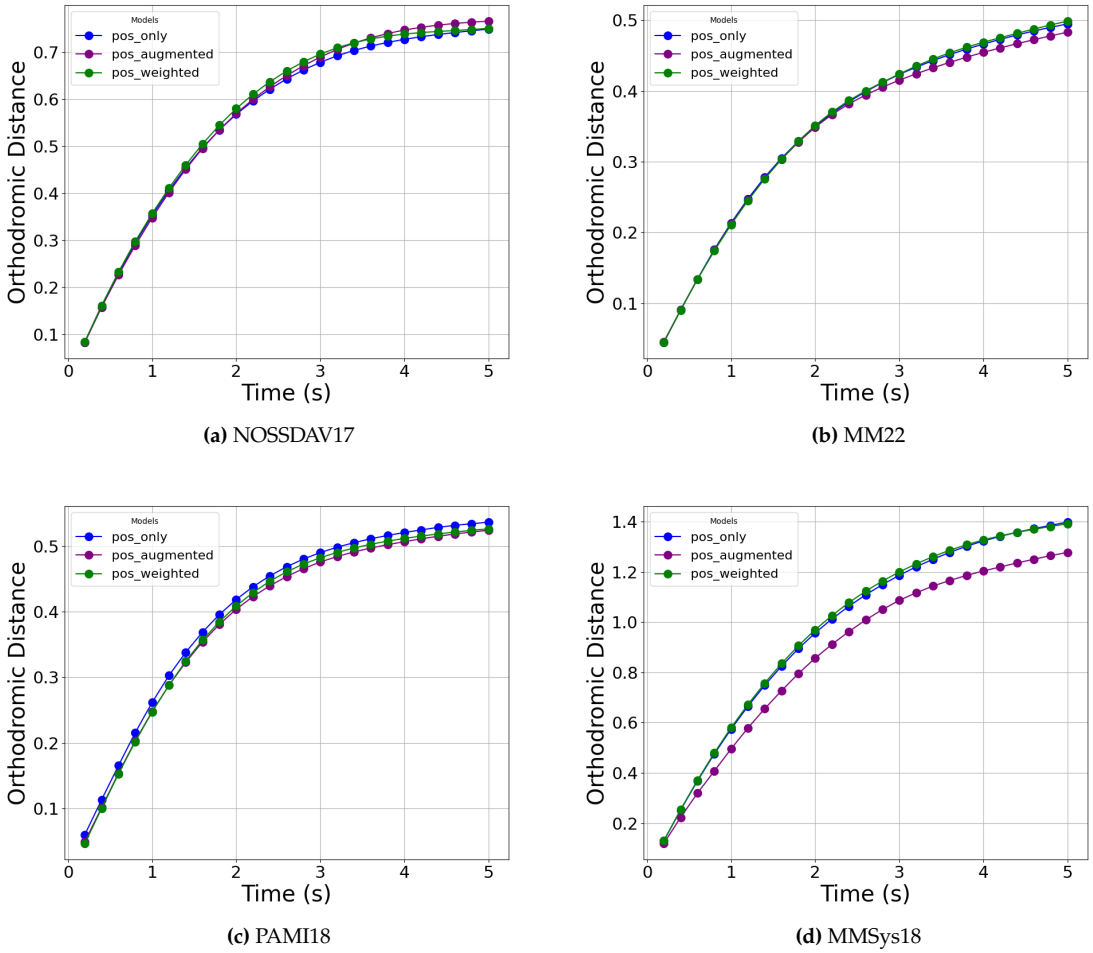
**Figure 6.2:** Performance comparison between AMH and VPT360 on (a) NOSSDAV17, (b) MM22, (c) PAMI18, and (d) MMSys18 datasets.

across less effective variations.

Figures 6.1, 6.2, and 6.3 illustrate the performance comparison across different datasets. Each figure contains four subplots, one for each dataset, i.e. NOSSDAV17 [17], MM22 [35], PAMI18 [78], and MMSys18 [14]. These subplots depict the average loss for all predictions across the videos of the dataset at each timestep up to the 5-second prediction horizon. Specifically, we compare the following models:

- Our proposed **E-AALSTM** versus its benchmark, **ALSTM** in Figure 6.1.
- Our proposed **AMH** versus its benchmark, **VPT360** in Figure 6.2.
- Our proposed **pos-augmented** and **pos-weighted** against their benchmark, **position-only** baseline in Figure 6.3.

In Figure 6.1 we observe that our proposed **E-AALSTM** performs similarly to its corresponding benchmark, **ALSTM**. E-AALSTM has a slightly lower accuracy for all timesteps across the entire prediction window for the MMSys18 dataset. These results suggest that the adaptive attention architecture may not effectively leverage trajectory




**(a)** NOSSDAV17

**(b)** MM22

**(c)** PAMI18

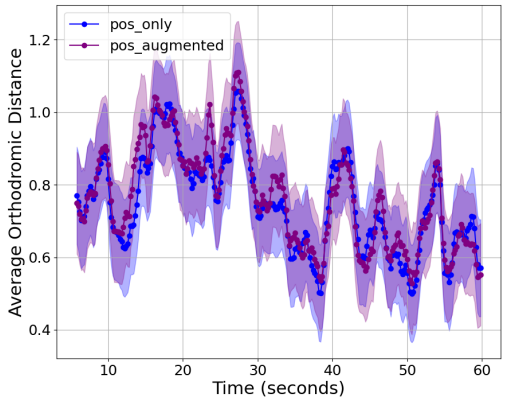
**(d)** MMSys18

**Figure 6.3:** Performance comparison between pos-only, pos-augmented, and pos-weighted on (a) NOSSDAV17, (b) MM22, (c) PAMI18, and (d) MMSys18 datasets.

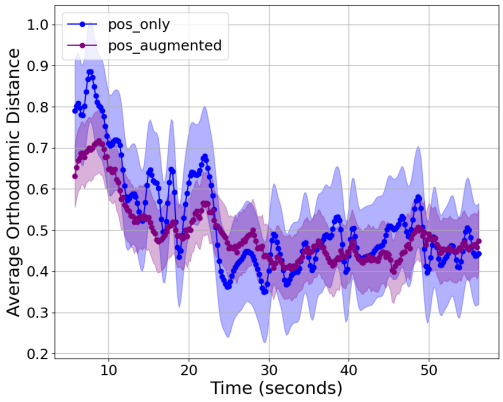
entropy metrics. The MMSys18 dataset, with its shorter videos, tends to feature higher trajectory entropy due to users rapidly exploring their environment at the start. By down-weighting high-entropy patterns, the model might inadvertently suppress information critical for these scenarios, resulting in reduced predictive performance.

We observe a similar trend when we compare the results for **AMH** and **VPT360**, as shown in Figure 6.2. The losses for these models are even more similar, possibly because the entropy metrics are only considered for the input window. These results further suggest that our proposed entropy-based adaptive attention layer might not effectively incorporate entropy information.

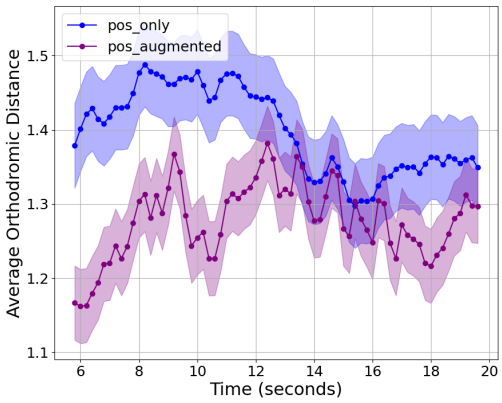
Lastly, in Figure 6.3, we compare the results for more straightforward integrations of entropy metrics, **pos-augmented** and **pos-weighted**, to the **position-only** baseline. While the position-only model with weighted loss does not show any notable differences from the position-only baseline, the pos-augmented model shows slightly better predictions beyond a 2-second window for the PAMI18 and MM22 dataset and notably better predictions as the prediction window increases for the MMSys18 dataset. The four plots also reveal a notable trend: the position-augmented model displays enhanced




**(a)** NOSSDAV17




**(b)** MM22

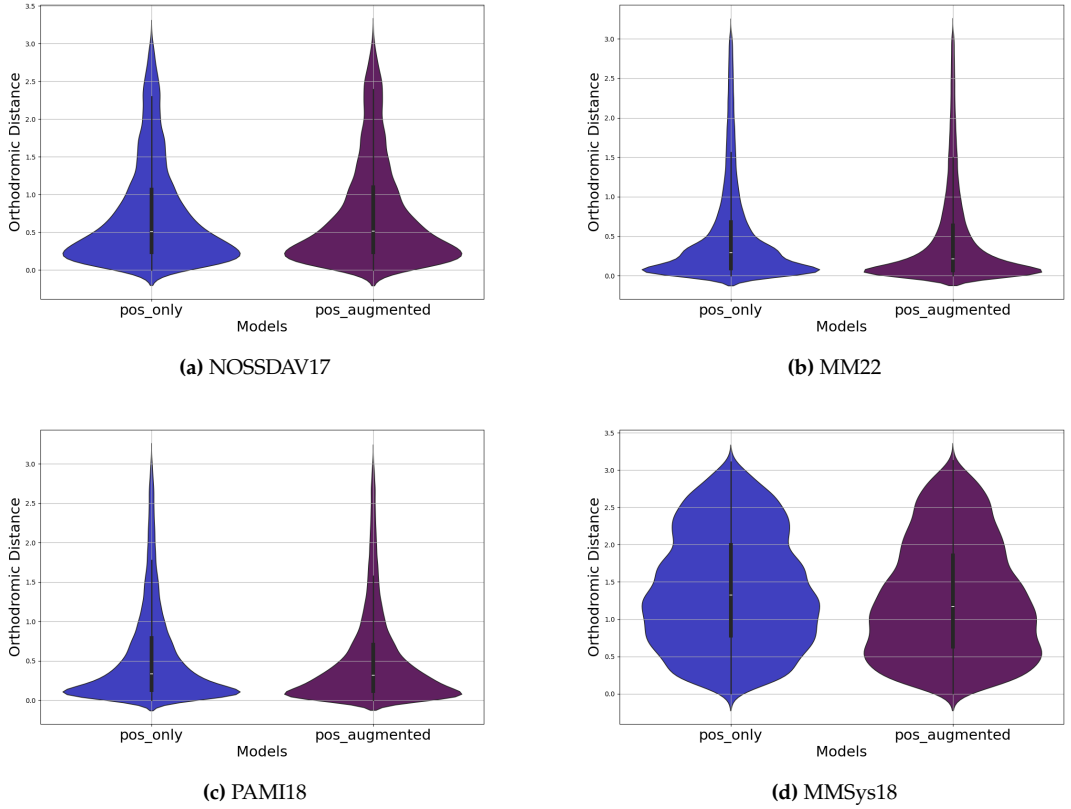



**(c)** MMSys18

**Figure 6.4:** Average prediction loss at each timestep for pos-only, and pos-augmented on (a) NOSSDAV17, (b) MM22, and (c) MMSys18 datasets, based on the predicted head pose 5 seconds after the input window.

performance over the position-only model as the mean AE exhibited by users in the testing videos increases. For the NOSSDAV17 dataset, which has the lowest mean AE of 1.01, the augmented model performs slightly worse for predictions beyond the 2.5-second window. In contrast, for the MMSys18 dataset which is characterized by a higher mean AE of 2.88, the pos-augmented model demonstrates significantly improved accuracy compared to the position-only baseline. We note again that all models tend to perform worse on the MMSys18 dataset, with losses exceeding 0.6 shortly after the first second, whereas, for other datasets, the mean error remains below 0.5–0.7 even at the end of the prediction horizon (5 seconds). This further confirms that the higher entropy in user trajectories, typical of shorter videos, introduces greater unpredictability, making accurate head pose prediction more challenging.

Given the above observation, we further compare the position-only baseline and the pos-augmented approach by analyzing the average orthodromic distance between the predicted head pose and the true head pose. Specifically, we average the prediction error 5 seconds after the input window, for each time step, across all test videos in the

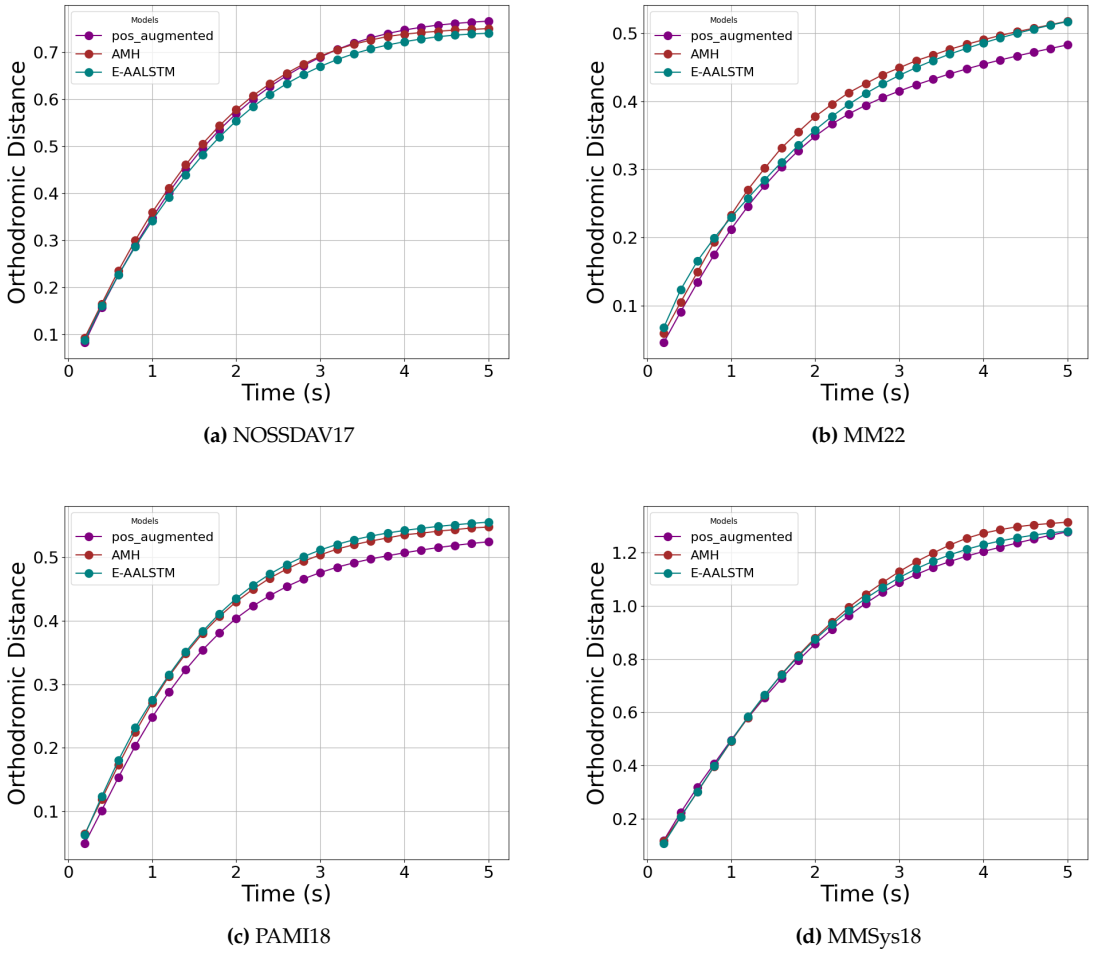



**(a)** NOSSDAV17 **(b)** MM22

**(c)** PAMI18 **(d)** MMSys18

**Figure 6.5:** Violin plots of Orthodromic distance for pos-only and pos-augmented on (a) NOSSDAV17, (b) MM22, (c) PAMI18, and (d) MMSys18 datasets

NOSSDAV17, MM22, and MMSys18 datasets. Additionally, we compute the standard deviation at each timestep to study the stability of the models. The pos-weighted model is excluded from the plots as its performance closely mirrors that of the position-only baseline. This analysis is illustrated in Figure 6.4. The PAMI18 dataset is excluded from this analysis because of the large variance in video lengths. The results for the NOSSDAV17 and MMSys18 datasets are in line with the previous results in Figure 6.3, indicating that the model augmented with viewer entropy data provides better accuracy for the MMSys18 dataset. This improved accuracy in the MMSys18 dataset is even more pronounced in the first half of the videos, suggesting that instantaneous entropy provides more valuable information early in the session when the user is exploring the environment and exhibiting more unpredictable behavior. Interestingly, although the pos-augmented model exhibits a slightly improved average orthodromic distance of 0.49 compared to 0.53 for the position-only model in the MM22 dataset, it shows significantly greater stability with a much lower variance in the results. To quantify this stability, we compute the coefficient of variation (CV) for the MM22 dataset, defined as the ratio of the standard deviation of the loss to its mean. The CV allows us to compare the stability of the models' performance independently of the overall error magnitude, as it normalizes the variance relative to the mean. The position-augmented approach yields a CV of 15.4%, whereas the position-only baseline has a CV of 23.4%. The lower CV indicates that the predictions of the pos-augmented model are less variable, meaning it produces more stable and reliable results. This




**(a)** NOSSDAV17

**(b)** MM22

**(c)** PAMI18

**(d)** MMSys18

**Figure 6.6:** Performance of pos-augmented, E-AALSTM, and AMH on (a) NOSSDAV17, (b) MM22, (c) PAMI18, and (d) MMSys18 datasets

improvement in consistency may be due to the incorporation of entropy-based features, which help the model account for unpredictability and variability in user behavior.

To further understand the distribution of losses for the last prediction step (5 seconds in the future), we plot violin plots of the orthodromic distance errors for the pos-augmented and pos-only models. This analysis highlights the spread of errors at the 5-second mark, offering insight into how the models' prediction errors are distributed and whether one model demonstrates more consistent performance over time. In Figure 6.5, for the NOSSDAV17 and PAMI18 datasets, the violin plots show nearly identical distributions, with minimal differences in the spread or central tendency of errors. For the MM22 dataset, the pos-only model shows a distinct, smaller secondary band in error distribution above the main cluster of losses. By contrast, the pos-augmented model reduces these higher error cases, as shown by the single, lower-density cluster of errors. This suggests that the entropy-based augmentation helps reduce the frequency of large deviations in final predictions, offering more stable performance. For MMSys18, where viewer trajectories have the highest mean entropy (2.88 AE), both models show broader distributions, reflecting high variability

in prediction errors. However, the pos-augmented model exhibits a skew towards lower error values compared to pos-only, which has a more balanced spread across the range of losses. This skew suggests that the pos-augmented approach may help the model reduce high-error predictions even when users exhibit highly unpredictable behavior, while pos-only remains vulnerable to a wider range of prediction errors.

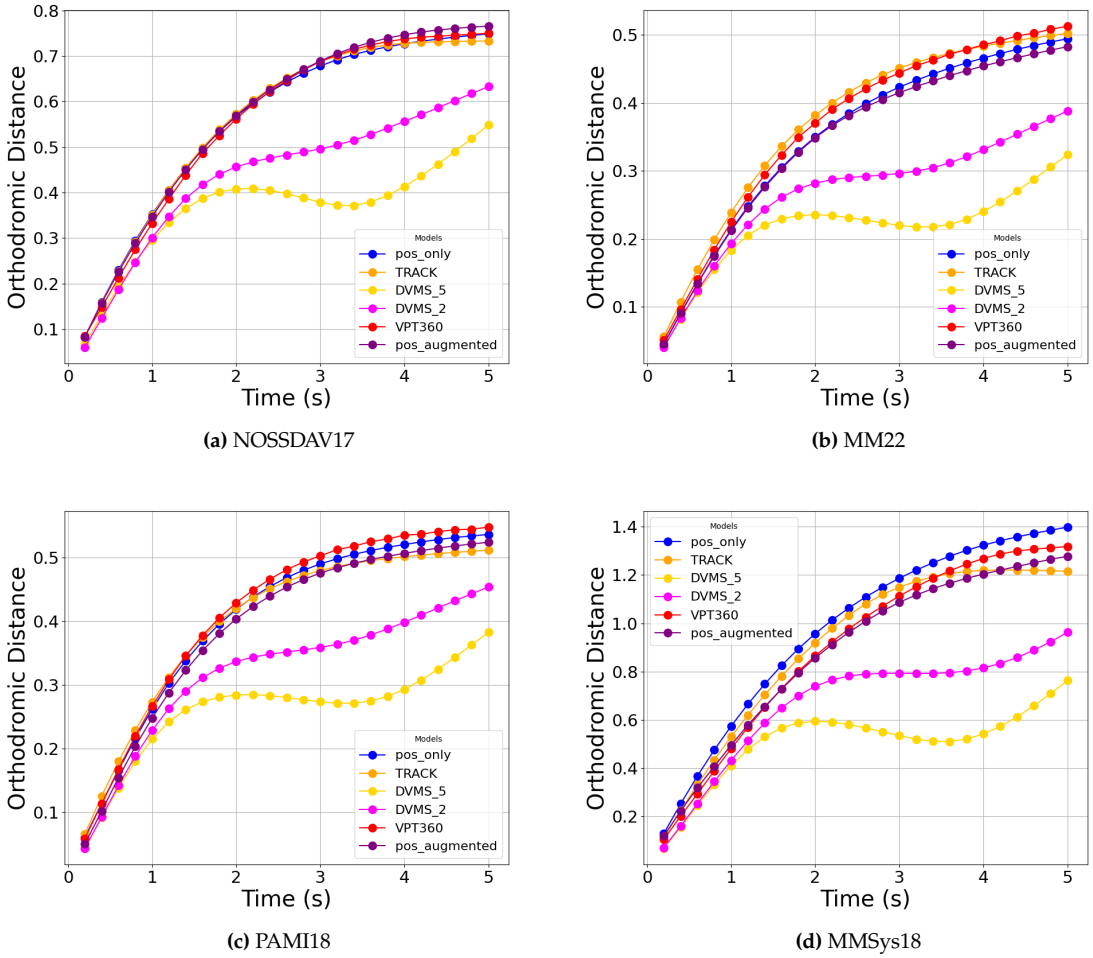
These results suggest that the pos-augmented model is able to better leverage head pose trajectory entropy information, in particular for data where users exhibit a higher degree of unpredictability, measured by the entropy of their trajectories, as seen in the case of the MMSys18 dataset. They also suggest that it is able to provide more stable predictions for the MM22 dataset.

Finally, to determine the best-performing proposed model for comparison with the existing state-of-the-art, we evaluate the performance of E-AALSTM, AMH, and pos-augmented. Figure 6.6 presents the error curves for each model on the NOSSDAV17 [17], MM22 [35], PAMI18 [78], and MMSys18 [14] datasets. While all models demonstrate comparable performance, the pos-augmented model modestly outperforms E-AALSTM and AMH across all datasets except NOSSDAV17. Consequently, we select the pos-augmented model for comparison with current state-of-the-art methods.

### 6.2.2. Comparison to the State-of-The-Art

In this section, we evaluate the performance of the pos-augmented model against the current state-of-the-art models, discussed in 6.1.2, i.e., **pos-only**, **TRACK**, **DVMS_5** and **DVMS_2**, and **VPT360**. We compare the orthodromic distance over a 5-second prediction horizon, averaged across all videos, to determine which models maintain lower predictive error. Figures 6.7 show the orthodromic distance curves for each model across the four datasets: NOSSDAV17 [17], MM22 [35], PAMI18 [78], and MMSys18 [14]. In all plots, DVMS_5 consistently achieves the lowest orthodromic distance, outperforming the other models at most time intervals. This demonstrates that the multiple prediction approach of DVMS, which generates multiple predictions during training and selects the best result, is more effective at capturing head pose trajectory dynamics. By incorporating uncertainty and refining predictions based on this approach, DVMS is able to provide more accurate and robust forecasts compared to the other models. For the pos-augmented model, we observe a slight underperformance toward the end of the prediction horizon for the NOSSDAV17 dataset, where user trajectories tend to exhibit the lowest actual entropy. Notably, pos-augmented achieves slightly better short-term predictions compared to TRACK, up to 3 seconds on PAMI18 and up to 4 seconds on MMSys18. Additionally, both seq2seq models (pos-only and pos-augmented) outperform VPT360 and TRACK on the MM22 dataset, indicating the suitability of the architecture for this dataset.

The results for the models at two key time points (2.6 seconds and 5 seconds into the prediction horizon) are summarized in Table 6.1. The values in the table represent the average orthodromic distance (error) at 2.6 seconds and 5 seconds into the prediction, which reflects how well each model predicts the user's head pose in the short-term and the long-term respectively. Lower values indicate better performance, meaning the predicted head pose is closer to the true head pose. While the table confirms our findings that both DVMS models consistently outperform the other models, it




**(a)** NOSSDAV17

**(b)** MM22

**(c)** PAMI18

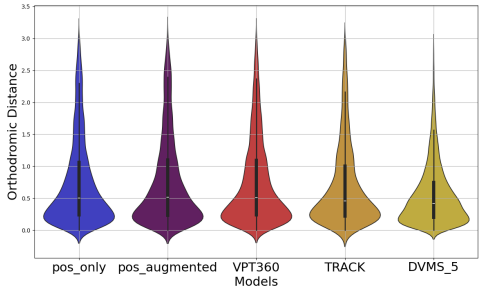
**(d)** MMSys18

**Figure 6.7:** Performance of pos-only, VPT360, TRACK, DVMS-5, and pos-augmented (a) NOSSDAV17, (b) MM22, (c) PAMI18, and (d) MMSys18 datasets

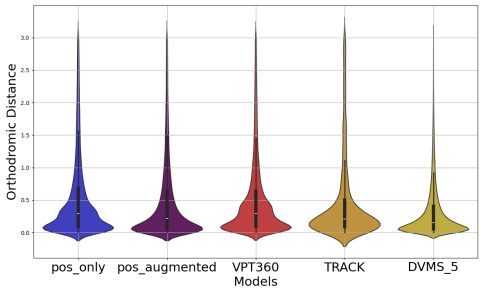
also highlights the differences between the pos-augmented model and other models. Notably, the **pos-augmented** model shows slightly better performance than **TRACK** at the 2.6-second mark for the PAMI18, MM22, and MMSys18 datasets. This suggests that at this early point in the prediction horizon, the pos-augmented model benefits from the integration of entropy information, while the saliency maps utilized by TRACK provide more valuable information as the prediction horizon extends to 5 seconds in the future. This is especially noticeable in the MMSys18 dataset, where user trajectories tend to exhibit higher entropy, providing more information for the model to utilize. However, as the prediction horizon extends to 5 seconds, the performance gap between pos-augmented and TRACK narrows. Overall, the results summarized in Table 6.1 confirm that DVMS_5 is the most effective model across all datasets, consistently outperforming the others in terms of orthodromic distance. The pos-augmented model performs competitively, especially for datasets where user trajectories exhibit higher entropy, but does not match the performance of DVMS_5. This suggests that while the pos-augmented model benefits from entropy integration, the approach used in DVMS (generating multiple predictions and selecting the best one) is more robust and leads to more accurate long-term predictions.

**Table 6.1:** Average Loss at 2.6 Seconds and 5 Seconds for Each Model and Dataset

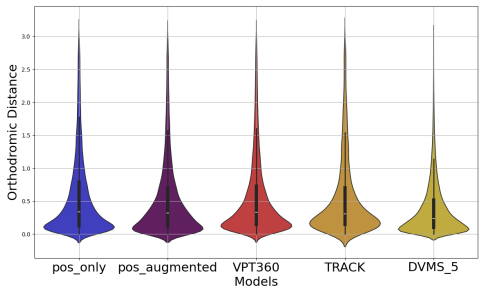
| Model | NOSSDAV17 | | MM22 | | PAMI18 | | MMSys18 | |
|---|---|---|---|---|---|---|---|---|
| | **2.6s** | **5s** | **2.6s** | **5s** | **2.6s** | **5s** | **2.6s** | **5s** |
| pos-only | 0.64 | 0.75 | 0.40 | 0.49 | 0.47 | 0.53 | 1.11 | 1.40 |
| TRACK | 0.65 | 0.73 | 0.43 | 0.50 | 0.46 | 0.51 | 1.08 | 1.22 |
| DVMS_5 | 0.39 | 0.55 | 0.23 | 0.32 | 0.28 | 0.38 | 0.57 | 0.76 |
| DVMS_2 | 0.48 | 0.63 | 0.29 | 0.38 | 0.35 | 0.45 | 0.79 | 0.96 |
| VPT360 | 0.65 | 0.75 | 0.42 | 0.51 | 0.48 | 0.55 | 1.03 | 1.32 |
| pos-augmented | 0.65 | 0.76 | 0.39 | 0.48 | 0.45 | 0.52 | 1.01 | 1.27 |



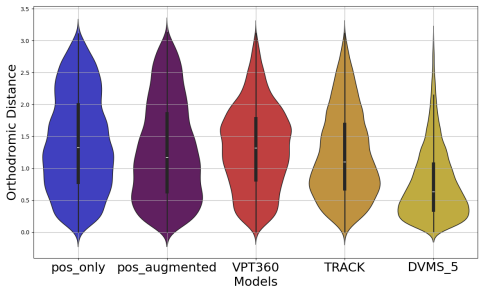

**(a)** NOSSDAV17




**(b)** MM22



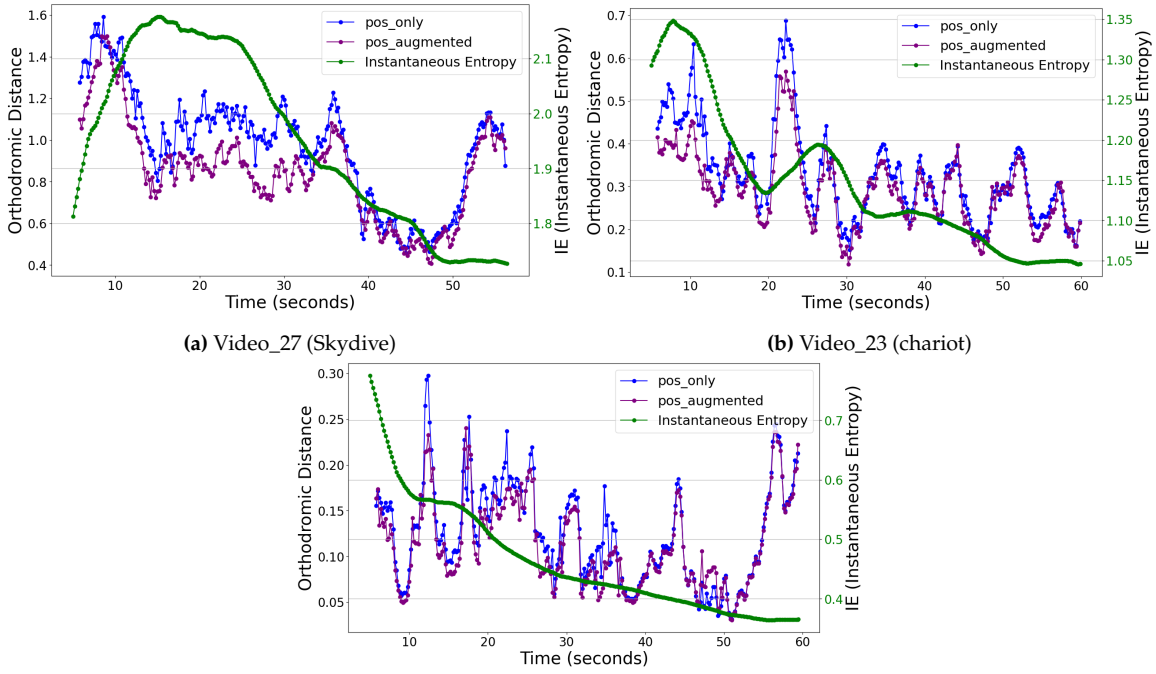

**(c)** PAMI18




**(d)** MMSys18

**Figure 6.8:** Violin plots of Orthodromic distance for pos-only and pos-augmented on (a) NOSSDAV17, (b) MM22, (c) PAMI18, and (d) MMSys18 datasets

To evaluate model stability, we also compute the violin plots of the errors for the last prediction step (5 seconds in the future) for the examined models, shown in Figure 6.8. We exclude DVMS_2 from these plots as the distribution is identical to that of DVMS_5 but with a slightly higher mean error. As expected DVMS_5 exhibits the most stable distributions, consistently skewed towards lower error values. For the NOSSDAV17 and PAMI18 datasets, the violin plots for all models, with the exception of DVMS_5, reveal nearly identical distributions, with minimal differences in the spread or central tendency of errors. However, for the MM22 dataset, both VPT360 and pos-only exhibit a prominent main cluster of errors centered around 0.08, with a secondary band visible above this range. In contrast, pos-augmented and TRACK lack this secondary band, as their distributions taper off more smoothly from their respective modes at 0.06 for pos-augmented and 0.13 for TRACK. While TRACK lacks the secondary band, its main error cluster is situated higher, reflecting a broader spread of errors compared




**(a)** Video_27 (Skydive)

**(b)** Video_23 (chariot)

**(c)** Video_13 (mario)

**Figure 6.9:** Orthodromic loss for the pos_augmented and pos_only models plotted along with the average instantaneous entropy of viewers across three videos from the MM22 dataset with varying levels of Actual Entropy (High AE: video_27, Medium AE: video_23, Low AE: video_13).

to the other models. For the MMSys18 dataset, we observe that the loss distribution for VPT360 is skewed towards higher losses, with its mode at 1.59, indicating a larger proportion of higher error values. In contrast, the TRACK model's distribution is skewed towards lower losses, but its central tendency occurs at higher error levels, with a mode of 1.17, compared to the pos-augmented model, which has a mode of 0.55. This suggests that, while the TRACK model has a tendency towards lower losses than VPT360, its errors still tend to be more concentrated in the higher error range compared to the pos-augmented model, which demonstrates the most favorable distribution with its mode of 0.34.

Finally, to identify the portions of the videos where pos_augmented outperforms pos_only and to understand why it might be more stable, we compare the losses across specific videos in the MM22 dataset. We select three videos based on the Actual Entropy (AE) exhibited by viewers: High AE (video_27, "skydive"), Medium AE (video_23, "chariot"), and Low AE (video_13, "mario"), the same ones chosen for our exploratory analysis in Section 4.2. For each video, we plot the orthodromic loss for the predicted head pose 5 seconds into the future (on the left y-axis), alongside the average instantaneous entropy of viewers throughout the video (on the right y-axis), as shown in Figure 6.9. Our analysis reveals that in video_27 (skydive), whenever users exhibit higher IE, the pos_only model's loss is notably higher compared to pos_augmented. However, when user trajectories have lower entropy, the losses for both models are quite similar. This is observed to a lesser extent in video_23 (chariot), where the pos_augmented model performs better at the start of the video where IE is particularly high, likely due to the viewers exploring the environment. In video_13, where viewers

have more predictable trajectories, both models perform comparably.

These findings suggest that integrating trajectory entropy into head pose prediction models can enhance performance, particularly in scenarios with highly unpredictable user behaviors. The pos-augmented model emerges as the only approach to show meaningful improvements, demonstrating enhanced stability and accuracy under high-entropy conditions, as observed in the MMSys18 dataset and certain segments of the MM22 dataset. While it remains less accurate than DVMS, the pos-augmented model outperforms other state-of-the-art models in terms of stability, especially during video segments with elevated instantaneous entropy. In contrast, adaptive attention models showed no significant improvements, suggesting that the benefits of entropy integration depend on the specific method of incorporation. These results highlight the potential of entropy-based augmentations to adaptively improve both accuracy and reliability in challenging, high-entropy scenarios. In the next chapter, we summarize our key findings, discuss their implications, and explore the limitations of our work, as well as propose future directions for research based on these insights.

# 7

# Discussion and Conclusion

In this chapter, we discuss the results in the context of the incorporation of viewer entropy, along with some of the limitations of the research project and future research possibilities. Lastly, we present the conclusions of this thesis.

## 7.1. Use of Entropy Metrics in Prediction

In this study, we explored the potential of user trajectory entropy to improve head pose prediction. Our initial exploratory analysis, conducted prior to testing the models, revealed a significant correlation between entropy metrics and prediction loss. Using three videos from the MM22 dataset (video_27 "skydive," video_23 "chariot," and video_13 "mario"), we observed a positive correlation between the Instantaneous Entropy of viewer trajectories and the prediction error. This finding motivated us to investigate whether incorporating entropy into head pose prediction models could improve stability and accuracy, particularly in segments where user behavior was highly unpredictable. To address this, we introduced two entropy-driven models: E-AALSTM and AMH. These models aimed to incorporate entropy information into adaptive attention mechanisms. The goal was to assign lower attention scores to high-entropy segments, where user behavior is more unpredictable and chaotic, making accurate predictions more challenging. By focusing more attention on low-entropy segments, where behavior is more consistent and predictable, we aimed to improve model stability. Additionally, we tested a simpler approach by augmenting head-pose data with entropy values in a seq2seq encoder-decoder architecture (referred to as the pos-augmented model). Our goal was to evaluate whether entropy could improve model stability and prediction accuracy. The incorporation of entropy into head pose prediction models demonstrated mixed results, with certain models benefiting from the added entropy-based features, while others showed no substantial improvement or even slight deterioration in performance. Notably, the experiments reveal that entropy-based metrics have nuanced impacts depending on the dataset and model configuration. Following our findings in Chapter 6, we observe that the proposed adaptive attention mechanisms (E-AALSTM and AMH) show limited capacity to leverage viewer entropy data effectively. This conclusion stems from the lack of significant deviations in performances between these adaptive models and their benchmarks, ALSTM and

VPT360. This suggests that the proposed adaptive attention mechanism may not capture the entropy information in a way that provides predictive improvements, and alternative mechanisms may be better suited for utilizing the entropy of user trajectories.

In contrast, the more straightforward approach of augmenting head-pose data by appending the entropy information in a seq2seq encoder-decoder architecture (pos-augmented) demonstrates promising potential. The analysis of violin plots (Figure 6.5) revealed that the pos-augmented model provided more stable predictions compared to the position-only model, especially for datasets with higher entropy, which indicates lower predictability. This was particularly evident in the MMSys18 dataset, and to a smaller extent in the MM22 dataset, where the pos-augmented model reduced the frequency of predictions that deviate greatly from the actual trajectory, showing a skew towards lower error values compared to the position-only model, which had a more spread-out error distribution. This reinforces the idea that entropy augmentation may help manage high-variance predictions, improving model stability when handling more unpredictable user trajectories. When comparing the pos-augmented model to state-of-the-art methods, we found that it modestly outperformed the VPT360 and TRACK models on the MM22 dataset and provided better predictions in the short term on the PAMI18 and MMSys18 datasets. This shows that while entropy-based models might not always outperform existing techniques, they can provide more stability and marginal improvements under specific conditions.

Additionally, when we examined specific video segments from the MM22 dataset, we observed that the pos-augmented model outperformed the position-only model during periods of high instantaneous entropy (e.g., video_27 "skydive" and video_23 "chariot"). In contrast, in video segments with lower entropy (e.g., video_13 "mario"), both models performed similarly. These findings emphasize the benefit of entropy augmentation in managing prediction errors during periods of high uncertainty.

While the pos-augmented model did not outperform state-of-the-art methods in all scenarios, it showed improvements in stability, particularly on datasets with high entropy. This supports the idea that entropy-based augmentations can help improve model performance in unpredictable environments. However, the effectiveness of entropy integration depends on the dataset's characteristics and how well the model can incorporate entropy information in a meaningful way.

## 7.2. Limitations

In this section, we discuss some of the limitations of our work based on our reflections while conducting this thesis.

**Implementation Constraints for State-of-The-Art Models:** The absence of readily available code for certain state-of-the-art (SOTA) models presented significant implementation challenges. As a result, most SOTA models had to be implemented from scratch, potentially leading to unintentional discrepancies in replication accuracy. This constraint limits the scope of the comparative study, as it restricts our ability to incorporate a wider range of models, particularly those based on non-deep learning algorithms for head pose prediction. Despite rigorous efforts to ensure correctness in

implementing SOTA models, slight deviations in architectural or training details may affect comparative results.

**Data Composition and Variability:** The datasets utilized in this thesis come from various sources, introducing potential inconsistencies in video characteristics that may influence results. For instance, two videos from different datasets with somewhat similar content—*skydive* from MM22 and *Touvet* from MMSys18—exhibit markedly different mean entropies in user trajectories, recorded at 1.73 and 3.34, respectively. This discrepancy may partly stem from differences in video lengths, as the MMSys18 dataset predominantly consists of shorter videos. Shorter videos often elicit higher entropy because users initially exhibit more random trajectories, rapidly exploring the environment to grasp the scene before the video ends. In contrast, longer videos, such as those in MM22, allow users more time to stabilize their focus, resulting in lower entropy. Furthermore, while the PAMI18 dataset includes both short and long videos, its design involves different users for each video, making it difficult to isolate the effect of video length on user trajectories. A more cohesive dataset with a controlled variety of short and long videos viewed by the same users would provide clearer insights into how video length and user behavior impact trajectory entropies and model performance.

**Hyperparameter Standardization and Limited Tuning** To maintain consistency, standardized hyperparameters were applied across all models, including a learning rate of 0.0001, an AdamW optimizer with a weight decay of 0.01, a batch size of 128, and a maximum of 500 epochs with early stopping. While this approach was designed to control for performance variability, it may unintentionally disadvantage certain models. Some architectures may have performed optimally with different settings, and the fixed parameters could skew the comparative results. Additionally, due to computational constraints, and to maintain consistency with previous studies, extensive hyperparameter tuning was not conducted. As a result, model performance may reflect sub-optimal configurations for certain architectures, particularly if they require different learning rates, dropout rates, or hidden layer sizes to achieve peak accuracy. This limitation affects the robustness of the results, as certain architectures may yield different outcomes with more refined hyperparameter optimization.

**Chosen Saliency Measures:** The saliency maps used in this work to calculate the entropy of saliency maps are generated based on user movements in the video, which may not always capture all the possible sources of attention in a scene. There are other approaches to generating saliency maps that can offer complementary or more accurate insights into visual focus. A more comprehensive evaluation using saliency maps generated using deep learning models [40] or by using optical flow [33] may offer additional insights into visual focus. The reliance on user-trajectory-based saliency maps, while effective in this context, may not fully account for the richness of other potential sources of visual saliency.

## 7.3. Future Work

In this section, we provide suggestions for future work based on our findings and the limitations of our work. One such suggestion is the investigation of alternative

methods for integrating entropy into predictive models. While the entropy-based attention mechanisms (E-AALSTM and AMH) showed limited improvements in model performance, the simpler pos-augmented approach showed modest promise in improving model stability. Future work could explore alternative methods for integrating entropy, such as incorporating it into different model architectures, or using it to augment other input features. This could help uncover novel ways of using entropy to enhance prediction accuracy and robustness.

The variability in video characteristics and user behavior across the datasets used in this study highlights the need for more cohesive datasets in future work. Specifically, a dataset designed to include both short and long videos viewed by the same users would provide a more controlled environment for analyzing trajectory dynamics and entropy effects. A dataset designed with these factors in mind would allow for clearer insights into the effects of video content on trajectory entropies. Additionally, future work could consider why the models incorporating entropy tend to show marginally worse performance on the NOSSDAV17 dataset and whether that remains consistent across other datasets where user trajectories exhibit low entropies.

Additionally, the use of user-trajectory-based saliency maps, while effective in this work, may have overlooked other potential sources of visual saliency. Future work could expand the scope to include alternative saliency measures, such as saliency maps generated using deep learning models [40] or optical flow analysis [33]. These methods could provide an alternative or complementary understanding of user trajectories, and improve the integration of saliency features in the models.

Finally, while we evaluated models based on Orthodromic distance to remain consistent with previous research and allow for quick comparison within the Unified Evaluation Framework [55], alternative evaluation metrics may provide deeper insights into model performances.

## 7.4. Conclusion

This study explored the role of entropy in predicting head pose trajectories, with a focus on entropy metrics derived from both saliency maps and user trajectories. We found that while the entropy of saliency maps did provide some insight into user focus, it exhibited an inverse relationship with the entropy of user trajectories. The entropy of user trajectories, particularly, emerged as a more reliable indicator of trajectory unpredictability, as evidenced by its closer correlation with higher prediction errors.

The results of this work suggest that incorporating user trajectory entropy into prediction models, specifically through straightforward augmentation methods like the pos-augmented model, can offer valuable improvements in model stability. This is particularly relevant when dealing with datasets characterized by unpredictable user behavior, where entropy augmentation helped reduce prediction deviations and provided more consistent results. The analysis of specific video segments further demonstrated that the pos-augmented model outperformed the position-only model during windows where user trajectories exhibit higher entropy, highlighting the potential of entropy to manage uncertainty. In contrast, the more complex entropy-driven attention mechanisms (E-AALSTM and AMH) did not show consistent

performance improvements across all models and datasets, pointing to the need for further exploration of more effective entropy integration strategies.

Ultimately, the study contributes to the understanding of how entropy, particularly from user trajectories, can be leveraged to enhance head pose prediction models. While the integration of entropy into deep learning architectures showed mixed results, straightforward approaches like the pos-augmented model demonstrated promise in improving prediction stability and could serve as a foundation for future work. Our findings underscore the importance of considering user behavior dynamics, dataset characteristics, and model architecture when incorporating entropy, and suggest several avenues for further research, including the exploration of alternative entropy integration methods and the development of more cohesive datasets for head pose prediction tasks.

# References


[1] Fares Abawi, Di Fu, and Stefan Wermter. *Unified Dynamic Scanpath Predictors Outperform Individually Trained Neural Models*. 2024. DOI: 10.48550/arXiv.2405.02929.

[2] Shahryar Afzal, Jiasi Chen, and K. K. Ramakrishnan. "Characterization of 360-degree Videos". In: *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*. VR/AR Network '17. Los Angeles, CA, USA: Association for Computing Machinery, 2017, pp. 1–6. ISBN: 9781450350556. DOI: 10.1145/3097895.3097896.

[3] Patrice Rondao Alface, Jean-François Macq, and Nico Verzijp. "Interactive omnidirectional video delivery: A bandwidth-effective approach". In: *Bell Labs Technical Journal* 16.4 (2012), pp. 135–147. DOI: 10.1002/bltj.20538.

[4] Michelle Annett et al. "How low should we go? Understanding the perception of latency while inking". In: *Graphics Interface* (2014), pp. 167–174.

[5] Roberto G. de A. Azevedo et al. "Visual Distortions in 360° Videos". In: *IEEE Transactions on Circuits and Systems for Video Technology* 30.8 (2020), pp. 2524–2537. ISSN: 1558-2205. DOI: 10.1109/tcsvt.2019.2927344.

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. 2016. arXiv: 1409.0473 [cs.CL]. URL: https://arxiv.org/abs/1409.0473.

[7] Yixuan Ban et al. "CUB360: Exploiting Cross-Users Behaviors for Viewport Prediction in 360 Video Adaptive Streaming". In: *2018 IEEE International Conference on Multimedia and Expo (ICME)*. 2018, pp. 1–6. DOI: 10.1109/ICME.2018.8486606.

[8] Paul Baumann and Silvia Santini. "On the use of instantaneous entropy to measure the momentary predictability of human mobility". In: *2013 IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. 2013, pp. 535–539. DOI: 10.1109/SPAWC.2013.6612107.

[9] Fang-Yi Chao, Cagri Ozcinar, and Aljosa Smolic. "Transformer-based Long-Term Viewport Prediction in 360° Video: Scanpath is All You Need". In: *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP)*. 2021, pp. 1–6. DOI: 10.1109/MMSP53017.2021.9733647.

[10] Jinyu Chen et al. "Sparkle: User-Aware Viewport Prediction in 360-Degree Video Streaming". In: *IEEE Transactions on Multimedia* 23 (2021), pp. 3853–3866. DOI: 10.1109/TMM.2020.3033127.

[11] Kyunghyun Cho et al. “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1724–1734. DOI: 10.3115/v1/D14-1179.

[12] Jonathan Cinnamon and Lindi Jahiu. “360-degree video for virtual place-based research: A review and research agenda”. In: *Computers, Environment and Urban Systems* 106 (2023), p. 102044. DOI: 10.1016/j.compenvurbsys.2023.102044.

[13] Xavier Corbillon et al. “Optimal Set of 360-Degree Videos for Viewport-Adaptive Streaming”. In: *Proceedings of the 25th ACM International Conference on Multimedia*. MM ’17. Mountain View, California, USA: Association for Computing Machinery, 2017, pp. 943–951. ISBN: 9781450349062. DOI: 10.1145/3123266.3123372.

[14] Erwan J. David et al. “A dataset of head and eye movements for 360° videos”. In: *Proceedings of the 9th ACM Multimedia Systems Conference*. MMSys ’18. Amsterdam, Netherlands: Association for Computing Machinery, 2018, pp. 432–437. ISBN: 9781450351928. DOI: 10.1145/3204949.3208139.

[15] Dmcq. *Spherical coordinate system, but with reversed notation for angles as often used in maths*. [Online: accessed 28-May-2024]. 2012. URL: https://commons.wikimedia.org/wiki/File:3D_Spherical_2.svg.

[16] Anup Doshi and Mohan Trivedi. “Head and eye gaze dynamics during visual attention shifts in complex environments (vol 12, pg 1, 2012)”. In: *Journal of vision* 12 (2012). DOI: 10.1167/12.2.9.

[17] Ching-Ling Fan et al. “Fixation Prediction for 360° Video Streaming in Head-Mounted Virtual Reality”. In: *Proceedings of the 27th Workshop on Network and Operating Systems Support for Digital Audio and Video*. NOSSDAV’17. Taipei, Taiwan: Association for Computing Machinery, 2017, pp. 67–72. ISBN: 9781450350037. DOI: 10.1145/3083165.3083180.

[18] Chelsea Finn, Pieter Abbeel, and Sergey Levine. “Model-agnostic meta-learning for fast adaptation of deep networks”. In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML’17. Sydney, NSW, Australia: JMLR.org, 2017, pp. 1126–1135.

[19] Chi-Wing Fu et al. “The Rhombic Dodecahedron Map: An Efficient Scheme for Encoding Panoramic Video”. In: *IEEE Transactions on Multimedia* 11.4 (2009), pp. 634–644. DOI: 10.1109/TMM.2009.2017626.

[20] Vamsidhar Reddy Gaddam et al. “Tiling in Interactive Panoramic Video: Approaches and Evaluation”. In: *IEEE Transactions on Multimedia* 18.9 (2016), pp. 1819–1831. DOI: 10.1109/TMM.2016.2586304.

[21] Tarek El-Ganainy and Mohamed Hefeeda. “Streaming Virtual Reality Content”. In: *arXiv* (2016). DOI: 10.48550/arXiv.1612.08350.

[22] Global Indusry Analysts, Inc. *Virtual Reality (VR) - Global Strategic Business Report*. [Online; accessed 28-May-2024]. 2024. URL: https://www.researchandmarkets.com/reports/3633908/virtual-reality-vr-global-strategic-business.

[23] A. Graves and J. Schmidhuber. "Framewise phoneme classification with bidirectional LSTM networks". In: *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* Vol. 4. 2005, 2047–2052 vol. 4. DOI: 10.1109/IJCNN.2005.1556215.

[24] Quentin Guimard et al. "Deep variational learning for multiple trajectory prediction of 360° head movements". In: *Proceedings of the 13th ACM Multimedia Systems Conference*. MMSys '22. Athlone, Ireland: Association for Computing Machinery, 2022, pp. 12–26. ISBN: 9781450392839. DOI: 10.1145/3524273.3528176.

[25] Ayah Hamad and Bochen Jia. "How Virtual Reality Technology Has Changed Our Lives: An Overview of the Current and Potential Applications and Limitations". In: *International Journal of Environmental Research and Public Health* 19 (2022), p. 11278. DOI: 10.3390/ijerph191811278.

[26] Bo Han. "Mobile Immersive Computing: Research Challenges and the Road Ahead". In: *IEEE Communications Magazine* 57.10 (2019), pp. 112–118. DOI: 10.1109/MCOM.001.1800876.

[27] Hristina Hristova et al. "Heterogeneous Spatial Quality for Omnidirectional Video". In: *IEEE Transactions on Circuits and Systems for Video Technology* 31.4 (2021), pp. 1411–1424. DOI: 10.1109/TCSVT.2020.3007620.

[28] Gazi Karam Illahi et al. "Learning to Predict Head Pose in Remotely-Rendered Virtual Reality". In: *Proceedings of the 14th ACM Multimedia Systems Conference*. MMSys '23. Vancouver, BC, Canada: Association for Computing Machinery, 2023, pp. 27–38. ISBN: 9798400701481. DOI: 10.1145/3587819.3590972.

[29] ITU-R. *Image parameter values for high dynamic range television for use in production and international programme exchange*. ITU-R Recommendation BT.2100. 2018. URL: https://www.itu.int/rec/R-REC-BT.1886.

[30] ITU-R. *Parameter values for the HDTV standards for production and international programme exchange*. ITU-R Recommendation BT.709. 2015. URL: https://www.itu.int/rec/R-REC-BT.709.

[31] ITU-R. *Reference electro-optical transfer function for flat panel displays used in HDTV studio production*. ITU-R Recommendation BT.1886. 2011. URL: https://www.itu.int/rec/R-REC-BT.1886.

[32] ITU-T. *Subjective video quality assessment methods for multimedia applications*. ITU-T Recommendation P.910. 2023. URL: https://www.itu.int/rec/T-REC-P.910.

[33] Muwei Jian et al. "Flow-Edge-Net: Video Saliency Detection Based on Optical Flow and Edge-Weighted Balance Loss". In: *IEEE Transactions on Computational Social Systems* 11.2 (2023), pp. 2026–2035. DOI: 10.1109/TCSS.2023.3270164.

[34] Zihang Jiang et al. *Few-shot Classification via Adaptive Attention*. 2020. arXiv: 2008.02465 [cs.CV]. URL: https://arxiv.org/abs/2008.02465.

[35] Yili Jin et al. "Where Are You Looking?: A Large-Scale Dataset of Head and Gaze Behavior for 360-Degree Videos and a Pilot Study". In: *Proceedings of the 30th ACM International Conference on Multimedia*. MM '22. ACM, 2022. DOI: 10.1145/3503161.3548200.

[36] Yoon Kim et al. *Structured Attention Networks*. 2017. arXiv: 1702.00887 [cs.CL]. URL: https://arxiv.org/abs/1702.00887.

[37] Evgeny Kuzyakov and David Pio. *Next-generation video encoding techniques for 360 video and VR*. [Online; accessed 21-July-2024]. 2016. URL: https://engineering.fb.com/2016/01/21/virtual-reality/next-generation-video-encoding-techniques-for-360-video-and-vr/.

[38] ChengDong Lan et al. "A self-attention model for viewport prediction based on distance constraint". In: *The Visual Computer* (2023). DOI: 10.1007/s00371-023-03149-6.

[39] Junjie Li, Yumei Wang, and Yu Liu. "Meta360: Exploring User-Specific and Robust Viewport Prediction in360-Degree Videos through Bi-Directional LSTM and Meta-Adaptation". In: *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 2023, pp. 652–661. DOI: 10.1109/ISMAR59233.2023.00080.

[40] Nian Liu et al. "Visual Saliency Transformer". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. CA, USA: IEEE Computer Society, 2021, pp. 4702–4712. DOI: 10.1109/ICCV48922.2021.00468.

[41] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*. 2017. URL: https://api.semanticscholar.org/CorpusID:53592270.

[42] Yiyun Lu, Yifei Zhu, and Zhi Wang. "Personalized 360-Degree Video Streaming: A Meta-Learning Approach". In: *Proceedings of the 30th ACM International Conference on Multimedia*. MM '22. Lisboa, Portugal: Association for Computing Machinery, 2022, pp. 3143–3151. ISBN: 9781450392037. DOI: 10.1145/3503161.3548047.

[43] Joschka Mütterlein. "The Three Pillars of Virtual Reality? Investigating the Roles of Immersion, Presence, and Interactivity". In: *Hawaii International Conference on System Sciences*. 2018. DOI: 10.24251/HICSS.2018.174.

[44] Albert Ng et al. "Designing for low-latency direct-touch input". In: *UIST '12: Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*. Massachusetts, USA: Association for Computing Machinery, 2012, pp. 453–464. ISBN: 9781450315807. DOI: 10.1145/2380116.2380174.

[45] Anh Nguyen, Zhisheng Yan, and Klara Nahrstedt. "Your Attention is Unique: Detecting 360-Degree Video Saliency in Head-Mounted Display for Head Movement Prediction". In: *Proceedings of the 26th ACM International Conference on Multimedia*. MM '18. Seoul, Republic of Korea: Association for Computing Machinery, 2018, pp. 1190–1198. ISBN: 9781450356657. DOI: 10.1145/3240508.3240669.

[46] Duc V. Nguyen et al. "An Optimal Tile-Based Approach for Viewport-Adaptive 360-Degree Video Streaming". In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9.1 (2019), pp. 29–42. DOI: 10.1109/JETCAS.2019.2899488.

[47] Cagri Ozcinar and Aljosa Smolic. "Visual Attention in Omnidirectional Video for Virtual Reality Applications". In: *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*. 2018, pp. 1–6. DOI: 10.1109/QoMEX.2018.8463418.

[48] Jounsup Park, Philip A. Chou, and Jenq-Neng Hwang. *Rate-Utility Optimized Streaming of Volumetric Media for Augmented Reality*. 2019. DOI: 10.1109/JETCAS.2019.2898622.

[49] Jounsup Park et al. "SEAWARE: Semantic Aware View Prediction System for 360-degree Video Streaming". In: *2020 IEEE International Symposium on Multimedia (ISM)*. 2020, pp. 57–64. DOI: 10.1109/ISM.2020.00016.

[50] Sohee Park et al. "Mosaic: Advancing User Quality of Experience in 360-Degree Video Streaming With Machine Learning". In: *IEEE Transactions on Network and Service Management* 18.1 (2021), pp. 1000–1015. DOI: 10.1109/TNSM.2021.3053183.

[51] Andrew Perkis et al. *QUALINET White Paper on Definitions of Immersive Media Experience (IMEx)*. 2020. eprint: 2007.07032.

[52] Stefano Petrangeli et al. "Improving Virtual Reality Streaming using HTTP/2". In: *Proceedings of the 8th ACM on Multimedia Systems Conference*. MMSys'17. Taipei, Taiwan: Association for Computing Machinery, 2017, pp. 225–228. ISBN: 9781450350020. DOI: 10.1145/3083187.3083224.

[53] Feng Qian et al. "Optimizing 360 video delivery over cellular networks". In: *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*. ATC '16. New York City, New York: Association for Computing Machinery, 2016, pp. 1–6. ISBN: 9781450342490. DOI: 10.1145/2980055.2980056.

[54] Werner Robitza and Lukas Krasula. *siti-tools*. VQEG GitHub repository. 2023. URL: https://github.com/VQEG/siti-tools.

[55] Miguel Fabián Romero Rondón et al. "A unified evaluation framework for head motion prediction methods in 360° videos". In: *Proceedings of the 11th ACM Multimedia Systems Conference*. MMSys '20. Istanbul, Turkey: Association for Computing Machinery, 2020, pp. 279–284. ISBN: 9781450368452. DOI: 10.1145/3339825.3394934.

[56] Miguel Fabián Romero Rondón et al. "TRACK: A New Method From a Re-Examination of Deep Architectures for Head Motion Prediction in 360° Videos". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.9 (2022), pp. 5681–5699. DOI: 10.1109/TPAMI.2021.3070520.

[57] Silvia Rossi, Alan Guedes, and Laura Toni. "Chapter 3 - Streaming and user behavior in omnidirectional videos". In: *Immersive Video Technologies*. Ed. by Giuseppe Valenzise et al. Academic Press, 2023, pp. 49–83. ISBN: 978-0-323-91755-1. DOI: https://doi.org/10.1016/B978-0-32-391755-1.00009-2.

[58] Silvia Rossi and Laura Toni. "Understanding user navigation in immersive experience: an information-theoretic analysis". In: *Proceedings of the 12th ACM International Workshop on Immersive Mixed and Virtual Environment Systems*. MMVE '20. Istanbul, Turkey: Association for Computing Machinery, 2020, pp. 19–24. ISBN: 9781450379472. DOI: 10.1145/3386293.3397115.

[59] Silvia Rossi, Laura Toni, and Pablo Cesar. "Correlation between Entropy and Prediction Error in VR Head Motion Trajectories". In: *Proceedings of the 2nd International Workshop on Interactive EXtended Reality*. IXR '23. Ottawa, ON, Canada: Association for Computing Machinery, 2023, pp. 29–36. ISBN: 9798400702808. DOI: 10.1145/3607546.3616805.

[60] Silvia Rossi et al. "Do Users Behave Similarly in VR? Investigation of the User Influence on the System Design". In: *ACM Trans. Multimedia Comput. Commun. Appl.* 16.2 (2020). ISSN: 1551-6857. DOI: 10.1145/3381846.

[61] Silvia Rossi et al. "Spherical Clustering of Users Navigating 360° Content". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019, pp. 4020–4024. DOI: 10.1109/ICASSP.2019.8683854.

[62] Marie-Laure Ryan. "Immersion vs. Interactivity: Virtual Reality and Literary Theory". In: *SubStance* 28.2 (1999), pp. 110–137. DOI: 10.1353/sub.1999.0015.

[63] Peiteng Shi, Markus Billeter, and Elmar Eisemann. "SalientGaze: Saliency-based gaze correction in Virtual Reality". In: *Computers  Graphics* 91 (2020). DOI: 10.1016/j.cag.2020.06.007.

[64] Vincent Sitzmann et al. "Saliency in VR: How Do People Explore Virtual Environments?" In: *IEEE Transactions on Visualization and Computer Graphics* 24.4 (2018), pp. 1633–1642. DOI: 10.1109/TVCG.2018.2793599.

[65] Mel Slater and Maria V. Sanchez-Vives. "Enhancing Our Lives with Immersive Virtual Reality". In: *Frontiers in Robotics and AI* 3 (2016). ISSN: 2296-9144. DOI: 10.3389/frobt.2016.00074.

[66] Iraj Sodagar. "The MPEG-DASH Standard for Multimedia Streaming Over the Internet". In: *IEEE MultiMedia* 18.4 (2011), pp. 62–67. DOI: 10.1109/MMUL.2011.71.

[67] Chaoming Song et al. "Limits of Predictability in Human Mobility". In: *Science* 327.5968 (2010), pp. 1018–1021. DOI: 10.1126/science.1177170.

[68] Gary J. Sullivan et al. "Overview of the High Efficiency Video Coding (HEVC) Standard". In: *IEEE Transactions on Circuits and Systems for Video Technology* 22.12 (2012), pp. 1649–1668. DOI: 10.1109/TCSVT.2012.2221191.

[69] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks". In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'14. Montreal, Canada: MIT Press, 2014, pp. 3104–3112.

[70] Samsung US. *Gear VR*. 2021. URL: https://www.samsung.com/us/support/mobile/virtual-reality/gear-vr/gear-vr-with-controller/.

[71] Rick Van Krevelen. *Augmented Reality: Technologies, Applications, and Limitations*. 2007. DOI: 10.13140/RG.2.1.1874.7929.

[72] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[73] HTC VIVE. *VIVE European Union | Discover Virtual Reality Beyond Imagination*. URL: https://www.vive.com/eu/.

[74] Thomas Weikert. "Head Motion Prediction in XR". Master's thesis. Master's Programme in ICT Innovation: Aalto University School of Electrical Engineering, 2021. URL: https://urn.fi/URN:NBN:fi:aalto-202202061781.

[75] Chenglei Wu et al. "A Dataset for Exploring User Behaviors in VR Spherical Video Streaming". In: *Proceedings of the 8th ACM on Multimedia Systems Conference*. MMSys'17. Taipei, Taiwan: Association for Computing Machinery, 2017, pp. 193–198. ISBN: 9781450350020. DOI: 10.1145/3083187.3083210.

[76] Mengbai Xiao et al. "BAS-360°: Exploring Spatial and Temporal Adaptability in 360-degree Videos over HTTP/2". In: *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*. 2018, pp. 953–961. DOI: 10.1109/INFOCOM.2018.8486390.

[77] Lan Xie, Xinggong Zhang, and Zongming Guo. "CLS: A Cross-user Learning based System for Improving QoE in 360-degree Video Adaptive Streaming". In: *Proceedings of the 26th ACM International Conference on Multimedia*. MM '18. Seoul, Republic of Korea: Association for Computing Machinery, 2018, pp. 564–572. ISBN: 9781450356657. DOI: 10.1145/3240508.3240556.

[78] Mai Xu et al. "Predicting Head Movement in Panoramic Video: A Deep Reinforcement Learning Approach". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.11 (2019), pp. 2693–2708. DOI: 10.1109/TPAMI.2018.2858783.

[79] Yanyu Xu et al. "Gaze Prediction in Dynamic 360° Immersive Videos". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5333–5342. DOI: 10.1109/CVPR.2018.00559.

[80] Abid Yaqoob, Ting Bi, and Gabriel-Miro Muntean. "A Survey on Adaptive 360° Video Streaming: Solutions, Challenges and Opportunities". In: *IEEE Communications Surveys & Tutorials* 22 (2020), pp. 2801–2838. DOI: 10.1109/COMST.2020.3006999.

[81] Alireza Zare et al. "HEVC-compliant Tile-based Streaming of Panoramic Video for Virtual Reality Applications". In: *Proceedings of the 24th ACM International Conference on Multimedia*. MM '16. Amsterdam, The Netherlands: Association for Computing Machinery, 2016, pp. 601–605. ISBN: 9781450336031. DOI: 10.1145/2964284.2967292.

[82] Jingbo Zhao et al. "Estimating the motion-to-photon latency in head mounted displays". In: *2017 IEEE Virtual Reality (VR)*. 2017, pp. 313–314. DOI: 10.1109/VR.2017.7892302.

[83] Chao Zhou, Zhenhua Li, and Yao Liu. "A Measurement Study of Oculus 360 Degree Video Streaming". In: *Proceedings of the 8th ACM on Multimedia Systems Conference*. MMSys'17. Taipei, Taiwan: Association for Computing Machinery, 2017, pp. 27–37. ISBN: 9781450350020. DOI: 10.1145/3083187.3083190.

[84] J. Ziv and A. Lempel. "Compression of individual sequences via variable-rate coding". In: *IEEE Transactions on Information Theory* 24.5 (1978), pp. 530–536. DOI: 10.1109/TIT.1978.1055934.