Deep learning for adaptive 360° video streaming in virtual reality

Quentin Guimard

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Deep learning pour le streaming adaptatif de vidéos à 360° en réalité virtuelle

Quentin GUIMARD

Laboratoire d'Informatique, de Signaux et Systèmes de Sophia Antipolis (I3S)
UMR7271 UCA CNRS

Présentée en vue de l'obtention du grade de docteur en Informatique **d'**Université Côte d'Azur

Dirigée par : Lucile SASSATELLI, Professeure des Universités, Université Côte d'Azur

Soutenue le : 18 décembre 2023

Devant le jury, composé de :

Patrick LE CALLET, Professeur des Universités, Nantes Université
Federica BATTISTI, Associate Professor, Université de Padoue
Gabriel-Miro MUNTEAN, Professor, Université de la ville de Dublin
Gwendal SIMON, Professeur des Universités, Synamedia
Aljosa SMOLIC, Professor, Haute École de Lucerne
Laura TONI, Associate Professor, University College de Londres

DEEP LEARNING POUR LE STREAMING ADAPTATIF DE VIDÉOS À 360° EN RÉALITÉ VIRTUELLE

Deep learning for adaptive 360° video streaming in virtual reality

Quentin GUIMARD

✉

Jury :

Président du jury
Patrick LE CALLET, Professeur des Universités, Nantes Université

Rapporteurs et rapportrices
Federica BATTISTI, Associate Professor, Université de Padoue
Gabriel-Miro MUNTEAN, Professor, Université de la ville de Dublin
Gwendal SIMON, Professeur des Universités, Synamedia

Examinateurs et examinatrices
Aljosa SMOLIC, Professor, Haute École de Lucerne
Laura TONI, Associate Professor, University College de Londres

Directrice de thèse
Lucile SASSATELLI, Professeure des Universités, Université Côte d'Azur

---

Université Côte d'Azur

Quentin GUIMARD

Deep learning pour le streaming adaptatif de vidéos à 360° en réalité virtuelle

xii+221 p.

Ce document a été préparé avec LaTeX2e et la classe `these-ISSS` version v. 2.10.

Impression : sommaire.tex – 15/3/2024 – 19:38

Révision pour la classe : these-ISSS.cls,v 2.10 2020/06/24 14:16:37 mpelleau

Deep learning pour le streaming adaptatif de vidéos à 360° en réalité virtuelle

Résumé

La réalité virtuelle (VR) a évolué de manière significative ces dernières années. Les casques immersifs devenant de plus en plus abordables et populaires, de nombreuses applications sont à l'horizon, des vidéos à 360° aux formations interactives en passant par les environnements virtuels collaboratifs. Cependant, pour atteindre des niveaux élevés de qualité perçue, la bande passante du réseau et les ressources de calcul nécessaires peuvent être supérieures de plusieurs ordres de grandeur à celles requises pour un contenu 2D traditionnel. Pour pallier ce problème, des stratégies de streaming qui adaptent le débit vidéo aux conditions du réseau et à l'orientation de la tête de la personne ont été mises en œuvre afin d'améliorer la qualité d'expérience. Étant donné que la plupart des algorithmes de débit adaptatif reposent sur l'utilisation d'une mémoire tampon vidéo suffisamment grande pour compenser les fluctuations de la bande passante, l'algorithme doit savoir où la personne regardera quelques secondes avant la lecture pour adapter correctement la qualité. La qualité d'expérience pour le streaming 360° dépend donc de la prédiction des mouvements de la tête en VR. Malheureusement, il s'agit d'un problème difficile en raison (i) du caractère aléatoire des mouvements humains, (ii) de la diversité des trajectoires de tête des personnes qui regardent des vidéos à 360° ce qui entraîne une ambiguïté entre les trajectoires passées, et (iii) des nombreux facteurs qui influencent le comportement, l'attention et les mouvements de la personne en VR. Afin de concevoir des systèmes de streaming VR qui s'adaptent mieux à chaque personne, il est important de comprendre les différents facteurs, leurs interactions et leurs effets sur le comportement humain. La collecte et l'exploitation de nouvelles données relatives à ces facteurs pourraient aider à désambiguïser les trajectoires la tête et à améliorer leur prédiction. Ce travail est divisé en quatre contributions principales. Premièrement, nous avons proposé un nouveau framework de deep learning variationnel pour prédire de multiples trajectoires possibles de mouvements de tête afin de mieux prendre en compte la diversité des trajectoires. Nous avons montré que notre modèle surpasse les performances de concurrents adaptés du domaine de la conduite autonome, réduisant l'erreur jusqu'à 41 % sur quatre datasets. Nous avons ensuite proposé un nouveau simulateur de streaming 360° afin de mesurer les gains système de notre framework et de permettre de comparer facilement les stratégies de streaming adaptatif. Nous avons montré que la prédiction de trajectoires multiples conduit à une plus grande équité entre les usagers, avec des gains de qualité atteignant jusqu'à 10 % pour 20 à 30 % des personnes. En parallèle, nous avons mené des expériences avec des personnes et des analyses statistiques pour mieux comprendre l'interaction entre le contenu immersif, l'attention et les émotions. Nous avons observé que le degré d'activation physiologique de la personne était corrélé à l'attention portée aux objets, et nous avons quantifié les effets des émotions sur la prédictibilité des mouvements de la tête. Enfin, nous avons voulu tirer parti des données liées aux émotions afin d'apprendre de meilleures représentations et d'améliorer la prédiction des mouvements de la tête. Inspirés par les travaux récents sur la distillation cross-modale et les modèles de fondation multimodaux, nous avons commencé à travailler sur une nouvelle architecture de deep learning multimodale capable d'apprendre des représentations transférables de modalités qui ne sont disponibles qu'au moment de l'apprentissage. Nous avons obtenu des résultats préliminaires qui surpassent de 21 % l'état de l'art existant tout en réduisant considérablement le nombre de paramètres.

Mots-clés : Apprentissage profond, Réseaux de neurones artificiels, Réalité virtuelle, Streaming, Régression, Multimedia

Deep learning for adaptive 360° video streaming in virtual reality

Abstract

Virtual reality (VR) has evolved significantly in recent years. As head-mounted displays become more affordable and popular, new opportunities for high-quality immersive experiences are opening up. A variety of exciting applications are on the horizon, from 360° videos to interactive training simulations and collaborative virtual environments. However, to achieve high levels of perceptual quality, the required network bandwidth and GPU computing resources can be orders of magnitude higher than those required for traditional 2D content. To mitigate this, adaptive streaming strategies have been implemented to improve the quality of experience (QoE) for people watching 360° videos over the Internet. This is done by adapting the video quality to the network conditions and the user's head orientation. Since most adaptive bitrate algorithms rely on using a large enough video buffer to compensate for bandwidth fluctuations, the algorithm needs to know where the person will be looking a few seconds before playback to make the appropriate quality decisions. Improving the QoE for 360° video streaming therefore depends on accurately predicting the user's viewport in VR. Unfortunately, viewport prediction is a challenging problem due to (i) the inherent randomness of human motion, (ii) the diversity of head trajectories among people watching 360° video, which leads to ambiguity between similar past trajectories, and (iii) the many factors that influence user behavior, attention, and movement in VR. In order to design VR streaming systems that can better adapt to each user, it is important to understand the different factors, their interactions, and their effects on human behavior. Collecting and exploiting additional data modalities related to these factors could help disambiguate head trajectories and improve viewport prediction. The work covered in this manuscript touches on many areas, including the design of various multimodal deep learning architectures applied to regression, dynamic optimization problems, time series forecasting, and user experiments along with associated statistical analyses. This work is divided into four main contributions. First, we studied the similarity between head motion trajectories and proposed a new variational deep learning framework for predicting multiple possible head motion trajectories to better account for trajectory diversity. While our framework is compatible with any sequence-to-sequence architecture, we implemented a flexible and lightweight stochastic prediction model and showed that it outperformed competitors adapted from the self-driving domain by up to 41% on four datasets. We then proposed a new trace-driven 360° video streaming simulator to measure the system gains of our framework and provide a way to easily compare adaptive streaming strategies. We showed that predicting multiple trajectories leads to higher fairness among simulated users, with gains for 20% to 30% of users reaching up to 10% in visual quality. In parallel, we conducted user experiments and statistical analyses to better understand the interaction between immersive content, attention, and emotions, as well as the effects of emotions on user motion. We observed that user arousal correlated with the accuracy of high-level saliency. We also quantified the effects of valence and arousal on the predictability of head movements and their interaction with spatial information. Finally, we wanted to take advantage of additional emotion-related data modalities to learn better representations and improve viewport prediction. Motivated by recent work on cross-modal knowledge distillation and multimodal foundation models, we initiated work on a new multimodal deep architecture able to learn transferable representations of modalities that are only available at training time. We obtained early results outperforming the existing state-of-the-art by up to 21% while greatly reducing the number of parameters.

Keywords: Deep learning, Artificial neural networks, Virtual reality, Streaming, Regression, Multimedia

Acknowledgements

Completing my PhD was only made possible thanks to the help and support of many people.

First and foremost, I would like to thank Lucile, my supervisor, who is the most dedicated person I have ever had the pleasure to work with. Thank you for always being available, helpful, and supportive. You were always able to find the right words to motivate me and put everything in perspective, especially when I felt lost. You will always be an inspiration to me.

I would also like to thank Federica Battisti, Gabriel Miro-Muntean, and Gwendal Simon for reviewing my manuscript and serving on my defense jury. I am very grateful for the time and effort they took to provide a very valuable feedback that helped to improve the manuscript. I would also like to thank Aljosa Smolic, Laura Toni, and Patrick Le Callet for serving as my defense examiners. I greatly appreciate the thoughtful consideration you all gave to my work, the insightful feedback you provided, and the challenging questions you asked.

The scientific contributions presented in this manuscript would not have been possible without my dear collaborators in Sophia, Florence, and Amsterdam. Special thanks go to Florent, Hui-Yin, Marco, Auriane, Hugo, Franz, Rémy, Lorenzo, Federico, Francesco, Alberto, Silvia, Pablo, Jiahuan, and Abdo.

I would also like to thank the helpful members of the administration, especially Nadia, who has been very efficient and considerate, always doing her best and pushing to get things done.

A very important thank you goes to all the people I have not mentioned yet, but who made a significant positive contribution to my doctoral experience. Thanks to all my friends for the ski trips, the hikes, the long discussions, the coffees, the beers, the parties, the barbecues, the raclettes, the games... The list is too long to mention them all, but I will try. Thank you Amélie, Ana Maria, Anderson, Arnab, Ashu, Azeez, Carlotta, Dalia, Diana, Elisa, Emilie, François, Georgios, Giulia, Hermes, Houssem, Ignacio, Irena, Irene, Jonas, Jules, Julie, Juliette, Kate, Laetitia, Lara, Lina, Lucile, Luigi, Marie, Michal, Miguel, Moonisa, Nina, Ninad, Pati, Pieter, Romain, Rudan, Sakshi, Sara, Sardor, Tarek, Tom, Tomas, Yacine, and Xuemei.

Last but not least, I would like to express my love and gratitude to my family, my sister Julie, my mother Nathalie, and my father Stéphane, who have always supported me and my work, especially during the months I spent with them in 2020 and 2021.

Table of contents

Chapter 1

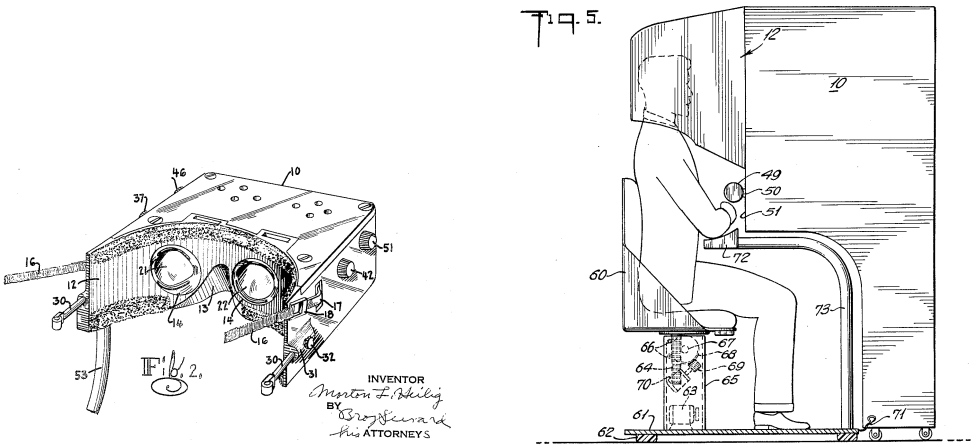
Introduction

In this chapter, we introduce the subject of this thesis. We first provide some historical context and background about virtual reality and head-mounted displays. We then elucidate the challenges hampering the widespread adoption of virtual reality, despite the recent technological developments. We propose to tackle these challenges by formulating objectives in the form of research questions. We finally list the scientific contributions that were made to answer these questions.

1.1 Context

The earliest published use of the term "virtual reality" dates back to 1938, where Antonin Artaud used the expression "*réalité virtuelle*" to describe the illusory nature of characters and objects in theater (Artaud, 1938). The meaning of "virtual reality" has since evolved to become what we know today, mostly thanks to the work of virtual reality pioneers (Zimmerman, Lanier, Blanchard, Bryson, & Harvill, 1986; Lanier, n.d.; Conn, Lanier, Minsky, Fisher, & Druin, 1989; Blanchard et al., 1990) and science fiction writers (Broderick, 1982; Krueger, 1983; *The Lawnmower Man*, 1992) in the 1980s. Nowadays, according to Merriam-Webster's disctionary (Merriam-Webster, 2023), virtual reality can be defined as "an artificial environment which is experienced through sensory stimuli (such as sights and sounds) provided by a computer and in which one's actions partially determine what happens in the environment".

The earliest form of virtual reality (VR) to approach this definition was envisioned by Morton Heilig (Heilig, 1955) in 1955, calling it the "Experience Theater". Over several years, he developed the Telesphere Mask (Heilig, U.S. Patent US2955156A, Oct. 1960) (Fig. 1.1a), "a telescopic television apparatus for individual use", and the Sensorama (Heilig, U.S. Patent US3050870A, Aug. 1962) (Fig. 1.1b), his vision for the future of cinema, with the help of his partner, Marianne Heilig. Patented in 1960, the Telesphere Mask can be considered as the first ever head-mounted VR display. "The spectator is given a complete sensation of reality, i.e. moving three dimensional images which may be in colour, with 100% peripheral vision, binaural sound, scents and air breezes", read the patent filing. Predating digital computing, the Sensorama was a mechanical device and one of the earliest known examples of immersive, multi-sensory technology. Unfortunately, Heilig did not manage to secure sufficient investment or sales, and both of his inventions were commercial failures.



(a) Telesphere Mask (1960) *(b) Sensorama (1962)*

Figure *1.1: Heilig's early VR prototypes.*

Other notable early prototypes of head-mounted VR displays include "The Sword of Damocles" (1968, owing its name to the fact that it had to be attached to a mechanical arm suspended from the ceiling), NASA's LEEP (1979, Large Expanse, Extra Perspective optical system) and VIEW (1985, Virtual Interactive Environment Workstation), the VPL EyePhone (1989), the Virtuality 1000 (1990) and 2000 (1994) series, the unreleased Sega VR (199X), the Forte VFX1 Headgear (1995) and the Nintendo Virtual Boy (1995).

From then on, VR technology has undergone significant development, especially in the last 10 years, with the advent of mass-market VR-capable head-mounted displays. Since the release of the Oculus Rift DK1 in 2013 (Fig. 1.2a), VR headsets have been getting more popular and affordable. The HTC Vive headsets (2016-), the Sony PlayStation VR (2016) and VR2 (2023), the Valve Index (2019), the Oculus (now Meta) Quest line of standalone headsets (2019-), and the recently-announced Apple Vision Pro (2023) are the most popular examples of modern VR heasets.

In 2023, Meta declared they had sold over 20 million Quest units (Fig. 1.2b). Revenue in the AR and VR market is projected to reach US$31.12bn in 2023 and is expected to rise to US$52.05bn by 2027 (Statista Market Insights, 2023). VR technology is rapidly evolving, opening up a wide array of applications across various fields, such as gaming and entertainment, education, healthcare, real estate and architecture, corporate training, or even art and design. The development of these new technologies brings new technical challenges.

1.2 Challenges

The evolution of VR technology has witnessed remarkable strides in recent years, with innovations in hardware and software leading to immersive experiences that were once considered the realm of science fiction. However, despite these advancements, the widespread

(a) Oculus Rift DK1 (2013) *(b) Meta Quest 3 (2023)*

Figure *1.2: First generation and latest generation of Oculus' (now Meta) VR headsets.*

adoption of VR remains hampered by a number of challenges that significantly impact the quality of user experience.

In the QUALINET White Paper on Definitions of Immersive Media Experience (Perkis et al., 2020), Quality of Experience (QoE) for immersive media is defined as "the degree of delight or annoyance of the user of an application or service which involves an immersive media experience. It results from the fulfillment of his or her expectations with respect to the utility and/or enjoyment of the application or service in the light of the user's personality and current state."

The challenges in achieving a high QoE in VR are intricately linked to human perceptual characteristics and a varity of influencing factors. Dizziness or nausea, commonly experienced in VR and usually referred to as "cybersickness", can heavily deteriorate the QoE. Prominent causes are locomotion, acceleration, and rotation in conflict with what the vestibular system perceives, and still imperfect displays (resolution, vergence-accomodation conflict,...) which lead to a loss of spatial awareness. It is also widely believed that a feeling of presence plays a vital role in enhancing the QoE in immersive media. This "sense of being there" involves a sense of agency while navigating and interacting within the virtual environment and allows users to perceive virtual objects as if they were real.

When watching 360° videos in a VR headset over the Internet, we argue that a smooth, high-resolution, experience is necessary to reduce cybersickness and strengthen the sense of presence. However, in order to achieve these high levels of perceptual quality, the necessary network bandwidth and GPU computing resources can be orders of magnitude higher than those required for traditional 2D content. This is mainly due to the fact that VR displays only stand a few centimeters from the eyes and need a very high resolution (25 pixels per degree for the Meta Quest 3, up to 70 for the Varjo VR-3 headset).

The QoE for people watching 360° videos over the Internet can be significantly improved by taking advantage of adaptive streaming strategies. While traditional 2D adaptive bitrate algorithms improve the QoE by dynamically adapting the bitrate of video segments to network conditions, 360° adaptive bitrate algorithms can take it a step further by also adapting the quality to the user's head orientation. Since the user can only see

one-third of the entire 360° scene at a time, the visual quality can be spatially adapted to match the user's field of view. This can be done through tile-based or view-based adaptive streaming, currently implemented in industry standards such as HLS and MPEG-DASH (Hosseini & Swaminathan, 2016a; H. S. Kim et al., 2018). Additional levers such as virtual walls that prevent the user from seeing certain areas, snap-changes selected by content creators to show important parts of the content, or even slow-downs can also be included in 360° adaptive streaming strategies (Dambra et al., 2018; Sassatelli et al., 2020).

The streaming of 360° videos can therefore be defined as an optimization problem where maximizing the QoE depending on the user and the state of the network can be done by dynamically adapting the bitrate of the video, both temporally and spatially, and carefully activating some of the aforementioned levers. We refer to these bitrate choices and lever activations as "quality decisions". Most adaptive bitrate algorithms rely on using a large enough video buffer to compensate for bandwidth fluctuations. This requires the quality decisions to be made a few seconds ahead of playback. For this reason, the algorithm needs to know where the person will be looking a few seconds in the future before making requests to the server.

Improving the QoE for 360° video streaming therefore depends on accurately predicting the user's viewport in VR. Unfortunately, viewport prediction is a challenging problem due to (i) the inherent randomness of human motion, (ii) the diversity of head trajectories among people watching 360° video, which leads to ambiguity between future trajectories that were similar in the past, and (iii) the many factors that influence user behavior, attention, and movement in VR.

1.3 Objectives

The main objective of the work presented in this manuscript is to address the challenges that arise when predicting the user's viewport in VR. We want to improve the quality of experience by designing VR streaming systems that can better adapt to each user. Specifically, we aim to answer the following research questions:

- *How can we consider the randomness and diversity of human motion when predicting head movements based on past head trajectories?*
- *What is the relationship between immersive content, emotions, and attention in virtual reality?*
- *Can we identify which factors influence human movement in VR and quantify their effects?*
- *How can we take advantage of these factors to learn better representations and improve viewport prediction?*

1.4 Contributions

We make the following contributions:

- First, in chapter 3, we present a pioneering approach for generating multiple plausible futures of head motion in 360° videos based on a shared past trajectory. We analyze the diversity of potential futures corresponding to similar past trajectories and address the limitations of existing predictors. We introduce the discrete variational multiple sequence (DVMS) learning framework, leveraging deep latent variable models to modulate the connection between past and future trajectories. We also conduct a detailed analysis of the learned latent space and its impact on trajectory prediction. Additionally, we devise a method to estimate the likelihoods of multiple predicted trajectories by exploiting the stationarity of prediction errors over the latent space. Our method outperforms competitors in the self-driving domain by up to 41% on prediction horizons up to 5 seconds, demonstrating superior performance at lower computational costs. This work represents the first exploration of multiple head motion prediction in 360° videos, contributing valuable insights and techniques to the field.

- Second, in chapter 4, we present SMART360, a 360° streaming simulation environment designed for comparing head motion prediction and Adaptive Bitrate (ABR) algorithms. Our contributions include the development of a comprehensive simulator with large datasets and baseline algorithms, along with transparent preprocessing pipelines for creating new input configurations. We provide detailed guidance on utilizing SMART360 for implementing and comparing motion prediction and adaptive bitrate strategies. Additionally, we conduct an extensive analysis, involving nearly 5 million simulations with diverse user-video pairs, network traces, and viewport prediction algorithms. Our findings demonstrate that predicting multiple trajectories under a constant bandwidth budget results in higher fairness between user-video pairs, reducing traces with the worst QoE. Furthermore, we show that choosing the best number of trajectories to predict yields significantly improved quality in the viewport and overall QoE, showcasing the effectiveness of our approach.

- Third, in chapter 5, we present a comprehensive contribution consisting of three key elements. First, we introduce PEM360, a novel dataset featuring user head movements, gaze recordings, emotional ratings, and physiological measurements in 360° videos. This dataset is enriched with high-level and low-level content-based saliency maps, enabling spatiotemporal analysis of the interconnections between content, user motion, and emotion. We provide open-access Python tools and notebooks for data processing and visualization, ensuring reproducibility. Secondly, we investigate the impact of emotions on saliency estimators in 360° videos, revealing that high-level saliency better predicts user attention under higher arousal levels. Lastly, we explore the relationships between user-centric and video-centric

measures and head motion predictability, validating hypotheses through a structural equation model (SEM). Our findings demonstrate that higher arousal leads to higher predictability, while higher valence leads to lower predictability, with head speed mediating this relationship. Furthermore, spatial information moderates the effect of arousal on predictability, providing valuable insights into emotion–video feature–predictability dynamics.

- Finally, in chapter 6, we present an ongoing work focused on developing a modular multimodal deep architecture for viewport prediction, leveraging cross-modal learning techniques. This architecture is designed to learn transferable cross-modal representations by jointly training on multiple modalities and can be easily expanded with additional modalities. Our contributions include the introduction of a new efficient multimodal architecture for viewport prediction and a benchmark of existing methods for online viewport prediction in 360° videos.

1.5 Publications

Guimard, Q., Robert, F., Bauce, C., Ducreux, A., Sassatelli, L., Wu, H.-Y., ... Gros, A. (2022a). On The Link Between Emotion, Attention And Content In Virtual Immersive Environments. In *2022 IEEE International Conference on Image Processing (ICIP)* (pp. 2521–2525). doi: 10.1109/ICIP46576.2022.9897903

Guimard, Q., Robert, F., Bauce, C., Ducreux, A., Sassatelli, L., Wu, H.-Y., ... Gros, A. (2022b). PEM360: A dataset of 360° videos with continuous Physiological measurements, subjective Emotional ratings and Motion traces. In *Proceedings of the 13th ACM Multimedia Systems Conference.* ACM. doi: 10.1145/3524273.3532895

Guimard, Q., & Sassatelli, L. (2022). Effects of Emotions on Head Motion Predictability in 360° Videos. In *Proceedings of the 14th International Workshop on Immersive Mixed and Virtual Environment Systems* (p. 37–43). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3534086.3534335

Guimard, Q., & Sassatelli, L. (2023). SMART360: Simulating Motion Prediction and Adaptive BitRate STrategies for 360° Video Streaming. In *Proceedings of the 14th Conference on ACM Multimedia Systems* (p. 384–390). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3587819.3592547

Guimard, Q., Sassatelli, L., Marchetti, F., Becattini, F., Seidenari, L., & Del Bimbo, A. (2024, jan). Deep variational learning for 360° adaptive streaming. *ACM Trans. Multimedia Comput. Commun. Appl.*. (Just Accepted) doi: 10.1145/3643031

Guimard, Q., Sassatelli, L., Marchetti, F., Becattini, F., Seidenari, L., & Del Bimbo, A. (2022). Deep Variational Learning for Multiple Trajectory Prediction of 360°

Head Movements. In *Proceedings of the 13th ACM Multimedia Systems Conference* (p. 12–26). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3524273.3528176

CHAPTER 2

Background

All models are wrong, but some are useful.

— George Box, Empirical Model-Building and Response Surfaces.

In this chapter, we take a deep dive in the scientific literature related to deep learning for deep learning for adaptive 360° video streaming in virtual reality. As this subject lies at the intersection of multiple fields, we split this chapter in three sections.
First, we discuss the evolution of adaptive streaming in a chronological fashion, before dwelling upon the question of adaptive streaming in immersive virtual environments.
Second, we provide the reader with a background of selected work on the closely related fields viewport prediction, human motion prediction, trajectory prediction, and time series forecasting–sequence modeling.
Third, we take an interest in works dealing with human attention and behavior in VR as well as their influencing factors.
This chapter is not destined to be an exhaustive review of the state of the art of each of the aforementioned fields. Instead, we make an attempt at mentioning (i) the existing work that we consider to be interesting and relevant to this thesis, (ii) the major works in each field to get a good general idea of the state of the art, and (iii) some recent work to catch a glimpse of the direction future work seems to be taking.

Contents

2.1 Adaptive streaming

In this section, we discuss the evolution of adaptive streaming, from the early stages to VR-adapted adaptive streaming. In Fig. 2.1, we provide a timeline showing what we consider to be the most impactful contributions to HTTP adaptive streaming.

2.1.1 Early stages

The idea of adaptive streaming, adapting the bit rate of the video to the network conditions to ensure a smooth experience, is not new and has been the subject of many works since the 1990s. As early as 1993, Kanakia, Mishra, and Reibman (1993) proposed an adaptive congestion control scheme, demonstrating that modulating the source rate of a video encoder based on delayed feedback from the network led to graceful degradation in picture quality during congestion. A year later, Bolot, Turletti, and Wakeman (1994) presented a rate control mechanism for packet video in the Internet, emphasizing the use of feedback mechanisms to adapt the output rate of video coders based on the state of the network. Their mechanism was implemented in the H.261 video coder of IVS. Building on this work, Bolot and Turletti (1994) developed a scalable feedback control mechanism for multicast video distribution in the Internet. Their innovative approach utilized a probing mechanism to solicit feedback information in a scalable manner, estimating the number of receivers, and separating congestion signals from the control algorithm. This strategy ensured effective multicast video distribution to a large number of participants, preventing congestion in the Internet and maximizing perceptual quality while minimizing bandwidth usage.

In 1995, Cen, Pu, Staehli, Cowan, and Walpole (1995) presented a distributed real-time MPEG video audio player designed for the diverse and variable Internet environment. Their approach utilized software feedback mechanisms for client/server synchronization, dynamic Quality-of-Service control, and system adaptiveness. The same year, Eleftheriadis and Anastassiou (1995) introduced the concept of Dynamic Rate Shaping, enabling the adaptation of compressed video bitstreams to dynamically varying rate and delay constraints. This technique decoupled the encoder and the network, ensuring universal interoperability and providing algorithms for dynamic rate shaping that could be implemented in software, allowing for widespread applicability in video-on-demand systems. Other notable contributions to the field at that time include the adaptive streaming service for streaming MPEG video over best-effort IP network environments designed by Ramanujan et al. (1997), and the Streaming Control Protocol (SCP) of Cen, Walpole, and Pu (1997).

2.1.2 First proposals and standards

Adaptive streaming continued to undergo significant development in the following years, with many commercial solutions for adaptive bitrate streaming over HTTP emerging in the late 2000s. Move Networks was the first to deploy its adaptive streaming solution

in 2006, based on a proprietary adaptive bit-rate technology, patented in 2010 (Brueck & Hurst, U.S. Patent US7818444B2, Oct. 2010). Large video files were broken up into many small files called streamlets, which were then delivered as a series of video segments using a highly efficient transmission protocol. Microsoft came up with Smooth Streaming (MSS) when they released their new Internet Information Server (IIS) 7.0 in 2008. Apple then release HTTP Live Streaming (HLS) in 2009. Adobe also released their own commercial solution for adaptive streaming in 2010 with the Flash 10.1 player, called HTTP Dynamic Streaming (HDS).

Standardization efforts led to the creation of Dynamic Adaptive Streaming over HTTP (DASH), developed under MPEG between 2010 and 2012. DASH specifies how the multimedia files must be segmented and described using a media presentation description (MPD) file. DASH is codec-agnostic, can be used with any protocol, and does not specify the adaptive bitrate streaming (ABR) logic.

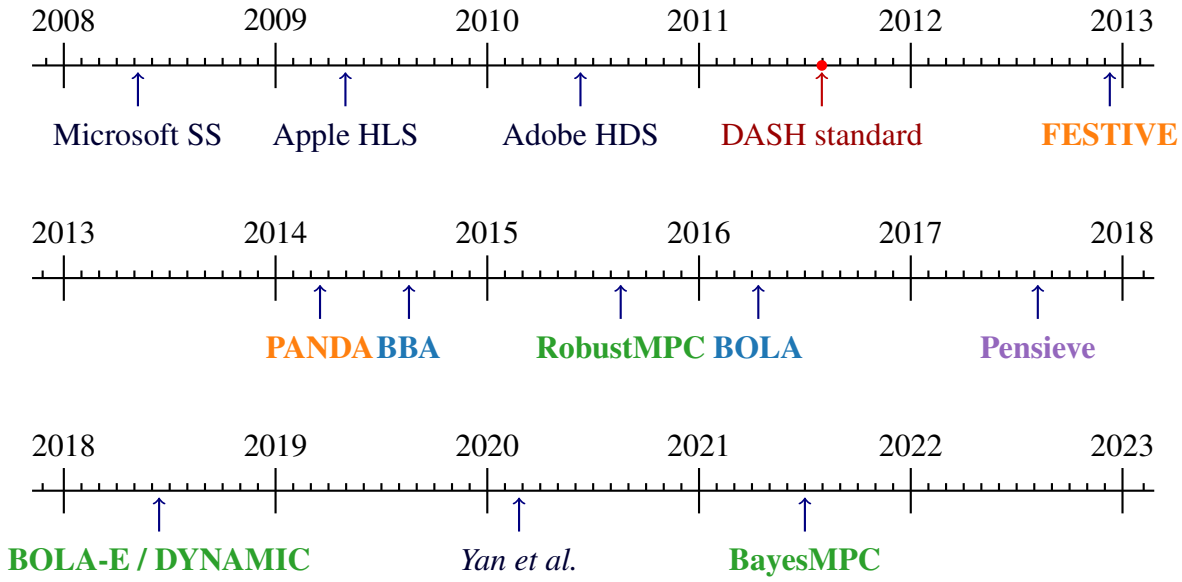


Figure 2.1: *Timeline of the key commercial and scientific contributions to HTTP adaptive streaming. ABR algorithms are highlighted in* ***bold****. Rate-based algorithms are colored in orange, buffer-based in blue, hybrid in green, and RL-based in purple.*

We mention some of the important research works contemporary to the standardization of DASH. Stockhammer (2011) delved into the specifications of DASH in 2011, providing design principles and examples. Akhshabi, Begen, and Dovrolis (2011) provided an evalutation of rate-adaptation mechanisms in adaptive streaming. They experimentally assessed commercial players (Smooth Streaming, Netflix) and an open-source player (OSMF) under different network conditions, identifying inefficiencies and differences between these players, especially concerning bandwidth adaptation and competition between players. Lederer, Müller, and Timmerer (2012) addressed the challenge of comparing adaptive streaming algorithms objectively due to the lack of a common dataset.

They introduced a DASH dataset and an open-source tool (DASHEncoder) for content generation. They explored segment lengths, HTTP server settings, and the advantages and problems associated with shorter segment lengths. Akhshabi, Anantakrishnan, Begen, and Dovrolis (2012) investigated the performance problems arising when multiple adaptive streaming players compete for available bandwidth. They identified player instability, unfairness, and bandwidth underutilization as issues and explored the root causes, focusing on the behavior of adaptive streaming players in their Steady-State. Mueller, Lederer, Timmerer, and Hellwagner (2013) explored the implementation of MPEG-DASH over HTTP 2.0 (SPDY). They highlighted DASH's ability to handle varying bandwidth conditions and its benefits in terms of NAT and Firewall traversal, flexibility, scalability, and reduced infrastructure costs. They conducted experimental evaluations comparing HTTP 2.0 with HTTP 1.1 and HTTP 1.0, assessing protocol overhead, performance under different round trip times, and DASH performance in a lab test scenario.

2.1.3 Adaptive bitrate (ABR) algorithms

When streaming multimedia with DASH, one or more representations (i.e., versions at different resolutions or bit rates) of multimedia files are typically available, and selection can be made based on network conditions, device capabilities and user preferences. Many algorithms have been proposed to get the best bitrate selection. The bitrate adaptation mechanisms can be implemented at different stages of content delivery. Several surveys provide interesting insights into adaptive streaming (Kua, Armitage, & Branch, 2017; Sani, Mauthe, & Edwards, 2017; Bentaleb, Taani, Begen, Timmerer, & Zimmermann, 2019; Jedari et al., 2021) from different perspectives. Kua et al. (2017) looked at emerging research into the application of client-side, server-side, and in-network rate adaptation techniques to support DASH-based content delivery. Sani et al. (2017) presented a comprehensive survey of the most significant research activities in the area of client-side HTTP-based adaptive video streaming. Bentaleb et al. (2019) examined more schemes and classified them based on the unique features of their adaptation logics. More recently, Jedari et al. (2021) presented an in-depth survey on video edge-C3 (edge caching, computing, and communication) challenges and state-of-the-art solutions in next-generation wireless and mobile networks. In this section, we choose to focus on client-side ABR algorithms, but it is important to note that server/edge-side approaches to adaptive streaming have also been the subject of many research works, including optimizing the encoding parameters (Toni, Aparicio-Pardo, Simon, Blanc, & Frossard, 2014; Toni et al., 2015) coordinating proxies to ensure fairness between clients (Petrangeli, Famaey, Claeys, Latré, & De Turck, 2015), optimizing edge caching (C. Li, Toni, Zou, Xiong, & Frossard, 2018; A. Zhang et al., 2021), taking advantage of edge computing (X. Ma, Li, Jiang, Muntean, & Zou, 2022; X. Ma, Li, Zou, et al., 2022), and content centric networks (Monks, Olaru, & Muntean, 2019). Some recent work also propose to improve base station selection in various environments (Anand, Togou, & Muntean, 2021; W. Shi et al., 2021) to optimize video transmission. Regarding client-side ABR algorithms, we can define three categories of ABR strategies: rate-based, which base their decisions upon bandwidth esti-

mation, buffer-based, which base their decisions upon buffer level, and hybrid approaches (among which there can be learning-based approaches), which can use both the estimated bandwidth and the buffer level.

2.1.3.1 Rate-based algorithms

As most of the recent ABR algorithms are now hybrid, we discuss two early important works in rate-based ABR streaming. J. Jiang, Sekar, and Zhang (2012) proposed FESTIVE to address the challenges related to bitrate adaptation in commercial video players when multiple players share a bottleneck link. They presented a principled understanding of bitrate adaptation and analyzed various commercial players using an abstract player model. Through this analysis, they identified the root causes of issues related to efficiency, fairness, and stability in adaptive streaming over HTTP. Leveraging these insights, the paper proposed a set of techniques that systematically balanced stability, fairness, and efficiency, leading to a robust framework for video adaptation. The authors demonstrated the effectiveness of one specific technique from this framework, showing significant improvements over current commercial players across diverse experimental scenarios. Z. Li et al. (2014) then challenged the conventional approach to HTTP-based adaptive streaming (HAS) by highlighting the limitations of using TCP throughput as a reliable reference for video bitrate selection, especially when multiple HAS clients compete at a network bottleneck. They proposed a novel "probe and adapt" principle for video bitrate adaptation, introducing PANDA, a client-side rate adaptation algorithm for HAS, as a practical embodiment of this principle. Unlike traditional methods, PANDA conducted trial increments of the data rate (probe) without sending auxiliary piggybacking traffic, effectively reducing video bitrate oscillation and other undesirable behaviors by over 75% without increasing the risk of buffer underrun.

2.1.3.2 Buffer-based algorithms

Research in the ABR field quickly showed that solely relying on bandwidth estimation to make quality decisions had major drawbacks. We discuss these drawbacks and the proposed solutions. T.-Y. Huang, Handigol, Heller, McKeown, and Johari (2012) identified the challenge of accurate client-side bandwidth estimation above the HTTP layer in popular video streaming services like Hulu, Netflix, and Vudu. They discovered the "downward spiral effect", where inaccurate estimates lead to variable and low-quality video. After investigating this phenomenon and presenting its root causes, T.-Y. Huang, Johari, McKeown, Trunnell, and Watson (2014) introduced BBA, an innovative approach by questioning the necessity of continuous capacity estimation. Instead of relying on bandwidth estimation, they advocated using buffer occupancy to guide rate selection, demonstrating that capacity estimation is only crucial during the startup phase. This approach significantly reduced the rebuffer rate compared to Netflix's default algorithm, maintaining similar average video rates and achieving higher rates in steady state. Spiteri, Urgaonkar, and Sitaraman (2016, 2020) then introduced BOLA, an innovative online con-

trol algorithm formulated as a utility maximization problem, utilizing Lyapunov optimization techniques. Unlike previous approaches, BOLA didn't require prediction of network bandwidth and adaptively selected bitrates for video segments without freezing or degrading quality excessively. Their algorithm achieved near-optimal utility, outperforming existing methods in simulated network environments. BOLA's effectiveness was validated through extensive empirical testing, demonstrating significantly improved utility. Moreover, it had immediate practical impact as it became part of the standard reference player dash.js, adopted by major video providers like Akamai, BBC, CBS, and Orange, enhancing the user experience in real-world video streaming scenarios and contributing to the evolving DASH standard for video transmission.

2.1.3.3 Hybrid algorithms

Most of the state-of-the-art ABR algorithms now consider both the estimated bandwidth and the buffer level to make quality decisions, as it usually allows to get the best of both worlds. These ABR algorithms are sometimes referred to as "hybrid" or "buffer-aware". In this section, we discuss "traditional" hybrid ABR algorithms, and we focus on reinforcement learning (RL)–based algorithms in Sec. 2.1.3.4.

Notable early work in hybrid ABR algorithms for HTTP adaptive streaming made use of feedback control mechanisms (De Cicco, Mascolo, & Palmisano, 2011; G. Tian & Liu, 2012; De Cicco, Caldaralo, Palmisano, & Mascolo, 2013) to dynamically adapt the content bitrate in order to provide the maximum QoE, given the available bandwidth. In 2012, G. Tian and Liu (2012) studied the responsiveness and smoothness trade-off in DASH, and showed that client-side buffered video time was a good feedback signal to guide video adaptation. From these findings, they proposed novel hybrid ABR algorithms that balance the needs for video rate smoothness and high bandwidth utilization. A year later, De Cicco et al. (2013) introduced ELASTIC, a client-side controller aiming at eliminating the on-off traffic pattern generated by existing client-side algorithms and avoiding the unfairness and underutilization issues associated with shared network bottlenecks.

An important milestone was reached when X. Yin, Sekar, and Sinopoli (2014) framed adaptive bitrate streaming as a model-based predictive control (MPC) problem, addressing fundamental questions about objectives, environment signals, and algorithm sensitivity. By adopting a control-theoretic abstraction, it shed light on these critical aspects, enhancing clarity in this complex area. X. Yin, Jindal, Sekar, and Sinopoli (2015) subsequently built on this foundation by developing a principled control-theoretic model and introducing a novel model predictive control algorithm, named RobustMPC. This new algorithm optimally combined throughput and buffer occupancy information, surpassing traditional methods. Additionally, a practical implementation in a reference video player validated their approach, offering a robust solution for client-side bitrate adaptation.

Notable works posterior to MPC include:

- A new class of prediction-based adaptation (PBA) algorithms (Zou et al., 2015), where the authors investigated the potential improvement in video Quality of Experience (QoE) if accurate bandwidth prediction were possible in cellular networks.

By assuming knowledge of bandwidth for the entire video session or even a few seconds into the future, their study revealed that existing streaming algorithms achieved only 69%-86% of optimal quality. Importantly, the research demonstrated that while prediction alone might not be sufficient, combining prediction with rate stabilization functions significantly enhanced QoE, reducing the gap with optimal quality to just 4%.

- The segment-aware rate adaptation (SARA) algorithm (Juluri, Tamarapalli, & Medhi, 2016), which took into account the significant variation in segment sizes for a given video bitrate to accurately predict the time required to download the next segment and ensure seamless playback.

- The spectrum-based quality adaptation (SQUAD) algorithm (C. Wang, Rizk, & Zink, 2016), designed to address several critical issues that contributed to the degradation of DASH performance with respect to the rate control loops of DASH and TCP.

- The cross session stateful predictor (CS2P) (Y. Sun et al., 2016), built on insights from an analysis the throughput characteristics in a large dataset of 20M+ sessions. CS2P leveraged data-driven approach to learn clusters of similar sessions, an initial throughput predictor, and a hidden Markov model–based midstream predictor modeling the stateful evolution of throughput.

- Oboe (Akhtar et al., 2018), a new technique to auto-tune the parameters of various ABR algorithms, such as MPC (X. Yin et al., 2015), BOLA (Spiteri et al., 2016) or Pensieve (H. Mao, Netravali, & Alizadeh, 2017), to different network conditions.

- BOLA-E and DYNAMIC (Spiteri, Sitaraman, & Sparacio, 2018, 2019), proposed as improvements to the solely buffer-based BOLA algorithm, integrating bandwidth estimation as well as a FAST SWITCHING algorithm able to replace segments that had already been downloaded. These algorithms are now part of the official DASH reference player *dash.js* and are being used by video providers in production environments. Along with these ABR algorithms, the authors provided Sabre, an open-source publicly available software tool to simulate adaptive streaming environments.

- The throughput and buffer occupancy-based adaptation (TBOA) algorithm (Yaqoob, Bi, & Muntean, 2019), which, in contrast to previous works, increased the bitrate aggressively to make efficient use of the available bandwidth and waited for the buffer to cross a certain level before decreasing the bitrate to obtain a steady performance.

- The elastic DASH-based bitrate adaptation (EDRA) scheme (Togou & Muntean, 2022), which focused on reducing the number of bitrate switches that can damage the QoE. EDRA was implemented in Sabre (Spiteri et al., 2018) and improved QoE and utility compared to BOLA and DYNAMIC.

In 2020, Yan et al. (2020) presented findings from a comprehensive randomized controlled trial evaluating video-streaming algorithms for bitrate selection and network prediction. Their study, conducted over a year and involving substantial real-world video streaming to over 63,000 users, revealed the challenges faced by sophisticated or machine-learned control schemes when dealing with the complex, heavy-tailed nature of network and user behavior. Despite previous successes in emulators or simulators, these learned algorithms were shown to struggle in real-world settings. The authors subsequently introduced a robust ABR algorithm outperforming other schemes by combining supervised learning with data from real deployments, creating a probabilistic predictor of upcoming chunk transmission times. This predictor informed a classical control policy (MPC), leading to improved performance. Additionally, they provided an open platform, sharing data and results, inviting other researchers to explore and develop new algorithms for bitrate selection, network prediction, and congestion control in video streaming applications.

As MPC-based ABR algorithms use the predicted throughputs to solve a QoE maximization problem for the next chunk's optimal bitrate, their performance heavily relies on throughput prediction accuracy. To mitigate this issue, Kan et al. (2021) proposed to take the uncertainty of throughput prediction into account. They introduced BayesMPC, a novel ABR algorithm that combines Bayesian neural networks (BNNs) and MPC. Unlike traditional methods, BayesMPC employed a BNN-based predictor capable of capturing both aleatoric uncertainty (e.g., noise) and epistemic uncertainty (resulting from limited training samples) when predicting future network throughput. By minimizing the generalization error using the negative log-likelihood loss function, the algorithm established a confidence region for future throughput. This uncertainty-aware approach informed a robust MPC strategy that maximized worst-case user QoE within this confidence region. Experimental results using real-world network trace datasets demonstrated the efficiency of the BNN-based predictor and the uncertainty-aware robust MPC strategy. BayesMPC outperformed other baselines in terms of overall QoE performance and generalization across diverse network and user conditions.

This new type of uncertainty-aware approach constitutes the core motivation behind the DVMS framework we propose in chapter 3, where we propose to take the uncertainty of the viewport prediction into account with a different method, BNNs being computationally-heavy.

2.1.3.4 Reinforcement learning–based algorithms

Many approaches have also studied the benefits of reinforcement learning (RL)–based ABR algorithms. Even though these algorithms can be rate-based, buffer-based, or hybrid, we choose to separate them for clarity and because they usually have a different approach to the adaptive streaming problem. Early works considering the use of RL to make quality decisions wanted to address the issues of traditional ABR algorithms that relied on fixed control rules and simplified models, which were rigid and less flexible across various network configurations, leading to suboptimal solutions. The idea behind

RL-based ABR algorithm was to dynamically learn optimal behaviors according to the current network conditions.

In 2014, Claeys, Latré, Famaey, and De Turck (2014); Claeys, Latré, Famaey, Wu, et al. (2014) proposed adaptive (frequency-adjusted) Q-learning-based HAS clients, which utilized tunable reward functions, allowing focus on different aspects of QoE. In 2015, van der Hooft, Petrangeli, Claeys, Famaey, and De Turck (2015) followed the same ideas, incorporating RL to existing rate adaptation heuristics based on real-time bandwidth characteristics. They focused on reducing average buffer filling while maintaining a high QoE. In 2016, Chiariotti, D'Aronco, Toni, and Frossard (2016) aimed to maximize long-term expected user satisfaction by formulating the problem as a Markov Decision Process (MDP) optimization, where clients selected video representations considering decoded quality, quality fluctuations, and rebuffering events. In 2017, H. Mao et al. (2017) introduced Pensieve, a novel ABR algorithm for client-side video players based on RL. Pensieve dynamically generated ABR strategies through RL by training a neural network model to select bitrates for video chunks based on past performance data. Pensieve adapted to diverse network conditions and QoE metrics without pre-programmed assumptions about the environment. Experimental comparisons demonstrated that Pensieve outperformed state-of-the-art ABR algorithms in various scenarios, improving average QoE by 12% to 25% and demonstrating superior generalization even on networks it was not explicitly trained for.

More recently, T. Huang et al. (2019) proposed to improve existing ABR algorithms by selecting video chunks based on perceptual video qualities instead of bitrates, training policies through expert trajectory imitation, and employing lifelong learning to continually adapt to changing network conditions. The system utilized a quality-driven neural network architecture, a specialized dataset, and QoE metrics estimation. They showed significant improvements in sample efficiency and training time, while outperforming existing methods. To address the challenge of unstable ABR performance in heterogeneous network conditions, T. Huang, Zhou, Zhang, Wu, and Sun (2022) presented A^{2}BR (Adaptation of ABR), a novel meta-RL approach for ABR algorithms in internet video streaming. It employed an online and offline stage, utilizing meta-RL to learn an initial meta-policy in various network conditions offline and tailoring ABR decisions to personalized network conditions online. A^{2}BR continually optimized the meta-policy for specific network environments efficiently and demonstrated superior adaptation to personalized QoE metrics and specific network conditions, outperforming recent ABR methods.

2.1.4 Adaptive streaming for virtual reality

Immersive media accentuates the challenges of adaptive streaming, with harsher bandwidth and latency requirements to achieve a reasonable QoE. Many approaches to design VR-specific adaptive streaming systems have been proposed over the years. A large part of VR adaptive streaming approaches are viewport-adaptive and rely on viewport prediction. In this section, we focus on advances in VR streaming systems. Scientific literature on viewport prediction and trajectory prediction in general is investigated in

Sec. 2.2.1. Recently, several surveys have explored 360° video streaming under different angles (Z. Chen, Li, & Zhang, 2018; D. He, Westphal, & Garcia-Luna-Aceves, 2018; C.-L. Fan, Lo, Pai, & Hsu, 2019; Zink, Sitaraman, & Nahrstedt, 2019; Azevedo et al., 2020; Yaqoob, Bi, & Muntean, 2020; Shafi, Shuai, & Younus, 2020; M. Xu, Li, Zhang, & Le Callet, 2020; Ruan & Xie, 2021; Chiariotti, 2021; Rossi, Guedes, & Toni, 2023), providing interesting insights into state-of-the-art solutions for 360° video acquisition, compression, delivery, and rendering, as well as quality assessment, prediction and behavioral analysis in VR.

2.1.4.1 Early works and aggressiveness tradeoff

Before the advent of 360° videos (also called omnidirectional or panoramic videos) and virtual reality streaming, early work on adaptive streaming with layered views for free viewpoint applications (Toni, Thomos, & Frossard, 2013) and interactive multiview videos (De Abreu et al., 2015; Toni & Frossard, 2017; X. Zhang, Toni, Frossard, Zhao, & Lin, 2019) has been carried out to provide high-quality interactive experiences. Pioneering work on adaptive streaming for 360° videos (Qian, Ji, Han, & Gopalakrishnan, 2016; Bao, Wu, Zhang, Ramli, & Liu, 2016; Hosseini & Swaminathan, 2016b; Corbillon, Simon, Devlic, & Chakareski, 2017; Ozcinar, De Abreu, & Smolic, 2017; Petrangeli, Swaminathan, Hosseini, & De Turck, 2017) then demonstrated the feasability of streaming schemes that delivers 360° videos in a viewport-aware manner, spatially adapting the quality of the video to the user's viewport. The topic of optimizing 360° video delivery rapidly took center stage with proposals of optimal transmission strategies to maximize user QoE (Rossi & Toni, 2017; Chakareski, Aksu, Corbillon, Simon, & Swaminathan, 2018; Ben Yahia, Le Louedec, Simon, & Nuaymi, 2018), while some work focused on optimizing the encoding parameters (Corbillon, Devlic, Simon, & Chakareski, 2017; Ozcinar, De Abreu, Knorr, & Smolic, 2017).

Nasrabadi, Mahzari, Beshay, and Prakash (2017b, 2017a) claimed viewport-adaptive streaming was too aggressive because it prevented buffering future video chunks for a duration longer than the interval that user's viewport was predictable, which made the streaming scheme vulnerable to bandwidth variations and caused freezes due to rebuffering. To solve this issue, they proposed using scalable video coding, alleviating the restrictions on buffer duration. More recently, Polakovič, Rozinaj, and Muntean (2022) followed this idea and proposed to extend the DASH spatial relationship description (SRD) by adding scalable video encoding to spatial tiling in conjunction with a novel tile-layering–based gaze adaptation algorithm. Almquist, Almquist, Krishnamoorthi, Carlsson, and Eager (2018) characterized this problem as the prefetch aggressiveness tradeoff: *How far ahead in time from playback should we prefetch immersive content?* On the one hand, prefetching late allows to benefit from good viewport prediction, but the buffer size will not be sufficient to absorb bandwidth variations. On the other hand, prefetching early allows to fill the buffer but is not really viewport-adaptive as the viewport cannot be predicted more than a few seconds in advance. The authors presented an optimization-based comparison of the prefetch aggressiveness tradeoffs for different video categories and

proposed a novel system design that allows both tradeoff objectives to be targeted simultaneously. They also provided insights into how to best design future delivery systems for 360° videos, allowing content providers to reduce bandwidth costs and improve users' playback experiences.

2.1.4.2 Heuristic-based bitrate adaptation

In recent years, many heuristics for 360° video bitrate adaptation have been proposed, mostly in a tile-based manner. Qian et al. (2016); Qian, Han, Xiao, and Gopalakrishnan (2018) focused on mobile 360° video delivery, proposing cellular-friendly tile-based streaming schemes that delivers only visible portions of 360° videos, based on head movement prediction and online algorithms that determine which spatial portions to fetch and their corresponding qualities. Bao et al. (2016) proposed to use the deviation of the prediction to decide the amount of redundancy needed to be streamed. Ozcinar, Cabrera, and Smolic (2018b, 2018a, 2019); Ozcinar, İmamoğlu, Wang, and Smolic (2021) proposed to make use of visual attention (VA) maps, a type of saliency map. They presented a novel user-centric framework for 360° video delivery, optimizing DASH representations of 360° videos considering their VA maps and taking advantage of VA maps for bitrate allocation algorithm and dynamic tiling. Hooft, Vega, Petrangeli, Wauters, and Turck (2019) proposed tile-based rate adaptation heuristics for 360° videos and introduced a feedback loop in the quality decision process, which allows the client to revise prior decisions based on more recent information on the viewport location. Yaqoob and Muntean (2020, 2021); Yaqoob, Togou, and Muntean (2022); Yaqoob and Muntean (2023) proposed several tile-based heuristics based on simple and lightweight content-agnostic prediction mechanisms, defining different regions around the viewport to prioritize the bitrate allocation.

2.1.4.3 Machine learning–based bitrate adaptation

In addition to heuristic-based approaches to adaptive streaming in VR, many machine learning–based approaches have been investigated. X. Jiang, Chiang, Zhao, and Ji (2018) combined supervised deep learning with recurrent neural networks (RNNs) to predict the user's viewport with an RL-based agent to determine the optimal tile bitrates to be streamed. Feng, Swaminathan, and Wei (2019); Feng, Liu, and Wei (2020); Feng, Bao, and Wei (2021); Feng, Li, and Wei (2021) focused on live VR streaming as most of the existing viewport prediction approaches targeted only the video-on-demand (VOD) use cases, requiring offline processing of the historical video and/or user data that was not available in the live streaming scenario. They proposed several content-based machine learning approaches to reach high prediction accuracy, obtain significant bandwidth savings, and achieve real-time performance with low processing delays, meeting the bandwidth and real-time requirements of live VR streaming. Hou, Dey, Zhang, and Budagavi (2021) considered a remote rendering case and presented ultra-low latency viewport prediction with deep learning to dynamically adapt the encoding settings. S. Park, Hoai,

Bhattacharya, and Das (2021) used a 3-dimensional convolutional network (3DCNN) to extract spatio-temporal features of videos and predict the viewport, then applied RL to learn a streaming policy able to adapt to the predicted behavior of a viewer and the dynamics of the network conditions. S. Park, Bhattacharya, Yang, Das, and Samaras (2021) combined a deep learning–based viewport prediction with a rate control mechanism that assigned rates to different tiles in the 360° frame such that the QoE was optimized subject to a given network capacity. They modeled the optimization as a multi-choice knapsack problem and solved it using a greedy approach. Chopra, Chakraborty, Mondal, and Chakraborty (2021) presented PARIMA, employing a pyramid-based bitrate allocation scheme informed by an online viewport prediction model that used past viewports of users along with the trajectories of prime objects as a representative of video content to predict future viewports. C. Wu, Wang, and Sun (2021) proposed to use deep RL (DRL) in conjunction with a dual-queue streaming framework, enabling the DRL agent to determine and change the tile download order without incurring overhead. They also designed a preference-aware DRL algorithm to incentivize the agent to learn preference-dependent ABR decisions efficiently. R. Zhang et al. (2021) formulated the multi-user buffer-aware VR video streaming problem as a stochastic game and leveraged a DRL algorithm to solve the formulated resource block allocation problem in a distributed manner.

2.1.4.4 Caching and edge devices for VR streaming

Recent advances in VR streaming include new delivery mechanisms taking advantage and edge caching and computing. Hou et al. (2021) aimed to enable truly mobile VR using wireless HMDs with rendering performed on edge devices. For live 360° video streaming, L. Sun, Mao, Zong, Liu, and Wang (2020) proposed to assign variable playback latencies to all the users in a streaming session to form a "streaming flock", and to use information from users in the front of the flock to improve the performance of both viewport prediction and caching on the edge servers. Various works (J. Liu, Simon, Corbillon, Chakareski, & Yang, 2020; R. Zhang et al., 2021; Z. Chen et al., 2022) designed intelligent delivery mechanisms for VR streaming taking advantage of cache-aided Mobile Edge Computing (MEC) networks, considering viewport prediction and communication resource allocation. Xiao et al. (2022) proposed a novel transcoding-enabled VR video caching and delivery framework for edge-enhanced next-generation wireless networks. (Ye et al., 2023) proposed a QoE-driven viewport reconstruction-based 360° video caching solution for tile-adaptive streaming.

2.1.4.5 Other forms of VR adaptive streaming

While most of the work discussed here relies on splitting the video in tiles and changing the quality of the tiles to spatially adapt the quality to the viewport, Hristova, Simon, Corbillon, Devlic, and Swaminathan (2021) investigated alternatives to tile-based encoding and proposed two novel approaches that prepare heterogeneous quality versions of a 360° video based on the Gaussian pyramid and the Laplace pyramid. In recent years, research

has also been going towards VR adaptive streaming with more than three degrees of freedom (3DoF), such as multi-viewpoint (MVP) 360-degree videos (Corbillon, De Simone, Simon, & Frossard, 2018), volumetric data (J. Park, Chou, & Hwang, 2019), point clouds (Hosseini & Timmerer, 2018; van der Hooft, Wauters, De Turck, Timmerer, & Hellwagner, 2019), as well as augmented reality (AR) applications (Petrangeli, Simon, Wang, & Swaminathan, 2019).

2.2 Viewport and trajectory prediction

As seen in Sec. 2.1.4, viewport-adaptive streaming in VR relies on viewport prediction. In this section, we aim to provide useful background on viewport prediction, and we progressively expand the scope to human motion prediction, trajectory prediction, and time series forecasting–sequence modeling. We illustrate how the fields nest inside each other in Fig. 2.2.

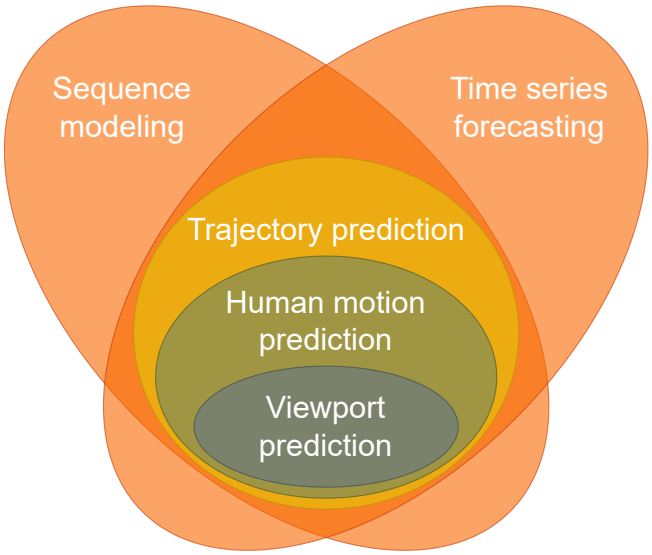


Figure 2.2: *Nested representation of the fields of viewport prediction, human motion prediction, trajectory prediction and time series forecasting–sequence modeling.*

Since the fields in Fig. 2.2 are very well-studied, we selected major works for viewport prediction as well as relevant works from the related fields that are interesting to consider for the scope of this thesis. Most of the approaches that we mention are deep learning approaches, as they are more relevant to our work and usually fare significantly better in

most benchmarks for trajectory prediction and time series forecasting tasks (Chang et al., 2019; H. Zhou et al., 2021).

2.2.1 Viewport prediction for 360° videos

In this section, we discuss some of the work that has been proposed for viewport prediction in 360° videos. We provide a list of viewport prediction methods that were proposed in the scientific literature in Table 2.1, sorted in chronological order of publication. This list contains the main methods until 2021, the time at which DVMS, the method we propose in chapter 3, was designed, as well as some more recent methods that we find relevant.

The first distinction we can make between methods for viewport prediction lies in how the problem is formulated. Some methods formulate this problem as a head trajectory prediction problem, and predict the orientation of the user's head, while other methods formulate this as a tile classification problem. We provide information about the prediction objective of the different methods in the "Objective" column. The "Prediction horizon" column tells us, for each method, how far ahead the prediction is done. For the selected methods, the interval is between 1 frame (30 milliseconds) and 10 seconds. The "Input modalities" column informs us on the type of data used to make predictions. Some methods rely on the past head orientations of the user, the past and/or future video frames, or even other users' known head orientations on the video. To further differentiate the prediction methods, we provide three binary columns:

- CU stands for "cross-users" and is set to "yes" if the viewport prediction method has access to data from other users who already watched the video. Comparison between CU and non-CU methods may not be fair, as CU methods have access to extra data. CU methods cannot work in a live streaming context (unless a technique like "flocking" (L. Sun et al., 2020) is used, which adds latency). CU is set to "no⋆" if the viewport prediction method cannot directly access other users' viewing data, but can benefit from some cross-user information, such as weight sharing in the case of federated learning.

- ML stands for "machine learning" and is set to "yes" if the method uses a model that learns parameters from the data. This includes linear regression, clustering, reinforcement learning or even deep learning methods.

- DL stands for "deep learning" and is set to "yes" if the method uses a deep neural network that learns parameters from the data. DL is set to "no⋆" if the method uses a pre-trained deep learning model as a feature extractor (such as the YOLO object detector), but does not train or propose any deep architecture.

Finally, we provide a short description of the method in the "Description" column, about the type of algorithm(s) used for prediction.

Let us now discuss some of the work presented in Table 2.1. Qian et al. (2016) were the first to predict head trajectories for 360° videos in a streaming context, and used

Table 2.1: List of methods for viewport prediction in 360° videos.

Reference	Objective	Prediction horizon	Input modalities	CU	ML	DL	Description
(Qian et al., 2016)	head orientation	2 seconds	head orientation	no	yes	no	linear regression
(Bao et al., 2016)	head orientation	1 second	head orientation	no	yes	no	linear regression / shallow MLP
(C.-L. Fan et al., 2017)	tile probabilities	1 second	head orientation, video frames	no	yes	yes	CNN + optical flow + LSTM
(Petrangeli et al., 2017)	head orientation	4 seconds	head orientation	no	no	no	constant movement
(Aladagli, Ekmekcioglu, Jarnikov, & Kondoz, 2017)	head orientation	2 seconds	video frame	no	no	no	GBVS saliency
(Y. Xu et al., 2018)	gaze position	1 second	gaze position, video frames	no	yes	yes	CNN + LSTM
(Ban et al., 2018)	tile probabilities	6 seconds	head orientation, other users head orientations	yes	yes	no	linear regression + KNN
(M. Xu, Song, et al., 2019)	head orientation	1 frame (30 ms)	head orientation, video frames	no	yes	yes	DRL + CNN + LSTM
(X. Jiang et al., 2018)	head orientation	3 seconds	head orientation	no	yes	yes	LSTM
(Nguyen, Yan, & Nahrstedt, 2018)	heatmap	2.5 seconds	head orientation, video frames	no	yes	yes	CNN + LSTM
(Qian et al., 2018)	head orientation	3 seconds	head orientation	no	yes	no	linear / ridge regression
(Y. Li, Xu, Xie, Ma, & Sun, 2019)	tiles in viewport	1 second	head orientation, video frames	no	yes	yes	CNN + optical flow + LSTM
(Petrangeli, Simon, & Swaminathan, 2018)	head orientation	10 seconds	head orientation, other users head orientations	yes	yes	no	clustering
(C. Li, Zhang, Liu, & Wang, 2019)	heatmap	10 seconds	head orientation, other users head orientations	yes	yes	yes	convLSTM + MLP
(Yu & Liu, 2019)	head orientation	3 seconds	head orientation	no	yes	yes	transformer
(Feng et al., 2019)	tiles in viewport	4 seconds	video frame	no	yes	no	GMM + Shi-Tomasi
(J. Park & Nahrstedt, 2019)	"view" probability	5 seconds	head orientation, other users head orientations	yes	yes	no	view transition model
(Feng et al., 2020)	tiles in viewport	2 seconds	head orientation, video frames	no	yes	yes	CNN + LSTM
(X. Zhang, Cheung, Le Callet, & Tan, 2020)	probability distribution of all positions	2 seconds	head orientation	yes	yes	no	view transition model
(Hou et al., 2021)	tile probabilities	0.2 second	head orientation	no	yes	yes	LSTM
(Nasrabadi, Samiei, & Prakash, 2020)	head orientation	10 seconds	head orientation, other users head orientations	yes	yes	no	clustering
(J. Chen, Luo, Hu, Wu, & Zhou, 2021)	tile probabilities	4 seconds	head orientation, other users head orientations	yes	no	no	hand-crafted
(S. Park, Bhattacharya, et al., 2021)	tile probabilities	2 seconds	head orientation, saliency maps, and motion maps	no	yes	yes	CNN+LSTM / 3DCNN+FC
(Feng, Bao, & Wei, 2021)	tiles in viewport	4 seconds	head orientation, video frames	no	yes	no*	YOLOv3 + Q-learning
(Romero Rondón, Sassatelli, Aparicio-Pardo, & Precioso, 2021)	head orientation	5 seconds	head orientation, saliency maps	no	yes	yes	LSTM
(Chopra et al., 2021)	head orientation	2 seconds	head orientation and video frames	no	yes	no*	YOLOv3 + ARIMA + passive aggressive regression
(Feng, Li, & Wei, 2021)	tiles in viewport	2 seconds	head orientation and video frames	no	yes	no*	3DCNN + Phrase2Vec
(Chao, Ozcinar, & Smolic, 2021)	head orientation	5 seconds	head orientation	no	yes	yes	transformer
(R. Zhang et al., 2021)	tiles in viewport	5 seconds	tiles in viewport	yes	yes	yes	DRL + ConvLSTM + federated learning
(Romero Rondon, Zanca, Melacci, Gori, & Sassatelli, 2021)	head orientation	5 seconds	head orientation	no	no	no*	physical model
(Chao, Ozcinar, & Smolic, 2022)	head orientation	5 seconds	head orientation	no*	yes	yes	transformer + federated learning

linear regression. C.-L. Fan et al. (2017) were the first to use saliency maps (in conjunction with motion maps and past head orientations) and long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997) deep networks to predict the future tiles to be seen by the user. We provide more background on saliency maps in Sec. 2.3.2. Yu and Liu (2019) were the first to propose an attention-based neural encoder-decoder (i.e., transformer (Vaswani et al., 2017)) to predict viewport in 360° videos. In 2020, as many viewport prediction methods had been proposed but were not comparing with each other, Romero Rondón, Sassatelli, Aparicio-Pardo, and Precioso (2020) proposed unified evaluation framework for these methods. They uniformized the sampling rate, the data formats and the metrics for several 360° datasets (C.-L. Fan et al., 2017; David, Gutiérrez, Coutrot, Da Silva, & Le Callet, 2018; Y. Xu et al., 2018; M. Xu, Song, et al., 2019; Nguyen et al., 2018; Y. Li et al., 2019). Building on this work, Romero Rondón et al. (2021) re-examined the existing deep learning models for viewport prediction in 360° videos and obtained the surprising result that they all performed worse than baselines using the user's trajectory. After analyzing the metrics, datasets and neural architectures, they identified the flaws of the existing methods and proposed a new model that used the user's past head orientations and the video content (not knowing other users' traces), establishing state-of-the-art performance.

While many deep learning methods have been proposed and proven to work well for head motion prediction, they can be perceived as "black boxes", with unexplainable outputs and internal mechanisms whose meaning are difficult to grasp. We find important to mention that J. Chen et al. (2021) and Romero Rondon et al. (2021), mentioned in Table 2.1, made a step in the opposite direction, proposing "white box" explainable models for viewport prediction that can be competitive with deep learning models, with parameters that have a physical meaning.

In contrast to previous "one-size-fits-all" approaches, some recent work has been investigating the use of federated learning (R. Zhang et al., 2021; Chao et al., 2022; Haseeb Ul Hassan, Brennan, Muntean, & McManis, 2023) to learn personalized user models for viewport prediction in a distributed manner. R. Zhang et al. (2021) and Chao et al. (2022) addressed the privacy concerns that arise with user data collection by constraining the personal scanpath data to the client-side.

The work we present in chapter 3 aims to better consider the uncertainty and randomness of human motion, and takes inspiration from other works in trajectory prediction, especially the idea of predicting multiple possible trajectories, described in Sec. 2.2.2.2 and Sec. 2.2.2.3.

2.2.2 Trajectory prediction

In this section, we provide some background on deep learning approaches to trajectory prediction. We first discuss models that predict the movements of the human body. We then focus on "social" trajectory prediction of human agents. Finally, we explore related work on vehicle trajectory prediction.

2.2.2.1 Human motion prediction

The work discussed in this section explicitly models the human body as part of the motion prediction.

Fragkiadaki, Levine, Felsen, and Malik (2015) proposed the encoder-recurrent-decoder (ERD) model for recognition and prediction of human body pose in videos and motion capture. ERDs extended previous LSTM models in the literature, incorporating nonlinear encoder and decoder networks before and after recurrent layers to jointly learn representations and their dynamics. The authors showed that representation learning was crucial for both labeling and prediction in space-time. Martinez, Black, and Romero (2017) showed that state-of-the-art performance could be achieved by a simple baseline that does not attempt to model motion at all. After investigating this result, analyzing previous RNN methods by looking at the architectures, loss functions, and training procedures, they proposed changes to the standard RNN models typically used for human motion, which resulted in a simple and scalable RNN architecture that obtained state-of-the-art performance on human motion prediction. Cao et al. (2020) took advantage from the fact that human movement is goal-directed and influenced by the spatial layout of the objects in the scene to design a three-stage framework that exploits scene context to predict human motion. Given a single scene image and 2D pose histories, their method first sampled multiple human motion goals, then planned 3D human paths towards each goal, and finally predicted 3D human pose sequences following each path.

Recent advances in pose forecasting aim to improve spatio-temporal modeling of human motion (Adeli et al., 2021; Z. Liu et al., 2021; Sofianos, Sampieri, Franco, & Galasso, 2021; Parsaeifard, Saadatnejad, Liu, Mordan, & Alahi, 2021). Adeli et al. (2021) proposed TRiPOD, a method based on graph attentional networks to model the human-human and human-object interactions both in the input space and the output space (decoded future output). Z. Liu et al. (2021) advocated to model motion contexts in the trajectory space instead of the traditional pose space, and used a semi-constrained graph convolution network (GCN) to explicitly encode skeletal connections and prior knowledge. Sofianos et al. (2021) also used a GCN to model human body dynamics, but proposed to factor the space-time graph connectivity into space and time affinity matrices, which bottlenecked the space-time cross-talk, while enabling full joint-joint and time-time correlations. Parsaeifard et al. (2021) proposed to learn decoupled representations for the global and local pose forecasting tasks, by using an LSTM encoder-decoder network for trajectory forecasting and a variational auto-encoder (VAE) to solve the local pose forecasting task.

When it comes to motion prediction in virtual environments, Gomes, Rossi, and Toni (2021) tackled the problem of point cloud prediction by proposing an end-to-end learning network to predict future frames in a point cloud sequence. Their pipeline included an initial layer learning topological information of point clouds as geometric features, followed by multiple Graph-RNN cells which learned point dynamics by processing each point jointly with its spatiotemporal neighbours. Zheng et al. (2022) studied of the benefits of leveraging the eye gaze for ego-centric human motion prediction with various

state-of-the-art architectures, arguing that eye gaze that served as a surrogate for inferring human intent. To realize the full potential of the gaze-informed prediction, they proposed a novel network architecture that enabled bidirectional communication between the gaze and motion branches.

2.2.2.2 Social trajectory prediction

In this section, we discuss various approaches to human trajectory prediction in social contexts. Pedestrians follow different trajectories to avoid obstacles and accommodate fellow pedestrians. Predicting the trajectories of pedestrians is a challenging problem due to the inherent properties of human motion in crowded environments. Following the recent success of RNN models for sequence prediction tasks, Alahi et al. (2016) proposed an LSTM model coupled with a "Social" pooling layer which could learn general human movement, typical interactions, and predict future trajectories. This was in contrast to traditional approaches which used hand-crafted functions.

Pedestrian trajectories are inherently multimodal: given a partial history, there is no single correct future prediction. For this reason many approaches started to generate multiple plausible future trajectories instead of one (Lee et al., 2017; Gupta, Johnson, Fei-Fei, Savarese, & Alahi, 2018; Sadeghian et al., 2019; Amirian, Hayet, & Pettré, 2019; Y. Huang, Bi, Li, Mao, & Wang, 2019; Ivanovic & Pavone, 2019; Kosaraju et al., 2019; Mangalam et al., 2020; Dendorfer, Ošep, & Leal-Taixé, 2020; H. Zhao & Wildes, 2021; H. Zhao et al., 2021). Lee et al. (2017) proposed to account for the multimodal nature of the future prediction by sampling potential future outcomes from the latent space of a conditional VAE (CVAE). Gupta et al. (2018) instead used a generative adversarial network (GAN) in conjunction with a variety loss to ensure the diversity of predictions. The use of such a loss function for trajectory prediction was analyzed by Thiede and Brahma (2019). Sadeghian et al. (2019) built on this idea and designed an attention module to incorporate context information in addition to the past agents' trajectories. While most of the existing methods ignored the temporal correlations of interactions between pedestrians, Y. Huang et al. (2019) proposed a spatial-temporal graph attention network to better capture spatio-temporal interactions. More recent approaches to trajectory prediction adopt a goal-driven approach, where a set of possible destinations of the agent are first estimated, and a set of possible trajectories going towards these goals are then generated (Dendorfer et al., 2020; H. Zhao & Wildes, 2021; H. Zhao et al., 2021).

2.2.2.3 Vehicle trajectory prediction

The idea to generate multiple plausible future trajectories, which can be applied to viewport prediction, as discussed in chapter 3, was also explored to predict the trajectories of vehicles, particularly in self-driving contexts. Similar to pedestrian trajectory prediction, vehicle trajectory prediction is an active research area (Chang et al., 2019). Some of the models discussed in Sec. 2.2.2.2 can actually predict both pedestrian and vehicle trajectories (Lee et al., 2017; H. Zhao et al., 2021). Generated vehicle trajectories are also

multimodal, as vehicles can choose to go in different directions, given a common past trajectory. While several approaches to vehicle trajectory prediction had been proposed, the best-performing ones which required extremely detailed input representations did not generalize to datasets they had not been trained on. Srikanth et al. (2019) proposed to use scene semantics as intermediate latent representations in their, as it allowed zero-shot transfer to unseen datasets. Marchetti, Becattini, Seidenari, and Del Bimbo (2020a) proposed to consider trajectory multimodality in a novel explainable way by using memory augmented neural networks, storing distinct past and future trajectories in a key-value register during training, and retrieving similar pasts to generate new futures at test time. We adapt this approach to predict multiple trajectories of head motion in chapter 3.

2.2.3 Sequence modeling and time series forecasting

In this section, we explore the most general case of sequence modeling and prediction. While time series forecasting can arguably be considered as a subset of sequence modeling, we choose to consider them separately for historical reasons. The term "sequence modeling" is mostly used to describe recent sequence-to-sequence deep models for natural language processing (NLP), while "time series forecasting" usually denotes a more traditional prediction task based on historical time stamped data.

2.2.3.1 Sequence modeling

Sequence modeling is the ability to model, interpret, make predictions about or generate any type of sequential data, such as audio, text, video, etc. In this section, we discuss some recent and influential work in sequence modeling.

Since LSTM networks (Hochreiter & Schmidhuber, 1997) were proposed in 1997 to solve the vanishing gradient problem, deep neural networks have come a long way. We first discuss recent improvements to RNNs. Sutskever, Vinyals, and Le (2014) were the first to present sequence-to-sequence encoder-decoder networks, a general end-to-end approach to sequence learning that makes minimal assumptions on the sequence structure, now widely used for a variety of tasks. Chung et al. (2015) proposed to include of latent random variables into the hidden state of RNNs by combining the elements of the variational auto-encoder to address sequence modeling problems. Realizing the power of high-level spatio-temporal graphs and sequence learning success of RNNs, Jain, Zamir, Savarese, and Saxena (2016) designed a generic method for casting spatiotemporal-graph as rich, scalable, and jointly trainable RNN mixtures. As many extensions of the VAE model to process sequential data with RNNs were proposed, Girin et al. (2021) provided a comprehensive review of a newly identified class of models: dynamical variational auto-encoders (DVAEs). We propose to explore DVAE models and what they can bring to head motion prediction in Sec. 3.7.2.

In recent years, transformer-based models (Vaswani et al., 2017) have become indispensable, as they now dominate most NLP tasks, as well as many computer vision tasks. Vaswani et al. (2017) first proposed the transformer model, based solely on atten-

tion mechanisms (Bahdanau, Cho, & Bengio, 2014). This new architecture avoided the recursion of RNNs, processed sentences as a whole, and learned relationships between words thanks to multi-head attention mechanisms and positional embeddings. Avoiding recursion allowed parallel computation (to reduce training time) and also improved performance for long dependency modeling. Devlin, Chang, Lee, and Toutanova (2019) then introduced masked language model (MLM) pre-training objective to learn deep bidirectional representations from unlabeled text. (Raffel et al., 2020) presented a comprehensive perspective on pre-training and transfer learning in NLP. They introduced a unified framework able to convert all text-based language problems into a text-to-text format and compared pre-training objectives, architectures, unlabeled data sets, transfer approaches, and other factors on dozens of language understanding tasks. Recently, transformer-based models have also been applied to pedestrian (see Sec. 2.2.2.2) and vehicle (see Sec. 2.2.2.3) trajectory prediction successfully (Giuliari, Hasan, Cristani, & Galasso, 2021; Y. Liu, Zhang, Fang, Jiang, & Zhou, 2021; Yuan, Weng, Ou, & Kitani, 2021; S. Shi, Jiang, Dai, & Schiele, 2022; Nayakanti et al., 2023).

Many transformer-based models have now been proposed, as the proposed multi-head self-attention is a very flexible architectural block that makes few assumptions about the relationship between its inputs. The weak inductive bias of transformer-based models allow them to outperform traditional CNNs and RNNs at sequence modeling tasks, given enough data. However, the all-to-all design of self-attention blocks makes transformer-based models scale quadratically with the number of inputs, in terms of both memory and computation. Several approaches to improve transformer efficiency have been proposed. Child, Gray, Radford, and Sutskever (2019) proposed sparse factorizations of the attention matrix which reduce the complexity. S. Wang, Li, Khabsa, Fang, and Ma (2020) proposed to further reduce the complexity of self-attention by demonstrating that the self-attention mechanism could be approximated by a low-rank matrix. Exploiting this finding, they proposed *Linformer*, based on a new self-attention mechanism that scales linearly with the number of inputs. Instead of changing the attention mechanism, Jaegle et al. (2021) proposed to use a cross-attention module to project an high-dimensional input byte array to a fixed-dimensional latent bottleneck, before processing it using a deep stack of transformer-style self-attention blocks in the latent space. The resulting architecture, named *Perceiver* was then extended to work with any kind of inputs and outputs (Jaegle et al., 2022), in an autoregressive mode (Hawthorne et al., 2022), and even in vision-language tasks (Z. Tang, Cho, Lei, & Bansal, 2023).

Due to their quadratic self-attention complexity transformers, do not scale very well to long sequence length. Recently, deep state-space models (SSMs) have been getting traction for very long sequence modeling thanks their efficient encoding of the recurrent structure. Deep SSMs leverage the state-space representation of linear systems that is commonly used in control theory, and propose to learn automatically the state, input, and output matrices. Gu et al. (2021) showed that deep SSMs actually struggle even on simple tasks, but can perform exceptionally well when equipped with HiPPO matrices (Gu, Dao, Ermon, Rudra, & Ré, 2020), special state matrices recently derived to solve a problem of continuous-time memorization. Gu, Goel, and Re (2022) proposed the first truly success-

ful deep SSM, the structured state space sequence model (S4), outperforming all previous transformer-based models on long range sequence modeling by a wide margin. Several extensions and variations to this model have since been proposed (Gupta, Gu, & Berant, 2022; Gu, Goel, Gupta, & Ré, 2022; Gu, Johnson, Timalsina, Rudra, & Re, 2023; Smith, Warrington, & Linderman, 2023).

2.2.3.2 Time series forecasting

Time series forecasting describes the general case of using a model to predict future values based on previously observed values. Yule (1921) first identified this as the "time-correlation problem", "the problem of elucidating (...) the relations subsisting between two quantities varying with the time", and proposed the aurogressive model (Yule, 1927) analyze Wolf sunspot numbers. With large amounts of consistent, quality data becoming available, more refined and complex time series forecasting models ermerged. Whittle (1951) proposed the autoregressive–moving-average (ARMA) model, combining an autoregressive model and a moving-average model, to describe stationary stochastic processes. Its generalization, ARIMA (Box & Jenkins, 1970), adds an initial differencing step to eliminate the non-stationarity of the mean function. ARIMA models have been a popular choice for time series forecasting, including stock price prediction (Ariyo, Adewumi, & Ayo, 2014). Other traditional time series forecasting approaches include exponential smoothing methods (Hyndman, Koehler, Snyder, & Grose, 2002) and random walk models (Kilian & Taylor, 2003). In 2018, Taylor and Letham (2018) proposed a practical approach to forecasting "at scale" using a modular regression model with interpretable parameters that can be intuitively adjusted by analysts with domain knowledge about the time series, outperforming previous traditional forecasting methods.

Recently, deep learning models have also been applied to univariate and multivariate time series forecasting. Lai, Chang, Yang, and Liu (2018) proposed the long- and short-term time-series network (LSTNet) for multivariate time series forecasting, combining RNNs and CNNs to extract short-term local dependency patterns among variables and to discover long-term patterns for time series trends. Oreshkin, Carpov, Chapados, and Bengio (2020) focused on univariate times series forecasting and proposed a deep architecture based on backward and forward residual links, as well as a very deep stack of fully-connected layers. As transformer-based models have become more popular in recent years, new transformer-based architectures have been proposed for time series forecasting. (S. Li et al., 2019; H. Zhou et al., 2021; Lim, Arik, Loeff, & Pfister, 2021; H. Wu, Xu, Wang, & Long, 2021; S. Liu et al., 2022; T. Zhou et al., 2022). S. Li et al. (2019) addressed two major weaknesses of the transformer model for time series forecasting: the locality-agnosticism due to point-wise dot product self-attention and the memory bottleneck due to the quadratic scaling problem discussed in Sec. 2.2.3.1. To solve these issues, they proposed to produce queries and keys with causal convolution to better consider local context and used a sparse self-attention mechanism (instead of all-to-all quadratic attention). H. Zhou et al. (2021) also used a sparse attention mechanism in combination with self-attention distilling to privilege dominating attention scores and further reduce

the complexity, as well as a generative decoder to directly predict multiple time-steps and avoid error accumulation, a common issue of autoregressive models. Lim et al. (2021) proposed to use recurrent layers for local processing in addition to the self-attention layers, better suited for long-term dependencies. Their architecture also included specialized components to select relevant features and a series of gating layers to suppress unnecessary components. Instead of using sparse attention, H. Wu et al. (2021) designed an auto-correlation mechanism inspired by stochastic process theory. S. Liu et al. (2022) explored multiresolution representations of time series by introducing a pyramidal attention module to summarize features at different resolutions and model the temporal dependencies of different ranges. T. Zhou et al. (2022) proposed to combine transformer with seasonal-trend decomposition method based on the Fourier transform.

While these transformer-based models have demonstrated considerable prediction accuracy improvements over traditional methods for time series forecasting, A. Zeng, Chen, Zhang, and Xu (2023) showed that simple one-layer linear models outperformed existing sophisticated transformer-based models, questioning the use of transformers for long-term time series forecasting. The authors found that most of the accuracy improvements came form direct multi-step (DMS) forecasting, compared to the traditional autoregressive iterated multi-step (IMS) forecasting. Following their findings, we develop a new DMS baseline for viewport prediction in chapter 6.

Recent trends in time series forecasting include using patch embeddings to represent time series (Nie, Nguyen, Sinthong, & Kalagnanam, 2023; Gong, Tang, & Liang, 2023) following the vision transformer (ViT) idea (Dosovitskiy et al., 2021), and incorporating stationary processeses into hierarchical structures with specialized attention mechanisms (Y. Yang et al., 2023).

2.3 Behavior, attention and influencing factors in VR

Understanding what drives user behavior and attention in VR is key to design better viewport prediction. We illustrate the interactions between content, emotions and user behavior in VR in Fig. 2.3. Immersive content directly affects the user's behavior by driving their visual attention. The content also affects the emotions felt by the user in VR, which in turn can affect their behavior. We call "influencing factors" some properties of the immersive content and the user emotional state that can significantly impact user behavior and, ultimately, quality of experience (QoE). In this section, we first explore studies that investigated human behavior in VR. Second, we focus on the influence of visual attention and saliency estimation in VR. Finally, we discuss the importance of the users' emotions in VR.

2.3.1 Human behavior in virtual environments

We first mention interesting work regarding human behavior in 360° videos. Almquist et al. (2018), who described the prefetch aggressiveness tradeoff discussed in 2.1.4.1,

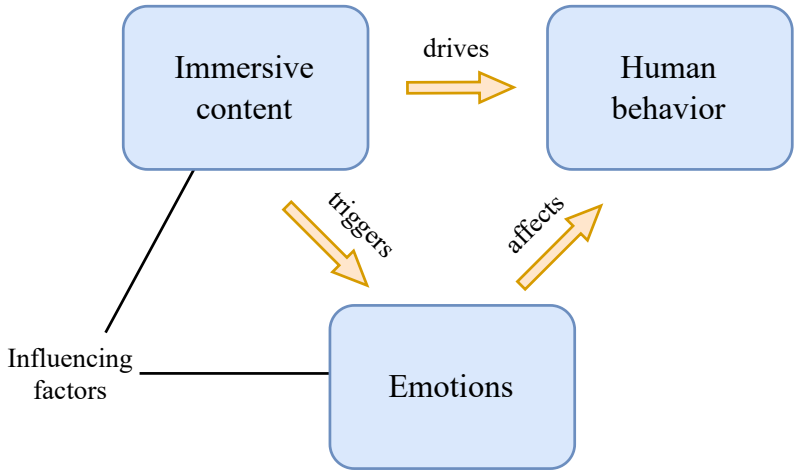


Figure *2.3: The interactions between content, emotions and user behavior in VR.*

proposed a taxonomy of 360° videos and characterized the user behavior for four different video categories: *static focus*, *moving focus*, *ride*, and *exploration*. They found that, while head movement was highly predictable in short time ranges, the predictability significantly differed among the video categories in longer time ranges. Rossi, De Simone, Frossard, and Toni (2019) proposed a graph-based method to identify clusters of users based on their common navigation patterns to better understand the behavior of users watching 360° videos. Rossi, Ozcinar, Smolic, and Toni (2020) investigated users' similarities when navigating within immersive content with different devices and found key differences of users' behavior across devices and content categories. Rossi and Toni (2020) highlighted the importance of looking at users' trajectories instead of more qualitative measures of users' interactions by studying the intra- and and inter-user variability of users in a VR system, using trajectory-based metrics adapted from information theory. This allowed them to identify consistent patterns across different contents (intra-user) and show that the transfer entropy better quantified behavioural similarity among users watching the same content rather than metrics based on spatial distribution (inter-user). A profusion of datasets of people watching 360° videos have now been proposed to test viewport prediction methods and study user behavior (Corbillon, De Simone, & Simon, 2017; Lo et al., 2017; C. Wu, Tan, Wang, & Yang, 2017; C.-L. Fan et al., 2017; David et al., 2018; Y. Xu et al., 2018; M. Xu, Song, et al., 2019; Nguyen et al., 2018; Y. Li et al., 2019; Nasrabadi et al., 2019; Rossi et al., 2020; Jin, Liu, Wang, & Cui, 2022).

Human behavior in six degrees of freedom (6DoF) virtual environments is also getting investigated (Zerman, Kulkarni, & Smolic, 2021; Rossi, Viola, Jansen, et al., 2021; Rossi, Viola, Toni, & Cesar, 2021, 2023). Zerman et al. (2021) analyzed the user behaviour for volumetric video consumption in an augmented reality (AR) setting. They showed that users spent most of their time looking at the frontal part of the volumetric video, indicating the importance of faces in visual attention. Rossi, Viola, Jansen, et al. (2021) investigated how users are affected by salient agents and narrative elements of the VR movie, showing that the motion during the VR experience was affected by the storytelling. While

more static and focused behaviour happened when participant had to complete a given task, exploration movements were more frequent when virtual characters were talking in the scene. Rossi, Viola, Toni, and Cesar (2021) showed the limitations of clustering algorithms for 3DoF in assessing user similarity in 6DoF and advocated needing new solutions to analyze 6DoF trajectories.

In chapter 5, we propose several contributions with the objective to gain a better understanding of user behavior in VR. Specifically, we investigate the link between immersive content, attention, emotions, and movements in VR. We discuss visual attention and emotions in VR in Sec. 2.3.2 and Sec. 2.3.3, respectively.

2.3.2 Saliency and attention in immersive media

In order to understand and predict user behavior in VR, it is crucial to know what attracts the users' attention. Visual attention in images and videos can be expressed with saliency maps, which highlight salient areas, areas that attract the viewer's visual attention. Saliency estimators aim to predict salient areas by generating saliency maps. While saliency maps are useful tools to predict human behavior in VR, they are also used objective quality assessment (Rai, Le Callet, & Guillotel, 2017), in order to efficiently drive encoding algorithms. In this section, we discuss recent work on visual attention and saliency estimation in VR.

Saliency estimators and their relationship with gaze and attention are well-studied (Itti, Koch, & Niebur, 1998; Cerf, Harel, Einhaeuser, & Koch, 2007; M. Jiang, Huang, Duan, & Zhao, 2015; Rai, Le Callet, & Cheung, 2016; Chaabouni & Precioso, 2019) and we choose to not go into details for the sake of brevity. However, while there are differences in the gaze patterns (fixations and saccades) between flat screen presentations and immersive content viewed in VR headset (David, Lebranchu, Perreira Da Silva, & Le Callet, 2022), research on saliency estimation and visual attention in 360° images and videos builds on existing saliency models. Recent research on saliency estimation in VR was enabled thanks to dedicated tools and datasets (Rai, Gutiérrez, & Le Callet, 2017; Ozcinar & Smolic, 2018; Gutiérrez, David, Coutrot, Da Silva, & Le Callet, 2018; Gutiérrez, David, Rai, & Le Callet, 2018).

2.3.2.1 Saliency in 360° images

Before the advent of immersive content and 360° images, traditional saliency models had already been extended to consider the depth factor of stereoscopic images, using machine learning (J. Wang, Da Silva, Le Callet, & Ricordel, 2013; Fang, Lin, et al., 2014) or hand-crafted features (Fang, Wang, Narwaria, Le Callet, & Lin, 2014). To generate saliency maps for saliency images De Abreu, Ozcinar, and Smolic (2017) proposed to use the viewport center (i.e., head motion) when gaze tracking data was not available. They also proposed fused saliency maps (FSM), a method to adapt previous saliency models considering the equatorial bias in 360° image viewing. Sitzmann et al. (2018) also showed the existence of a particular fixation bias in VR, which they then used to

adapt existing saliency estimators to 360° images. Monroy, Lutz, Chalasani, and Smolic (2018) used deep learning to estimate saliency by extending traditional 2D deep saliency estimators (CNNs) to 360° images. Battisti, Baldoni, Brizzi, and Carli (2018); Battisti and Carli (2019); Mazumdar and Battisti (2019); Mazumdar, Arru, Carli, and Battisti (2019); Mazumdar, Lamichhane, Carli, and Battisti (2019) proposed several models for saliency estimation in 360° images using hand-crafted features, such as low-level saliency features, depth information, and higher-level features such as human faces and semantic information. Mazumdar, Arru, Carli, and Battisti (2021) further investigated the influence of human faces on visual attention in 360° images. Their study confirmed previous research on saliency estimation in 2D images, as they observed that the presence of faces also attracts human attention in 360° images. However, the authors also found that giving equal importance to all the detected faces did not improve saliency estimation, showing that more refined models were needed.

2.3.2.2 Saliency in 360° videos

Recently, Chao, Battisti, Lebreton, and Raake (2023) presented a comprehensive literature review of omnidirectional (i.e., 360°) video saliency, providing information on the different approaches that have been taken, and key challenges that have been raised compared to traditional 2D contents.

Ozcinar and Smolic (2018) were the first to propose a new dataset for 360° video saliency estimation. They analyzed viewer behavior and compared the performance of state-of-the-art 2D and 360° image saliency models. Many proposed to estimate 360° video saliency with deep learning in the following years (Cheng et al., 2018; Z. Zhang, Xu, Yu, & Gao, 2018; Nguyen et al., 2018; Chao, Ozcinar, Zhang, et al., 2020; Qiao, Xu, Wang, & Borji, 2021; Dahou, Tliba, McGuinness, & O'Connor, 2021; Yun, Lee, & Kim, 2022; Q. Yang et al., 2023; Cokelek, Imamoglu, Ozcinar, Erdem, & Erdem, 2023). Cheng et al. (2018) proposed to solve the saliency estimation problem with weakly-supervised deep neural network using a new ConvLSTM-based cube padding technique, which extended CNN architectures to accommodate the distortions inherent to 360° videos. Z. Zhang et al. (2018) adapted the U-Net (CNN) architecture by defininig the convolutional on a spherical crown that rotated along the 360° sphere. Nguyen et al. (2018) leveraged pre-trained 2D saliency models to improve 360° video saliency estimation. Qiao et al. (2021) had the idea to generate viewport-dependent saliency maps, unlike previous methods that generated full frame saliency maps, and achieved this through the use of a multi-task deep neural network that considered both the viewport and the full 360° video frame. Dahou et al. (2021) proposed a two-stream architecture, leveraging an attention mechanism and "expert" models to improve over previous approaches. Recently, vision transformer (ViT)–based architectures have been adopted for 360° video saliency estimation, taking advantage of powerful pre-trained models (Yun et al., 2022; Cokelek et al., 2023).

All of the models we just mentioned only consider visual content as an input to generate saliency maps (and may train on gaze data). Other VR-specific modalities, such as

directional sound, might influence visual attention. Chao, Ozcinar, Wang, et al. (2020); Hirway, Qiao, and Murray (2022); J. Li, Zhai, Zhu, Zhou, and Zhang (2022); Singla et al. (2023); Q. Yang et al. (2023) presented new audio-visual datasets of people watching with different types of audio (usually no audio, mono audio, or ambisonics). As expected, Chao, Ozcinar, Wang, et al. (2020) found that compared to only perceiving visual cues, perceiving visual cues with salient object sound could draw more visual attention to the objects making sound and guide viewing behaviour when such objects were not in the user's field of view. Hirway et al. (2022) found that the participants paid attention to a wider range of visual elements when the sound was spatial in nature. J. Li et al. (2022) found that visual attention was drawn to and concentrated on the sound source with the presence of sound, especially when there were several visually salient objects and only one sound source. Singla et al. (2023) also found that subjects concentrated more on the directions sound was coming from when higher-order ambisonics was played. They also noted that the different types of audio did not have an influence on cybersickness. Following these studies, Chao, Ozcinar, Zhang, et al. (2020); Q. Yang et al. (2023) proposed new audio-visual saliency models, improving over previous models that only considered the video. Due to the novelty of these approaches and the scarcity of datasets, which makes it more difficult to develop and compare new models, we did not choose to consider directional sound in our work. However, we believe that this modality should be considered in future work on viewport prediction.

2.3.3 Emotions in virtual reality

While video content and visual saliency play an important role in driving human attention and behavior in VR, modeling the effect of emotions might lead to a better understanding of the users' behavior (Schupp et al., 2007; S. Fan et al., 2018). Emotions can be described with a two-dimensional circumplex model (Russell, 1980), using valence as the x-axis (positiveness-negativeness of emotions) and arousal as the y-axis (intensity of emotions). These can be measured with subjective methods (Mehrabian & Russell, 1974; Bradley & Lang, 1994) and physiological signals (Bradley & Lang, 2000; Nasoz, Alvarez, Lisetti, & Finkelstein, 2004).

Baños et al. (2008) explored the impact of stereoscopy on presence and emotions in virtual environments. While previous literature had shown that stereoscopic displays enhanced subjective feelings of presence, they found no significant influence, as similar emotional reactions are elicited by both monoscopic and stereoscopic presentation. However, the authors confirmed that a strong sense of presence was correlated with strong emotional reactions. Felnhofer et al. (2015) then used virtual environments to induce specific emotional states and showed that the type of emotional reaction did not have an effect on the level of presence. They also found that electrodermal activity (EDA) seemed to be a poor indicator of presence as it was not significantly correlated with self-reported presence.

In recent years several tools and datasets have been proposed (B. J. Li, Bailenson, Pines, Greenleaf, & Williams, 2017; W. Tang, Wu, Vigier, & Da Silva, 2020; Toet, Heijn,

Brouwer, Mioch, & van Erp, 2020; Xue, Ali, Zhang, Ding, & Cesar, 2021) to investigate emotions and their influence on user behavior in VR. In these studies, people were watching 360° videos in a VR headset and the emotions were measured using subjective metrics such as self-assessed valence and arousal (Bradley & Lang, 1994) and objective physiological measurements, such as EDA (Boucsein, 2012). B. J. Li et al. (2017) found that the standard deviation of yaw (horizontal head movements) positively correlated with valence, while a significant positive relationship was found between head pitch (vertical movements) and arousal. While Pallavicini, Pepe, and Minissi (2019) found that, compared to 2D displays, playing video games with head-mounted displays led to a higher sense of presence and self-reported feelings of happiness and surprise, Voigt-Antons et al. (2020) found no significant differences in presence and emotional ratings between 2D and head-mounted displays. W. Tang et al. (2020) observed that the presence of negative images had a significant impact on visual attention, resulting in more visual agitation and avoidance behavior from larger, longer and faster saccades when people were showed negative 360° images. Barreda-Ángeles, Aleix-Guillaume, and Pereda-Baños (2020) proposed to investigate the claim of VR being an "empathy machine", as it had been described before. While they observed a direct positive effect of spatial presence on perspective taking and empathic concern, they also observed an indirect negative effect of immersive presentation on empathic concern through enjoyment. This meant that the enjoyment of pleasurable aspects associated to VR experiences may hinder the affective dimension of empathy toward the characters, pointing out the need to carefully consider the targeted reactions from the audience when designing VR experiences. Xue, El Ali, Zhang, Ding, and Cesar (2021) proposed a new tool to collect real-time, continuous emotion annotation in 360° videos and used it to collect a new dataset (Xue, Ali, Zhang, et al., 2021), also containing behavioral and physiological information, as well as subjective ratings. From this data, (Xue, Ali, Ding, & Cesar, 2021) found significant correlations between head motion rotation data, as well as some eye movement features, with valence and arousal ratings. They also showed that their new fine-grained emotion labels provided greater insight into how head and eye movements related to emotions during 360° video watching in VR. Jicol et al. (2021) studied the effects of human factors such as emotions and agency on the sense of presence. They showed that the dominant emotion induced by a virtual environment was positively correlated with presence, and that agency had a significant positive effect on presence, while also moderating the effects of emotions on presence.

The dataset we present in chapter 5 and the dataset presented by Xue, Ali, Zhang, et al. (2021) were collected concurrently, and are quite similar. However, the analyses we present to investigate the relationship between immersive content, attention, emotions, and movements in VR have not been proposed before.

CHAPTER 3

Deep variational learning for multiple trajectory prediction of 360° head movements

Prediction of head movements in immersive media is key to design efficient streaming systems able to focus the bandwidth budget on visible areas of the content.
Numerous proposals have therefore been made in the recent years to predict 360° images and videos. However, the performance of these models is limited by a main characteristic of the head motion data: its intrinsic uncertainty.
In this chapter, we present an approach to generate multiple plausible futures of head motion in 360° videos, given a common past trajectory. Our method provides likelihood estimates of every predicted trajectory, enabling direct integration in streaming optimization. To the best of our knowledge, this is the first work that considers the problem of multiple head motion prediction for 360° video streaming.
We first quantify this uncertainty from the data. We then introduce our discrete variational multiple sequence (DVMS) learning framework, which builds on deep latent variable models. We design a training procedure to obtain a flexible and lightweight stochastic prediction model compatible with sequence-to-sequence recurrent neural architectures. Experimental results on four different datasets show that our method DVMS outperforms competitors adapted from the self-driving domain by up to 41% on prediction horizons up to 5 sec., at lower computational and memory costs.
To understand how the learned features account for the motion uncertainty, we analyze the structure of the learned latent space and connect it with the physical properties of the trajectories. From this analysis, we design a method to estimate the respective likelihoods of the multiple predicted trajectories, by exploiting the stationarity of the distribution of the prediction error over the latent space.

Experimental results on three datasets show the quality of these estimates, and how they depend on the video category.
Additionally, we present a brief exploration of other methods we considered to consider the uncertainty of head motion data. We show promising results, highlighting the need to pursue further research in this direction.

Contents

3.1 Introduction

As explained in Sec. 2.1.4, predicting head movements in VR is crucial for viewport-adaptive streaming, significantly improving the quality of experience in constrained network conditions. Numerous works have therefore looked into the problem of head motion prediction in 360° images and videos in the last couple of years (J. Chen et al., 2021; Romero Rondón et al., 2021; R. Zhang et al., 2021; Chao et al., 2021). However, the performance of existing prediction models is limited by a main characteristic of the head motion data: its intrinsic uncertainty. Very few models have considered this characteristic so far (H. Hu, Xu, Zhang, & Guo, 2019; X. Fan et al., 2021; L. Yang et al., 2022), but only heuristically for 360° videos. We illustrate this uncertainty in Fig. 3.1-left, showing that close past trajectories often lead to diverse/distant future trajectories. This is exemplified for two different users in Fig. 3.10-left. This has long been identified in other application domains such as autonomous driving (Marchetti et al., 2020a; Berlincioni, Becattini, Seidenari, & Del Bimbo, 2021) or human pose estimation (Rupprecht et al., 2017). Such an ambiguity in the data (a same input may be mapped to several outputs) leads to degraded performance and over-fitting. Considering uncertainty in optimization of resource allocation is therefore key to improve systems' performance, as shown in robotic planning (Ha & Schmidhuber, 2018) and regular video streaming considering bandwidth uncertainty (Yan et al., 2020; Kan et al., 2021).

In this chapter, we present an approach to generate multiple plausible futures of head motion in 360° videos, given a common past trajectory. Our method provides likelihood estimates of every predicted trajectory, enabling direct integration in streaming optimization. To the best of our knowledge, this is the first work that considers the problem of multiple head motion prediction in 360° videos. Our contributions are:

- We first analyze head motion data and show the substantial diversity of futures corresponding to close past trajectories, and the shortcomings of a recent predictor in such cases.
- We introduce our discrete variational multiple sequence (DVMS) learning framework, which builds on deep latent variable models. The latent variable is designed to modulate the function connecting the past to the future. Each sample of the latent variable leads to a different plausible future. We design a training procedure to obtain a flexible and lightweight stochastic prediction model compatible with sequence-to-sequence recurrent neural architectures. Experimental results on four different datasets show that our method DVMS outperforms competitors adapted from the self-driving domain by up to 41% on prediction horizons up to 5 sec., at lower computational and memory costs.
- We provide a detailed analysis of both the learned latent space where the encoding of the past trajectories lies, and of the impact of z on the connection between past and predicted trajectories. For both analyses, we connect latent space locations and values of z with physical properties.

- We design a method to estimate the respective likelihoods of the multiple predicted trajectories, by showing that the distribution of the prediction error over the latent space has some stationarity, which we exploit. Experimental results on three datasets show the quality of these estimates, and how they depend on the video category.

Part of the work and ideas presented in this chapter are the outcome of a 1-month research stay at the Media Integration and Communication Center (MICC, part of the University of Florence), carried out in collaboration with Dott. Mag. Francesco Marchetti, Dr. Federico Becattini, Prof. Lorenzo Seidenari, and Prof. Alberto Del Bimbo. Part of the work presented in this chapter was the object of a conference paper presented at the 13th ACM Multimedia Systems Conference (MMSys '22) (Guimard, Sassatelli, et al., 2022). The in-depth analyses of the latent space of our model were submitted as part of a journal extension to this article and accepted with minor revisions in the ACM Transactions on Multimedia Computing, Communications, and Applications journal (TOMM) (Guimard et al., 2024).

In Sec. 3.2, we present recent work on point-wise and uncertainty-aware prediction of head motion, as well as relevant work on trajectory prediction from the domains of robotics and autonomous driving. In Sec. 3.3, we formulate the prediction problem we tackle, position formally the contribution in the framework of 360° streaming optimization, and motivate the approach with analysis of head motion data. In Sec. 3.4, we (i) provide necessary background on deep generative models, (ii) present our DVMS stochastic prediction model, emphasizing its generality and exemplifying it with a simple recurrent architecture, and (iii) present experimental results of DVMS on four datasets. In Sec. 3.5, we give an analysis of the DVMS latent space and shows how it can be exploited to estimate trajectory likelihoods, with an experimental assessment. In Sec. 3.6, we discuss the limitations of this work and the perspectives it opens for streaming optimization In Sec. 3.7 we briefly explore other approaches to consider uncertainty, including uncertainty quantification and the formalization of variational models for sequences. We finally conclude the chapter in Sec. 3.8.

We provide an analysis of the system gains of DVMS in a 360° system in chapter 4. The code associated with this chapter is publicly available at https://gitlab.com/DVMS_/DVMS.

3.2 Related work

We first review head motion prediction in 360° videos, with methods producing point-wise trajectory estimates, and methods considering motion uncertainty. We then discuss recent relevant work on multiple trajectory prediction in the domain of robotics with human pose estimation and autonomous driving systems.

3.2.1 Head motion prediction in 360° videos

Point-wise prediction:

Several approaches have relied on simple regressors or hand-crafted features to produce single trajectory prediction. For example, J. Chen et al. (2021) observed an equatorial posture attraction, and that video genre affects user behavior similarity. They then proposed a FoV prediction algorithm that explicitly balance between the current user's history and the history of other traces, for horizon of up to 4 seconds. Their method requires to have traces of previous users available for every video. Y. Mao, Sun, Liu, and Wang (2020) presented a coding scheme for interactive applications based on 360° video content, such as VR gaming or conferencing. They also adopt a simple linear FoV prediction method for horizons of 100ms. Recently, Chopra et al. (2021) extracted trajectories of moving objects in the video and combine them with autoregressive-filtered past user trajectories to predict future trajectories.

Regression with deep neural networks (DNNs) have also been investigated in several works. S. Park, Bhattacharya, et al. (2021) designed a point-wise prediction method fed with the video content and the past trajectory of the current user. They then fed the predicted FoV coordinates to a model-predictive control (MPC)–based streaming logic. Hou et al. (2021) also considered streaming optimization with FoV prediction based on an LSTM architecture predicting the tiles in FoV over the next 2 seconds. Both approaches are similar to a baseline considered by (Romero Rondón et al., 2021). They first re-examined existing deep-learning approaches and showed that they achieve worse or similar performance as simple baselines (predicting the future position equal to the last past or predicting only from the past head coordinates and not considering the video content). Then they proposed a new deep architecture establishing state-of-the-art performance for head prediction on horizons of up to 5 seconds. Their method enables prediction for new videos where no previous user trace is available. Feng, Li, and Wei (2021) also considered FoV prediction for live content. They underlined that the challenge is to find features of the video content and user behavior that have high correlation with the user's future FoV. They designed a FoV prediction method by collecting the user's real-time trajectory and the semantic description of the attended video regions. FoV is then predicted by finding the tiles with semantic description similar enough with the user's past trajectory encoded as a phrase. R. Zhang et al. (2021) designed a federated learning approach to predict the viewing probability of every tile, considering the video catalog known per user and focus on personalized model training from other users traces. Yu and Liu (2019) proposed LSTM-based architectures with attention to predict future FoV up to 3 sec. ahead, given past FoV coordinates. Chao et al. (2021) proposed a similar approach but with a transformer-based architecture, and showed that it can outperform previous approaches. Finally, several approaches are based on a deep reinforcement learning (DRL) framework where FoV prediction, bandwidth prediction and tile quality decisions are not achieved separately but jointly. C. Wu et al. (2021) proposed one of the most recent such DRL-based approaches for end-to-end control of 360° video streaming. The download horizon is up to 5 seconds.

Considering prediction uncertainty:

How users explore in VR and what commonalities do their viewing patterns exhibit have fostered a lot of interest in the last few years (Sitzmann et al., 2018; David et al., 2018; Almquist et al., 2018). Almquist et al. (2018) showed that the viewing congruence heavily depends on the type of scene, while other works (Sitzmann et al., 2018; David et al., 2018) have shown that, upon entering a new scene, the user first goes through an exploration phase where movements are not strongly correlated with the visual content.

To study and cope with user movement uncertainty, several approaches have relied on hand-crafted adaptations. X. Fan et al. (2021) studied spurious head movements that are not related to the scene content. They ran user experiments and attempted to automatically classify such movements. H. Hu et al. (2019) dealt with the uncertainty of FoV prediction by designing a FoV prediction method with a probabilistic model to prefetch video segments into the playback buffer, and enabled chunk replacements to maximize quality in the FoV. Prediction is achieved with a linear regressor trained on data from which are also extracted the parameters of the Gaussian distribution of prediction errors. Feng et al. (2019) developed a FoV prediction scheme for live 360° videos that consider various levels of synchronization of the user with the moving objects in the scene. From object detection and optical flow calculation, they linearly predicted future FoV, and dynamically adapted the size of the predicted region to cope with arbitrary moves. X. Zhang et al. (2020) considered FoV prediction over 50-300ms. They proposed a Markov model that learns stationary and transition distributions between discrete angle positions from past users' traces on this video, from the saliency map, and considering human head physical constraints. In contrast to these works relying on single trajectory prediction trying to consider the error distribution around a single mode, our method provides diverse trajectories by design, additionally to their estimated likelihood.

Recent works in 2D adaptive streaming have presented deep learning approaches to consider prediction uncertainty (Yan et al., 2020; Kan et al., 2021). In contrast with the vast majority of approaches considering point-wise estimates of future bandwidth for adaptive streaming, both consider the uncertainty of bandwidth prediction in the decision problem of what encoding rate to choose for the next video chunks to send. Both derive probability distribution of the future throughputs, that they feed into an MPC algorithm. Yan et al. (2020) designed a neural network to output a discretized probability distribution of predicted transmission times. Kan et al. (2021) considered Bayesian neural networks (BNNs) to output the probability distribution of future throughput, given the network's historical throughput.

In another recent work, L. Yang et al. (2022) considered predicting multiple head trajectories but only for 360° images, not videos as we do. They consider head trajectory as a succession of fixations and saccades, and intend to learn to capture the uncertainty of head trajectories across different subjects. They resort to a Bayesian neural networks (BNNs) approach, to predict, given an input 360° image, multiple head trajectories by sampling the weights of the neural network predictor, the inter-subject variance being modeled with a latent variable conditioning the weight distribution. This approach is the closest to our work, but it differs from ours in several aspects. It considers 360° images, not videos as

we do. It generates trajectories for the entire viewing duration, and is meant to model the intrinsic variability between the users, generating the trajectory uncertainty. In our work, we generate future trajectories online over a prediction horizon of 5 seconds and considering past motion of the current user only. We therefore cope not only with inter-user variability, but also with intrinsic uncertainty of the data in how past is correlated with future motion, data uncertainty often referred to as aleatoric uncertainty. Also, BNNs are computationally-heavy (the approach from L. Yang et al. (2022) is not real-time) and fit accurately but to just one mode in the data (Fort, Hu, & Lakshminarayanan, 2020). In this chapter, we consider a lightweight approach to multiple trajectory prediction, able to predict multiple modes for the future trajectory.

3.2.2 Multiple trajectory prediction in robotics

Prediction of 360° head motion is closely related to human pose prediction, and more generally to human motion prediction. In the field of robotics, a major challenge is the study of human pose motion, with the aim of generating the future movement of a collection of joints that represent human body. Fragkiadaki et al. (2015) developed an encoder-decoder model based on a recurrent neural network (RNN) to process the temporal dynamics of human pose. Afterwards, Jain et al. (2016) combined the ability of temporal modeling of RNNs with a spatio-temporal graph to model the interactions between humans and the environment. Many of the state-of-the-art models are based on graph neural networks (GNN) and its evolutions such as the graph convolutional network (GCN) and graph attentional network (GAT) (Adeli et al., 2021; Sofianos et al., 2021; Z. Liu et al., 2021). In these models, each joint is represented as a node and each relation between joints as an edge. In the literature, this problem is still handled in single-modal setting, despite a recent attempt to better consider randomness (Parsaeifard et al., 2021).

The problem of predicting a set of multiple and diverse trajectories has been extensively studied in the field of autonomous driving. There the task is to forecast future positions of moving agents such as cars and pedestrians. Compared to head motion prediction, where predictions are guided by content and user attitude, trajectory forecasting is a more constrained task due to social behavioral rules (Alahi et al., 2016; Lee et al., 2017; Gupta et al., 2018; Sadeghian et al., 2019; Ivanovic & Pavone, 2019; Yuan et al., 2021), inertia of moving agents and environmental constraints (Lee et al., 2017; Srikanth et al., 2019; Berlincioni et al., 2021; Chang et al., 2019; Marchetti, Becattini, Seidenari, & Del Bimbo, 2020b). Nonetheless, the ability to forecast a multimodal prediction is of fundamental importance for planning secure trajectories for autonomous vehicles.

The first method to generate multiple predictions has been DESIRE (Lee et al., 2017), which samples plausible trajectories from the latent space of a conditional variational auto-encoder (CVAE), in combination with RNN encoders and decoders, as well as a combination of reconstruction and Kullback–Leibler divergence (KLD) losses, which do not explicitly enforce diversity. Gupta et al. (2018) then proposed Social-GAN, which uses a generative adversarial model to sample multiple outcomes by injecting random noise in an encoder-decoder architecture. Diversity is enforced with the introduction of

a variety loss, which optimizes only the best prediction thus leaving the model free to explore the output space with multiple outcomes. The usage of a variety loss is now a common approach for generating multimodal predictions, not only for trajectory forecasting (Marchetti et al., 2020b; Y. Huang et al., 2019; Kosaraju et al., 2019; Amirian et al., 2019; Guan, Yuan, Kitani, & Rhinehart, 2020; De Divitiis, Becattini, Baecchi, & Del Bimbo, 2021; Walker, Doersch, Gupta, & Hebert, 2016). In the present chapter, we leverage this domain knowledge by considering the variety loss to enable the training of our DVMS model aiming to produce diverse plausible trajectories.

An extension of such loss, the multimodality Loss, has been introduced by Berlincioni et al. (2021), where the authors rely on synthetic data to generate multiple ground truth futures and directly optimize the model to output multiple adequate predictions. This approach requires the ability to generate synthetic samples but replaces the exploration step with an explicit supervision signal.

A recent trend in multimodal trajectory forecasting for autonomous driving is to divide the problem into two steps: first, possible goals or endpoints are estimated and then actual trajectories are regressed to reach such intents (H. Zhao et al., 2021; Dendorfer et al., 2020; Mangalam et al., 2020). Similarly, other approaches use a set of anchors to guide motion prediction following some previously observed samples (H. Zhao & Wildes, 2021; Marchetti et al., 2020b). We believe that such approaches are less suited for 360° head motion prediction, since motion is mostly guided by content and user attitude rather than constrained maneuvers.

3.3 Motivation behind multiple prediction of head trajectories

We first outline the 360° adaptive streaming problem we aim to solve in Sec. 3.3.1, then formally define the head motion prediction problem in Sec. 3.3.2. In Sec. 3.3.3, we analyze head motion data to quantify the diversity of futures corresponding to similar past trajectories, and show the need for multiple future predictions.

3.3.1 360° adaptive streaming problem

The core motivation for our contribution is to improve adaptive streaming for 360° videos by taking into account randomness of the environment in the optimization of resource allocation. Specifically, considering the optimization of spatial heterogeneous quality in streaming 360° videos, one has to consider the variations of network bandwidth and human head position, which both cannot be predicted perfectly. Such stochastic optimization can generally be approached in two ways. First, RL-based approaches (H. Mao et al., 2017; C. Wu et al., 2021) do not split the problem into environment prediction and resource allocation, but rather tackle it end-to-end. Other recent works show the benefit, for regular video streaming (Yan et al., 2020; Kan et al., 2021), of splitting the problem

and designing a DNN to produce stochastic predictions of bandwidth, which are then considered as parameters in model predictive control (MPC). For example, Yan et al. (2020) used dynamic programming to maximize the expected cumulative quality of experience (QoE) as shown in Eq. 3.1, where H is the look-ahead horizon for download, B_j is the playback buffer's level at chunk j, $QoE(\cdot)$ is the QoE function, K_i^s is chunk i in quality s, and $T(K_j^s)$ is the stochastic download time of this chunk.

$$\max_{K_i^s,\dots,K_{i+H-1}^s} \sum_{j=i}^{i+H-1} \sum_{t_j} Pr[T(K_j^s) = t_j] QoE\left(K_j^s, K_{j-1}^s, B_j, t_j\right) \tag{3.1}$$

$$\max_{\{K_{i,l}^s\}_l,\dots,\{K_{i+H-1,l}^s\}_l} \sum_{j=i}^{i+H-1} \sum_{l=1}^{L} \sum_{t_j} Pr[l \in FoV(j)] \\ Pr[T(K_j^s) = t_j] QoE\left(\{K_{j,l}^s\}_l, \{K_{j-1,l}^s\}_l, B_j, t_j\right) \tag{3.2}$$

Such formulation enables buffering of H chunks to absorb bandwidth variations. Kan et al. (2021) formulated this optimization by projecting the estimated bandwidth distribution onto a confidence interval. In the case of 360° streaming, the equivalent problem can be formulated, incorporating the distribution of the FoV position over the look-head horizon (Sassatelli, Winckler, Fisichella, Aparicio, & Pinna-Déry, 2019) as shown in Eq. 3.2, with $l \in \{1, L\}$ denoting the tile index, if we consider a tile-based formulation.

In this chapter, we provide a stochastic tool, the DVMS learning framework presented in Sec. 3.4.2, to estimate the distribution $Pr\left[l \in FoV(j)\right]$. To do so, we make a proposal to predict several K trajectories (series of centers of FoV) $\mathbf{y}_{t:t+H}^k$, for $k \in \{1, K\}$, with their estimated likelihood $Pr[\mathbf{y}_{t:t+H}^k | \mathbf{x}_{0:t}]$. If the problem is tile-based as above, then we can obtain $Pr\left[l \in FoV(j)\right]$ in Eq. 3.3.

$$\begin{aligned} Pr\left[l \in FoV(j)\right] &= \sum_{k=1}^{K} Pr[l \in FoV(j) | \mathbf{y}_{i:i+H-1}^k] Pr[\mathbf{y}_{i:i+H-1}^k | \mathbf{x}_{0:i}] \\ &= \sum_{k:l \in \text{ FoV of center } y_j^k} Pr[\mathbf{y}_{i:i+H-1}^k | \mathbf{x}_{0:i}] \end{aligned} \tag{3.3}$$

Once we have at our disposal multiple trajectory estimates and their respective likelihoods, we can express the distribution of the FoV position as a function of these estimates as seen in Eq. 3.3. This distribution can in turn be used in conjunction with the appropriate QoE function by an adaptive bitrate (ABR) algorithm to solve the optimization problem as formulated in Eq. 3.2. In this chapter, we focus on the design and evaluation of the prediction methods. An evaluation of the system gains is provided in chapter 4.

3.3.2 Head motion prediction problem

The problem we consider is formally described as follows. We consider that a given 360° video v of duration T seconds is being watched by a user u. The head trajectory of the

user is denoted $\mathbf{x}_{0:T}^{u,v}$, with $\mathbf{x}$ storing the head coordinates on the unit sphere (as, e.g., Euler angles, Euclidean coordinates or quaternions).

Online single prediction problem: At any time t in $[0, T]$, predict $\mathbf{x}_{t:t+H}^{u,v}$ with an estimate $\mathbf{y}_{t:t+H}^{u,v}$, that is predict the future trajectory over a prediction horizon H, assuming only $\mathbf{x}_{0:t}^{u,v}$ is known.

That is, we do not assume any knowledge of traces other than u on this video v. Hence, for lighter notations, we drop indices u and v from $\mathbf{x}_{0:t}^{u,v}$ and only write $\mathbf{x}_{0:t}$ and $\mathbf{y}_{0:t}$.

Online multiple future prediction problem: At any time t in $[0, T]$, predict K possible future trajectories $\mathbf{y}_{t:t+H}^{k}$, for $k = 1, \ldots, K$, to estimate $\mathbf{x}_{t:t+H}$.

This is the general problem definition considered in related work (Babaeizadeh, Finn, Erhan, Campbell, & Levine, 2018; Bhattacharyya, Schiele, & Fritz, 2018). However, for optimization of heterogeneous quality decisions in a video streaming system, it is also important to estimate the likelihood of every such possible future trajectory. We therefore augment the multiple future prediction problem with estimation of likelihood $Pr[\mathbf{y}_{t:t+H}^{k}|\mathbf{x}_{0:t}]$. This is addressed in Sec. 3.5.2 thanks to our variational model proposed in Sec. 3.4.2.

3.3.3 Analysis of the need for multiple prediction in head motion data

We now analyze the need for multiple prediction from two perspectives: from the data only, and from the performance of a given predictor on this data. We consider data from the test set of the MMSys18 dataset, described in Sec. 3.4.4.3. In what follows, past (resp. future) trajectories are considered over a horizon of 1 sec. (resp. 5 sec.) as done in recent work (Chao et al., 2021; Romero Rondón et al., 2021). We exclude the 5% shortest past trajectories and the 5% shortest future trajectories from this analysis, as they may skew the distance calculations between pairs of trajectories.

Distance metric: We compare two trajectories P_1 and P_2 of equal length L using the average point-wise great-circle distance, defined in Eq. 3.4. We consider pairs of trajectories with the lowest distance to be the closest to each other.

$$d(P_1, P_2) = \frac{2}{L} \cdot \sum_{p_1 \in P_1, p_2 \in P_2} \arcsin\left(\frac{\|p_1 - p_2\|_2}{2}\right) \tag{3.4}$$

First, we investigate how the distance between past trajectories relates to the distance between their corresponding true futures. To do so, for each timestamp of each video in the dataset, we consider all pairs of users, and select 200 pairs per video with the closest past trajectories. Every pair of users yields the distance between both past trajectories, and the distance between both respective true future trajectories. Fig. 3.1-left represents the scatter plot of both distances for every pair. We observe that, for 200 pairs of closest past trajectories per video, **90%** of the corresponding future pairs have a distance more than twice the distance between their past trajectories (above the $y = 2x$ line). Also, we observe that for close past elements, more distant futures are produced, on this dataset, for exploration-type videos *PortoRiverside* and *PlanEnergyBioLab*. Specifically, 81% of the points are above the $y = 4x$ line for *PortoRiverside* and 85% of the points are above

the $y = 4x$ line for *PlanEnergyBioLab*. Fig. 3.1-right represents the distance between past trajectories (continuous) and the distance between future trajectories (dashed), for every N-th pair of closest past trajectories for each video, with $N \leq 5000$ (distances are smoothed with a moving average). It confirms that the distance between future trajectories is generally higher that the distance between past trajectories, with a greater difference obtained for exploration videos.
Finding: This is an indication that relatively close past trajectories may lead to distinct/-farther apart future trajectories, which may create difficulties when attempting to train a prediction model on such data. Indeed, a (neural) regressor trained with the regular mean square error (MSE) cannot map similar inputs to different outputs.

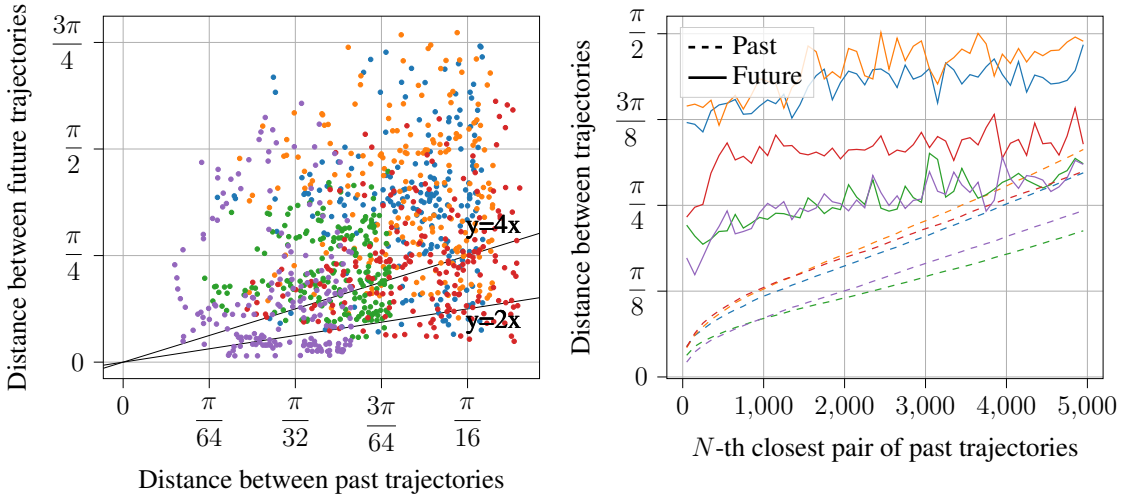


Figure *3.1: Distances between pairs of past and future trajectories for pairs of close past trajectories on the test videos of the MMSys18 dataset. The colors are associated with the video IDs and are the following: blue: PortoRiverside, orange: PlanEnergyBioLab, green: Waterpark, red: Warship, purple: Turtle.*

Second, we investigate predictions made by a recent deep predictor on this data. Thanks to the reproducible framework provided by Romero Rondón et al. (2021), we consider their prediction model named TRACK. For the same pairs of closest past trajectories as in Fig. 3.1, Fig. 3.2 represents how the distance between predicted future trajectories match the distance between their corresponding true futures. Fig. 3.2-left shows that, given close past trajectories, the predicted trajectories are much closer together than the true future trajectories. Specifically, for all of the videos, the proportion of points above $y = 2x$ ranges from **83%** (*Turtle*) to **99%** (*PlanEnergyBioLab*), and the proportion of points above $y = 4x$ ranges from 40% (*Turtle*) to 88% (*PortoRiverside* and *PlanEnergyBioLab*). This is confirmed in Fig. 3.2-right showing the difference between the average in-between true futures distances and in-between predicted futures distances.
Finding: This is an indication that predicted trajectories have less diversity than true trajectories.

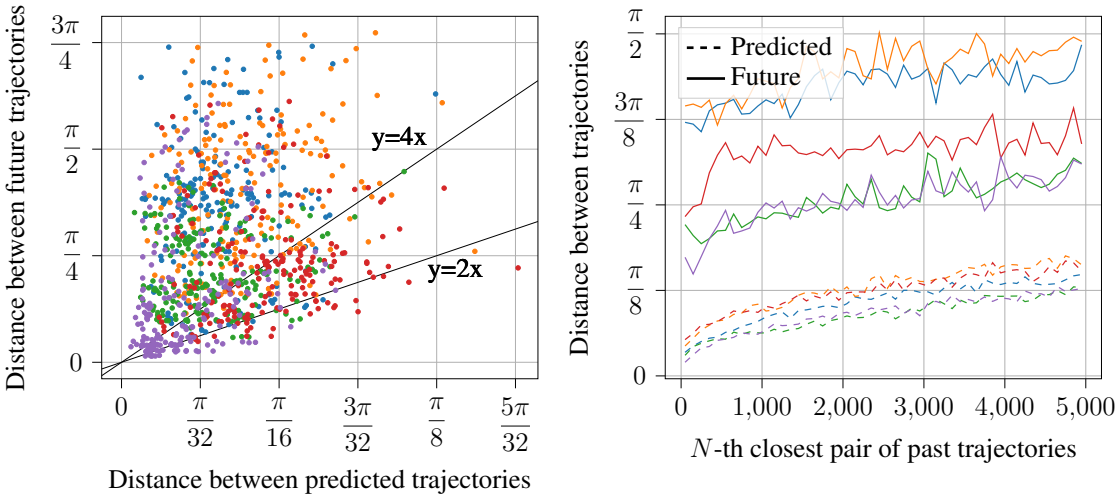


Figure *3.2: Distances between pairs of predicted and true future trajectories for pairs of close past trajectories on the test videos of the MMSys18 dataset. The colors are associated with the video IDs and are the following: blue: PortoRiverside, orange: PlanEnergyBioLab, green: Waterpark, red: Warship, purple: Turtle.*

Finally, we investigate the connection between multimodality (ratio of distance between the true futures over the distance between their pasts) and prediction error. Fig. 3.3 shows the evolution of the average prediction error for pairs of trajectories where $\frac{d_{future}}{d_{past}} \geq 0.8$ (i.e., ratio ≥ 0.8) against the ratio of distance between the true futures over the distance between their pasts. As the ratio increases, the prediction error also tends to increase for most videos.

Finding: This is an indication that the predictor is less accurate when there is more diversity in the true futures than in the past trajectories.

While this finding might have been expected, we considered important to experimentally verify that (i) there is diversity in head motion traces as seen in Fig. 3.1, (ii) the diversity of ground truth data is not properly reproduced by a refined recent predictor considering both past motion and visual content as shown in Fig. 3.2, and (iii) this diversity of futures indeed contributes to the error of single trajectory predictor as illustrated in Fig. 3.3.

In this section, we have seen that (i) we need to estimate the distribution of future viewports for streaming optimization, (ii) future trajectories exhibit a significant diversity relatively to their close respective past trajectories, and (iii) the prediction error increases in such cases. All these observations provide strong justification for the development of multiple trajectory prediction methods.

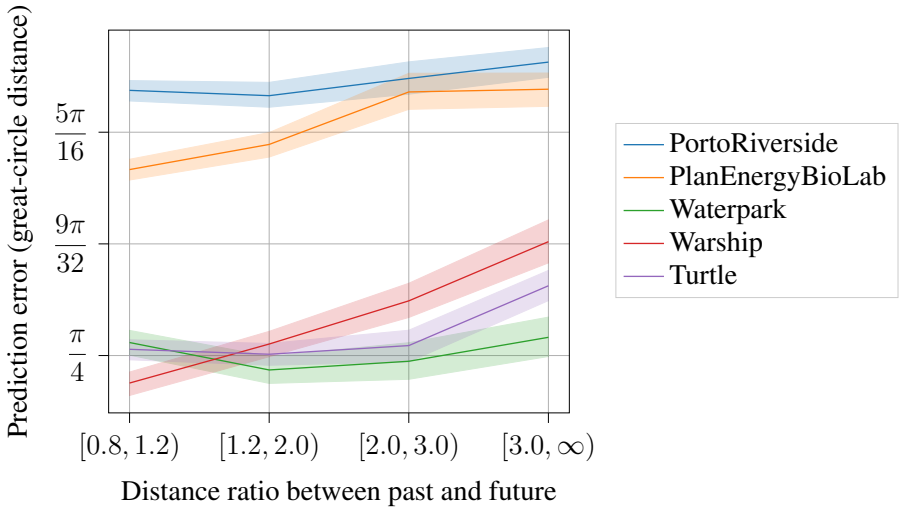


Figure *3.3: Prediction error of TRACK (Romero Rondón et al., 2021) against the ratio of distances between pairs of future trajectories over distances between pairs of their corresponding past trajectories.*

3.4 Deep stochastic prediction of multiple head trajectories

We first provide necessary background on deep generative models in Sec. 3.4.1. We present our proposal for a multiple prediction framework in Sec. 3.4.2, exemplified with an architecture in Sec. 3.4.3. Sec. 3.4.4 presents performance results.

3.4.1 Background on deep generative models for sequences

Trajectory prediction can be cast into conditional sequence generation, for which we provide some background next. Deep generative approaches are meant to generate data, modeling either explicitly or implicitly training data distributions. Variational auto-encoders (VAEs) (Kingma & Welling, 2014; Rezende, Mohamed, & Wierstra, 2014) and generative adversarial networks (GANs) (Goodfellow et al., 2020) are two prominent such families of approaches.

In this work, we focus on the VAE family owing to their capability not to narrowly focus on a few modes of the data distribution, hence being a better fit to the characteristics of head motion data as described in Sec. 3.3.3. VAE frameworks aim to enable the generation of high-dimensional data samples by sampling a normally distributed low dimensional latent variable $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. A sample x is then generated by passing z through a high-capacity model, particularly a deep neural network. The latent variable is meant to capture the minimum number of independent random dimensions from the data, while the decoding by a neural network of z into x is meant to capture the complex dependencies in a sample (Kingma & Welling, 2014). The typical representation of a VAE is illustrated in Fig. 3.4. Denoting the decoder's parameters with θ, the generative model is typically

defined with $p(z)$ and $p_\theta(x|z)$. For the decoder network to be trained, the posterior distribution $p(z|x)$ is required, but can only be approximated with the output distribution, named the approximate posterior $q_\phi(z|x)$, of another neural network of parameters ϕ and usually referred to as the *encoder* or *inference network*. VAEs can also be declined into conditional VAEs (CVAEs) when the goal is to generate output variables y from input variables x, drawing z from a prior distribution $p_\theta(z|x)$ to generate y from the decoder with $p_\theta(y|x,z)$ (Sohn, Lee, & Yan, 2015).

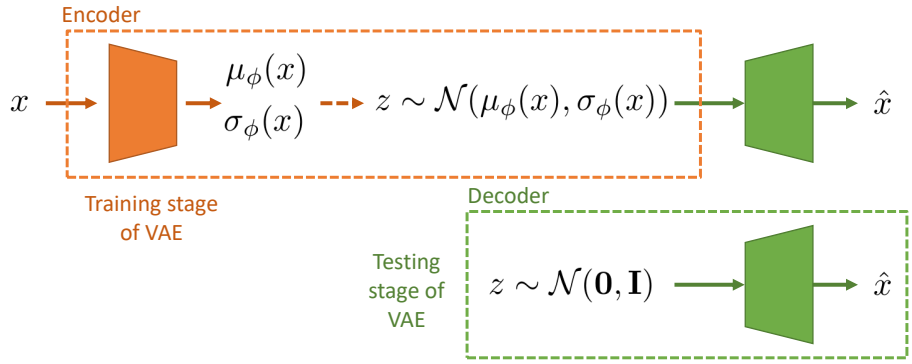


Figure *3.4: Schematic representation of a VAE.*

When considering sequence prediction formalized in Sec. 5.6.1.1, prediction of a time series over a certain horizon is often made conditionally to the past of the time series. This is often translated into sequence-to-sequence architectures, where a so-called *encoder* processes the past (even at test time, different to VAEs), produces an intermediate embedding, which is then decoded into a future trajectory (see Fig. 3.5). The architecture of the encoder and decoder networks are often based on recurrent neural networks (RNNs), such as LSTM or GRU (Sutskever et al., 2014; Romero Rondón et al., 2021). Note that the concept of encoder of past trajectory in a sequence-to-sequence architecture is different from the term *encoder* used in a variational context (aka *inference network*, as mentioned above).

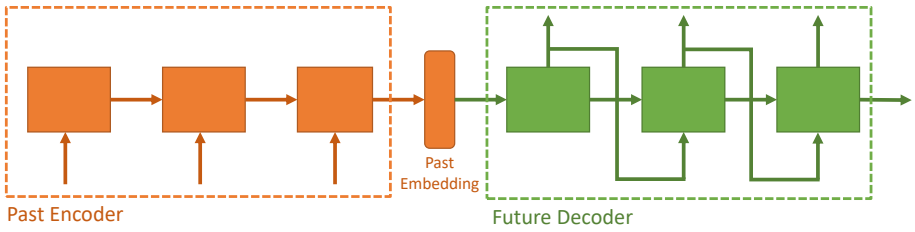


Figure *3.5: A sequence-to-sequence architecture.*

Deep variational learning has initially been designed for image data. More recently, variational approaches have been proposed for sequence data and so-called *structured output prediction*. These approaches are diverse depending on where the random latent variables are considered in the recurrent architectures (Chung et al., 2015; Babaeizadeh et al., 2018; Parsaeifard et al., 2021).

For example, Babaeizadeh et al. (2018) performed conditional video prediction to predict future frames until final video time T, conditioned on c initial frames, by sampling from $p(x_{c:T}|x_{0:c-1})$. A random latent vector z is picked at random from the prior distribution $p(z) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ at test time (while the training is made as usual with z sampled from the approximate posterior $q_\phi(z|x_{0:T})$). They show the performance in multiple future frame sequence prediction, specifically in PSNR and SSIM of the 10% best sequences obtained from 100 samples of z.

Parsaeifard et al. (2021) considered the randomness in human pose forecasting, which they decompose into trajectory forecasting and local pose forecasting. They advocate that the latter has higher randomness, which they tackle by considering an LSTM-based sequence-to-sequence architecture as done by Martinez et al. (2017), but they set the initial hidden state of the decoder to a latent vector z drawn from $\mathcal{N}(\mu(h_t), \sigma(h_t))$ where h_t is the latest hidden state of the encoder of the past coordinates, and $\mu(\cdot)$ and $\sigma(\cdot)$ are functions implemented with fully connected layers.

The training of such RNN-based VAEs can be difficult to converge and unstable (Babaeizadeh et al., 2018; Bhattacharyya et al., 2018). This is particularly due to the fact that during training, z is sampled from the approximate posterior $q_\phi(z|x_{0:T})$ while it can only be sampled from $p_\theta(z|x_{0:c-1})$ in test, and despite a KL divergence component in the training loss meant to nudge $q_\phi(z|x_{0:T})$ towards $p_\theta(z|x_{0:c-1})$.

With this background on variational approaches for sequence generation, we now present our learning framework for multiple prediction of head trajectories.

3.4.2 Discrete Variational Multiple Sequence (DVMS) prediction

We now present a new learning framework for multiple head motion trajectory prediction, named discrete variational multiple sequence (DVMS). It builds on deep latent variable models like VAEs. DVMS is designed to be compatible with any sequence-to-sequence architecture. The rationale for such design is as follows. Our goal is to design a framework for multiple prediction of head motion for deep architectures, which provides key properties:

(P1) sufficiently diverse predictions $\mathbf{y}^k_{t:t+H}$, for $k = 1, \ldots, K$,

(P2) state-of-the-art performance when $K = 1$,

(P3) estimates of likelihoods of the predicted trajectories,

(P4) flexibility and low computational cost.

Generative model: The probabilistic graphical model of DVMS is depicted in Fig. 3.6. For any encoder fed with past sequence $\mathbf{x}_{0:t}$, an embedding h_t is produced. This embedding is then concatenated with a unique latent variable z. The latent variable is key in our DVMS proposal. This latent variable is meant to capture the variations in the function relating the future sequence to the past sequence, hence acting as a parameter in the past-to-future mapping. The resulting concatenation produces the first hidden state g_t of the

decoder. Considering that the encoder is made of recurrent connections with hidden state h_t, the generative model writes as Eq. 3.5, where $\mathbb{U}_{\mathcal{Z}_K}$ denotes the uniform distribution over discrete set $\mathcal{Z}_K$, and MLP stands for multi-layer perceptron to denote one or several fully connected (FC) layers.

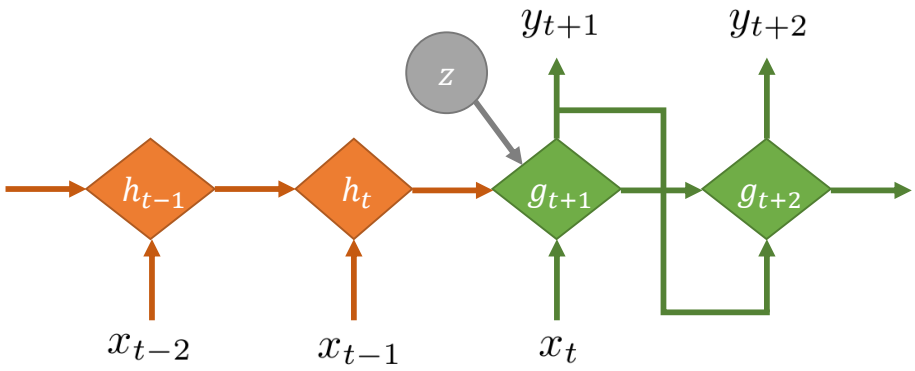


Figure *3.6: Probabilistic graphical model of the proposed stochastic discrete variational multiple sequence (DVMS) prediction framework. A random variable is represented with a circle, a deterministic state with a diamond.*

To generate multiple prediction, every $z_k \in \mathcal{Z}_K$ generates a future trajectory $\mathbf{y}^k_{t:t+H}$. To enable diverse predictions (P1), we do not constrain the distribution $p(z)$ we sample from to be conditioned on $x_{0:t}$ in test, in contrast to what was done by Parsaeifard et al. (2021), but instead draw z uniformly in $\mathcal{Z} \in [-1, 1]^d$ (where d is the dimension of vector z). To meet (P3), z is drawn from a discrete set $\mathcal{Z}$ with K elements. Indeed, z codes for latent features parameterizing the expression of the future trajectory from the past trajectory. In other words, different values of z allow for the representation of different *modes* of future trajectories, given the same past. If there is some stationarity in how likely is every trajectory produced from every z_k, then we can exploit this stationarity for likelihood estimation (P3). We therefore consider a discrete fixed set of possible z values to ease this exploitation, which we describe in Sec. 3.5.2.

$$\begin{aligned} h_t &= \text{RNN}_{enc}\left(h_{t-1}, \mathbf{x}_{t-1}\right), \quad h_0 = 0 \\ z &\sim \mathbb{U}_{\mathcal{Z}_K} \\ g_{t+1} &= \text{MLP}(h_t, z) \\ \mathbf{y}_t &= \mathbf{x}_t \\ \mathbf{y}_{t+s} &= \text{FC}(g_{t+s}) + \mathbf{y}_{t+s-1}, \quad \text{for } s \geq 1 \\ g_{t+s} &= \text{RNN}_{dec}\left(g_{t+s-1}, \mathbf{y}_{t+s-1}\right), \quad \text{for } s \geq 2 \end{aligned} \tag{3.5}$$

Training procedure: To ensure (P2), we enforce the prior distribution $p(z)$ z is sampled from at training time to be the same as in test (contrary to work from Babaeizadeh et al. (2018)), i.e., we do not consider an inference network. This allows to avoid the mismatch between $p(z)$ and $q(z|x_{0:T})$, which impedes the training convergence, as described in Sec. 3.4.1. However, doing so also adds noise to the sequence decoder which, if trained with

gradient descent performed over every sampled trajectory obtained from z_k, for all $k \in \{1, K\}$, learns to discard the z input and only produces a single trajectory corresponding to the baseline, as described by Babaeizadeh et al. (2018). To avoid this phenomenon, we instead train our architecture with the *best of many samples* (BMS) loss (Bhattacharyya et al., 2018), also named the *variety loss* (Gupta et al., 2018; Thiede & Brahma, 2019), defined in Eq. 3.6.

$$\mathcal{L}(\mathbf{x}_{0:t}, \theta) = \min_{k \in \{1,K\}} D\left(\mathbf{y}^k_{t:t+H}, \mathbf{x}_{t:t+H}\right) \tag{3.6}$$

where $D(\cdot)$ can be any distance between two trajectories on the sphere. This loss thus consists, for every past trajectory sample, in selecting sample z_{k^*} generating the best match to the single ground truth future. The gradient descent is hence performed only on a single k^* sample out of the trajectories generated by the model. This prevents the architecture from learning to discard z as being an uninformative input for prediction.

DVMS is flexible (P4) because it can be used with any sequence-to-sequence architecture, being it an architecture processing video content (Romero Rondón et al., 2021) in case of streaming of stored content, or an architecture processing only the past user's trajectory (Chao et al., 2021) in case of live streaming. Indeed, Bayesian methods like BNN (Neal, 2012) and Monte-Carlo dropout (Gal & Ghahramani, 2016) require to change how every network weight is considered in train (generating multiple weight samples). In contrast, DVMS only consists in adding a latent variable to modulate the initial state of the sequence-to-sequence decoder with a random component, independently of the actual structure of the sequence-to-sequence encoder and decoder.

DVMS is also lightweight (P4) because the additional training cost, w.r.t. the original sequence-to-sequence architecture, only comes from the latent variable z to be concatenated with the encoder's last hidden state (MLP to learn in Eq. 3.5). This additional cost is also limited because we do not learn an approximate posterior $q(z|\mathbf{x}_{0:t})$, that is an additional neural network (named *inference network* in Fig. 3.4 and used only in train), but rather directly sample z from $\mathcal{U}_{\mathcal{Z}_K}$ both in test and train.

All 4 properties (P1)-(P4) are experimentally evaluated in Sec. 3.4.4 and 3.5.3.

3.4.3 Proposal of a DVMS-based architecture

To demonstrate the interest of the proposed DVMS learning framework of multiple head trajectory prediction, we propose a simple architecture akin to those presented in (Parsaeifard et al., 2021, Fig. 2) or (Romero Rondón et al., 2021, Fig. 4) in this section. This architecture is of type sequence-to-sequence and is represented in Fig. 3.7. It is however simplified compared to the previous literature, as we consider double-stacked gated recurrent units (GRU) instead of single or double-stacked LSTM.

Here, we purposefully do not consider the visual content in order to simplify the presentation and analysis of our contribution, which is on the variational framework DVMS for multiple future prediction, and not on a specific neural architecture. This means that other architectures can be incorporated in our framework, such as based on more advanced recurrent techniques like transformers (Chao et al., 2021) or fusion of multimodal input

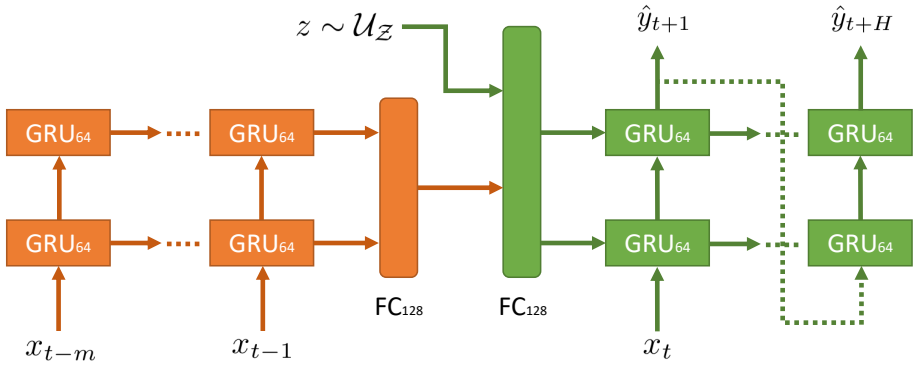


Figure *3.7: Proposed example of a DVMS-based architecture.*

considering the visual content (Romero Rondón et al., 2021). This compatibility is further discussed in Sec. 3.6.

Architecture: We set $d = 1$ as the dimension of z. The encoder is made of a doubly-stacked GRU with 64 neurons (and default GRU activations). The final GRU's hidden state is then fed to a 128-neuron fully connected layer. The output of this layer is concatenated with z and fed to another 128-neuron fully connected layer. The decoder is also a doubly-stacked GRU with same hyper-parameters as the encoder. Not shown on the diagram for simplicity, the output of the GRU decoder is fed to a fully connected layer that reduces the dimension to 3. This output is added to the last known position, computing the next position in a residual manner. Using residuals improves training stability and overall test performance. The past sequence is restricted to $\mathbf{x}_{t-m:t}$ with $m = 1$sec., matching recent work (Chao et al., 2021; Romero Rondón et al., 2021), and we set $H = 5$sec. as the prediction horizon. The sampling rate of the scanpaths is 5Hz, thus the past sequences are 5 sample-long, and the future sequences are 25 sample-long.

Training procedure: The model is trained using the loss described in Eq. 3.6. Distance $D(\cdot)$ is taken as the cumulated Euclidean distance, that is $D\left(\mathbf{y}^k_{t:t+H}, \mathbf{x}_{t:t+H}\right) = \sum_{s=0.2}^{H} \left\|\mathbf{y}^k_{t+s} - \mathbf{x}^k_{t+s}\right\|^2$. The optimizer is Adam with weight decay (AdamW), with a learning rate of 5×10^{-4} and a batch size of 64.

3.4.4 Results on multiple trajectory prediction

In this section we assess (P1) the diversity of predictions, (P2) the performance for $K = 1$, and (P4) the computational cost. Likelihood estimation (P3) is addressed in Sec. 3.5.2.

3.4.4.1 Notation

For simplicity, we introduce a new notation to indicate the number of trajectories the model is generating. As our model can be trained to generate any number of K trajectories, we note DVMS-**K** the version of the model that predicts K future head trajectories. For example, DVMS-**1** generates one trajectories and DVMS-**5** generates five trajectories.

3.4.4.2 Experimental settings

Datasets: We consider four datasets of 360° videos with head motion traces:

- MMSys18 (David et al., 2018): head motion traces of 57 subjects watching 19 360° videos, all lasting 20 seconds.
- CVPR18 (Y. Xu et al., 2018): head motion traces of 45 subjects watching 208 360° videos lasting from 15 to more than 80 seconds (36 seconds on average).
- PAMI18 (M. Xu, Song, et al., 2019): head motion traces of 58 subjects watching 76 360° videos, lasting from 10 to 80 seconds (25 seconds on average).
- MM18 (Nguyen et al., 2018): head motion traces of 48 subjects watching 9 360° videos, lasting from 19 to 49 seconds (30 seconds on average).

For all of these datasets, we use the same split as described in the supplemental material from Romero Rondón et al. (2021), such that there is no overlap between the videos in the train and test sets of CVPR18, PAMI18, and MM18, as well as no overlap between the users of MMSys18. Additionally, we do not make predictions for the first 6 seconds of the video with any of the considered competitors, as done by Romero Rondón et al. (2021) to skip the user's initial exploration phase.

Metrics: When it comes to evaluating the quality of the multiple predictions, the major challenge is that several plausible futures may correspond to a single input, but the datasets provide only a single ground-truth future. The best way to assess (P1) is therefore to check if the known ground truth is covered by one of the few predictions, while the others can efficiently explore the search space to cover the futures of close inputs. This can be done by using the *winner-take-all* or *best of many samples* (BMS) metric (Bhattacharyya et al., 2018). Therefore, as is usually done as standard practice in multiple sequence prediction (Babaeizadeh et al., 2018; Marchetti et al., 2020a; Srikanth et al., 2019), we report the BMS metric. Specifically, BMS at prediction step s is defined in Eq. 3.7, where the great-circle distance between points P_1 and P_2 on the unit sphere is $\text{gcd}(P1, P2) = \arccos(\sin\phi_1 \sin\phi_2 + \cos\phi_1 \cos\phi_2 \cos\lambda)$ with ϕ the latitude and λ the absolute difference in longitude, and k^* is defined in Eq. 3.8.

$$\frac{1}{U}\frac{1}{V}\frac{1}{T}\sum_{u}\sum_{v}\sum_{t}\text{gcd}(y^{k^*}_{u,v,t+s}, x_{u,v,t+s}) \tag{3.7}$$

$$k^*(u,v,t) = \arg\min_{k}\sum_{s=0.2}^{H}\text{gcd}(y^{k}_{u,v,t+s}, x_{u,v,t+s}) \tag{3.8}$$

We report the BMS metric in figures, and we report in a more compact form in tables the average prediction error, which is the average over $s \leq H$ of the BMS metric, similarly to what Marchetti et al. (2020a) reported. For $K = 1$, the BMS metric is equal to the great-circle distance, hence enabling the assessment of (P2) with the same metric as used for single sequence prediction (Chao et al., 2021; Romero Rondón et al., 2021).

Competitors: We compare our models with four competitors. As no competitor exists so far for multiple prediction of head motion, we adapt a recent method from the autonomous driving domain.

- *Trivial-static*: *Trivial-static* is a trivial baseline already shown to outperform previous viewport prediction architectures by Romero Rondón et al. (2021). The predicted head positions are equal to the last known head position.

- *Deep-position-only*: *Deep-position-only* is a deep learning baseline introduced by Romero Rondón et al. (2021). It is a simple sequence-to-sequence LSTM taking past head positions as input. Additional details can be found in section 3.2 of (Romero Rondón et al., 2021). Thanks to the reproducible framework they published (Romero Rondón et al., 2020), we were able to directly evaluate *Deep-position-only* with the provided code and model weights and achieve the same performance as reported.

- MANTRA-*adapted*: MANTRA is an approach described by Marchetti et al. (2020a) to predict the trajectory of moving agents in a self-driving context. It uses an auto-encoder in conjunction with a memory network. The auto-encoder is first trained to reconstruct future trajectories from past and future trajectories. A memory-writing controller is then trained to fill the memory with embeddings from the encoder. The memory takes the form of a (key, value) dictionary, where the embeddings of past trajectories are the keys that are used to retrieve the values, embeddings of future trajectories.

  At prediction time, embeddings of yet unseen past trajectories are computed and matched with keys from the memory. The K most similar keys are used to retrieve the K corresponding values, which are then fused with the embedding of the actual past and decoded into K predicted future trajectories.

  Memory is built at training time with the following procedure. During training, if none of the predicted trajectories is close enough (defined by a manual threshold) to the ground truth future trajectory, the embeddings (past and future) of this trajectory are added as new key and value to the memory. The loss for the writing controller is designed so that it only writes relevant trajectories into the memory. Embeddings that are too similar and do not help to decrease the prediction error are not added to the memory. At test time, the memory is read-only and filled with embeddings from the training set. For this model to work properly, the trajectories have to be normalized so that they are translation and rotation-invariant.

  Building from this approach, we build a MANTRA-*adapted* model as a multiple trajectory prediction baseline to be compared to our proposed model. The changes from the original MANTRA model are described as follows. The trajectories are 3-dimensional instead of 2-dimensional. We adapt the manual distance thresholds used for the writing controller with values that fit our data and give an acceptable memory size. We do not normalize the trajectories in the same way. As there is

no rotation invariance in head motion, we carried out several tests with translation invariance (separating yaw and pitch). The results were best when only re-centering on yaw (longitude). The results were worse with re-centering both axes, only pitch or with no re-centering. Since video cue is not considered in DVMS, thus not providing any contextual information or map, MANTRA-*adapted* does not employ any contextual cue either, such as the "Iterative Refinement Module" (Marchetti et al., 2020a), which normally integrates information from the map.

- VPT360: VPT360 is the recurrent transformer-based viewport prediction architecture presented by Chao et al. (2021). We do not reproduce their results because the code is not available at the time of writing, but we report the values presented in their work (Chao et al., 2021) on the MMSys18 dataset and compare DVMS with VPT360 on the exact same settings.

3.4.4.3 Experimental results

Prediction error: Fig. 3.8 shows the prediction error (great-circle distance, BMS metric of DVMS for $K > 1$) of DVMS against against state-of-the-art single trajectory predictors on all four datasets. The shaded area represents the 95% confidence interval. Detailed prediction results showing the average displacement error (ADE) on the same datasets are also available in Tables 3.1, 3.2, 3.3 and 3.4.

*Table 3.1: Prediction error over all $s \leq H$ on the MMSys18 dataset. Lowest prediction error for a given K is underlined, lowest prediction error for all K is highlighted in **bold**.*

Method		Average prediction error				
		$s \leq 1s$	$s \leq 2s$	$s \leq 3s$	$s \leq 4s$	$s \leq 5s$
Trivial-static ($K = 1$)		0.322	0.522	0.674	0.792	0.883
Deep-position-only ($K = 1$)		0.261	0.450	0.598	0.721	0.818
VPT360 (reported) ($K = 1$)		0.239	0.438	0.603	0.726	0.809
MANTRA-adapted	$K = 1$	0.333	0.621	0.828	0.967	1.066
	$K = 2$	0.296	0.515	0.651	0.743	0.824
	$K = 3$	0.290	0.472	0.575	0.645	0.717
	$K = 4$	0.287	0.453	0.539	0.592	0.659
	$K = 5$	0.274	0.433	0.515	0.566	0.625
DVMS (ours)	$K = 1$	0.245	0.432	0.581	0.700	0.790
	$K = 2$	0.262	0.424	0.516	0.566	0.613
	$K = 3$	0.228	0.372	0.439	0.465	0.501
	$K = 4$	0.218	0.352	0.402	0.418	0.452
	$K = 5$	**0.216**	**0.343**	**0.386**	**0.397**	**0.432**

We observe that DVMS-**1** slightly outperforms both *Deep-position-only* (by **3.4%**) and VPT360 (by **2.3%**) on the MMSys18 dataset for when looking at the average prediction error over 5 seconds (prediction step $s \leq 5$sec.). DVMS-**1** also slightly outperforms

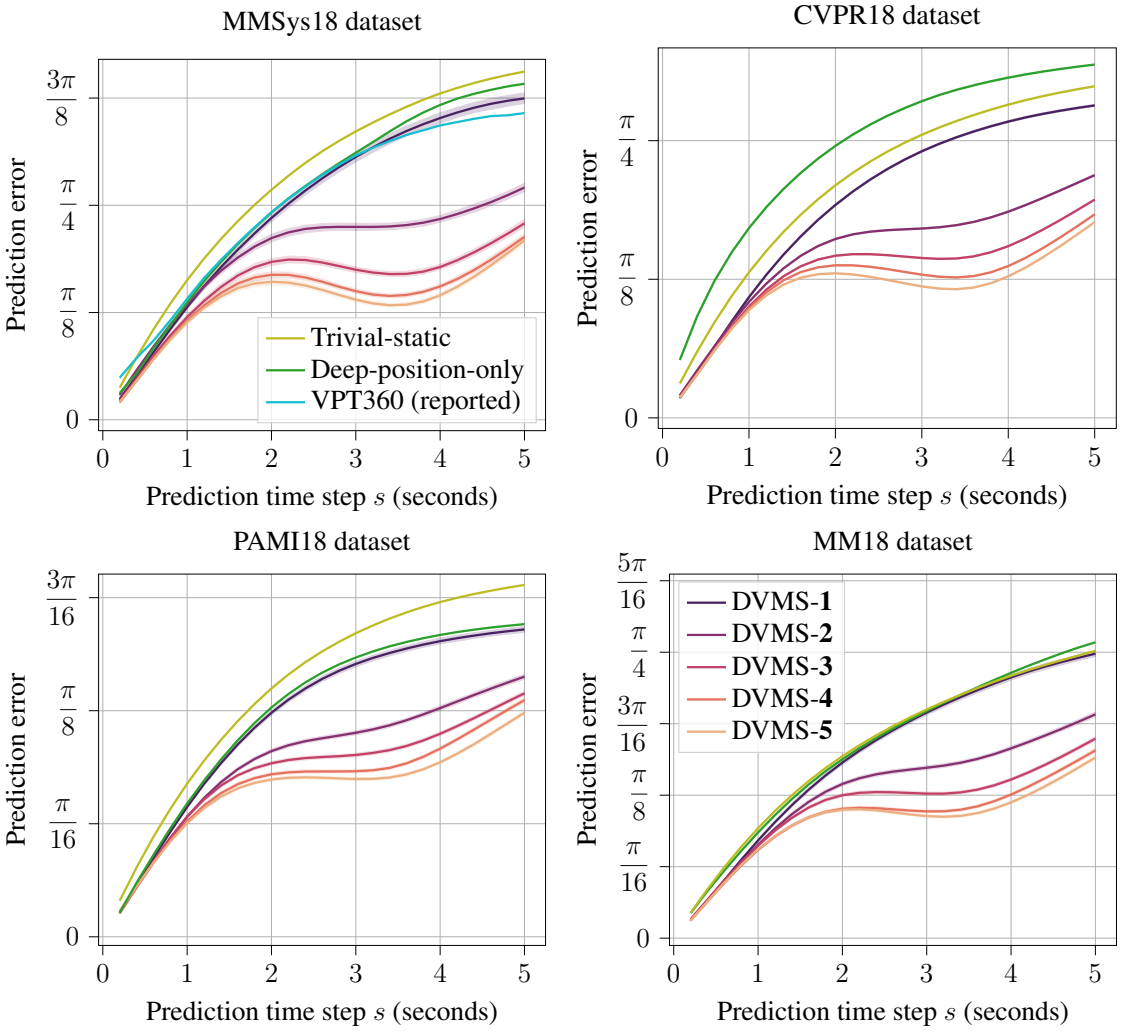


Figure *3.8: Prediction error (great-circle distance) of DVMS (ours) against state-of-the-art single trajectory predictors on the four evaluated datasets. Colors have the same meaning across all subfigures.*

Deep-position-only on the PAMI18 and MM18 datasets, by **2.1%** and **2.8%**, respectively, hence meeting (P2). On CVPR18, DVMS largely outperforms *Deep-position-only* by **19.5%**, which may suggest that *Deep-position-only* was not properly trained on this dataset. DVMS-**1** also consistently outperforms the *Trivial-static* baseline (by **8.3%** on average), which is expected. For (P1), we observe for $K = 2$ a **25%** reduction in prediction error for $s \leq 5$sec. with DVMS, compared to the single prediction competitors *Deep-position-only* and VPT360. For higher K, the error reduction increases, and tends to saturate for $K = 4$ then $K = 5$. DVMS hence meets both (P1) and (P2) on these datasets.

Fig. 3.9 compares the performance of DVMS with the MANTRA-*adapted* competitor on the same datasets. Detailed prediction results can also be found in in Tables 3.1, 3.2, 3.3 and 3.4. We first notice that for every $K = 1, \ldots, 5$, DVMS consistently yields a lower

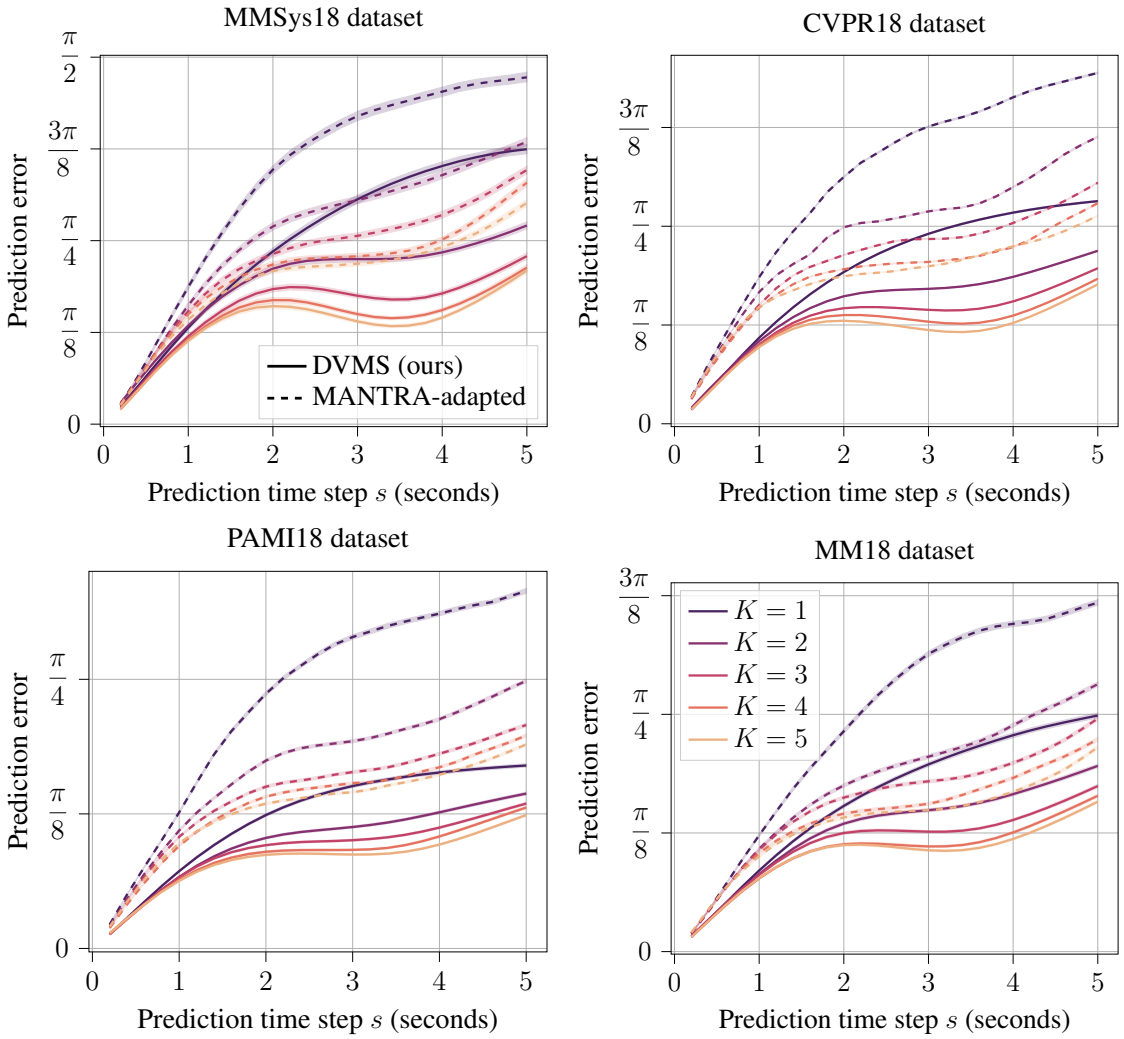


Figure *3.9: Prediction error (BMS metric) of DVMS (ours, solid lines) against MANTRA-adapted (dashed lines) on the four evaluated datasets. Colors have the same meaning across all subfigures.*

prediction error than MANTRA-*adapted*. Also, we observe that MANTRA-*adapted* does not match the state of the art performance of *Deep-position-only* for $K = 1$. Over all datasets, for $s \leq 5$sec., the prediction gains of DVMS over MANTRA-*adapted* range from **25.9% to 47.3%** (average 36.0%) for $K = 1$, from **25.6% to 40.9%** (average 32.7%) for $K = 2$, from **27.4% to 36.8%** (average 32.6%) for $K = 3$, from **27.2% to 36.7%** (average 32.6%) for $K = 4$, and from **26.0% to 37.5%** (average 32.8%) for $K = 5$.

Constructing futures by combining past with future pieces from the training set does not seem sufficient for MANTRA-*adapted* to produce diverse enough futures, compared with DVMS which instead modulates the initial state of the sequence decoder with a random component. The results on all four datasets therefore show that DVMS is able to produce diverse predictions (P1), outperforming the multiple prediction competitor

*Table 3.2: Prediction error over all $s \leq H$ on the CVPR18 dataset. Lowest prediction error for a given K is underlined, lowest prediction error for all K is highlighted in **bold**.*

Method		Average prediction error				
		$s \leq 1s$	$s \leq 2s$	$s \leq 3s$	$s \leq 4s$	$s \leq 5s$
Trivial-static ($K = 1$)		0.263	0.417	0.528	0.610	0.672
Deep-position-only ($K = 1$)		0.369	0.529	0.637	0.713	0.768
MANTRA-adapted	$K = 1$	0.351	0.594	0.767	0.887	0.981
	$K = 2$	0.323	0.503	0.608	0.678	0.755
	$K = 3$	0.298	0.456	0.544	0.598	0.656
	$K = 4$	0.292	0.433	0.500	0.543	0.596
	$K = 5$	0.303	0.426	0.487	0.533	0.581
DVMS (ours)	$K = 1$	0.200	0.355	0.470	0.555	0.618
	$K = 2$	0.200	0.326	0.394	0.435	0.477
	$K = 3$	0.190	0.305	0.357	0.383	0.419
	$K = 4$	0.187	0.295	0.337	0.355	0.387
	$K = 5$	**0.186**	**0.287**	**0.321**	**0.335**	**0.366**

*Table 3.3: Prediction error over all $s \leq H$ on the PAMI18 dataset. Lowest prediction error for a given K is underlined, lowest prediction error for all K is highlighted in **bold**.*

Method		Average prediction error				
		$s \leq 1s$	$s \leq 2s$	$s \leq 3s$	$s \leq 4s$	$s \leq 5s$
Trivial-static ($K = 1$)		0.169	0.270	0.345	0.399	0.439
Deep-position-only ($K = 1$)		0.140	0.239	0.311	0.361	0.396
MANTRA-adapted	$K = 1$	0.236	0.429	0.571	0.666	0.736
	$K = 2$	0.211	0.343	0.426	0.479	0.530
	$K = 3$	0.202	0.313	0.375	0.417	0.457
	$K = 4$	0.186	0.291	0.351	0.389	0.428
	$K = 5$	0.194	0.290	0.342	0.378	0.413
DVMS (ours)	$K = 1$	0.135	0.233	0.304	0.353	0.388
	$K = 2$	0.127	0.207	0.253	0.284	0.313
	$K = 3$	0.128	0.202	0.238	0.262	0.289
	$K = 4$	**0.124**	0.192	0.224	0.244	0.271
	$K = 5$	0.125	**0.189**	**0.218**	**0.235**	**0.258**

MANTRA-*adapted*, while providing comparable performance to state-of-the-art single trajectory predictors when $K = 1$ (P2).

We ran experiments where DVMS was tested on different datasets than it was trained on to assess its generalization capabilities. We report the cross-dataset performance in Table 3.5. We can see that models trained on smaller datasets such as MMSys18 and MM18 struggle to generalize to other datasets, with a prediction error usually higher than the other models. However, we observe that models trained on CVPR18 (the largest of the four datasets) tend to generalize well to other datasets, even outperforming models

*Table 3.4: Prediction error over all $s \leq H$ on the MM18 dataset. Lowest prediction error for a given K is underlined, lowest prediction error for all K is highlighted in **bold**.*

Method		Average prediction error				
		$s \leq 1s$	$s \leq 2s$	$s \leq 3s$	$s \leq 4s$	$s \leq 5s$
Trivial-static ($K = 1$)		0.190	0.309	0.400	0.471	0.530
Deep-position-only ($K = 1$)		0.183	0.301	0.392	0.466	0.529
MANTRA-adapted	$K = 1$	0.221	0.411	0.572	0.693	0.779
	$K = 2$	0.204	0.341	0.431	0.499	0.565
	$K = 3$	0.204	0.330	0.402	0.450	0.501
	$K = 4$	0.202	0.310	0.366	0.409	0.456
	$K = 5$	0.199	0.302	0.356	0.391	0.435
DVMS (ours)	$K = 1$	0.159	0.282	0.377	0.453	0.514
	$K = 2$	0.155	0.262	0.326	0.369	0.410
	$K = 3$	0.156	0.254	0.302	0.330	0.364
	$K = 4$	**0.148**	**0.236**	0.275	0.298	0.332
	$K = 5$	0.149	0.237	**0.273**	**0.292**	**0.322**

trained MMSys18 and MM18 on their own test datasets in some cases, without any kind of dataset-specific fine-tuning. We recommend using the CVPR18 dataset to train DVMS, as models trained on CVPR18 are always the best or second-best performing on any test dataset.

Fig. 3.10 shows qualitative examples of multiple trajectory prediction. It shows similar past trajectories of two different users, yielding distant future trajectories. When $K = 2$, we observe that DVMS produces different plausible trajectories, where one matches best the first user and the other the second user.

Computational cost: Hardware used to train and test the methods is a Nvidia RTX 3080 with 10GB of video RAM on a station with 128GB of RAM. Table 3.6 shows that DVMS and MANTRA-*adapted* have significantly less weights than both single prediction methods *Deep-position-only* and VPT360. While DVMS has more neural network parameters than MANTRA-*adapted*, the execution time to generate a trajectory at test time is 14% less than MANTRA-*adapted*. This is due to MANTRA-*adapted* having to do memory lookup.

Indeed, MANTRA-*adapted* has an extra memory, which DVMS does not, and the size of this memory, shown in Table 3.7 in percentage of the training set size, varies with the target accuracy and the dataset (and hence cannot be generalized to other datasets before actual training). Also, training MANTRA-*adapted* requires two phases, the first to train the auto-encoder, the second for the memory writing controller.

Hence, on four datasets of head motion data, DVMS achieves better prediction performance for lower computational resources than single prediction methods VPT360 and *Deep-position-only*, and for lower or equivalent resources than MANTRA-*adapted*.

*Table 3.5: Prediction error over $s \leq 5s$ when training and testing on different datasets. For a given test dataset and a given K, the lowest prediction error is highlighted in **bold**, the second lowest prediction error is underlined.*

		Test dataset			
	Train dataset	MMSys18	CVPR18	PAMI18	MM18
$K = 1$	MMSys18	0.790	0.695	0.466	0.706
	CVPR18	**0.778**	**0.618**	0.397	0.561
	PAMI18	0.789	0.643	**0.388**	0.718
	MM18	0.881	0.796	0.802	**0.514**
$K = 2$	MMSys18	0.613	0.569	0.420	0.542
	CVPR18	**0.581**	**0.477**	0.321	0.437
	PAMI18	0.604	0.493	**0.313**	0.501
	MM18	0.698	0.606	0.530	**0.410**
$K = 3$	MMSys18	**0.501**	0.491	0.371	0.414
	CVPR18	0.503	**0.419**	0.295	**0.362**
	PAMI18	0.530	0.442	**0.289**	0.440
	MM18	0.597	0.533	0.487	0.364
$K = 4$	MMSys18	**0.452**	0.444	0.350	0.354
	CVPR18	0.460	**0.387**	0.275	0.341
	PAMI18	0.476	0.404	**0.271**	0.407
	MM18	0.531	0.481	0.435	**0.332**
$K = 5$	MMSys18	0.432	0.418	0.330	0.347
	CVPR18	**0.427**	**0.366**	0.261	**0.321**
	PAMI18	0.451	0.385	**0.258**	0.376
	MM18	0.483	0.434	0.364	0.322

Table 3.6: Computational cost of the different models.

Model	# parameters	Latency (ms)
Trivial-static	*0*	*< 0.01*
Deep-position-only	4.21M	46.98
VPT360	6.3M	N/A
MANTRA-adapted	**76k**	6.81
DVMS (ours)	110k	**5.87**

3.5 Analysis of the DVMS latent space and likelihood estimation

In this section, we first analyze the structure of the latent space learned from the trajectory data and we connect latent space locations and values of z with physical properties. We

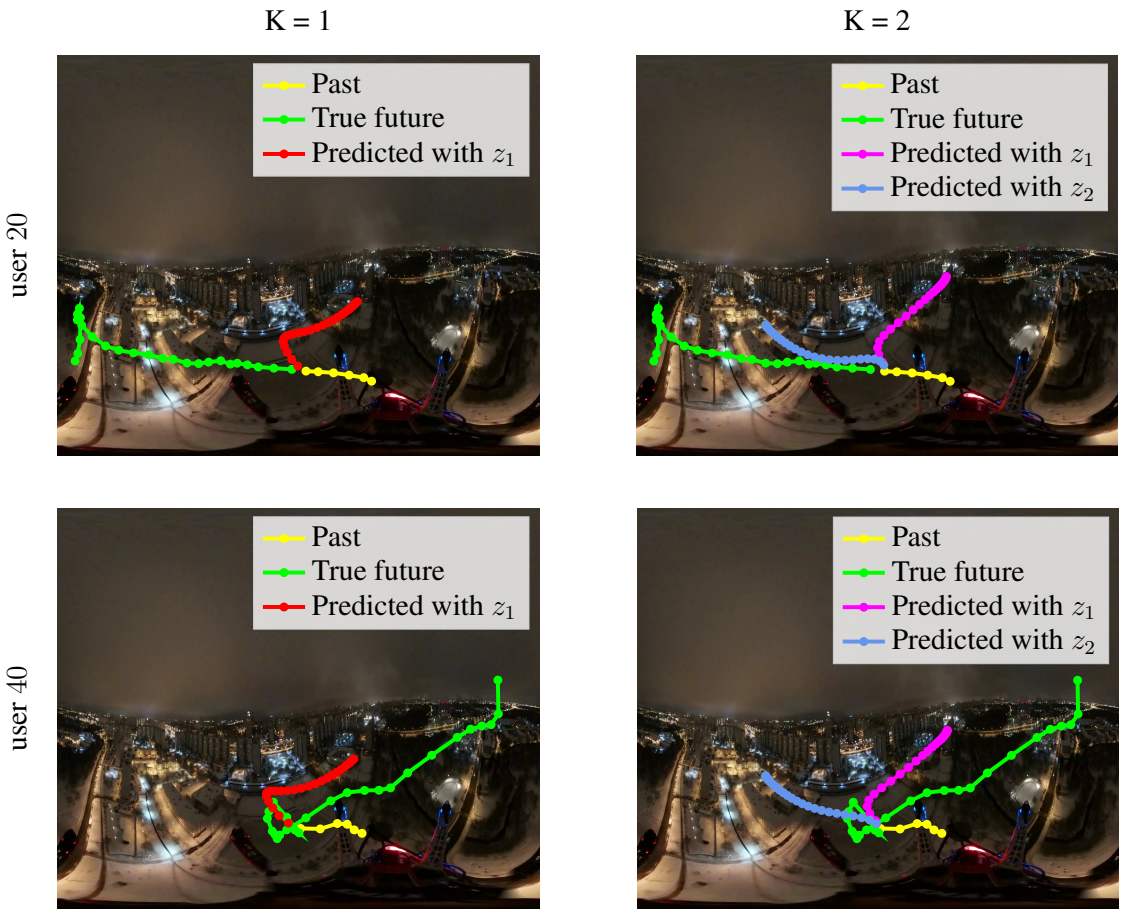


Figure *3.10: Examples of generated trajectories. Two different users (rows) have close past trajectories for the same timestamp $t = 8$sec. of the same video DroneFlight (MMSys18 dataset), but their future trajectories are significantly different. Predicting only one future (left column) does not enable good prediction of both futures, while predicting multiple (right column) does.*

then present our method to estimate the likelihood of every of the K generated trajectories, instrumental to deploy DVMS in a streaming system (evaluated in chapter 4).

3.5.1 Linking latent space features to trajectory properties

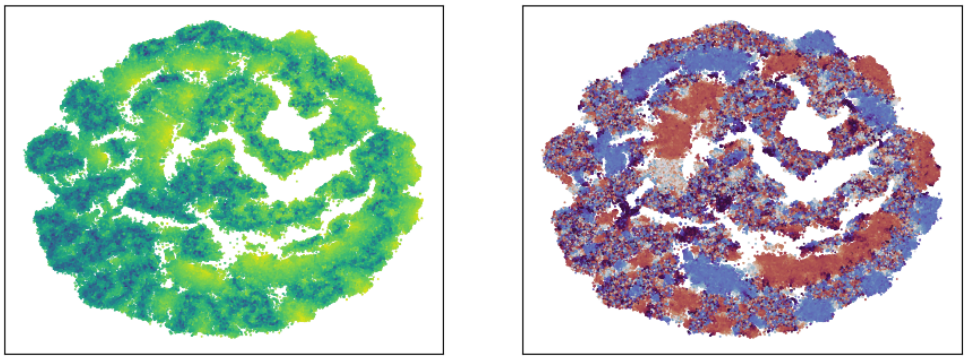
The model learns a representation of the past trajectory before being combined with z to generate a future trajectory. In this section, we first show what trajectory properties the encoder is able to perceive, and then we analyze the impact that different values of z can have on the generated trajectories when combined with the output of the encoder.

Table 3.7: Memory size (in number and percentage of training samples) of the MANTRA-adapted method for different number of predicted trajectories K across all the datasets.

Dataset Training set size	MMSys18 12600	CVPR18 560342	PAMI18 271440	MM18 32160
$K = 1$	6413 (50.90%)	258758 (46.18%)	79503 (29.29%)	10520 (32.71%)
$K = 2$	3601 (28.58%)	142113 (25.36%)	34564 (12.73%)	4575 (14.23%)
$K = 3$	2041 (16.20%)	83702 (14.94%)	20059 (7.39%)	2489 (7.74%)
$K = 4$	1227 (9.74%)	50265 (8.97%)	10470 (3.86%)	1431 (4.45%)
$K = 5$	**811** **(6.44%)**	**36325** **(6.48%)**	**7632** **(2.81%)**	**797** **(2.48%)**

3.5.1.1 Learned representation of past trajectories

We define as "embedding of the past trajectory" the output of the last layer of the encoder of DVMS, i.e., the output of the last orange layer in Fig. 3.7 (128 dimensions). Fig. 3.11 shows a 2D representation (obtained with t-SNE (van der Maaten & Hinton, 2008)) of all the embeddings of the test set of the CVPR18 dataset obtained with a model trained to predict three trajectories ($K = 3$) on the train set of the CVPR18 dataset. Each dot corresponds to a past trajectory, and embeddings that are similar in the 128-dimensional space should be close to each other on the 2D representation.



(a) Colored by past trajectory speed *(b) Colored by past trajectory direction*

Figure *3.11: 2D representation of the embeddings of past trajectories learned by the encoder on the CVPR dataset.*

The embeddings dots were colored according to the speed of their corresponding past trajectories on Fig. 3.11a. Blue dots represent embeddings of low-speed trajectories and yellow dots represent embeddings of high-speed trajectories. We can see some areas

where the colors are well separated, with yellow "high-speed areas" and blue "low-speed areas".

The embeddings dots were colored according to the direction of their corresponding past trajectories on Fig. 3.11b. Blue dots represent embeddings of trajectories going to the left and red dots represent embeddings of trajectories going to the right. We can see some areas where the colors are well separated, with some plain blue (resp. red) areas where all the embeddings correspond to trajectories going to the left (resp. right).

We can observe a correspondance between "high-speed areas" and areas where the direction is clearly left or right. "Low-speed areas" correlate with areas where the direction seems random. From these observations, we can deduce that the model is able to differentiate and represent different speeds and angles in its latent space. Specifically, it can easily identify high speed trajectories coming from left or right.

3.5.1.2 Impact of z

Depending on weight initialization, training set and data order, the model will map different trajectory features to z. Here we show an example for a model trained to predict three trajectories ($K = 3$) on the CVPR dataset.

To understand and evaluate the impact of z in the model, we show the estimated probability density functions (PDF) of the speed and direction of the output trajectories generated with each z_k. The approximate probability density functions are obtained through kernel density estimation (KDE), which we consider to be easier to read and understand than histograms for our data.

Fig. 3.12 shows the impact of z on trajectory speed. Fig. 3.12-left shows the distribution of the past and future trajectory speeds. Fig. 3.12-center shows the distribution of the output trajectory speeds generated with each z_k and that corresponding to the ground truth future (GT). Fig. 3.12-right shows the distribution of the ratios between past and future speed. A ratio of 10^0 (1) means that the generated/future trajectory has the same speed as its corresponding past. A ratio greater (resp. lower) than 1 means that the generated/future trajectory has a greater (resp. lower) speed than its corresponding past.

We can see that the predicted speed is always at least slightly lower than the actual speed (which we see for all K on all datasets), most likely because predicting higher speed (longer) trajectories will lead to higher error on average. The model learns to be conservative by predicting shorter (lower speed) trajectories. We also see much less variance in predicted speed than in actual speed (which we also see for all K on all datasets), but having more predicted trajectories (higher K) allows for more diversity overall, since each z can cover different parts of the distribution. In this example, the model learns to "specialize" z: z_2 generates low speed trajectories while z_1 and z_3 give similar speeds, usually in the same order of magnitude as GT.

Fig. 3.13 shows the impact of z on trajectory direction. Fig. 3.13-left shows the distribution of the past and future trajectory directions. Fig. 3.13-center shows the distribution of the output trajectory directions generated with each z_k compared to the ground truth future (GT). Fig. 3.13-right shows the distribution of the differences between past

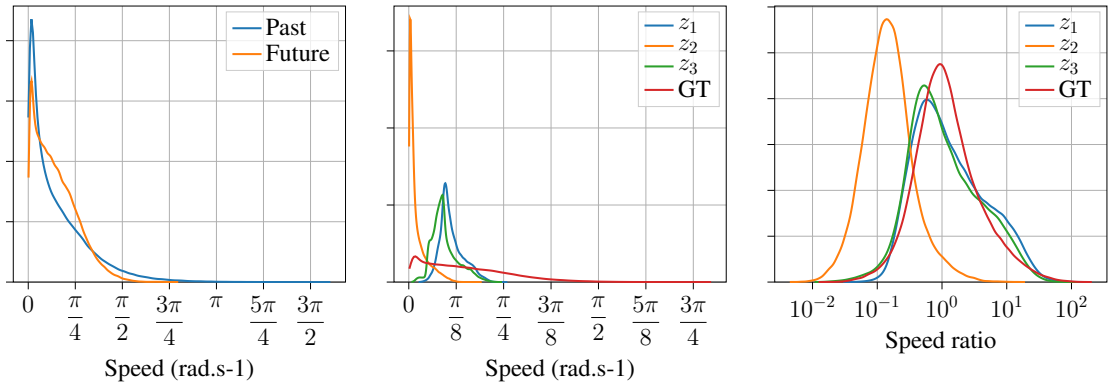


Figure *3.12: Distribution of the trajectory speeds depending on z.*

and future direction. A difference of 0 means that the generated/future trajectory kept going in the same direction as its corresponding past. A difference greater (resp. lower) than 0 means that the generated/future trajectory turned right (resp. left) relative to its corresponding past.

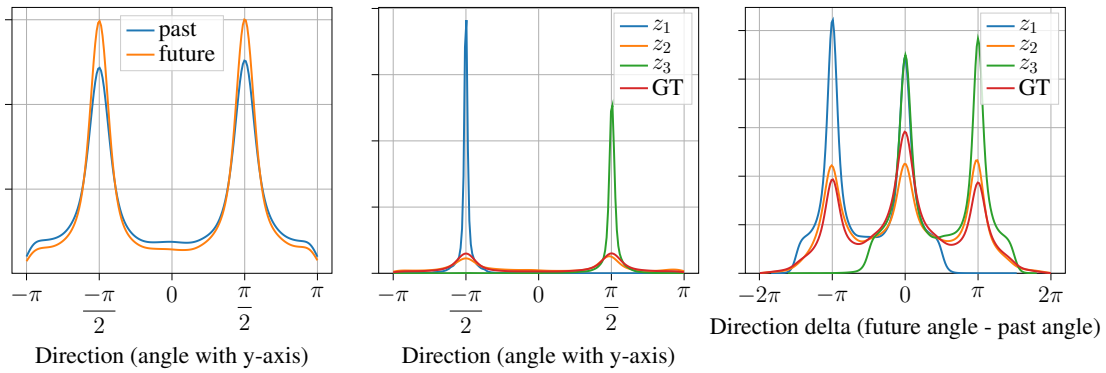


Figure *3.13: Distribution of the trajectory directions depending on z.*

The direction peaks that we observe on Fig. 3.13-left and Fig. 3.13-center indicate more trajectories going left ($\frac{-\pi}{2}$) or right ($\frac{\pi}{2}$) than in other directions. This means there is a lot more horizontal head movement than vertical head movement, which is expected in head motion data. We can discern 3 peaks on Fig. 3.13-right. The first peak at $-\pi$ corresponds to cases where the past was going to the right but the future is going to the left, the second peak at 0 corresponds to cases where the past and the future go in the same direction, and the third peak at $+\pi$ corresponds to cases where the past was going to the left but the future is going to the right.

In this example, while z_1 and z_3 give similar speeds, the generated trajectories have completely different directions. With z_1, the generated trajectories always go in the left direction. With z_3, the generated trajectories always go in the right direction. The direc-

tion of trajectories generated with z_2 follow a distribution close to GT. We can see that in order to go left, trajectories generated with z_1 will either continue in the same direction as the past or turn around to go left if the past was going right. Similarly, trajectories generated with z_3 will either continue or turn around in order to always go right.

In conclusion, DVMS learns to efficiently use the values of z by specializing them. Each z_k will generate a different "type" of future trajectory. In this example, we have seen that one z_k is used to generate low speed futures, while the other two are used to generate higher-speed futures, but with opposite directions.

3.5.2 Exploiting properties of z to estimate trajectory likelihood

We have seen that the model learns to differentiate trajectories based on their features such as speed (length) and direction to project them into a meaningful latent space, and that the model also learns to specialize provided values of z into specific types of trajectories. In this section, we present how we can take advantage the latent space properties of our variational predictor DVMS to estimate the likelihood of the multiple predicted trajectories. First, we present the general approach to likelihood estimation in Sec. 3.5.2.1. Second, we study in Sec 3.5.2.2 a stationarity hypothesis we make to produce our estimator. Third, we demonstrate in Sec. 3.5.3 the performance of our likelihood estimator, including disaggregated results and analysis over video categories.

3.5.2.1 Definition of the likelihood estimator

For a regression problem, the likelihood $Pr[\mathbf{y}^k_{t:t+H}|\mathbf{x}_{0:t}]$ of a future trajectory can be expressed with $\exp^{-D(\mathbf{y}_{t:t+H},\mathbf{x}_{t:t+H})}$, hence estimating the likelihood is equivalent to estimating the distance of a trajectory to the ground truth, that is the negative log-likelihood. We denote by $err^k_{u,v,t}$ the error of the k-th generated trajectory $\mathbf{y}^k_{t:t+H}$, the motion of user u on video v at timestamp t, defined in Eq. 3.9.

$$err^k_{u,v,t} = D\left(\mathbf{y}^k_{t:t+H}, \mathbf{x}_{t:t+H}\right) \tag{3.9}$$

With a variational framework, a standard approach to estimate the likelihood would be to rely on the model (whose parameters are set from the training data) and on the known past $\mathbf{x}_{0:t}$.

In this work, we argue that this is not sufficient, and that other available information must be considered, namely the past generated trajectories $\mathbf{y}^k_{s:\min(s+H,t)}$, for all $k \in \{1, K\}$ and $s \in [0, t]$, and the errors obtained by every such trajectory when compared to the available ground truth at t $\mathbf{x}_{s:\min(s+H,t)}$. Indeed, these errors are informative of which z_k, for $k \in \{1, K\}$, have best coded the latent features connecting the future trajectory with the past trajectory.

If the errors over the various z_k, for $k \in \{1, K\}$, have sufficient stationarity in time, then we can exploit such stationarity to estimate the likelihood of the predicted trajectories. We therefore define an estimate the estimate $\widehat{err}^k_{u,v,t}(r)$ of $err^k_{u,v,t}$ in Eq. 3.10.

$$\widehat{err}^k_{u,v,t}(r) = D\left(\mathbf{y}^k_{t-r:\min(t-r+H,t)}, \mathbf{x}_{t-r:\min(t-r+H,t)}\right) \tag{3.10}$$

where r is a past window of size controlling the age of the trajectory ground truth to produce the error estimate. Let us recall that the z-space is discrete, with $\mathcal{Z}_K = \{z_k\}_{k=1}^K$. This means that $err_{u,v,t}^k$ is predicted by the error produced by the trajectory $\mathbf{y}_{t-r:t-r+H}^k$ generated with the same z_k and predicted at time $t-r$ over a horizon H, but with the error only counted on the timestamps for which the ground truth $\mathbf{x}^{u,v}$ is available, i.e., on $[t-r, \min(t-r+H, t)]$.

The accuracy of this estimator therefore depends on the stationarity in time of the distribution of the error over the latent values z_k, for $k \in \{1, K\}$. We study this stationarity next.

3.5.2.2 Study of the stationarity of error distribution in the latent space of DVMS

We first analyze how $err_{u,v,t}^k$ evolves over $t = t_{start} : T$, for given test videos v and users u from the MMSys18 dataset. The MMSys18 test set is made of 5 videos of 4 categories according to the taxonomy established by Almquist et al. (2018) (Romero Rondón et al., 2021): exploration (*PortoRiverside* and *PlanEnergyBioLab*), moving focus (*Turtle*), static focus with camera motion (*WaterPark*), static focus without camera motion (*Warship*). Fig. 3.14-left shows an example for video *PortoRiverside* and user $u = 56$. We observe that, for every $t \in \{t_{start} : T\}$, the future trajectory $\mathbf{y}_{t:t+H}^3$ produced by latent value z_3 consistently yields the lowest prediction error. Fig. 3.14-right shows that, for video *Turtle* and user $u = 28$, the lowest error is consistently produced by latent value z_2.

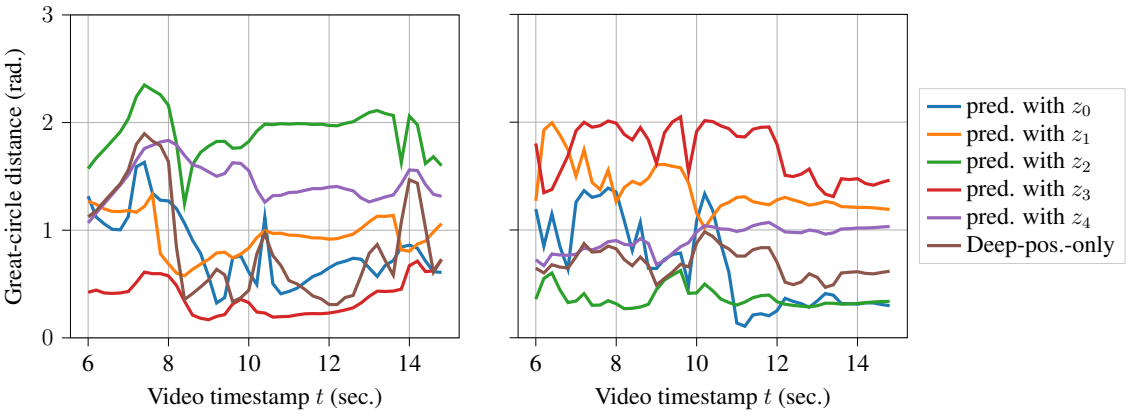


Figure *3.14: Prediction error for different latent values over time, test set of the MMSys18 dataset. Left: video PortoRiverside, user 56. Right: video Turtle, user 28.*

Second, as Fig. 3.14 only shows examples obtained for specific choices of (v, u), we investigate how representative these cases are. To do so, we define a *latent stationarity matrix* (LSM) per video v defined in Eq. 3.11, where U is the total number of user traces per video.

$$A_{ij}^v = \frac{1}{UK} \sum_{u=1}^{U} \sum_{k=0}^{K} \left(err_{u,v,i}^k - err_{u,v,j}^k \right)^2, (i,j) \in \{t_{start}, \ldots, T\}^2 \tag{3.11}$$

For every user u, the main term in the summation represents the difference, between timestamps i and j, in how the error is distributed in the discrete latent space $\mathcal{Z}_K$. Fig. 3.15 depicts such error differences as heatmaps for all the five videos of the test set of the MMSys18 dataset. Each heatmap shows how the distribution of the error over all $k \in 1, \ldots, K$ varies over time for each video. The more variation in prediction error over time for each k (i.e., the more variation in Fig. 3.14 over all users), the higher the value of A^v_{ij}.

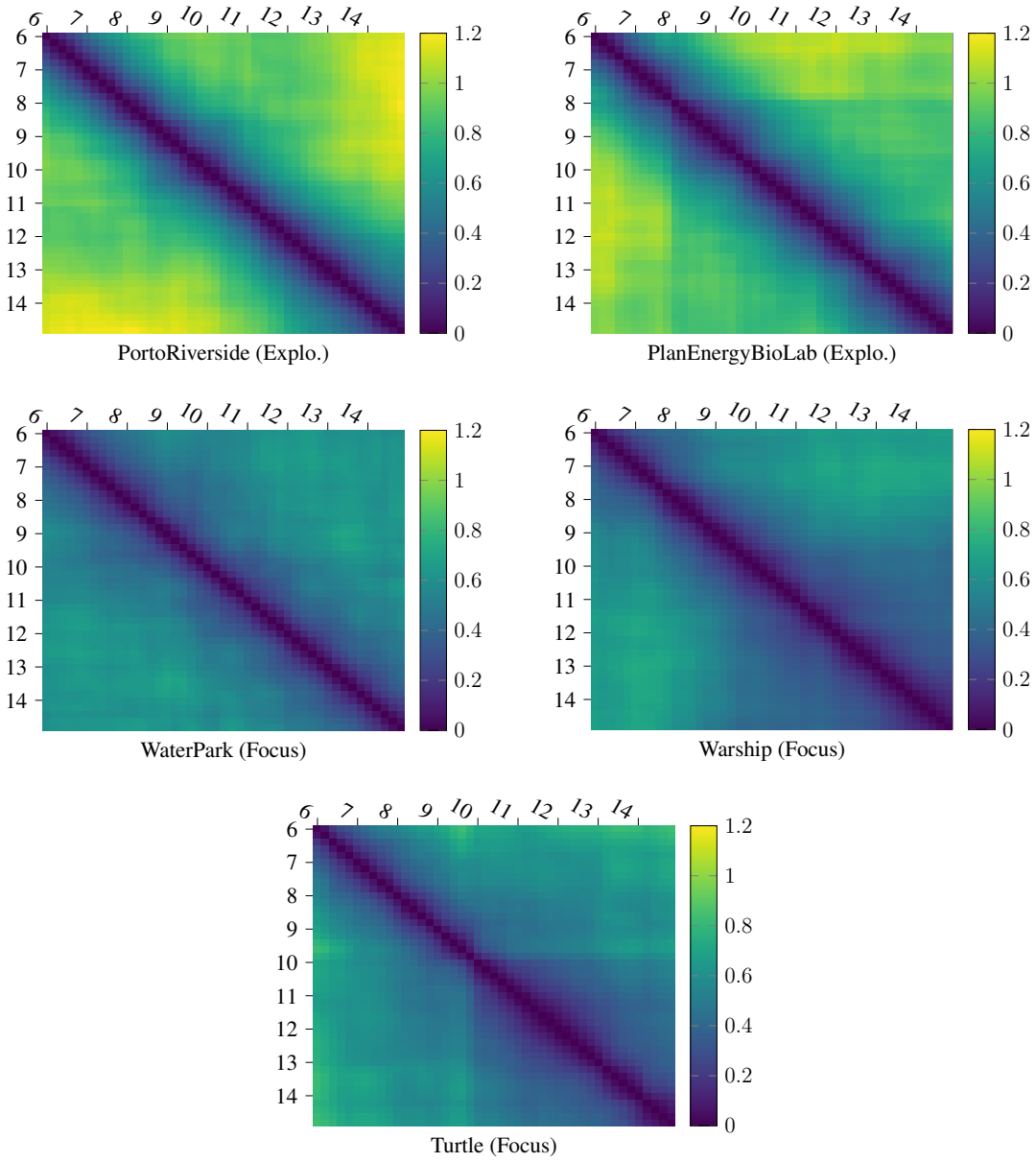


Figure *3.15: Latent Stationarity Matrix (LSM). The color scale codes for the error difference A^v_{ij}. Axes are in seconds ($t = 6$ sec. to $t = 15$ sec. so $t + H \leq T = 20$ sec.). Test videos of the MMSys18 dataset.*

First, we observe that, when the timestamps i and j are equal, error difference is null, which is expected. Also, it is interesting to observe that the closer i and j, the lower the error distance. This shows that the results illustrated in Fig. 3.14 for specific (u, v) pairs are general: the prediction error yielded by z_k varies more or less slowly over time (depending on the videos). Such stationarity may hence be exploited to produce error (i.e., likelihood) estimates. Similar qualitative results hold on the other datasets, but we do not show these maps for the sake of brevity. However, estimation results on all 3 datasets are shown in Sec. 3.5.3.

Second, it is interesting to observe that the level of stationarity/speed of variation of the error produced by every z_k, for $k \in \{1, K\}$, depends on the video category. For both exploration-type videos *PortoRiverside* and *PlanEnergyBioLab*, the error difference for a given timestamp difference is significantly higher than for the focus-type videos. The stationarity of latent errors is hence lower in exploration videos, and error estimates from the past timestamps should yields better estimates in Focus videos, which we investigate in the next section.

3.5.3 Results on trajectory likelihood estimation

Datasets: The results below are obtained on the three largest datasets among those considered in Sec. 3.4.4, namely MMSys18, CVPR18, and PAMI18.
Metrics: We measure the quality of the trajectory likelihood/error estimate with its Pearson correlation coefficient with the ground truth. Indeed, as the motivation for the present contribution is to benefit from such estimates in a stochastic formulation of resource optimization (see Sec. 3.3.1), we want to evaluate how much are the produced estimates linearly correlated with the ground truth error. If the correlation coefficient was 1, we would obtain the true likelihood. For any past window size r, for every tuple (v, u, t), we are interested in how the corresponding K samples $\left\{(err^k_{u,v,t}, \widehat{err}^k_{u,v,t}(r))\right\}_{k=1}^{K}$ are correlated. To allow considering a larger number of samples to obtain more confident estimates of the correlation coefficient, we consider $\left\{(err^k_{u,v,t}, \widehat{err}^k_{u,v,t}(r))\right\}_{k,u,t}$. However, to do so without artificially increasing the correlation coefficient owing to different average values of the samples over the (u, t) tuples, we first normalize $\left\{(err^k_{u,v,t}, \widehat{err}^k_{u,v,t}(r))\right\}_{k=1}^{K}$ independently for every (u, t). The correlation coefficient for every value of r is then computed on the normalized pairs of ground truth error and error estimate.
Results: Figures 3.16-3.18 show the evolution of the Pearson correlation coefficient with the past window size $r \in [0, H]$, with the shaded area representing the 95% confidence interval. To assess the correlation strength, we follow the recommendation from recent literature (Xue, Ali, Ding, & Cesar, 2021; Akoglu, 2018): low: $0.1 \leq |\text{corr}| < 0.3$; moderate; $0.3 \leq |\text{corr}| < 0.6$; high: $0.6 \leq |\text{corr}| \leq 1.0$.

It is important to note that the LSM is a novel type of characterization for 360° video. While previous characterizations were directly based on the video content or the user traces (Almquist et al., 2018; Nasrabadi et al., 2019; Rossi & Toni, 2020), the LSM represents characteristics of the latent space in connection with prediction performance.

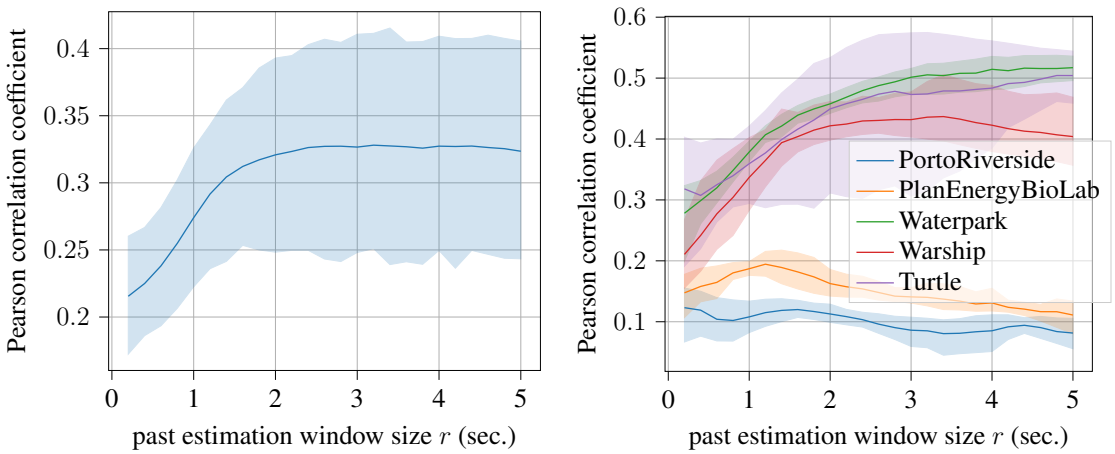


Figure *3.16: Correlation between estimated and ground truth error of predicted trajectories, on the MMSys18 dataset. Left: average over all test videos. Right: average per test video.*

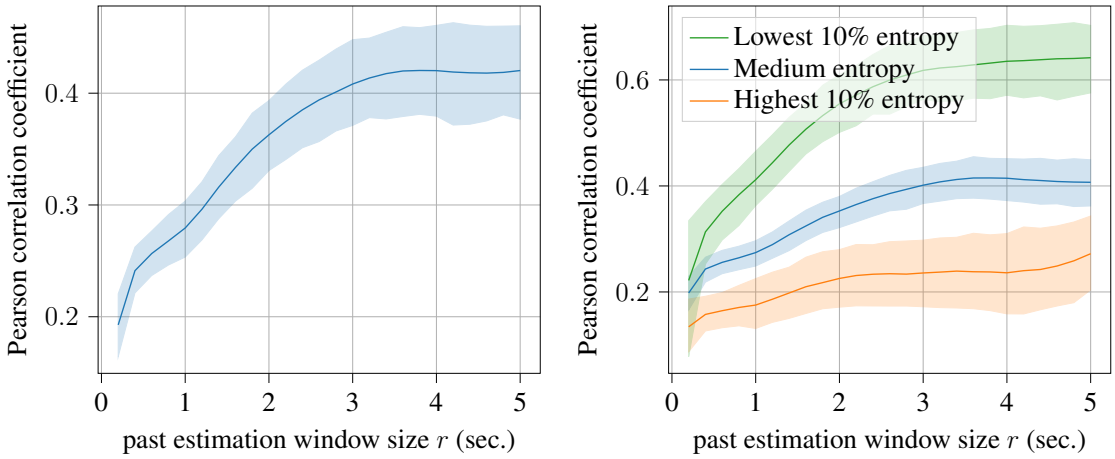


Figure *3.17: Correlation between estimated and ground truth error, on the PAMI18 dataset. Left: average over all test videos. Right: average per group.*

For the test sets of the PAMI18 and CVPR18 datasets, Fig. 3.17-left and 3.18-left show that there is a moderate significant correlation of the estimates with the ground truth errors. For the test set of the MMSys18 dataset, there is a moderate to low correlation, the significance being low possibly owing to the very low number of videos in the test set (only 5). We also observe that the correlation generally increases with r for the MMSys18 and PAMI18 datasets in Fig. 3.16 and 3.17 (reaching maximum level for $r \geq 2$ sec. and $r \geq 3.5$ sec., respectively), while for the CVPR18 dataset in Fig. 3.18, the average correlation reaches a maximum for $r = 3$ seconds then decreases.

It is interesting to disaggregate the results and analyze the correlation per video type. Regarding video types, as in Section 3.5.2.2, we follow the taxonomy of Almquist et al. (2018), who define the following categories: *static focus*, *moving focus*, *ride*, and *exploration*. Fig. 3.16-right shows the correlation results for each of the 5 test videos in the

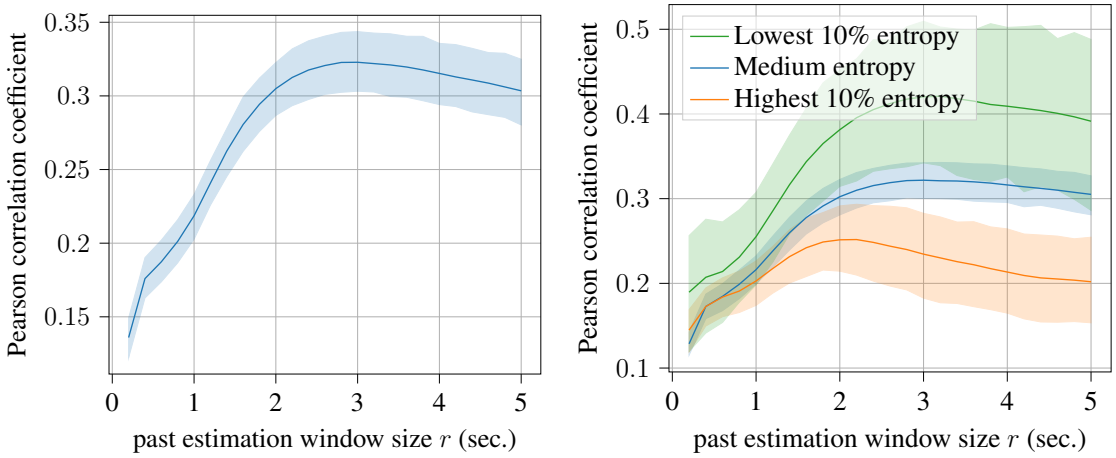


Figure *3.18: Correlation between estimated and ground truth error, on the CVPR18 dataset. Left: average over all test videos. Right: average per group.*

MMSys18 dataset. For both videos of type exploration, *PortoRiverside* and *PlanEnergy-Biolab*, the correlation is significantly low. However for the focus-type videos (*Warship*, *WaterPark* and *Turtle*), the correlation is moderate, being significant for the first two for $2 \leq r \leq 3$ sec., while for *Turtle*, the performance in unstable (it is worth noting that *Turtle* is of type moving focus video, which is not present in the training set). We also notice that the maximum correlation is obtained for $r \leq 1.5$ sec. for exploration videos, while the maximum is obtained for $r \geq 3$ sec. for focus-type videos.

The other two datasets have 16 (PAMI18) and 42 (CVPR18) test videos. Therefore, to categorize the video automatically between exploration and focus-type video, we resort to the method presented by Romero Rondón et al. (2021), consisting in associating exploration (resp. focus) videos with low (resp. high) entropy values of the saliency maps obtained from the user traces. Fig. 3.17-right and 3.18-right show the correlation results broken down into the 10% of videos with highest entropy, 10% with lowest entropy, and the rest. In Fig. 3.17-right, we observe that for PAMI18 focus videos, the correlation is strong and significant for $r \geq 3$ sec., while it is low significant for exploration videos and moderate significant for the rest. Similar trends can be observed with CVPR18 videos, where the correlation is low and significant for exploration videos, and moderate and significant for the rest. For exploration videos, the correlation is maximum for $r \sim 2$ sec., while it is maximum for $r \sim 3$ sec. for the rest.

Therefore, we have shown that our method is able to predict multiple diverse trajectories, providing estimates of their respective likelihood to leverage in stochastic optimization of resource allocation. The estimates are shown to correlate with ground truth, the level of correlation and past window size r providing highest correlation depending on the video category. In chapter 4, we show how multiple trajectories and their estimated likelihoods can be used in an 360° streaming system with trace-driven simulations in a new simulator that we propose and make publicly available.

3.6 Discussion

To the best of our knowledge, this work presents the first proposal to generate multiple plausible 360° head trajectories. Our DVMS learning framework therefore establishes a first baseline for comparison, and paves the way for more principled approaches to stochastic optimization of 360° streaming, and immersive streaming in general.

In particular, DVMS can be adapted to 6DoF immersive environments where both head-gaze and translational motion need to be predicted. It will also be most relevant for foveated rendering and streaming for ultra-high resolution eye-tracker equipped head-mounted displays, where the level of uncertainty is increased by the need to predict a restricted foveal area where the human focuses and which moves rapidly. It is also instrumental to enable the automatic triggering of interactive strategies when the predictability of the user motion is evaluated to be too low (Sassatelli et al., 2020). Also, we have shown that the latent variable with discrete values is learned to generate different types of trajectory, for example with low speed or with high speed with two different positions. Considering DVMS with content-aware architectures in connection to user and video profiles will enable to investigate more intricate connections between scene video content, user state and motion predictability, as preliminarily investigated in chapter 5.

Considering uncertainty in prediction is often approached with Bayesian neural networks (Neal, 2012; Kan et al., 2021; L. Yang et al., 2022). However, these approaches are computationally intensive and do not allow to capture mode diversity in the data (Fort et al., 2020). Alternatives exist to better learn data diversity, e.g., Monte-Carlo dropout (Gal & Ghahramani, 2016), or approaches based on memory networks like (Marchetti et al., 2020a). In contrast, our DVMS method builds on deep latent variable models, and proves lightweight, flexible and suited to the head motion data diversity. It confirms the interest in investigating more dynamic VAEs, to possibly design proper inference networks and conditional prior $p(z|\mathbf{x}_{0:t})$ in test. We discuss this approach in Sec. 3.7.2.

We have exemplified the DVMS prediction framework with a video-agnostic neural architecture. A direct perspective is to investigate DVMS performance when used in conjunction with a content-aware architecture. DVMS is compatible by design with sequence-to-sequence architectures, such as those used in the head motion prediction literature (Romero Rondón et al., 2021; Nguyen et al., 2018; Y. Xu et al., 2018).

3.7 Other investigated approaches to consider uncertainty

The idea behind DVMS was to alleviate the problem of uncertainty caused by the intrinsic randomness of head motion data. To do so, we proposed a stochastic model capable of estimating the probability distribution of future viewports by (i) predicting multiple possible trajectories of future head motion (Sec. 3.4), (ii) estimating the likelihood of these trajectories (Sec. 3.5.2). In this section, we provide a quick exploration of other ideas to reach the same objective.

The work discussed in this section is the outcome of two Master's theses internships, which were co-supervised by Prof. Lucile Sassatelli and myself. We worked on uncertainty quantification using the variational information bottleneck with Hugo Bell who was a Master student at the University of Edimburgh. We worked on formal approaches to variational sequence prediction with dynamical VAEs with Franz Franco Gallo who was a Master student at Université Côte d'Azur.

3.7.1 Uncertainty quantification with the variational information bottleneck

Predicting future head movements and being able to provide an indication of the confidence that we can have in the prediction would essentially achieve our desired objective. For this reason, we started to look into ways to quantify the uncertainty of our predictions. Uncertainty can be classified into two categories (Kendall & Gal, 2017):

- Aleatoric uncertainty, also known as statistical uncertainty, pertains to the inherent randomness and variability in experimental outcomes, exemplified by phenomena like coin flipping. This randomness originates from intrinsic stochastic elements, leading to outcomes like heads or tails, which cannot be definitively predicted even with the best possible model.

- In contrast, epistemic uncertainty, or systematic uncertainty, arises from a lack of knowledge about the underlying model. It represents the ignorance of the agent or decision maker regarding the true nature of the phenomenon, rather than an inherent randomness.

We can roughly simplify these definitions to fit our head prediction problem: aleatoric uncertainty originates from the randomness of the data and is irreducible, while epistemic uncertainty comes from the incorrectness of our model, and may reduce with more data.

Following work from A. A. Alemi, Fischer, and Dillon (2018), we looked into how we could leverage the variational information bottleneck (VIB) to quantify the uncertainty of our predictions. We provide a more detailed background as well as our methods and results in the next sections.

3.7.1.1 Background

The variational information bottleneck (VIB) (A. A. Alemi, Fischer, Dillon, & Murphy, 2017) establishes a variational approximation of the Information Bottleneck (IB) (Tishby & Zaslavsky, 2015). The VIB operates in the realm of supervised learning, mirroring the role of β-VAE (Burgess et al., 2018) in unsupervised learning. Both methodologies find their foundation in information-theoretic principles (A. Alemi et al., 2018). In the context of supervised learning, the IB addresses the challenge as a representation learning task. It aims to discover a probabilistic mapping from input data X to a latent representation Z, ensuring that this representation retains the capability to predict the corresponding labels

Y. This process operates under a constraint limiting the overall complexity of the learned representation.

In other words, the objective of the IB is to only learn the most useful latent representation of the input, while discarding irrelevant information. This is done by maximizing the mutual information between the latent representation and the output, while minimizing the mutual information between the input and the latent representation. While the mutual information optimization problem of the IB is intractable in general, A. A. Alemi et al. (2017) derived a simple tractable variational bound. In practice, the VIB is implemented as follows:

- a (learned) stochastic encoder transforms the input X into some encoding Z,
- a (learned) variational decoder outputs predicted labels $\hat{Y}$ from the sampled code Z,
- the loss function combines the cross-entropy between the outputs $\hat{Y}$ and the ground truth labels Y and the Kullback–Leibler divergence (KLD) between the conditional distribution of the latent codes learned by the encoder (given the inputs) and the density of the latent space estimated by a (learned) variational marginal.

A. A. Alemi et al. (2018) proposed two metrics to quantify the (epistemic) uncertainty with the VIB: (i) the entropy of the classifier H, and (ii) the KLD between the conditional distribution over latent codes given the input and the code space defined by the learned marginal. (also known as the rate R).

The VIB and the uncertainty quantification metrics that we just discussed were defined in the context of classification tasks. While the VIB appears to be easily transferrable to our encoder-decoder sequence-to-sequence framework (we can introduce a bottleneck between the encoder and the decoder), this is part of exciting new work to apply this approach to regression tasks (Lyu, Aminian, & Rodrigues, 2021; Ngampruetikorn & Schwab, 2022).

3.7.1.2 Methods

In order to use the VIB approach to uncertainty quantification, we chose to start with the adaptation of a simple model: the *deep-position-only* baseline introduced by Romero Rondón et al. (2021), discussed in Sec. 3.4.4.2. We adapted *deep-position-only* by introducing latent random variables between the encoder and decoder LSTMs. We briefly explain how this new model (namely, *pos-only-VIB*) works, and provide an illustration in Fig. 3.19.

First, instead of generating a single, deterministic hidden state, the LSTM encoder generates the parameters of a Gaussian distribution, our latent space. Second, the decoder is fed with samples from this distribution to generate future head trajectories.

To estimate epistemic uncertainty with *pos-only-VIB*, we used the rate R described by A. A. Alemi et al. (2018). To do so, we needed to estimate the density of the latent space with a marginal, as described in Sec. 3.7.1.1. We used Gaussian mixture of 200

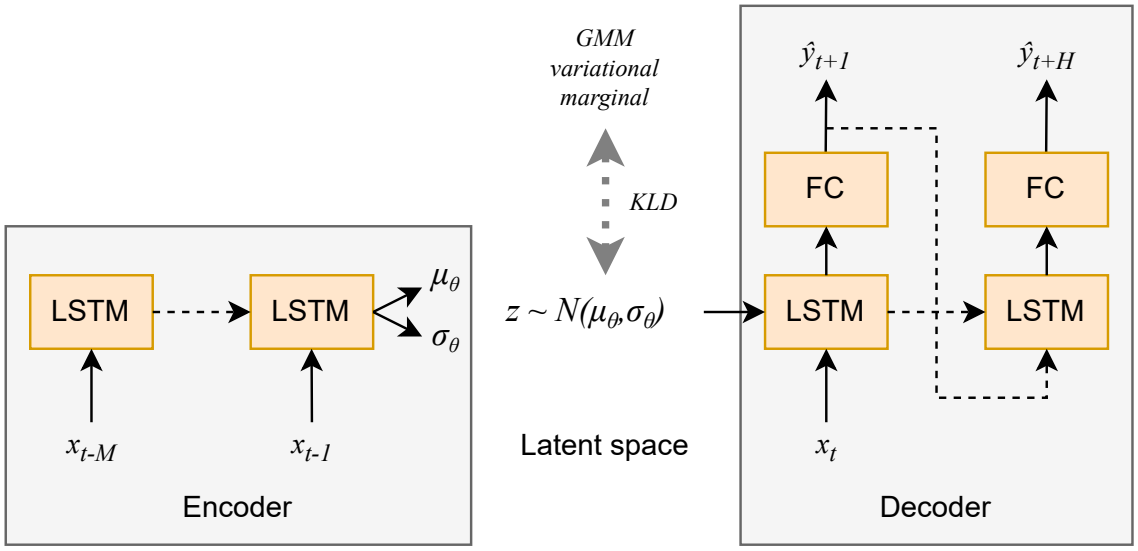


Figure *3.19: Architecture of the pos-only-VIB model.*

components, following A. A. Alemi et al. (2018). To estimate the aleatoric uncertainty, we followed the approach of Sinha et al. (2021), repeatedly sampling from the latent distribution for a test input, generating the corresponding network outputs and observing their empirical variance.

To evaluate the uncertainty estimates of our approach, we must assess how *well-calibrated* our model is, i.e., how well our uncertainty estimates align with true epistemic and aleatoric uncertainty. To evaluate our epistemic uncertainty estimates, we used the rate R for out-of-distribution (OOD) detection against an induced *distributional shift* (Ovadia et al., 2019), by using users and videos unseen in training on head motion datasets (MMSys18 and CVPR18, see Sec. 3.4.4.2) and by using synthetic data generated from different distributions. We used the AUROC metric as a threshold-free metric to measure the calibration of our model for epistemic uncertainty estimates (A. A. Alemi et al., 2018; van Amersfoort, Smith, Teh, & Gal, 2020; Postels et al., 2021). To evaluate our aleatoric uncertainty estimates, we measured the correlation between the empirical variance of our outputs (when repeatedly sampling from the latent space) and the prediction error on in-distribution (ID) trajectories (because aleatoric uncertainty is irreducible), following Postels et al. (2021).

3.7.1.3 Results

Regarding epistemic uncertainty quantification, the AUROC stayed around 0.5 for all cases on real world datasets, indicating there was no threshold on the rate R that allowed to differentiate between ID and OOD samples. On "easily" differentiable synthetic trajectories which were generated using radically different distributions, R was a good indicator of epistemic uncertainty, as the AUROC reached 0.9. However, the AUROC

stayed around 0.5 for more "difficult" cases of synthetic trajectories that were generated with speed and curvature constraints to look like realistic trajectories.

Regarding aleatoric uncertainty quantification, we found a strong correlation between our estimates and the prediction error, with Pearson and Spearman correlation scores of 0.912 and 0.794 respectively. The accuracy of of our aleatoric uncertainty estimates seemed to vary with the type of video (lower accuracy for exploration-type videos).

We also investigated a non-variational density estimation approach to uncertainty quantification, following the idea of Postels et al. (2021), fitting a conditional normalizing flow (CNF) (Ardizzone, Lüth, Kruse, Rother, & Köthe, 2019) to training set activations, allowing the exact evaluation of the log-likelihood of test inputs (Kingma & Dhariwal, 2018). However, this approach was found to be unsuitable in its current form for our purposes, as it assumes a uniform distribution over the output space (which is not the case for the head motion prediction task).

Although alternative methods for density estimation using Gaussian mixture models (often in the form of mixture density networks (Bishop, 1994)) have been developed in the uncertainty quantification literature, practical issues have been found with these models such as non-convergence for high-dimensional problems (Postels, Ferroni, Coskun, Navab, & Tombari, 2019) and mode collapse (Makansi, Ilg, Cicek, & Brox, 2019).

Overall, the outcomes of our experiments demonstrated the efficacy of our model in accurately quantifying aleatoric uncertainty in real head motion trajectories. However, its success in quantifying epistemic uncertainty was observed solely in the case of highly simplified synthetic trajectories. Through visualizations of epistemic uncertainty estimates and model latent representations (not shown here), we hypothesized that the failures in epistemic uncertainty estimation could be attributed to a recently identified issue called feature collapse (van Amersfoort et al., 2020), affecting the representations generated by our model.

In summary, despite negative results on epistemic uncertainty quantification, we think that our approach yielded promising outcomes, encouraging further exploration. This highlights the need for in-depth examinations of the acquired learned representations and the creation of more efficient representation techniques, pointing toward avenues for future development.

3.7.2 Dynamical variational auto-encoders

With DVMS, we took inspiration from RNN-based VAEs to build a (discrete) variational model for sequence prediction (see Sec. 3.4 for a background on deep generative models and the design of DVMS). Dynamical VAEs were recently formalized as a new class of models by Girin et al. (2021), extending VAE to sequence modeling. In their comprehensive review, they detailed the temporal dependencies of the latent variables in the inference (encoder) and generative (decoder) networks of various DVAE models. In this section, we investigate the potential of DVAE approaches compared to our DVMS approach. We first provide a background on DVAE in Sec. 3.7.2.1, then we provide some details on the specific DVAE approaches that we chose to explore for our head motion

prediction tasks, with graphical models of their generative and inference networks in Sec. 3.7.2.2 and Sec. 3.7.2.3.

3.7.2.1 Background on DVAE

To better understand what DVAE models are and how they are related to different classes of models, we look at the proposed taxonomy in Fig. 3.20, originally proposed by Girin et al. (2021).

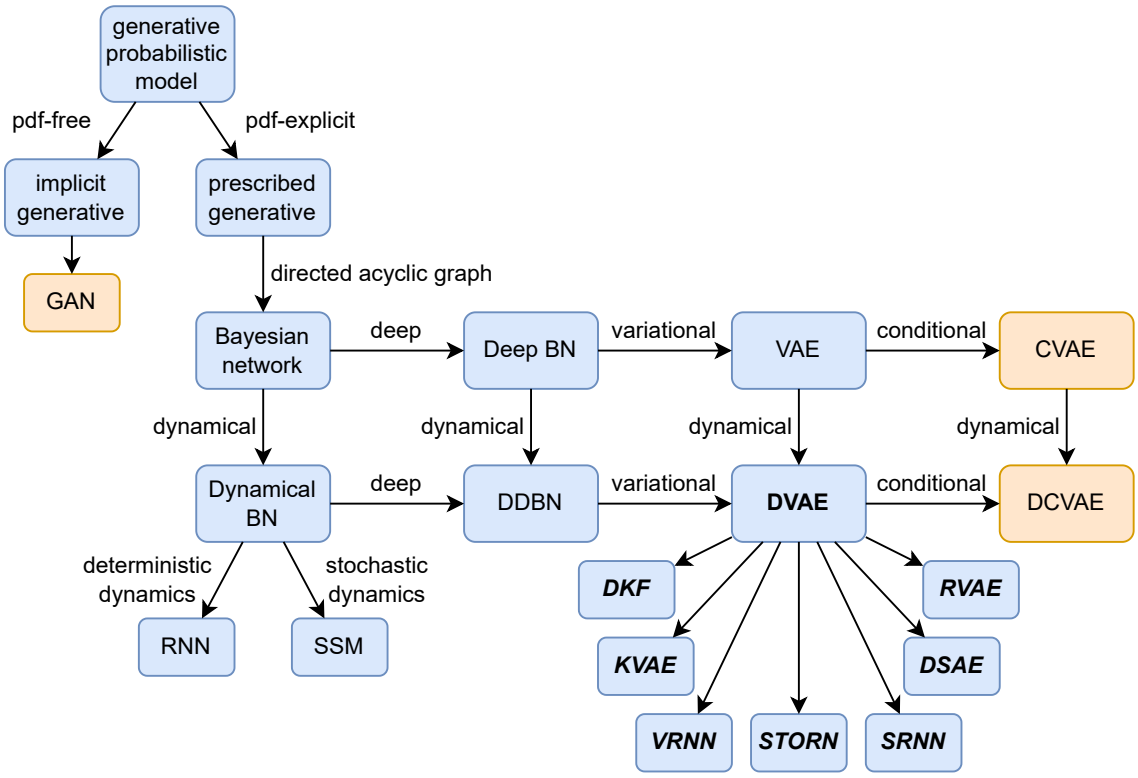


Figure *3.20: Graphical taxonomy of generative probabilistic models, adapted from Girin et al. (2021).*

We left the original elements of the proposed taxonomy of generative probabilistic models in blue. In orange, we added elements that we consider relevant here:

(i) An example of implicit generative model (Mohamed & Lakshminarayanan, 2017), that can generate data "directly". Implicit generative models can be seen as "decoder-only", as they do not have an explicit inference (encoder) network that learns an approximate posterior distribution of the latent space. Generative adversarial networks (GANs) constitute the most popular example (Goodfellow et al., 2014, 2020) of this type of model.

(ii) Conditional variational auto-encoders (CVAEs), which are considered as beyond the scope of the DVAE review, but that we consider relevant to our head motion prediction problem (e.g., we can condition the reconstruction of the future

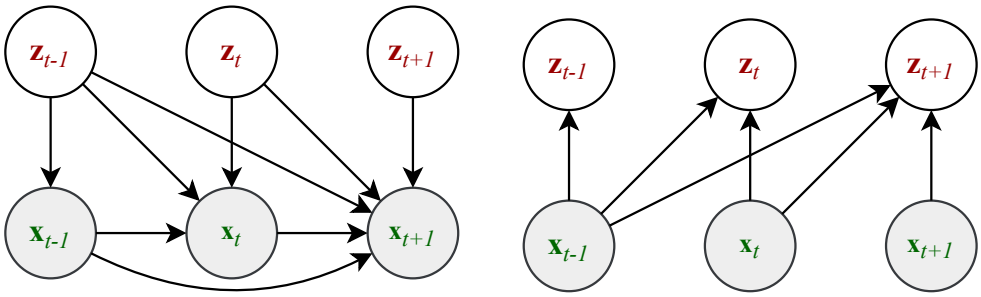
trajectory on the past trajectory or the video content). DVAEs could also be seen as a special case where the built-in temporal dependency in the latent variable itself constitutes a conditioning.

We highlighted the focus of the DVAE review in **bold**, and examples of DVAE models they explore in ***bold italics***.

We position DVMS in this taxonomy. DVMS does not fall in the DVAE class, as it is not a VAE. DVMS is an implicit generation model with no inference network. While DVMS is designed for sequence modeling with temporal dependencies, another important difference with DVAEs is the time-independence of z in our case: unlike DVAEs, we only draw z once between the encoder and the decoder (like in a regular VAE), as seen in Fig. 3.6. As an implicit generation model, it introduces noise in the model from a prior distribution without approximating a posterior, losing the need to train with the variational lower bound. A parallel and can be drawn between DVMS and the generator of a conditional GAN in this sense. In fact, the variety loss that we use to enforce diversity was first proposed for SocialGAN (Gupta et al., 2018), an implicit generation model.

3.7.2.2 Stochastic recurrent networks (STORN)

As a first baseline, we looked into stochastic recurrent networks (STORN) (Bayer & Osendorfer, 2015). We made this choice because this model was the most straightforward way to apply the DVAE framework to DVMS or *deep-position-only* (Sec. 3.4.4.2), due to architectural similarities. It is interesting to note that VRAE, that we briefly discuss in Sec. 3.4.1, can been considered a simplified version of STORN.We provide graphical models of the generation and inference networks of STORN in Fig. 3.21.



(a) generation *(b) inference*

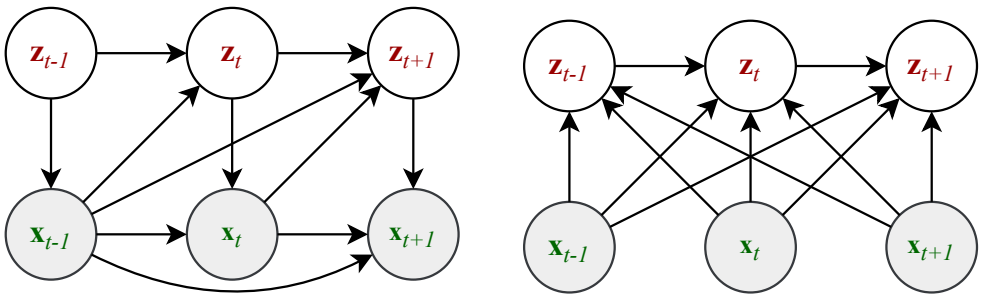
Figure *3.21: STORN graphical models showing the temporal dependencies, adapted from Girin et al. (2021).*

At generation time (decoding, Fig. 3.21a), x_t depends on $x_{1:t-1}$ and $z_{1:t}$ (we chose to omit the internal deterministic state h_t for brevity), while z_t is considered to be independent and identically distributed (i.i.d.) with a standard Gaussian distribution. At inference time (encoding, Fig. 3.21b), z_t depends on $x_{1:t}$ (we chose to omit the internal state g_t for brevity).

With this formulation, we do not expect to reach good prediction performance, since z_t will be sampled from a standard Gaussian distribution that does not hold any information about the past, and cannot represent multiple modes of the data. When training with a mean square error (MSE) loss combined with the KLD, the model did not converge. This is not a surprising result, as we introduce random noise at each decoding step. Training with the variety loss would partially solve this problem, but likely lead to overfitting due to a large difference between the approximate posterior learned at inference and the standard Gaussian prior used at inference time. Unfortunately, we did not have time to pursue further experiments with STORN due to a lack of time.

3.7.2.3 Stochastic recurrent neural networks (SRNN)

Our objective was to adapt stochastic recurrent neural networks (SRNN) to our problem (Fraccaro, Sø nderby, Paquet, & Winther, 2016). According to the authors, their objective was to "glue (or stack) a deterministic recurrent neural network and a state space model together to form a stochastic and sequential neural generative model". Among all investigated DVAE models, SRNN was found to be the best performing model on the analysis-synthesis task, on both speech and human motion data (Girin et al., 2021). With the help of the graphical models shown in Fig. 3.22, we further detail the motivations behind an adaptation of SRNN to head motion prediction.



(a) generation *(b) inference*

Figure *3.22: SRNN graphical models showing the temporal dependencies, adapted from Girin et al. (2021).*

At generation time (decoding, Fig. 3.22a), x_t depends on $x_{1:t-1}$ and z_t (we chose to omit the internal deterministic state h_t for brevity). A key difference with STORN is the introduction of the parameterization of the prior: z_t is dependent on z_{t-1} and $x_{1:t-1}$, which means that the parameters of the distribution are learned, instead of being a standard Gaussian. At inference time (encoding, Fig. 3.22b), z_t depends on z_{t-1} and $x_{1:T}$, with T the full length of the sequence (we chose to omit the internal states h_t and g_t for brevity).

With this formulation, we can expect to reach a better prediction performance, for two main reasons:

(i) During inference, we can learn bidirectional dependencies (z_t depends on past and future $x_{1:T}$), providing the model with a higher learning capacity.

(ii) More importantly, during generation, the samples are not drawn from a standard Gaussian distribution, but from a learned approximate prior that depends on past values of x and z, reducing the difference with the approximate posterior learned at inference.

Reaching better prediction performance with SRNN might not be a straightforward path however, due to likely instabilities during training. Appropriate losses and training procedures must be considered (Babaeizadeh et al., 2018). Unfortunately, we were not able to implement and test an SRNN-based model to fit our task, due to a lack of time. We believe that further research deserves to be carried out in this direction.

3.8 Conclusion

In this chapter, we have presented the first method for multiple head motion prediction in 360° videos, motivated by the user motion uncertainty yielding a high diversity of future trajectories. Our main contribution is a new learning framework, called DVMS, which builds on deep latent variable models and allows to predict multiple future trajectories from a given past. We have designed a training procedure to obtain a flexible and lightweight stochastic prediction model compatible with sequence-to-sequence architectures. We have analyzed the structure of the learned latent space and the impact of the latent variable on the generated futures, and are able to connect them with physical properties of the trajectories. We have assessed DVMS on 4 datasets and show that it outperforms competitors adapted from the self-driving domain by up to 41%, on prediction horizons up to 5 seconds. By exploiting the stationarity of the prediction error over the latent space, our method provides likelihood estimates of every predicted trajectory, enabling direct integration in streaming optimization. In a more exploratory section, we have provided leads into interesting research directions for uncertainty quantification for sequence-to-sequence architectures and sequence prediction with variational auto-encoders. DVMS paves the way for multiple head motion prediction in 360° videos, and an evaluation of the gains when predicted trajectories and their likelihoods are used by a 360° adaptive streaming system is provided in chapter 4.

CHAPTER 4

Simulating motion prediction and adaptive bitrate strategies for 360° video streaming

Adaptive bitrate (ABR) algorithms are used in streaming media to adjust video or audio quality based on the viewer's network conditions to provide a smooth playback experience. With the rise of virtual reality (VR) headsets, 360° video streaming is growing rapidly and requires efficient ABR strategies to also adapt the video quality to the user's head position.

However, research in this field is often difficult to compare due to a lack of reproducible simulations. To address this problem, we provide SMART360, a 360° streaming simulation environment to compare motion prediction and adaptive bitrates strategies.

We provide sample inputs and baseline algorithms along with the simulator, as well as examples of results and visualizations that can be obtained with SMART360. The code and data are made publicly available.

This new simulator enables an extensive evaluation of the interest of our DVMS proposal for a streaming system. On real-world user, video, and networking data, we show that predicting multiple trajectories yields higher fairness between the traces, the gains for 20 to 30% of the users reaching up to 10% in visual quality for the best number K of trajectories to generate.

Contents

4.1 Introduction

As seen in Sec. 2.1.4, adaptive streaming of 360° videos has been the subject of many research works in the past few years (Qian et al., 2016; Corbillon, Simon, et al., 2017; S. Park, Hoai, et al., 2021). While these works present new approaches and methods to improve the quality of experience (QoE) or bandwidth usage of adaptive streaming systems, it is often difficult to compare them fairly, as the code for simulating them is not always provided.

We proposed the discrete variational multiple sequence learning framework (DVMS) in chapter 3 as a way to predict multiple possible future trajectories of head motion for people watching 360° videos. Evaluating the practical gains of this framework in a streaming system would only be possible through (i) user experiments where a video would be streamed over a network with a variable bandwidth and DVMS would be used to predict head positions in real time, or (ii) simulations of real-world streaming situations with network traces and simulated users with head motion traces.

Unfortunately, user experiments can be very costly and take a lot of time, cannot test millions of cases like simulations can, and are difficult to reproduce. With the many public datasets of head motion traces and network traces, comparing motion prediction and adaptive bitrate strategies for 360° video streaming should be an easy task. However, despite the wide availability of such datasets, we could not find any suitable public software tool to simulate a realistic tile-based VR adaptive streaming system, flexible enough to work with any 360° video, head motion and network trace.

For these reasons, we present SMART360, a 360° streaming simulation environment that can be used to compare head motion prediction and ABR algorithms. Our contributions are the following:

- We provide a new simulator[*], equipped with large datasets and baseline algorithms that builds upon the existing solutions, with explanations about the code structure and logic.
- We also make the preprocessing pipeline available[†] for transparency and to give the ability to easily create new input configurations for the simulator.
- We explain in detail how SMART360 can be used by researchers to implement and compare existing and new motion prediction and adaptive bitrate strategies and show examples of metrics and visualizations that can readily be used to evaluate their algorithms. SMART360 can be used to implement any kind of viewport prediction algorithm to work with tile-based adaptive streaming.
- We provide an in-depth analysis of the performance of multiple trajectory prediction with the DVMS framework when incorporated in a streaming system, obtaining results from nearly 5 million simulations using 3378 head motion traces from 132

[*] https://gitlab.com/SMART360/SMART360-simulator
[†] https://gitlab.com/SMART360/SMART360-preprocessing

different users watching 94 different videos, 40 different network traces with 5 different buffer settings and 7 viewport prediction algorithms, including state-of-the-art competitors and variants of DVMS. Our results show that predicting multiple trajectories (under constant bandwidth budget) yields a higher fairness between the traces of the user-video pairs, with less traces with the worst quality of experience (QoE) level, and a close (resp. slightly higher) number of traces with maximum QoE when predicting with $K = 5$ (resp. choosing the best K per trace) futures with DVMS. We also quantify that choosing the best K yields up to a 10% higher quality in the FoV (up to a 5% better QoE) for the 20% to 30% of traces with the highest prediction errors.

The work presented in this chapter was the object of a conference paper presented at the *Open Dataset and Software* track of the 14th ACM Multimedia Systems Conference (MMSys '22) (Guimard & Sassatelli, 2023). The simulation results showing the system gains of DVMS were submitted as part of a journal extension to the DVMS article (Guimard, Sassatelli, et al., 2022) presented in chapter 3 and accepted with minor revisions in the ACM Transactions on Multimedia Computing, Communications, and Applications journal (TOMM) (Guimard et al., 2024).

The chapter is organized as follows: Sec. 4.2 makes an inventory of the existing tools to compare 360° adaptive streaming strategies and algorithms and exhibits their shortcomings. Sec. 4.3 presents the simulator inputs and the preprocessing pipeline for these inputs. Sec. 4.4 details the the architecture of the proposed simulator, explaining the file and object structure, as well as the algorithmic logic behind SMART360. Sec. 4.5 explains how SMART360 can be used to compare motion predictors and adaptive bitrate algorithms, and serves a guide to improve reproducibility. Sec. 4.6 provides simulation results when prediction multiple head trajectories with DVMS, and shows the benefits of our proposed framework. the extensive evaluation of our DVMS proposal in a streaming system. Sec. 4.7 discusses how the simulator can still be improved and what the simulation results of DVMS tell us for future work on viewport prediction for 360° video streaming, and Sec. 4.8 concludes the chapter.

4.2 Related work

As a result of the lack of reproducible simulations for most of the 360° adaptive streaming research, several tools have been made available in recent years in an effort to improve reproducibility in this field.

Ribezzo, De Cicco, Palmisano, and Mascolo (2020) released TAPAS-360° [*], an open-source emulator that enables designing and experimenting omnidirectional video streaming algorithms. Unfortunately, TAPAS-360° does not support tile-based streaming, but works with a set of pre-defined "views". This makes it impossible to use with tile-based bitrate adaptation algorithms, which are the most common type of bitrate adaptation algorithms for 360° video streaming.

[*] https://github.com/c3lab/tapas360

Spiteri (Spiteri, 2021) released Sabre360*, a simulation testbed for 360° videos as an extension of Sabre† (Spiteri et al., 2019), an open-source simulation environment for ABR algorithms. While Sabre360 can be used to compare adaptive bitrate algorithms, it has some drawbacks: (i) it does not implement stalls, but plays "blank tiles" instead, (ii) it is built around a "view" system that only supports one kind of tiling layout (4x4 tiles), and (iii) the ABR optimization for quality allocation is done between each tile download, and makes an individual request for each tile of each segment, which is not realistic. In a real-world scenario, the ABR has to plan in advance and make a single request for several tiles.

X. Jiang et al. (2018) provide code for simulating 360° bitrate adaptation and motion prediction along with Plato‡, but the lack of documentation and obscure file structure makes it difficult to use, precluding other researchers from using it and test new algorithms.

Finally, Chopra et al. (2021) provide the code for PARIMA§, which allows to test and compare their model to some baselines with QoE metrics, but it is not a streaming simulation since it does not consider network aspects.

Our simulator takes a lot of inspiration from Sabre360, which we consider to be the closest solution to the problem we want to solve. Our work aims at rectifying any shortcomings the existing solutions may have for comparing motion prediction and adaptive bitrate strategies in the context of 360° streaming.

4.3 Data preprocessing

The objective of SMART360 is to provide a simulation environment that enables the comparison of ABR and viewport prediction algorithms when streaming 360° videos with network constraints. In this section, we describe the necessary inputs the simulator needs to perform this task, as well as the preprocessing pipeline the data undergoes before being used by the simulator.

4.3.1 Simulator inputs

All the input data for the SMART360 simulator is provided in the `config/` directory of the simulator repository. The input data uses the same JSON format as Sabre360 (Spiteri, 2021). The data provided in the `SMART360-simulator/config/` directory is split in two types: *real* and *synthetic* data. The *real* data is extracted from multiple public datasets and is described in the following subsections. The *synthetic* data contains simple cases of network traces with constant bandwidth and manifests describing uniformly-sized 360°

*https://github.com/UMass-LIDS/sabre360
†https://github.com/UMass-LIDS/sabre
‡https://github.com/federerjiang/Plato
§https://github.com/sarthak-chakraborty/PARIMA

videos. The user head motion traces found in the *synthetic* directory are copied from *real* data.

4.3.1.1 Network traces

The network traces describe the available bandwidth and the latency over time in different situations. They allow for realistic simulations where the bandwidth is highly variable. The network traces provided in the *real* input data are the same as the ones used in Sabre360, and come from the 4G/LTE dataset published by van der Hooft et al. (2016). They are made of 40 traces of bandwidth measurements along several routes in the city of Ghent, Belgium.

For the comparisons between ABR algorithms to be relevant, we need to be in a situation where the algorithm has to adapt to the network constraints. On the one hand, if the bandwidth is very high relative to the video bitrate, there is no need for ABR streaming, as we can just download everything in the highest quality without any rebuffering (stall) event. On the other hand, if the bandwidth is very low relative to the video bitrate, ABR streaming is not so useful either, as we can only download everything in the lowest quality. To make for a relevant comparison between ABR algorithms, we provide a *Jupyter* notebook to scale the network traces relatively to the video bitrates, as illustrated in Fig. 4.1. This notebook is available in the `SMART360-simulator/notebooks/` directory.

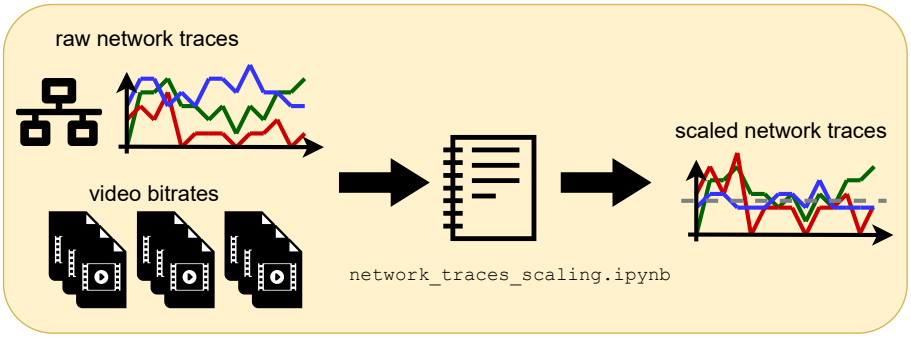


Figure *4.1: Network trace scaling principle.*

4.3.1.2 User head motion traces

The user head motion traces describe the behavior of people watching 360° videos. They contain the coordinates of the head orientation over time. This allows calculating which tiles are visible to the user at any given time during the video. We provide 3518 head motion traces from users watching 94 different videos, extracted from three of the datasets used by Romero Rondón et al. (2020) in their framework to evaluate head motion prediction methods in 360° videos*. The traces have a 5 Hz sampling rate and use a 3D

*https://gitlab.com/miguelfromeror/head-motion-prediction

Cartesian coordinate system, where the orientation of the head is represented as a point on the unit sphere.

We provide a *Python* script, available in the `SMART360-preprocessing/` root directory to convert the traces from their original CSV format to a JSON format similar to the one used in Sabre360.

4.3.1.3 Video manifests

The video manifests describe the video files to be streamed over the Internet. In the case of 360° tiled videos, the manifests describe the tiling layout and the different quality levels of encoding. The SMART360 simulator uses the video manifest to get the size of each downloaded tile. We provide simplified JSON video manifests for the 94 videos mentioned in Sec. 4.3.1.2 in the same format as the one used in Sabre360. We detail the preprocessing steps to obtain the video manifests from MP4 video files in the next subsection.

4.3.2 Preprocessing pipeline

The preprocessing pipeline is based on TOUCAN-preprocessing*, a *Java* command line application to convert a regular 360° videos into DASH-SRD described videos, using *FFmpeg* and *MP4Box*, released by Dambra et al. (2018). We have made some changes to simplify the original pipeline, update the encoding parameters, and adapt the input and output formats to our problem. The preprocessing pipeline is described in Fig. 4.2 and detailed in the following subsections.

4.3.2.1 Video tiling and re-encoding

First, the MP4 videos are split into tiles using the *FFmpeg* crop filter. Since cropping the videos requires re-encoding them, we choose to re-encode the video tiles in different quality levels while tiling them. The tiling layouts and quality levels are configurable settings that can be specified in an XML file for each video.

The videos are re-encoded with *libx265*, using the HEVC compression standard. Different quality levels are achieved using different constant rate factors (CRFs). CRF is a method of video compression that is designed to maintain a constant level of perceived quality, as opposed to constant bitrate (CBR) encoding, but similar to using a constant quantization parameter (CQP). Unlike CQP, CRF adjusts the QP to compress different frames by varying amounts by taking motion into account. For high-motion frames, the QP is increased to compress the frame more, and for low-motion frames, the QP is lowered to reduce compression. This leads to a varying bitrate allocation over time, resulting in a more efficient use of the available bandwidth. While constant bitrate and constrained CRF may be better suited for streaming to avoid bitrate variations, CRF is better suited than CQP (the most popular encoding mode to compare adaptive bitrate strategies in 360°

*https://github.com/UCA4SVR/TOUCAN-preprocessing

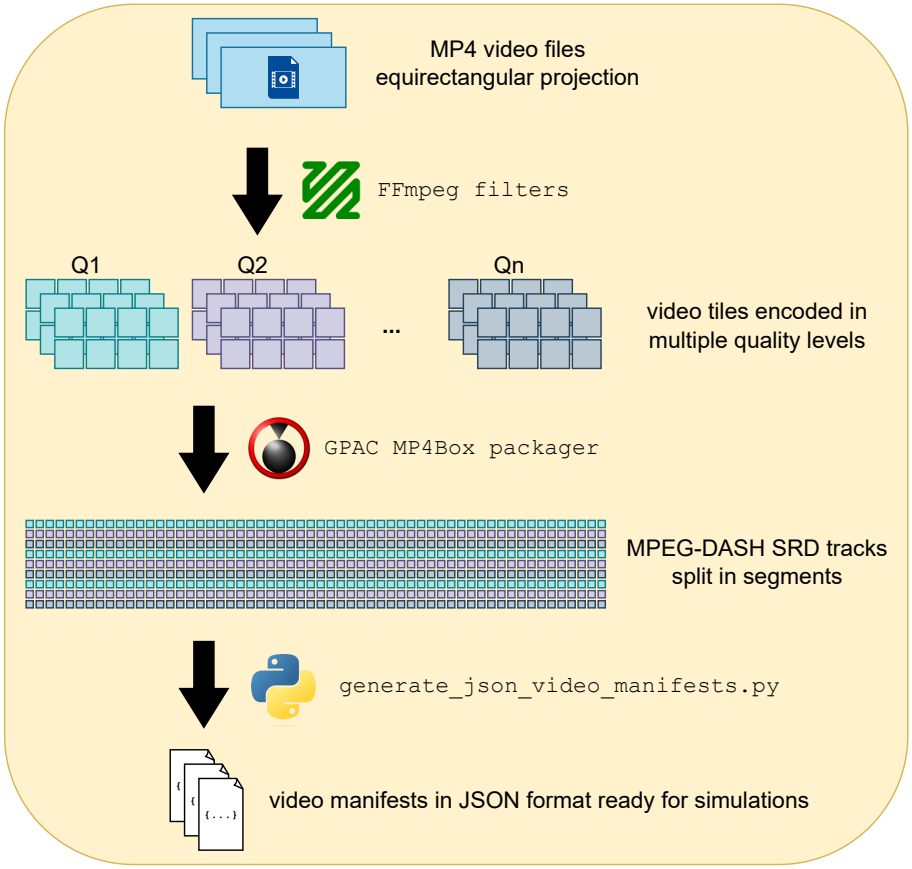


Figure *4.2: SMART360 video preprocessing pipeline.*

videos (Yaqoob et al., 2020)), as it results in a more constant bitrate (Robitza, 2019). CRF was chosen as the best compromise between bitrate stability (better than CQP), efficiency (not wasting bits like constant bitrate), and encoding time (not needing multiple passes like constrained CRF).

4.3.2.2 DASH packaging

Once the videos are cropped into the desired tiling layouts and encoded in the appropriate quality levels, we use the *MP4Box* multimedia packager to obtain a DASH-SRD compliant video split in segments. The segment duration is also a configurable parameter that can be specified in the same XML file as mentioned in Sec. 4.3.2.1. The output files generated by this preprocessing step are MP4 tracks and an XML manifest for each video, which correspond to the files that can be streamed over the Internet.

4.3.2.3 JSON file generation

Finally, we provide a *Python* script, available in the `SMART360-preprocessing/` root directory to build the JSON manifests that can be used by the simulator. This script

simply reads the files that were previously generated and keeps only the information that is relevant for the simulations to put them in the JSON video manifests described in 4.3.1.3.

4.4 Simulator architecture

The SMART360 simulator architecture is based on the architecture of the Sabre360 simulator, with substantial differences. The changes mainly aim at rectifying the shortcomings formulated in Sec. 4.2, namely: (i) introducing actual stall events that pause the video playback instead of playing blank tiles, (ii) re-thinking the coordinate system and modifying the headset model to support any rectangular tiling layout, and (iii) re-designing the simulator and ABR logic, enabling the planning of quality allocation for multiple tiles and segments in advance.

4.4.1 File and object structure

We present a simplified class diagram in Fig. 4.3, where we choose to only keep the relevant attributes and methods of the SMART360 simulator. The classes are separated in multiple files located at `SMART360-simulator/simulator/`. The classes highlighted in red in the diagram, *BandwidthEstimator*, *TiledABR*, and *ViewportPredictor* are classes that can be easily extended to implement new algorithms. We detail the file structure and classes of the simulator in the following subsections.

4.4.1.1 Session

The `session.py` file contains the *Session* and *SessionInfo* classes. The *Session* class is the main class that contains all the objects necessary to the simulation. The *Session::run* method is the entry point of the simulator and is described in Algo. 1. The *SessionInfo* class is mainly used to access information and objects like the buffer, log file, or viewport predictor from other objects.

4.4.1.2 Buffer

The `buffer.py` file contains the *TiledBuffer* class. This class contains the buffer in the form of a two-dimensional *NumPy* array of size $B \times T$, where B is the buffer size (in number of segments) and T is the number of tiles in the video. This class also provides methods to update the buffer.

4.4.1.3 Headset

The `headset.py` file contains the *HeadsetModel* and *HeadsetConfig* classes. These classes contain information about the headset configuration (tile layout, FoV size) and provide methods to calculate which tiles are visible, given the user's head coordinates. Unlike most existing tools, the tile calculation considers the distortion produced by the

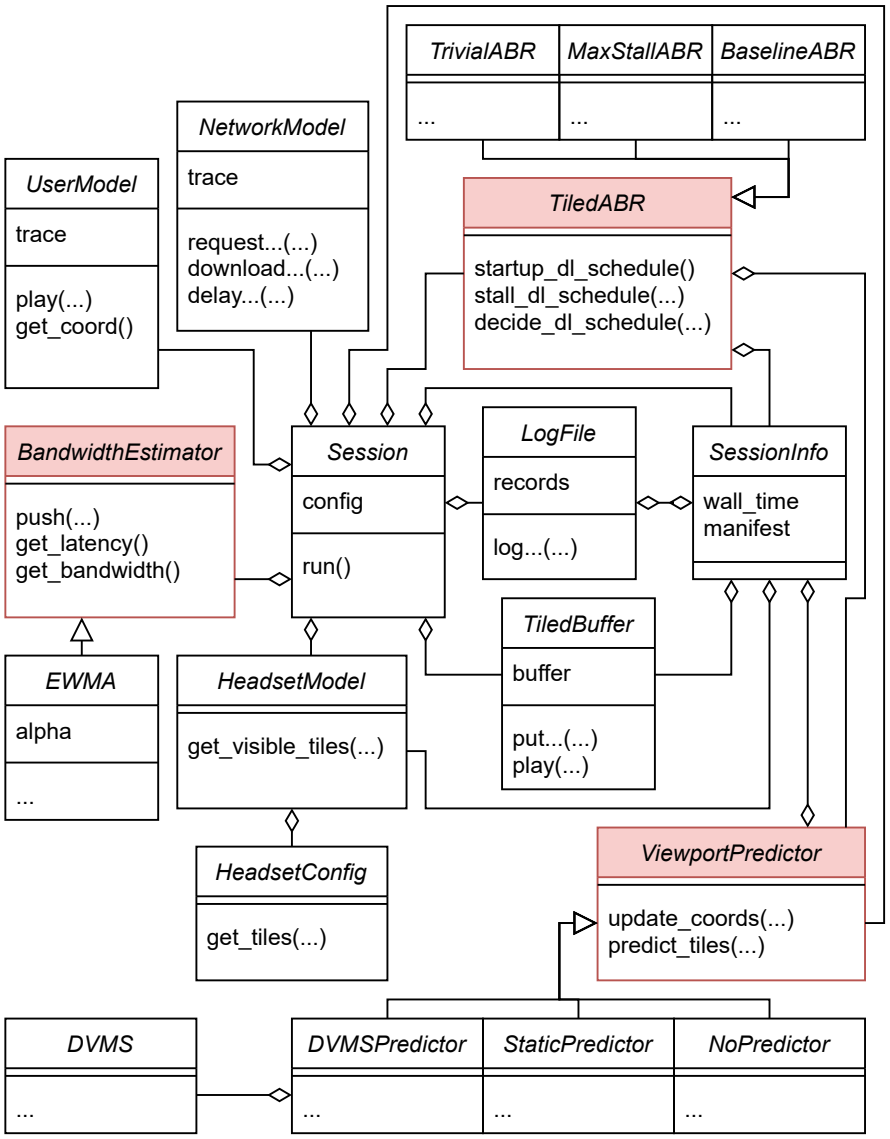


Figure *4.3: UML class diagram of the SMART360 simulator. All aggregation relationships are one-to-one.*

equirectangular projection. The headset configuration is loaded from JSON file located in the `SMART360-simulator/config/` directory.

4.4.1.4 User

The `user.py` file contains the *UserModel* class. This class handles the user head motion trace and is used to get head motion coordinates updates.

4.4.1.5 Network

The `network.py` file contains the *NetworkModel* class. This class handles the network trace and provides methods to download groups of tiles in compliance with the bandwidth and latency information present in the network trace.

4.4.1.6 Bitrate adaptation

The `br_adaptation.py` file contains the *TiledABR* abstract class and its subclasses. This class is responsible for deciding which tiles of which segments will be downloaded in which quality, and in which order. We provide three simple ABR strategies with no buffer replacements. *TrivialABR* tries to download all tiles in the lowest quality and fills the buffer as quickly as possible. *MaxStallABR* is provided for experimental purposes to calculate the maximum possible stall ratio for a user, in the case where we only download tiles once they are missing in the viewport and causing a stall event. We also provide *BaselineABR*, a simple ABR strategy with some rate-based and buffer-based elements.

4.4.1.7 Viewport prediction

The `vp_prediction.py` file contains the *ViewportPredictor* abstract class and its subclasses. This class is used to make predictions about user head movements and the resulting viewports. These predictions can in turn be used by the ABR algorithms. We provide two baseline viewport predictors as well as an implementation of a deep learning predictor. *NoPredictor* gives equal probabilities for all tiles. *StaticPredictor* assumes the user will not move and gives higher probabilities to tiles that were inside the viewport. It is based on the *Trivial-static* baseline described and evaluated in Sec. 3.4.4. *DVMSPredictor* uses the DVMS-based deep learning model described in Sec. 3.4.3 to make predictions.

4.4.1.8 Bandwidth estimation

The `bw_estimation.py` file contains the *BandwidthEstimator* abstract class and its *EWMA* subclass. This class can be used to make estimates of the future bandwidth and latency of the network, useful for ABR planning. The *EWMA* subclass makes latency and bandwidth estimates following an exponentially weighted moving average model, as done in the *dash.js* reference player*, but in a simplified manner.

4.4.1.9 Logging

The `_logging.py` file contains the *LogFile* class. This class provides methods to add simulation information and measurements to a list of records, that is then written to a JSON log file at the end of the simulation. New methods can easily be implemented to include more information and measurements.

*https://github.com/Dash-Industry-Forum/dash.js

4.4.1.10 Log parsing

The `parse_session_logs.py` file, located in the `log_parsing/` directory, consists of a post-processing pipeline that reads the JSON log file and builds data frames stored in *Feather* files. This file format produces very lightweight files that are quick to read and write compared to the raw log files.

4.4.1.11 Notebooks

There are two notebooks in the `notebooks/` directory. The first notebook, `network_traces_scaling.ipynb`, is described in Sec. 4.3.1.1, and the second notebook, `output_metrics.ipynb`, gives examples of possible visualizations of the SMART360 output metrics, as shown in Sec. 4.5.3.

4.4.2 Simulator logic

In this section, we describe the algorithmic flow of the SMART360 simulator. As mentioned in Sec. 4.4.1.1, the *Session::run* method is the entry point to the simulator. We describe the logic behind this method in Algo. 1. For the sake of readability, the algorithms described in Algo. 1 and Algo. 2 are simplified versions of the methods, where only the most relevant steps are shown.

The three ABR functions that appear on lines 3 and 9 of Algo. 1, and line 11 of Algo. 2 refer to the three methods of the *TiledABR* abstract class that have to be implemented by subclasses, as explained in Sec. 4.5.1. These functions return download schedules, noted skd. A download schedule is an ordered list of elements that each contain s, the segment number, t, the tile number, and q, the quality level. In the case of startup and stall (lines 3 of Algo. 1 and line 11 of Algo. 2), the full schedule must be downloaded before the video playback can be resumed. In the case of the regular ABR decision function (line 9 of Algo. 1), elements are downloaded in the same order as given by the schedule during Δ_{DL} seconds.

On line 5 of Algo. 1 and lines 13 and 17 of Algo. 2, "download" implies using the *NetworkModel* with the appropriate latency and bandwidth, as well as putting the downloaded tiles in the buffer.

In Algo. 2, we give some detail behind the logic of one the most complex methods of the simulator, *Session::play_and_download*. This method enables the simulation of video playback and tile download at the same time, while also making sure that the *NetworkModel* and the *UserModel* stay synchronized. This method brings two improvements over Sabre360:

- the ABR algorithm has to plan and make individual requests for downloading groups of tiles every Δ_{DL} seconds, which is more realistic than the very frequent ABR optimizations and requests in Sabre360;
- stall events can happen and stall periods can be measured, which we also consider more realistic than the video not pausing and showing blank tiles in Sabre360. We

Algorithm 1 Simplified `run` method

```
1:  l ← video length
2:  p ← 0                                              ▷ video play head
3:  skd_startup ← ABR_STARTUP                          ▷ startup schedule
4:  for all s, t, q in skd_startup do
5:      download tile t from segment s in quality q
6:  end for
7:  while p < l do
8:      bw_est ← network bandwidth estimation
9:      skd ← ABR_DECIDE(bw_est, Δ_DL)                 ▷ download schedule
10:     PLAY_AND_DOWNLOAD(skd, Δ_DL)                   ▷ see Algo. 2
11: end while
```

have chosen for stall events to happen in SMART360 only if a tile that should be visible to the user is not present in the buffer. This means that the video does not stop if tiles are missing from the buffer but are not in the user's field of view.

4.5 Using SMART360 to compare motion predictors and adaptive bitrate algorithms

In this section, we explain how researchers can use the SMART360 simulation environment to implement new ABR strategies and motion prediction algorithms for 360° video streaming and compare them.

4.5.1 Implementing an ABR strategy within SMART360

To implement a new ABR strategy, one only needs to create a new subclass of *TiledABR* (see Sec. 4.4.1.6) that implements three methods. Each one of these methods returns a download schedule containing elements composed of s, the segment number, t, the tile number, and q, the quality level. In addition to the method parameters, the ABR class can access other information such as the buffer content, the video manifest, or a viewport predictor.

- *startup_dl_schedule()* is called at the beginning of the simulation. It must return a schedule of what to download before the video playback starts;
- *decide_dl_schedule(bw_{est}, Δ_{DL})* is the main ABR decision method. It is called every Δ_{DL} seconds and must return a schedule of what to download in the next Δ_{DL} seconds, given the estimated bandwidth;
- *stall_dl_schedule($\mathcal{T}_{missing}$)* is called whenever a stall event happens. When the video playback is paused during this event, the list of missing tiles in the user's

Algorithm 2 Simplified `play_and_download` method

procedure PLAY_AND_DOWNLOAD(skd, Δ_{DL})
  $\Delta_{left} \leftarrow \Delta_{DL}$
  while $\Delta_{left} > 0$ **do**
    $\tau_{coord} \leftarrow$ time until next user coord. update
    $\tau_{segment} \leftarrow$ time until next video segment
    $\tau \leftarrow \min(\Delta_{left}, \tau_{coord}, \tau_{segment})$
    $\mathcal{T}_{buf} \leftarrow$ set of tiles in buffer for current segment
    $\mathcal{T}_{visible} \leftarrow$ set of visible tiles calculated from coord.
    $\mathcal{T}_{missing} \leftarrow \mathcal{T}_{visible} - \mathcal{T}_{buf} \cap \mathcal{T}_{visible}$
    if $\mathcal{T}_{missing} \neq \emptyset$ **then** ▷ stall event
      $skd_{stall} \leftarrow$ ABR_STALL($\mathcal{T}_{missing}$) ▷ stall schedule
      for all s, t, q in $skd_{startup}$ **do**
        download tile t from segment s in quality q
      end for
    end if
    if $|skd| > 0$ **then**
      download (s, t, q) schedule elements for τ seconds
      remove downloaded elements from skd
    end if
    $p \leftarrow p + \tau$
  end while
end procedure

field of view is passed as a parameter and the method must return a schedule of what to download. The video playback can only resume if all the missing tiles and everything in the stall schedule has been downloaded.

4.5.2 Implementing a motion predictor within SMART360

SMART360 also allows the implementation of head motion prediction algorithms, in the form of a viewport predictor that can in turn be used by the ABR algorithm. To implement a new motion predictor, one only needs to create a new subclass of *ViewportPredictor* (see Sec. 4.4.1.7) that can implement two methods:

- *predict_tiles(*s*)* has to be implemented by the subclass. The parameter s corresponds to the segment number for which we want to make predictions. This method returns a list of length T, where each element corresponds to the score given to each tile. A higher score means a higher probability of being present in the user's viewport during segment s;
- *update_coord(*$coord$*)* can be implemented, but is not mandatory. This method allows updating the motion predictor with new head coordinates that can be used to make predictions.

As of right now, the only information that can be used for predictions is the past head coordinates of the user. However, SMART360 could easily be extended to include video information such as saliency maps for head motion prediction.

4.5.3 SMART360 output metrics

SMART360 brings many QoE-related metrics and visualizations, as well as some network-related metrics. The logs that we provide already enable numerous types of insightful visualizations, as shown in Fig. 4.4, and can easily be extended to include more information and measurements. In this section, we show examples of figures that can be produced with SMART360 to compare ABR and head motion prediction algorithms. The figures presented in this section are extracted from the `output_metrics.ipynb` notebook, and new visualizations can readily be generated from the same data frames without needing to extend the logs. These figures show examples for 34 user head motion traces on one specific hand-made network trace, which alternates between 0 and 4 Mbps, for one video with hand-made quality levels corresponding to 1, 2, 4, 8, and 16 Mbps. The buffer size is set to 10 seconds and B_{min} (see Section 4.6.1) is set to 1 second. The shaded areas represent the 95% confidence interval.

- Fig. 4.4a compares the **average visible quality** when using two different viewport predictors over one video for all users who have watched this video. The average visible quality is computed by calculating the average quality level of the tiles that are inside the user's viewport at each point in time. In this example, each tile

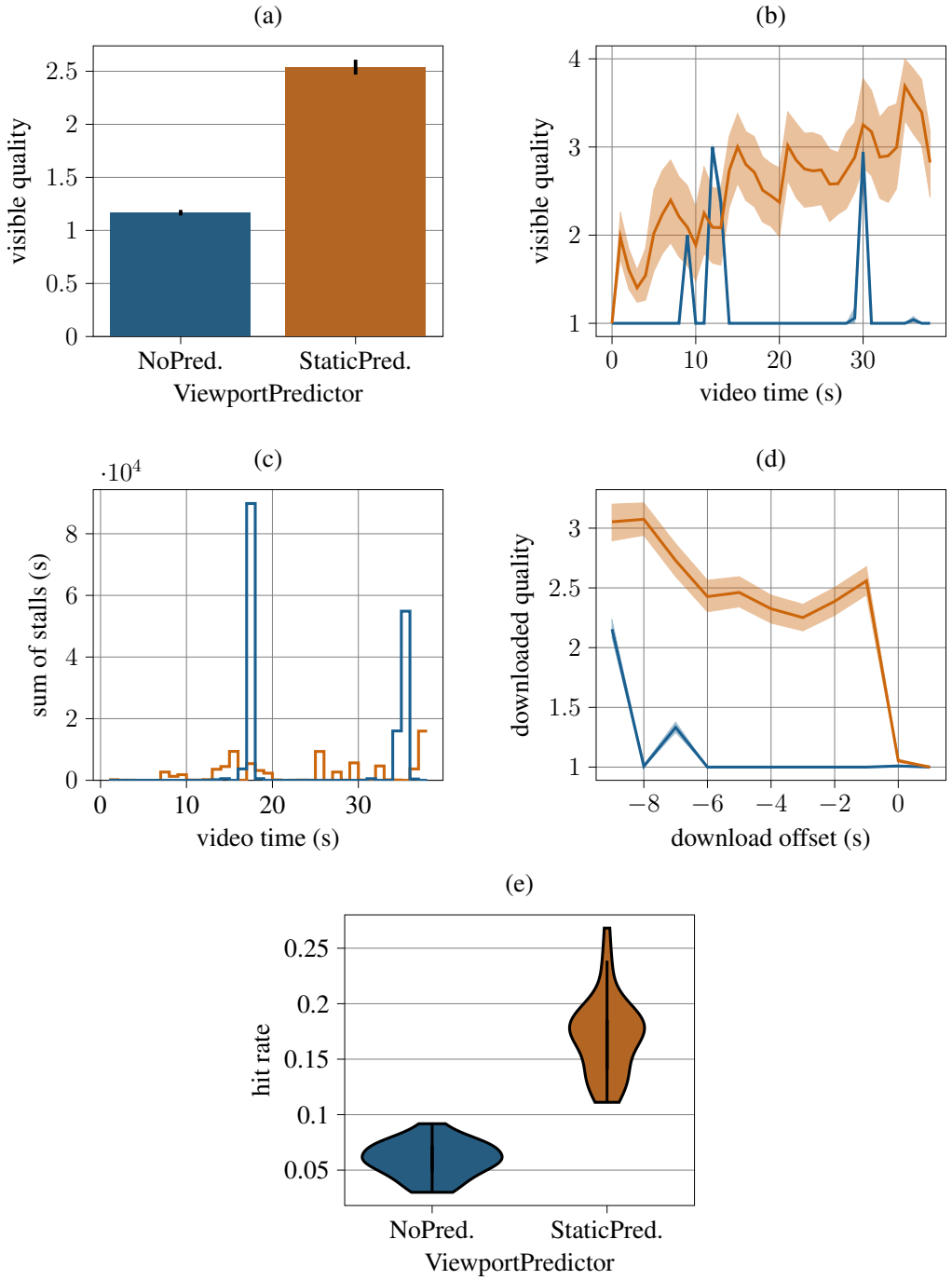


Figure *4.4: Some examples of visualizations from SMART360 simulation output metrics. (a) average visible quality, (b) average visible quality against video timestamp, (c) sum of all user stalls against video timestamp, (d) average downloaded quality against download offset, (e) bandwidth efficiency. Colors have the same meaning across all subfigures.*

can be downloaded five quality levels ranging from 1 to 5, and more details about

the simulation settings can be found in the notebook. Here, we can see that the *StaticPredictor* gives higher average visible quality than *NoPredictor*.

- Fig. 4.4b compares the **average visible quality** when using two different viewport predictors **against** the **video timestamp** for all users who have watched this video. The quality levels and simulation settings are the same as in Fig. 4.4a. Here, we can see with more detail when, in the video, *StaticPredictor* gives higher average visible quality than *NoPredictor*.

- Fig. 4.4c compares the sum of **stall periods** when using two different viewport predictors **against** the **video timestamp** for all users who have watched this video. The simulation settings are the same as in Fig. 4.4a. Here, we can see with precision exactly when, in the video, the stalls are happening, and that *StaticPredictor* gives fewer stall periods than *NoPredictor*.

- Fig. 4.4d compares the **average quality** of downloaded tiles that end up in the user's viewport when using two different viewport predictors **against** the "**download offset**" for all users who have watched this video. The download offset is inversely proportional to the buffer level: a download offset of -6 means that the tile was downloaded 6 seconds before it was played. The quality levels and simulation settings are the same as in Fig. 4.4a. Here, we can understand better how the ABR strategy works and how the prediction impacts its behavior regarding the buffer level, and that *StaticPredictor* gives higher average visible quality than *NoPredictor*, regardless of the buffer level.

- Fig. 4.4e compares the **distribution** of the "**hit rate**" when using two different viewport predictors over one video for all users who have watched this video. This figure shows the estimated density of the distribution on the x-axis for different values of hit rate on the y-axis. The hit rate is calculated by dividing the number of bits that appeared in the user's viewport by the total number of bits that were downloaded, it can be seen as a form of bandwidth efficiency. The simulation settings are the same as in Fig. 4.4a. Here, we can see that *StaticPredictor* is more efficient than *NoPredictor* and wastes less bandwidth.

With Figures 4.4b, 4.4c, and 4.4d, we can get an understanding of the ABR algorithm behavior for each viewport prediction algorithm. As mentioned above, the bandwidth alternates between 0 Mbps for 5 seconds and 4 Mbps for 5 seconds.

Since it must spread the quality equally, *NoPredictor* only has budget to first fill the buffer with tiles of the lowest quality (Fig. 4.4d), which gives a poor average visible quality (Fig. 4.4b). Once the buffer has reached a sufficient level, *NoPredictor* has enough budget to start downloading tiles of higher quality for the whole frame (Fig. 4.4d). This leads to an improved average visual quality (first blue peak on Fig. 4.4b). However, the bandwidth drops to 0 and the buffer depletes, which causes a stall (Fig. 4.4c). Once the bandwidth is back to 4 Mbps, the same cycle repeats (Fig. 4.4b and Fig. 4.4c).

Since it can focus the quality in some areas, *StaticPredictor* starts downloading higher quality tiles much earlier (Fig. 4.4d), which gives an increased average visible quality (Fig. 4.4b). Once the bandwidth drops to 0, the buffer does not deplete as fast, since the bandwidth was not wasted by downloading high quality tiles far away from the viewport, which reduces also stall periods (Fig. 4.4c).

The notebook also includes metrics on the spatial and temporal quality variance of 360° videos, as well fairness metrics based on the QoE fairness index described by Hoßfeld, Skorin-Kapov, Heegaard, and Varela (2017).

4.6 360° video streaming with DVMS

This section presents an extensive evaluation of the interest of our discrete variational multiple sequence learning framework (DVMS), proposed in chapter 3, in a 360° streaming system. We first detail the implementation of DVMS in our simulation environment. We then present the simulation settings and results, identifying the possible gains on different metrics and quantifying them.

4.6.1 DVMS implementation in SMART360

From trajectory and likelihood to tile scores: In the SMART360 simulator, the client uses an ABR algorithm to make requests for new tiles every Δ_{DL} seconds. The ABR makes its quality allocation for incoming segments decisions based on tile scores, given by the viewport prediction algorithm. Before the prediction is made, all the tile scores are initialized to 0. DVMS outputs the predicted head positions for a given segment. In our case, DVMS outputs $5 \cdot K$ points (with K the number of predicted trajectories), because the segments are one-second-long and the head motion trace sampling rate is 5 Hz. For each predicted position (FoV center), we calculate the list of tiles belonging to this FoV. The scores of the tiles belonging to this FoV are updated as follows: $\alpha \mathrel{+}= \frac{\mathcal{L}_k}{5 \cdot K}$, with α being the score of any tile belonging to the FoV calculated from a position of a predicted trajectory of likelihood $\mathcal{L}_k$. Since $\sum_{k \in K} \mathcal{L}_k = 1$, the maximum score for a tile that belong to all the predicted FoVs is 1. The remaining tiles are given a score inversely proportional to the distance to the viewport.

Adaptive bitrate (ABR) algorithm: The objective of our ABR algorithm is to maximize the expected QoE given the predicted viewport, the estimated network bandwidth, and the buffer level. This task is achieved by selecting the right tiles to download in the right quality, such that the quality inside of the user's viewport is as high as possible, without any stall event. With SMART360, the ABR algorithm is called periodically: every Δ_{DL} seconds, the ABR algorithm is used to produce a download schedule that will be sent as a request to the server.

We chose to implement a simple "hybrid" ABR algorithm, considering both the estimated bandwidth and the buffer level (see Sec. 2.1.3.3), named *BaselineABR* for tile-based streaming that can demonstrate the advantages of multiple trajectory prediction.

This algorithm was kept simple for an easier understanding of the streaming behavior with different viewport prediction algorithms. The objective of *BaselineABR* is to maximize the expected viewport quality, while maintaining a minimum buffer level B_{min}. A simplified version of the *BaselineABR* is described in Algo. 3.

Algorithm 3 Simplified *BaselineABR* logic

Input: Available bandwidth budget b, Tile scores $\alpha_{s,t}$, Indices of empty (segments, tiles) in buffer (S, T)
Parameters: Minimum buffer level B_{min}, Quality levels Q_k, $k = 1, ..., 5$, Score threshold $p = 0.2$
Output: Download schedule skd
$(S_{min}, T_{min}) = s_t < B_{min}, s_t \in (S, T)$ ▷ Indices of empty (segments, tiles) inferior to B_{min}
if COST$(S_{min}, T_{min}, Q_1) > b$ **then**
    $skd \leftarrow S_{min}, T_{min}, Q_1$ ▷ Request all under B_{min} anyway
else
    $skd \leftarrow S, T, Q_5$ ▷ Initialize schedule with max quality
    while COST$(skd) > b$ **do**
        $S, T, Q_k \leftarrow skd$
        $Q_k = \begin{cases} Q_k & \text{if } \alpha_{S,T} > p \textbf{ or } (Q_k == Q_1 \textbf{ and } s_t \in (S_{min}, T_{min})) \\ Q_{k-1} & \text{otherwise (remove from schedule if already min quality)} \end{cases}$
        $p = \min(p + 0.2, 1)$
    end while
end if
$skd \leftarrow$ SORT$(skd, \alpha_{s,t})$ ▷ Sort with highest tile scores first

4.6.2 Simulation settings

The results presented in Sec. 4.6.3 summarize metrics from **4,729,000 simulations** using **3,378 head motion traces** of **132 different users** watching **94 different videos**, **40 different network traces** with **5 different buffer settings** and **7 different viewport prediction algorithms**.

4.6.2.1 Videos

The simulations were run on 94 different videos coming from the test sets of three datasets the DVMS model was trained on (see Sec. 3.4.4.2). There are 5 videos from the MMSys18 (David et al., 2018) dataset, 74 videos from the CVPR18 (Y. Xu et al., 2018) dataset, and 15 videos from the PAMI18 (M. Xu, Song, et al., 2019) dataset. The average video duration is around 30 seconds (range 17-64).

The original video files of each dataset were retrieved, split in a **12x6 tile layout** and re-encoded with *libx265*, using the HEVC compression standard. The tiles were each

encoded in **five different quality levels** with different **constant rate factors (CRFs): 16, 22, 28, 34, and 40**, which each quality level being approximately twice the bitrate of the previous one. Finally, the videos were packaged in **1 second segments** for streaming delivery.

4.6.2.2 Head motion traces

Each video was watched by an average of approximately 36 users (can vary for each video, range 28-58), which gives a total 3,378 head motion traces, coming from 132 different unique users. 145 traces (29 different users) come from the MMSys18 dataset, 2363 traces (45 different users) come from the CVPR18 dataset, and 870 traces (58 different users) come from the PAMI18 dataset. Each trace contains the head positions of the user with a 5 Hz sampling rate. These traces were not included in the training of the model, as they come from the test sets of the datasets.

4.6.2.3 Network traces

The simulations were run on 40 different 4G network traces from the 4G/LTE dataset published by van der Hooft et al. (2016). They are made of 40 traces of bandwidth and latency measurements along several routes in the city of Ghent, Belgium. For the comparisons between ABR algorithms to be relevant, the average bandwidths of the network traces were scaled to approximately match the video bit rates, because we need to be in a situation where the algorithm has to adapt to the network constraints to make a difference in quality.

4.6.2.4 Buffer settings

The maximum size of the **buffer** was always set to **10 segments** (i.e., 10 seconds), because it is not possible to show the benefit of viewport prediction with larger values, since we only predict the future head positions 5 seconds ahead.

The simulations were run for 5 different values of B_{min}, the ABR buffer constraint (see Sec. 4.6.1): 1, 2, 3, 4, and 5 seconds. With this parameter, we can tune the behavior of the ABR algorithm and show results for different levels of aggressiveness.

4.6.2.5 Prediction algorithms

7 different prediction algorithms were tested in the simulations:

- *NoPred*: no assumption is made about the viewport location and the same score is given to all tiles before ABR allocation. This serves as a baseline for comparison.
- *StaticPred*: we assume that the person will stay still and that the viewport will not change in the near future. Tiles present in the viewport are given a score of 1.0, and remaining tiles are given a score inversely proportional to the distance to the viewport. This serves as a baseline for comparison.

- DVMS, $K = 1$ (DVMS-**1**): DVMS is used to predict one trajectory of the future head positions. Since there is only one trajectory, the likelihood of this trajectory is set to 1.0 and the tile scores are computed as described in Sec. 4.6.1.
- DVMS, $K > 1$ (2 to 5): DVMS is used to predict K possible trajectories of the future head positions. The respective likelihoods are based on the past error as described in Sec. 3.5.2. The tile scores are computed as described in Sec. 4.6.1.

For DVMS-based prediction algorithms, we choose to reuse the DVMS-**K** notation introduced in Sec. 3.4.4.1 to improve readability.

4.6.2.6 Metrics

We report results on two metrics in the following section:
Viewport quality: For each video segment, a person sees multiple tiles that were downloaded at a certain quality level. The quality level of a tile is approximately logarithmically proportional to its bitrate, because of the choice of CRFs that was made in Sec. 4.6.2. The viewport quality metric is computed as the weighted average of the quality of the tiles that were seen during a segment. The weights are proportional to the duration for which this tile was visible during the segment.
Normalized QoE: After reviewing several references including QoE functions combining various components, we decided to make our own in order to avoid subjective hyper-parameter choices required by existing formula. We defined our own QoE function, the normalized QoE (Eq. 4.1) combines the viewport quality, the stall periods, the spatial quality variance, and the temporal quality variance in one metric, with a value between 0 and 1. T is the duration of the video and S is the stall duration over the simulation. $\overline{VQ}$ is the average viewport quality over the video, VQ_{max} is the maximum possible viewport quality for a frame. $\overline{SQV}$ is the average spatial quality variance (standard deviation of quality levels of the tiles in a viewport) over the video, SQV_{max} is the maximum possible spatial quality variance for a frame. TQV is the temporal quality variance (mean of absolute differences between average viewport quality of segments) over the video, TQV_{max} is the maximum possible temporal quality variance over a video.

$$QoE = \frac{T \cdot \overline{VQ}}{VQ_{max} \cdot (T + S)} \cdot \left(1 - \frac{\overline{SQV}}{2 \cdot SQV_{max}}\right) \cdot \left(1 - \frac{TQV}{2 \cdot TQV_{max}}\right) \tag{4.1}$$

This normalized QoE includes the four main usual components of 360° video streaming: the viewport quality, the stall periods, the spatial quality variance, and the temporal quality variance, without the need for additional factors or hyper-parameters for each component. A QoE of 1 indicates the highest possible viewport quality without any stall period and with no spatial or temporal quality variance. Stall periods only appear in the denominator of the first term and are consequently considered as time spent at a quality level of 0. A high spatial or temporal quality variance also reduces the QoE by multiplying it by a factor between 0.5 (highest variance) and 1 (lowest variance). All these choices

were made to avoid negative and unbounded QoE values, and allow to keep the QoE value between 0 and 1. This makes comparisons and fairness computation easier (when using the QoE fairness metric described by Hoßfeld et al. (2017)).

4.6.3 Results

In this section, we compare the results for simulations with the viewport prediction algorithms presented in Sec. 4.6.2.5. For DVMS, only $K = 1$, $K = 5$, and $K = best$ are shown for better readability. The choice of number of trajectories to predict (K) is related to prediction uncertainty, where factors such as video features such as spatial and temporal information as well as user emotions can have a strong impact, as seen in Sec. 5.6. The "*best* K" shows the potential gains if we were able to use this information to choose the best K (ranging from 1 to 5) for each (user, video) pair, but even more gains could be found by dynamically adapting K during video playback, using head speed or past prediction error.

4.6.3.1 Viewport quality and QoE gains of DVMS

We present results showing viewport quality and QoE gains in Fig. 4.5 and Fig. 4.6, where the simulations were run with $B_{min} = 1$, which is where differences between viewport prediction algorithms are the most visible. The order of performance between predictors is nearly identical for all values of B_{min}. As B_{min} increases, the advantage of viewport prediction slowly decreases. We provide detailed results in Tables 4.1 and 4.2 for completeness.

Fig. 4.5 shows the average viewport quality across all segments during the simulations. The average viewport quality is not necessarily an integer, each bin contains the number of segments with an average viewport quality lower or equal to its tick label, but greater than the preceding bins. *NoPred* (i.e., uniform spread of the quality budget) gives the most segments with the lowest quality and the fewest segments with the highest quality. We can see that *StaticPred* already gives significant quality improvements. This figure also illustrates a key difference between single and multiple trajectory prediction: DVMS-**5** gives fewer segments with a very low viewport quality, but also fewer segments with a very high viewport quality than DVMS-**1**. While DVMS-**1** leads to more segments with very low quality than DVMS-**5**, it is still fewer very low quality segments than *StaticPred*. While DVMS-**5** leads to fewer segments with very high quality than DVMS-**1**, it is still more very high quality segments than *StaticPred*. DVMS-**best** gives the best of both worlds with less segments with very low quality and more segments with very high quality.

Fig. 4.6-left shows the viewport quality gain over *NoPred* for all the played segments of the simulations. We can see that DVMS-**5** has an average viewport quality gain (**48.0%**) slightly better than the gain of DVMS-**1** (46.7%). The median gain is better: 50% of segments have a viewport quality improvement over 20.4% with DVMS-**5**, while 50% of segments have a viewport quality improvement over 14.3% with DVMS-**1**.

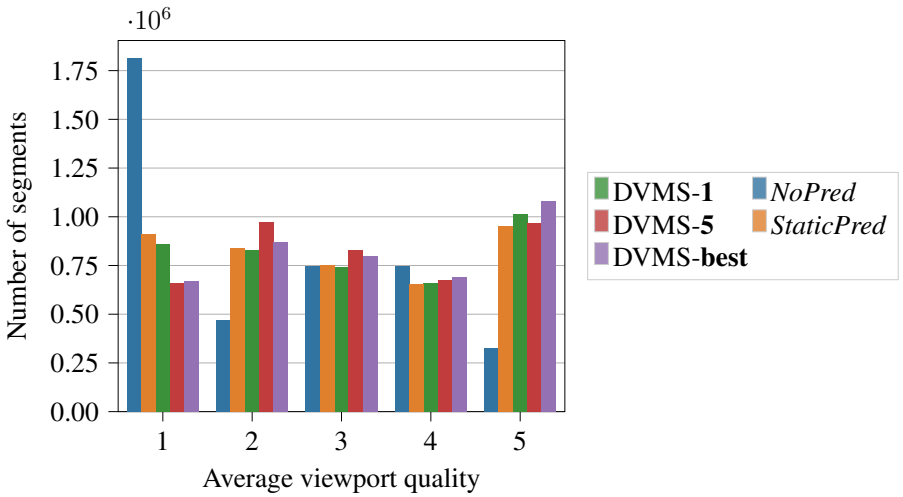


Figure *4.5: Segment quality distribution comparing different viewport prediction algorithms over all simulations with* $B_{min} = 1s$.

Improvements are also more evenly distributed with DVMS-**5**, with **61.2%** of segments having an increased viewport quality (20.0% decreased, 18.8% unchanged), while 57.2% of segments had an increased viewport quality (20.4% decreased, 22.4% unchanged) with DVMS-**1**.

Fig. 4.6-right shows the QoE gain over *NoPred* for all simulations. We can see that DVMS-**5** has an average QoE gain (15.6%) slightly worse than the gain of DVMS-**1** (16.2%), but the median gain is slightly better: 50% of segments have a QoE improvement over 10.0% with DVMS-**5**, while 50% of segments have a QoE improvement over 9.8% with DVMS-**1**. Improvements are once again more evenly distributed with DVMS-**5**, with **71.9%** of simulations having an increased QoE, while **70.0%** of simulations had an increased QoE with DVMS-**1**.

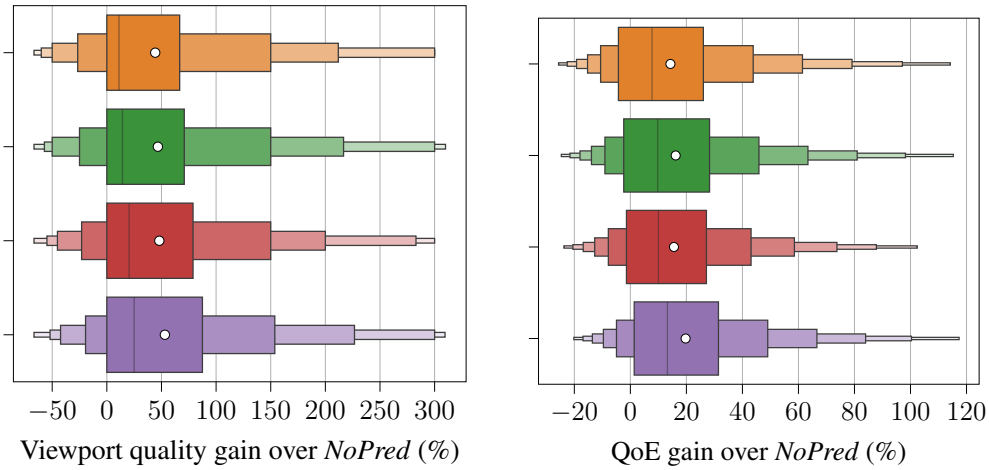


Figure *4.6: Viewport quality and QoE gains over all simulations with* $B_{min} = 1s$. *Colors are the same as in Fig. 4.5.*

*Table 4.1: Visual quality gains (in %) over NoPred for all segments during simulations for different values of B_{min}. We report average and median gains in the "Avg. / Med." columns. We report the proportion of segments (in %) for which there was an increase / decrease in viewport quality over NoPred in the "Inc. / Dec." columns (some segments keep the same quality). Best results are highlighted in **bold**, second best are underlined.*

	$B_{min} = 1$		$B_{min} = 2$		$B_{min} = 3$		$B_{min} = 4$		$B_{min} = 5$	
	Avg. / Med.	Inc. / Dec.	Avg. / Med.	Inc. / Dec.	Avg. / Med.	Inc. / Dec.	Avg. / Med.	Inc. / Dec.	Avg. / Med.	Inc. / Dec.
StaticPred	44.3 / 11.1	54.7 / 22.2	39.8 / 4.8	51.2 / 22.1	36.1 / 0.0	48.7 / 22.1	32.6 / 0.0	45.3 / 21.8	29.1 / **0.0**	42.2 / 21.5
DVMS-**1**	46.7 / 14.3	57.2 / 20.4	41.4 / 9.6	53.4 / 20.3	37.3 / 1.9	50.6 / 20.2	34.0 / 0.0	47.3 / 20.0	31.0 / **0.0**	44.5 / 19.6
DVMS-**5**	48.0 / 20.4	61.2 / 20.0	42.3 / 14.3	58.0 / 19.8	36.9 / 9.7	54.1 / 19.6	33.6 / 2.2	50.8 / 19.3	30.4 / **0.0**	47.5 / 18.9
DVMS-**best**	**53.1 / 25.0**	**62.8 / 18.4**	**47.7 / 17.1**	**59.6 / 18.2**	**43.2 / 12.2**	**56.1 / 18.1**	**39.8 / 6.6**	**52.6 / 17.8**	**36.2 / 0.0**	**49.3 / 17.5**

*Table 4.2: QoE gains (in %) over NoPred for all simulations for different values of B_{min}. We report average and median gains in the "Avg. / Med." columns. We report the proportion of simulations (in %) for which there was an increase in QoE over NoPred in the "Inc." columns (no Dec. column because it can be inferred from Inc., as no simulations keep the same QoE). Best results are highlighted in **bold**, second best are underlined.*

	$B_{min} = 1$		$B_{min} = 2$		$B_{min} = 3$		$B_{min} = 4$		$B_{min} = 5$	
	Avg. / Med.	Inc.	Avg. / Med.	Inc.	Avg. / Med.	Inc.	Avg. / Med.	Inc.	Avg. / Med.	Inc.
StaticPred	14.3 / 7.8	65.8	13.2 / 7.1	65.0	12.2 / 6.1	63.3	11.0 / 5.3	62.2	10.0 / 4.5	61.0
DVMS-**1**	16.2 / 9.8	70.0	14.8 / 8.9	68.9	13.8 / 7.9	67.5	12.9 / 7.2	66.5	12.0 / 6.7	65.6
DVMS-**5**	15.6 / 10.0	71.9	14.2 / 9.2	71.1	12.8 / 8.1	69.3	12.1 / 7.6	68.8	11.3 / 7.1	67.9
DVMS-**best**	**19.7 / 13.2**	**77.9**	**18.4 / 12.4**	**77.3**	**17.4 / 11.4**	**76.2**	**16.6 / 10.9**	**75.7**	**15.5 / 10.2**	**74.7**

Over all B_{min}, DVMS-**5** gives similar improvements on average than DVMS-**1** but leads to better fairness between users than DVMS-**1**: there are more cases of segments where the quality is improved, slightly fewer cases with very high quality, considerably fewer cases with very low quality. This could be the effect of the aggressiveness of DVMS-**1**. Single trajectory is a double-edged sword as it completely focuses the quality where the single prediction is: if the prediction is accurate prediction, we can have very high quality segments, but if the prediction is inaccurate, we will have very low quality segments. Predicting more trajectories would be a more conservative approach, as spreading the quality more prevents from catastrophic failure, albeit at the cost of less very high quality segments.

4.6.3.2 Link with prediction error

To confirm that this different distribution of quality gains between DVMS-**1** and DVMS-**5** is indeed due to the prediction error, we first look at the link between normalized QoE and prediction error in Fig. 4.7. The x-axis of this figure is the prediction centile: the average prediction error for all (users, videos) was sorted and equally distributed in 100 sorted bins, so that each bin has the same number of (users, videos). Unsurprisingly, there is a clear decreasing trend across all K: the QoE decreases when the prediction error increases.

We now show the average viewport quality (resp. QoE) of segments (resp. simulations) against DVMS prediction error deciles in Fig. 4.8-left (resp. 4.8-right). The process for deciles is the same as for centiles, but there are only 10 bins instead of 100. We can

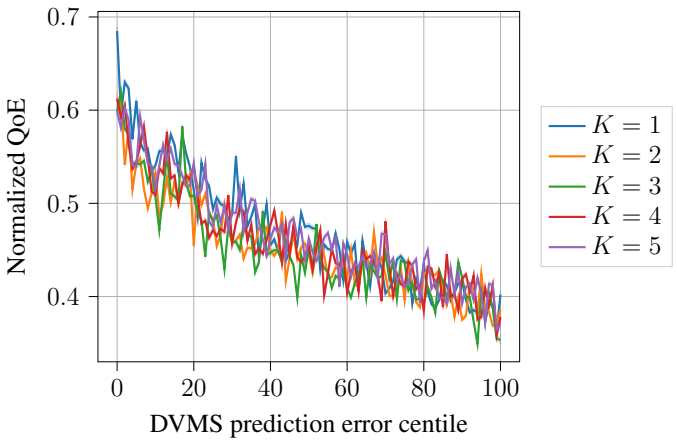


Figure *4.7: Average normalized QoE against DVMS prediction error centiles for different values of K.*

see that predicting one trajectory is better than predicting 5 trajectories when head motion is predictable and that the single trajectory is very accurate. However, when head motion becomes less predictable and prediction error increases, predicting 5 trajectories leads to higher viewport quality and QoE. As previously shown in Fig. 4.6, predicting multiple trajectories increases fairness between users, as individual bad predictions have a smaller negative effect on viewport quality and QoE than in the case of single trajectory prediction. Finally, we can see that the potential gain in visual quality (resp. QoE) when using DVMS-**best** over DVMS-**1** is around 10% (resp. 5%) for roughly 30% of the data, when the prediction error is at its highest (deciles 7-10).

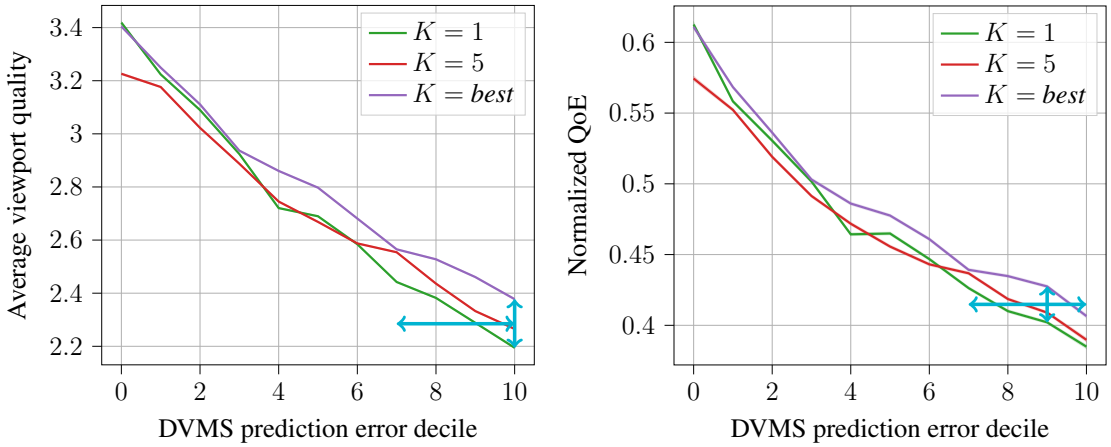


Figure *4.8: Average viewport quality and QoE against DVMS prediction error deciles. Horizontal blue arrow: 30% of the users, vertical arrow: 10% gain in visual quality, 5% gain in QoE.*

4.7 Discussion

With these simulations, we have shown the type of gains that DVMS can bring, particularly to reduce the number of (video,user) traces with lowest visual quality and QoE, while maintaining the number of traces with high-quality levels. Beyond enabling such a fairness increase, maximizing the QoE requires to dynamically adapt the number K of predicted trajectories to both the type of scene and the current state of the user, that is to the (video,user) pair, but also if possible over time, considering changing types of scene and user attentional states. The more predictable the user motion (i.e., in sync with the content and with an attention-driving content with a low number of points of interest), the less the need for a high K. We propose to explore these ideas in chapter 6, motivated by this work and based on the results that we obtain in chapter 5.

We believe that SMART360 successfully addresses the shortcomings of the existing solutions to realistically simulate 360° streaming systems and efficiently compare ABR and head motion prediction algorithms. With SMART360, we yearn to encourage reproducible research by providing transparent code that can be adapted and improved. Possible improvements include but are not limited to: considering the percentage of each tile actually in the viewport for more accurate measurements of the visible quality, instead of counting them as inside the viewport regardless of the proportion of the tile actually in the viewport; giving the ability to the viewport predictor to use more information than the past head coordinates, such as video saliency maps; using multiple threads and communication between threads when events occur during the simulation instead of the monolithic structure of the *Session::play_and_download* method.

4.8 Conclusion

In this chapter, we have presented SMART360, a new simulation environment for 360° video streaming that allows comparing different motion prediction and adaptive bitrate strategies with numerous metrics and graphical visualizations.

SMART360 overcomes the drawbacks of the few existing alternative tools by providing highly-configurable code, with many inputs and settings, as well as offering a realistic streaming behavior, with stall events and ABR planning.

We have described the inputs and outputs of the simulator, as well as its internal structure. We have explained how new motion predictors and adaptive bitrate algorithms can be implemented inside the simulation environment to be evaluated and compared.

Thanks to SMART360, we are able to deploy an extensive system evaluation of our proposed DVMS framework, considering four different datasets of user, video and network bandwidth traces. We show that predicting multiple trajectories yields a higher fairness between the traces, the gains for 20% to 30% of the users reaching up to 10% in visual quality for the best number K of trajectories to generate for a given trace. Finding the ideal K should consider video characteristics in connection with user emotional and attentional states, and can be made even more effective by adapting K over time consid-

ering the evolution of these variables. We explore these ideas by proposing new ways to consider emotions in conjunction with video content in chapter 6.

We believe that SMART360 can improve the reproducibility of research regarding 360° video motion prediction and adaptive streaming algorithms, and make future comparisons of new strategies easier for researchers.

CHAPTER 5

Investigating the link between immersive content, attention, emotion, and movements in virtual reality

From a user perspective, immersive content can elicit more intense emotions than flat-screen presentations. From a system perspective, efficient storage and distribution remain challenging, and must consider user attention. Understanding the connection between user attention, user emotions and immersive content is therefore key. While 360° videos viewed in a VR headset are gaining popularity, it is necessary to reduce the bandwidth required to stream these immersive videos and achieve a satisfactory quality of experience. This requires predicting the user's head motion in advance, which has been addressed by a number of recent prediction methods that consider the video content and the user's previous motion, as shown in the previous chapters. However, human motion is a complex process that can depend on many more parameters, including the type of attentional state the user is in and their emotions, which can be difficult to capture.
In this chapter, we present three contributions stemming from user experiments aiming at better understanding the link between immersive content, attention, emotion, and movements in virtual reality.
First, we present PEM360, a new dataset of user head movements and gaze recordings in 360° videos, along with self-reported emotional ratings of valence and arousal, and continuous physiological measurement of electrodermal activity and heart rate. The stimuli are selected to enable the spatiotemporal analysis of the connection between content, user motion and emotion. We describe and

provide a set of software tools to process the various data modalities, and introduce a joint instantaneous visualization of user attention and emotion we name Emotional maps. We exemplify new types of analyses the PEM360 dataset enables. The entire data and code are made available in a reproducible framework. Second, using this new dataset, we make a first step towards understanding the connection between user emotion and visual attention, in the form of content-based saliency maps. To the best of our knowledge, this is the first work to investigate the tri-partite connection between user attention, user emotion and visual content in immersive environments. To do so, we use PEM360 to analyze how different types of saliency, both low-level and high-level, are related with the user's state in 360° videos. Specifically, we study how the accuracy of saliency estimators in predicting user attention depends on user-reported and physiologically-sensed emotional perceptions. Our results show that high-level saliency better predicts user attention for higher levels of arousal. We discuss how this work serves as a first step to understand and predict user attention and intents in immersive interactive environments.
Finally, using PEM360 as well as CEAP-360VR, another dataset, we investigate the effects of user emotions on the predictability of head motion, in connection with video-centric parameters. We formulate and verify hypotheses, and construct a structural equation model of emotion, motion and predictability. We show that the prediction error is higher for higher valence ratings, and that this relationship is mediated by head speed. We also show that the prediction error is lower for higher arousal, but that spatial information moderates the effect of arousal on predictability. This work opens the path to better capture important factors in human motion, to help improve the training process of head motion predictors, which is investigated in chapter 6.

Contents

5.1 Introduction

As explored in Sec. 2.3.2, visual attention and content-based saliency estimation in virtual reality (VR) is of crucial for several reasons. First, saliency maps are a useful tool to predict user behavior in VR and constitute an important part of viewport-adaptive streaming systems (C.-L. Fan et al., 2017; S. Park, Hoai, et al., 2021; Romero Rondón et al., 2021). Second, saliency maps are used to automate quality assessment with objective quality estimators (M. Xu, Li, Chen, Wang, & Guan, 2019), which are key to enable efficient storage and distribution of content with high perceptual quality (Brunnström et al., 2013).

Studies have shown that emotional stimuli attracts attention (Schupp et al., 2007; S. Fan et al., 2018), but recent work indicates that low-level saliency features may better explain visual attention than emotional stimuli (Hedger, Garner, & Adams, 2019). While saliency estimation in 360° images and videos is well-studied with many proposed approaches (De Abreu et al., 2017; Mazumdar et al., 2021; Ozcinar & Smolic, 2018; Chao, Ozcinar, Zhang, et al., 2020), little research has investigated the link between content-based saliency and emotions in VR. For these reasons, we state the need to better understand the attentional and emotional processes of users in an immersive environment, and how these processes relate to the content.

As seen in Sec. 2.2.1, viewport prediction is also well-studied, and many different prediction approaches have been investigated. However, existing viewport prediction methods do not consider how emotions may affect the predictability in virtual environments. As the effects of emotions may be interlinked with features of the video content, we aim to propose an approach considering both user-centric parameters (emotion and motion) and video-centric parameters.

In this chapter, we propose to investigate the following research questions:

- *How do emotions impact the accuracy of saliency estimators?*
- *What are user-centric parameters (emotion and motion) and video-centric parameters impacting the head motion predictability in immersive 360° videos, and what are the relationships?*

Answering the first question will allow us to better understand the connection between user attention, user emotions and immersive visual content. Answering the second question is important to understand how well the human motion can be captured, and how we can improve prediction approaches, by explicitly modeling the impact these parameters can have when designing prediction models, or by changing the architectures or training losses of the deep models to learn directly from these parameters. We propose an exploration into new ways to consider the emotional data in viewport prediction models in chapter 6.

To answer these questions, we make the following contributions:

- A new dataset of user head movements and gaze recordings in 360° videos, named PEM360, along with self-reported emotional ratings of valence and arousal, and

continuous physiological measurement of electrodermal activity and heart rate. The stimuli are selected based on high-level and low-level content saliency to enable the spatiotemporal analysis of the connection between content, user motion and emotion. The dataset comes with a set of software tools to pre-process the data of gaze, electrodermal activity and content, and to visualize jointly instantaneous heat maps of gaze and arousal level superimposed on the frame, which we name *Emotional maps*. The entire collection of artifacts is presented as Python tools and notebooks to enable reproducibility of the data processing. The dataset and tools are now available in a public GitLab repository*.

- An investigation into how emotions affect the accuracy of saliency estimators, both with low-level and high-level saliency, in 360° videos. Our results show that high-level saliency better predicts user attention for higher levels of arousal.

- An investigation into the connection between a subset of user-centric and video-centric measures and head motion predictability. We consider two datasets (including PEM360) where the user movements and subjective emotions are available. The considered measures are valence and arousal graded by every user on every video, head motion speed, spatial information (SI) and temporal information (TI), shown to provide important insights into this type of emotion-video feature-predictability relationships. We formulate three hypotheses that we verify, and model the data with a directed graph of causal relationships formalized in a structural equation model (SEM). We show that the prediction error is generally lower (higher predictability) for users having provided higher arousal ratings. We also show that the prediction error is higher for higher valence ratings, and that this relationship is mediated by head speed. Finally, we exhibit an interaction effect between SI and arousal, SI being shown to moderate the effect of arousal on the prediction error.

The work presented in this chapter was the object of two conference and one workshop papers. Our dataset and software tools were presented at the *Open Dataset and Software* track of the 13th ACM Multimedia Systems Conference (MMSys '22) (Guimard, Robert, et al., 2022b). The analysis highlighting the link between emotion, attention and content in virtual immersive environments was presented at the 2022 IEEE International Conference on Image Processing (ICIP) (Guimard, Robert, et al., 2022a). The investigation into the effects of head motion on predictability in VR was presented at the 14th International Workshop on Immersive Mixed and Virtual Environment Systems (MMVE '22) (Guimard & Sassatelli, 2022).

Section 5.2 positions our approach with respect to existing works. Section 5.3 outlines the design of our user study. Section 5.4 presents the dataset and software tools that we make available to the research community. Section 5.5.1 provides analyses of our data, including an investigation into the link between attention, emotion and content. Section 5.6 studies the effects of emotions on head motion predictability, combining our collected data with another recent dataset. Finally, section 5.7 concludes the chapter.

*https://gitlab.com/PEM360/PEM360/

5.2 Related work

Sensing and analyzing emotions in immersive environments has spurred interest with several studies looking into the state of presence as well as emotional states in virtual reality (Baños et al., 2008; Felnhofer et al., 2015; Pallavicini et al., 2019; Voigt-Antons et al., 2020). More recently, studies and public datasets considered behavioral data (head and eye movements) in addition to emotional ratings (B. J. Li et al., 2017; W. Tang et al., 2020; Toet et al., 2020; Xue, Ali, Zhang, et al., 2021; Xue, Ali, Ding, & Cesar, 2021).

5.2.1 Sensing emotions in VR environments

Human emotions are commonly decomposed along two main dimensions: valence, representing the negative or positive nature of an emotion (unpleasant-pleasant), and arousal, representing the intensity of the perceived emotion (calm-excited) (Russell, 1980; Barrett, 1998).

The first reference database (B. J. Li et al., 2017) providing emotional ratings and motion recordings of 360° videos is made of 73 VR videos on which 95 users rated valence and arousal using the self-assessment manikin (SAM) tool (Bradley & Lang, 1994) after experiencing each video. Their head positions were continuously recorded. A dataset of self-reported emotions of 19 users watching thirty-six 360° images is collected by W. Tang et al. (2020), with eye motion recorded. However, ratings made in retrospect cannot represent the variety of states a user goes through during the experience (Voigt-Antons et al., 2020), limiting potential analyses and interpretations. Recent works have therefore proposed tools enabling a continuous collection of self-reports inside the immersive environment (Toet et al., 2020; Xue, Ali, Zhang, et al., 2021). The data collected in these recent works also comprise physiological measurements of heart rate and electrodermal activity (EDA, as skin conductance), which has been shown to reliably represent user instantaneous arousal (Boucsein, 2012). Toet et al. (2020) presented a new emotions rating tool, named EmojiGrid, tested on 40 users viewing 62 videos from the reference database of B. J. Li et al. (2017). While they provide the per-user per-video valence and arousal ratings, only time averages are made available for EDA, and no gaze or head motion traces. Xue, Ali, Zhang, et al. (2021) introduced a continuous grading tool of valence and arousal. They provide a dataset of 11 immersive videos from the same database (B. J. Li et al., 2017) experienced by 32 users. Subjective emotional ratings, physiological measurements (including EDA) and head and gaze movements are continuously collected and made available. This latter work (Xue, Ali, Zhang, et al., 2021) is closer to ours and has been made partly concurrently. Our dataset is however complementary and enables other types of studies. We provide EDA streams at a higher rate, acquired at 16Hz, compared to 4Hz in the work of Xue, Ali, Zhang, et al. (2021). In the aforementioned objective of understanding the connection between attended regions and instantaneous emotions, like arousal, it is important to enable the detection of several peaks of the phasic component of EDA per second, requiring hence a higher acquisition rate. Also, we sample seven video stimuli from the same reference database (B. J. Li et al., 2017) for our experiments, so

that specific criteria on saliency are met, as detailed in Sec. 5.3.1. Out of the seven videos, six differ from the videos selected by Xue, Ali, Zhang, et al. (2021). Our dataset therefore enriches the existing datasets and enables extensive analysis to gain new insights on the connection between attention, emotion and content.

5.2.2 Correlating user emotion with motion

Understanding how different types and levels of emotions correspond to specific types of motion has already been investigated (B. J. Li et al., 2017; W. Tang et al., 2020; Xue, Ali, Ding, & Cesar, 2021). Results from B. J. Li et al. (2017) show some level of correlation between (time) average arousal and average pitch angle, and between yaw angle standard deviation and valence. Correlations analyzed from continuous ratings by Xue, Ali, Ding, and Cesar (2021) also show moderate correlation between pitch and arousal on segments of 5 to 10 seconds, pitch and valence, but negative correlation between yaw standard deviation and valence, while results from W. Tang et al. (2020) show a significant impact of negative images on eye behavior.

5.2.3 Correlation of user attention with the spatial content

While previously mentioned works have focused on the analysis of user emotion and motion based on coarse-grained categorization of the entire content (high/low positive/negative valence and high/low arousal), other works have focused on the impact of specific regions on the user's attention, described with low-level (LL) saliency or emotional aspects. LL saliency refers to pixel-level features (e.g., edges, luminance, motion). Cerf et al. (2007) showed that human eye movements are influenced both by LL and high-level (HL) saliency (related to higher semantic concepts such as objects and faces), possibly based on the emotional content. Chaabouni and Precioso (2019) showed that user interest estimators fed with gaze recordings are shown to be weak because highly dependent on LL saliency, which is independent of the user interest. They found that normalizing fixations density with LL saliency significantly improves the interest estimators based on gaze data. The authors took care of selecting emotion-neutral images in their experiments not to interfere with the interest-guided task. Hedger et al. (2019) re-examined previous results suggesting that emotional faces in an image attract more user attention/fixations outside awareness. They showed that facial expressions had no effect on attentional allocation, which can instead be explained by the higher LL saliency.

5.2.4 Prediction of head movements in VR

The prediction of head movements in VR, or viewport prediction has been heavily investigated in recent years, as seen in Sec. 2.2.1. In this chapter, we consider the prediction methods presented by Romero Rondón et al. (2021) to benchmark our approach. Considering the analysis of how the video content or user emotions impact the head motion predictability, to the best of our knowledge only Romero Rondón et al. (2021) analyzed

the prediction performance disaggregated over video categories. However, to the best of our knowledge, no previous research has looked at motion predictability based on felt emotions, nor did they formalize the relationship between predictability and video features.

5.2.5 Positioning of our contributions

In this chapter, we first present a first step towards understanding the connection between user emotion and predictability of motion from content saliency. Specifically, we analyze how the accuracy of LL and HL saliency estimators depend on the user's self-reported and physiologically-sensed emotional perceptions.

Secondly, no work has so far investigated and formalized the effect of emotions and video features on head motion predictability, that is on the performance of prediction methods. To do so, we consider the state-of-the-art predictors introduced by Romero Rondón et al. (2021), as motivated below, and formulate working hypotheses from initial results obtained in the works we previously mentioned (B. J. Li et al., 2017; Xue, Ali, Ding, & Cesar, 2021).

5.3 User study design

We conducted a controlled, indoor laboratory experiment where users watched 360° videos in a VR headset. We collected eye movement (EM), head movement (HM), heart rate (HR) and skin conductance (EDA) data as well as emotion annotations of valence and arousal. The user experiment has been approved by the university ethics committee.

5.3.1 Stimuli

The videos are selected to enable several levels of content analysis and description, to correlate with user motion and emotion. User attention in relation with the visual content is described with saliency maps, obtained either from gaze locations, or estimated from the content. Here we consider two levels of content description as two types of saliency maps, and select the videos so that for each, the overlap between both saliency maps is limited. Specifically, we consider low-level (LL) and high-level (HL) saliency. Low-level saliency maps are made up of a combination of colors, intensity and orientations as defined by Itti et al. (1998). Since we are dealing with videos and not images, we combine this definition with the one of optical flow (Horn & Schunck, 1981), because we also consider motion in the video to be part of the low-level saliency. High-level saliency maps are composed of high-level semantic features, such as faces, cars, or animals. Inspiring from Chopra et al. (2021), high-level saliency is obtained from YOLOv4 object detector, with object bounding boxes being used as binary saliency maps.

We selected 7 videos from the reference database of B. J. Li et al. (2017). The selected videos should have a range of valence and arousal as wide as possible, and the LL saliency

should be evenly distributed both within and outside object bounding boxes characterizing HL saliency. To select these videos we compare (i) the number of pixels inside and outside objects, and (ii) the per-pixel LL saliency (ranging between 0 and 255), computed as the total LL saliency inside and outside objects normalized with the corresponding number of pixels. Fig. 5.1 demonstrates this in videos 13 and 73. The number of pixels with such minimum LL saliency inside and outside objects is equivalent over time, as is the per-pixel LL saliency in both areas. Fig. 5.2 shows a frame where regions with high LL saliency can be seen outside of the detected objects. Table 5.1 lists the video details.

Table 5.1: Details of selected videos for our dataset. Videos YouTubeIDs are clickable links, otherwise accessible at youtube.com/watch?v=[YouTubeID]. Ratings of valence and arousal are between 1 and 9.

ID	Valence	Arousal	Start (s)	End (s)	Duration (s)	YouTubeID
12	7	4.6	5	103	98	T-aOVE22lEw
13	4.92	4.08	4	131	127	GJGfxfGEa9Y
17	5.22	5	5	69	64	g7btxyIbQQ0
23	7.2	3.2	8	143	135	CDfsFuDuHds
27	6	1.6	60	180	120	QxxXu_B–ZA
73	6.27	6.18	9	70	61	bUiP-iGN6oI
32	6.57	1.57	40	130	90	-bIrUYM-GjU

5.3.2 Equipment

Recordings of head and eye movements have been made with a FOVE headset, equipped with an eye-tracker with a 120Hz acquisition rate, and tethered to a desktop computer. A Unity3D scene was used with a 360° sphere object to display the videos. We use the FOVE Unity plugin to record head and gaze positions.

Recordings of EDA and optical pulse have been made with a Shimmer3 GSR+ sensor with a frequency range of 15.9Hz and 51.2Hz, respectively. All of the measurements were resampled to 100Hz for analysis. The apparatus is depicted in Fig. 5.3.

5.3.3 Participants

The experiment was carried out with a total of 34 users. The data of three participants was removed from the dataset due to corrupt or incomplete files. No other outlier removal procedure was implemented. Therefore, we include the data from 31 participants (10 women, 20 men, 1 non-binary; 18-29 years old, M=24, SD=3.26). 19 of them had a normal vision, 9 had corrected to normal vision and 3 did not have a normal vision. Most of them played games but rarely or never in VR, and the majority have seen only one or two 360° videos before the experiment. Participants received monetary compensation for their time. The seven videos were experienced by all 31 users for their entire duration (60 to 135 seconds, see Table 5.1).

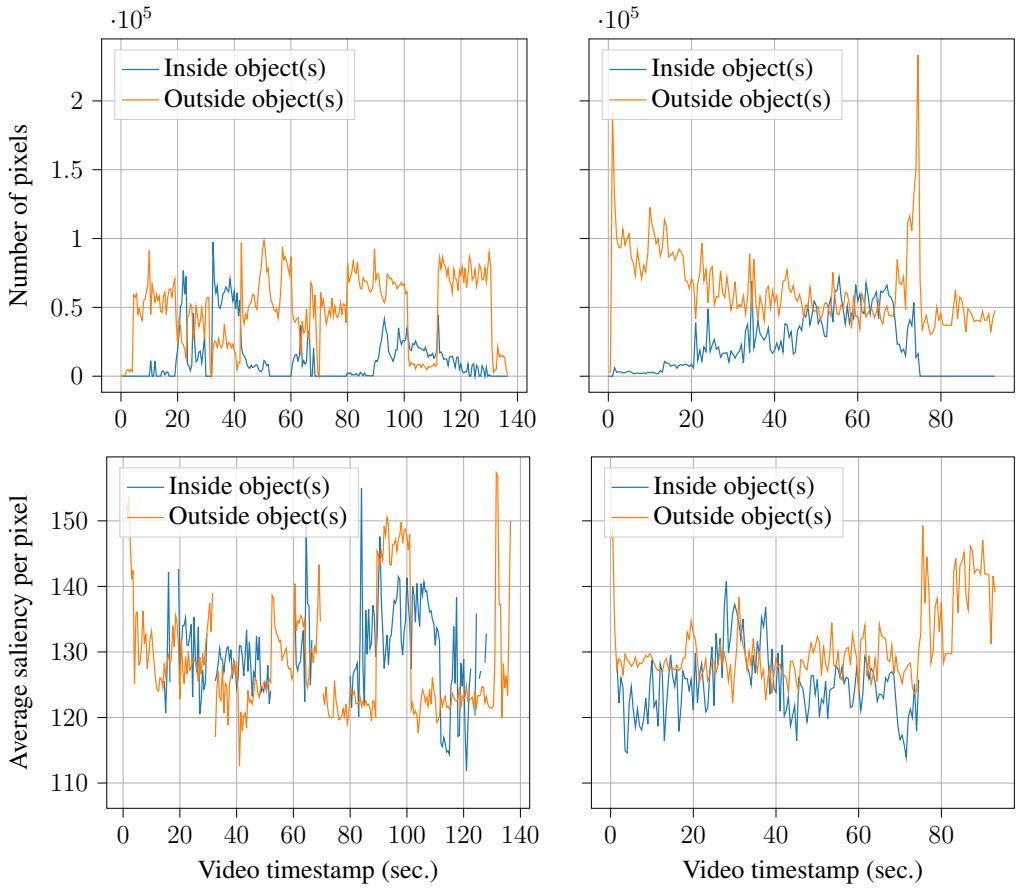


Figure *5.1: HL and LL saliency characterization of video 13 (left) and video 73 (right). Top: number of pixels inside and outside objects. Bottom: average LL saliency per pixel inside and outside objects.*

Figure *5.2: HL and LL saliency visualization for frame 2145 of video 13 (top) and frame 3630 of video 73 (bottom). Left: the frame. Center: HL saliency (detected objects, human on top, animals at bottom). Right: LL saliency.*

5.3.4 Procedure

The lab experiment started with a pre-questionnaire assessing the user's background with VR and checking for visual deficiencies. Eye tracking calibration was done using the

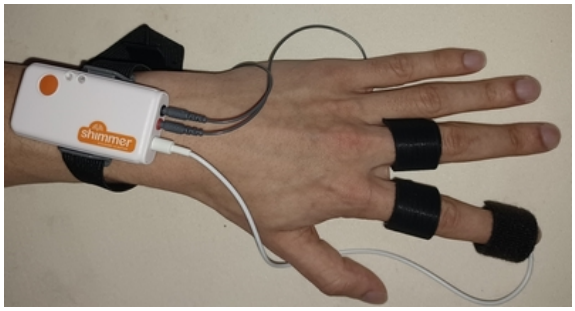


Figure *5.3: Shimmer3 GSR+ used to record EDA and optical pulse. Gray wires connect the EDA sensor, white wire connects the pulse sensor.*

FOVE software for each user before beginning the experiment to make sure the eye tracking data is properly recorded. The VR experiment systematically started with a low-arousal (relaxing) video (ID 32) to bring EDA and HR levels to a user-relative baseline. The remaining six VR videos were then experienced in a random order by every user. Users were in standing position during the experience and could freely explore in 360° while holding the back of a chair to maintain balance and orientation. The videos were played without audio. After each viewing, the headset was removed and the user was asked to rate their emotions (valence and arousal) using the self-assessment manikin (SAM, see Fig. 5.4). An at least 1-min break outside the headset was observed between every video.

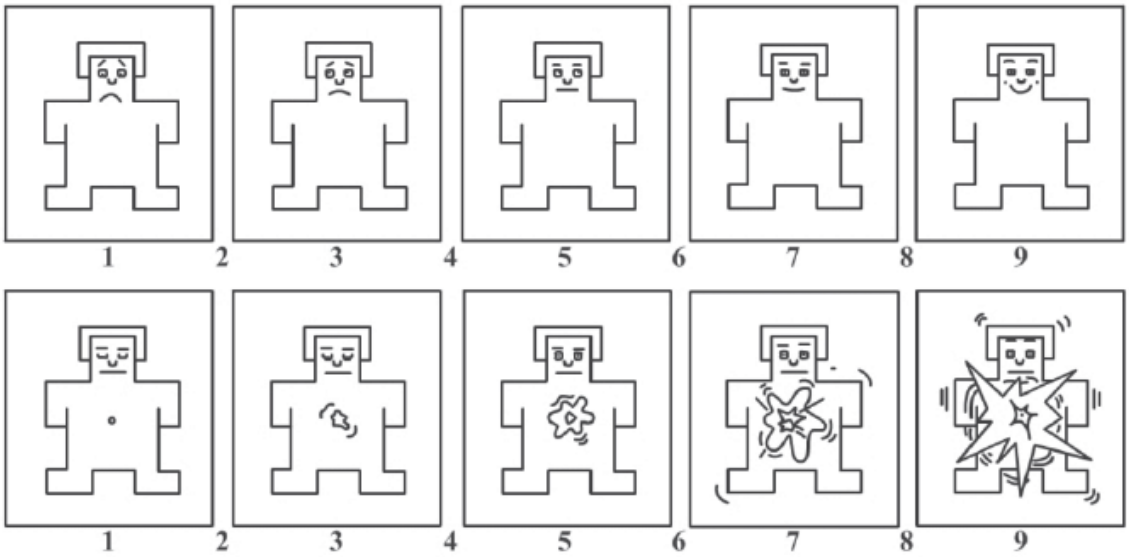


Figure *5.4: Self-Assessment Manikin (SAM) scale for rating of valence (top row) and arousal (bottom row). Taken from (Bradley & Lang, 1994).*

5.4 Dataset and tools

Along with the data, we provide a Jupyter notebook to reproduce the entire processing of head and gaze data, EDA, ratings of valence and arousal, and the code to produce saliency maps from the content.

5.4.1 Dataset structure

The resulting dataset PEM360 is provided with the structure shown in Fig. 5.5. The `raw_data` folder contains 34 folders, one for each user. User folders contain a Shimmer CSV file containing the EDA and optical pulse data recorded over all the 360° videos experienced by the user, and seven CSV files, one per video, containing the gaze and head motion data recorded during the corresponding video. Entries in the CSV files include system timestamps to synchronize the data modalities for analysis.

Valence and arousal ratings of each user for each video are stored in the root folder under `graded_valence_arousal.csv`. Finally, the root folder `PEM360` also contains the Python Jupyter notebook providing the software tools described below, and the entire data processing workflow to reproduce the analysis presented in Sec. 5.5.1 and Sec. 5.5.2.

```
PEM360/
├─ emotional_maps/
│   ├─ compute_emotional_map.py
│   ├─ requirements.txt
│   └─ utils_shimmer.py
├─ raw_data/
│   ├─ LICENSE
│   ├─ user_01/
│   │   ├─ shimmer.csv
│   │   ├─ video_12.csv
│   │   ├─ video_13.csv
│   │   ├─ video_17.csv
│   │   ├─ video_23.csv
│   │   ├─ video_27.csv
│   │   ├─ video_32.csv
│   │   └─ video_73.csv
│   ├─ user_02/
├─ saliency_maps/
│   ├─ compute_gt_saliency.ipynb
│   ├─ compute_hl_saliency.py
│   ├─ compute_ll_saliency.py
│   ├─ nfov.py
│   ├─ pySaliencyMap.py
│   ├─ pySaliencyMapDefs.py
│   └─ requirements.txt
├─ LICENSE
├─ README.md
├─ design_of_experiments.csv
├─ graded_valence_arousal.csv
├─ notebook.ipynb
├─ requirements.txt
└─ resample_data.ipynb
```

Figure *5.5: Folder structure of the dataset with main files.*

5.4.2 Processing gaze data

For both HM and EM, 3D positions are logged in Cartesian coordinates $(x, y, z) \in \mathbb{R}^3$. We provide functions:

- to convert the positions from Cartesian to Eulerian (ϕ, θ, ψ) denoting respectively yaw, pitch and roll (function `cartesian_to_eulerian()`),

- to obtain speed and acceleration over yaw and pitch (function `get_speed_acc_yaw_pitch()`),
- to obtain global speed and acceleration by computing the derivatives of the orthodromic distance (function `get_speed_acc()`),
- to represent rotational motion with quaternions (hence enabling to compute non-linear motion on the sphere as changes in quaternion rotational axis) (function `convert_to_quaternion()`).

5.4.3 Processing EDA data

The EDA signal is the raw measurement of skin electrical conductance in micro-Siemens (μS). Two main components can be distinguished in an EDA signal (Braithwaite, Watson, Jones, & Rowe, 2013; Boucsein, 2012): the tonic level, also called skin conductance level (SCL), varies slowly and represents slow autonomic changes that may not be associated with stimulus presentation; and the phasic level, which represents faster changes in EDA, and can better reflect the impact of successive stimuli. Raw EDA, phasic and tonic components are shown in Fig. 5.6-top and 5.6-center. We use the Python toolbox Neurokit (Makowski et al., 2021) to process EDA data, which uses the cvxEDA method to extract the phasic component. Finally, the physiological arousal to be analyzed in connection with experimental stimuli can be assessed from several metrics on the phasic level, such as peak frequency, duration and amplitude. This is called the skin conductance response (SCR), and can be defined in several ways. In our code, we choose to compute instantaneous SCR as the absolute value of the first-order time derivative of the phasic component, shown in 5.6-bottom. Note however that the code can easily be modified to implement other definitions of SCR from the phasic component. The obtained SCR is therefore a time series for every user-video pair. This enables analysis with SCR averaged over time for each such pair (as often done), or on a time-dependent basis.

5.4.4 Processing video content

As introduced in Sec. 5.3.1, we use LL and HL saliency models designed for regular flat images. We therefore apply them on FoV projections of the entire frame. We first uniformly sample 100 points on the unit sphere and project them on the equirectangular frame using the equirectangular-toolbox (Mutha, 2017). Each "patch" is made of a projection centered on one of these points, it is a 512x512 image corresponding to a 108°x108° FoV. These patches can overlap each other and are separately given to the appropriate models for both LL and HL saliency. For LL saliency, we use a Python implementation of Itti's saliency map (Kimura, 2020), which also allows the combination of Itti saliency with the optical flow between consecutive frames, which we do by using separate extractors for each patch. For HL saliency, we use the TensorFlow 2 implementation of YOLOv4 (TensorfFlow, 2021). For each patch given to the YOLO model, we create a binary saliency map equal to 1 inside the bounding boxes of the detected objects. For both LL and HL saliency, the overlapping patches are back-projected by addition onto

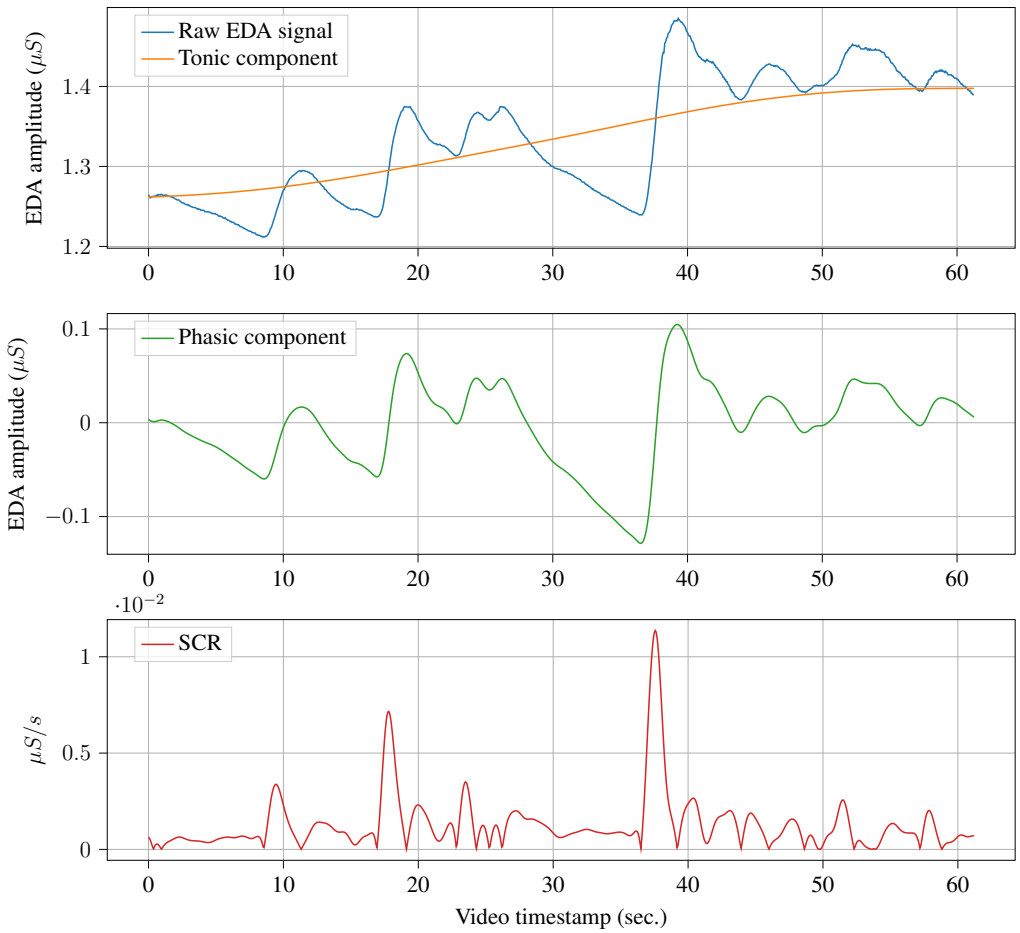


Figure *5.6: EDA signal recorded for user 03 while watching video 73. The three graphs from the top show the raw EDA data and the tonic component, the phasic component and the SCR (absolute value of phasic first derivative).*

the equirectangular frame to obtain a single (LL or HL) saliency map per frame. For LL saliency, the back-projection is normalized by dividing the value of each pixel by the number of patches it belongs to. The final value of a given pixel is the average over all existing projections for this pixel. For HL saliency, the back-projection is normalized by clipping the value of each pixel between 0 and 1. The final value of a given pixel is the maximum over all existing projections for this pixel. Finally, the saliency maps are downscaled by a factor of 5 both horizontally and vertically (from 1920x1080 to 384x216) for storage space reasons. The LL saliency is downscaled using average pooling over blocks of 5x5 pixels, whereas HL saliency is downscaled using max pooling over blocks of the same size. The files are stored in HDF5 format and can be accessed from a link given in the repository mentioned in Sec. 5.1, but can also be re-computed from the provided code.

5.4.5 Instantaneous visualization of gaze and emotions: *Emotional maps*

Figure *5.7: Emotional map visualizing instantaneous gaze locations (luminance) and user arousal (from blue to red for low to high SCR). Example with high arousal in a roller-coaster video.*

As previously discussed, the stimuli choice and experimental procedure are designed to collect data enabling a time-dependent analysis of the connection between attention, emotion and content. That is why we provide a tool for the experimenter to play the 360° video and visualize the instantaneous gaze locations and arousal (SCR) of a given user from the recorded data. This tool implements a new way of visualizing arousal in connection with gaze, which we name *emotional maps*. An *emotional map* is a 4D-array represented as a frame where:

- pixel luminance reflects the time the user spent attending the area over a past window of T seconds. A Gaussian kernel of parameter σ is convolved with every gaze location, and accumulated over the sliding window of T seconds. A bright (resp. dark) area can therefore reflect a fixation (resp. a saccade).
- pixel color represents the user's SCR, from blue (low arousal) to red (high arousal).

Emotional maps generated from a record with our tool are accumulated into videos. Each point persists on the video for P seconds, creating a trail to easily visualize the gaze path and arousal changes. An example of such a video frame is shown in Fig. 5.7*. The script `compute_emotional_map.py` creates the emotional maps and blends them with the frames to produce the resulting video visualization from records of gaze and EDA data. We believe this tool can lead to important qualitative insights for diverse disciplines (including neuroscience) on the connection between visual attention and emotion.

*Demonstration of a resulting video is accessible at https://tinyurl.com/25vjwk2s.

5.5 Analyses of the collected data

5.5.1 Preliminary analysis

In this section we first verify the validity of our data and correspondence with the original dataset and between arousal and EDA. We then exemplify possible analyses of correlation between motion and emotion, and between attention, content saliency and emotion.

5.5.1.1 Data validation

Reliability of the collected ratings We verify the reliability of the collected arousal and valence by assessing the similarity of the user ratings for each video. This is achieved with the intra-class correlation coefficient (ICC), with classes corresponding to the 360° videos. ICC estimates based on mean ratings with a two-way mixed effects model are 0.96 (95% CI 0.87-0.99) for arousal and 0.88 (95% CI 0.72-0.98) for valence. According to Koo and Li's guidelines (Koo & Li, 2016), this is excellent and good inter-rater agreement, respectively.

Agreement between collected ratings and original dataset Fig. 5.8 shows the valence and arousal ratings of our users as a boxplot for each video, along with a red dot representing the corresponding average values available in the original dataset (B. J. Li et al., 2017). We observe the good agreement between both sets, as the latter are all the times but one in the inter-quartile value range of our data. We also compute the median of the root square difference of averages of our valence and arousal ratings with the corresponding averages from (B. J. Li et al., 2017). This median is 1.17 (within a range of 1 to 9), showing the agreement between both.

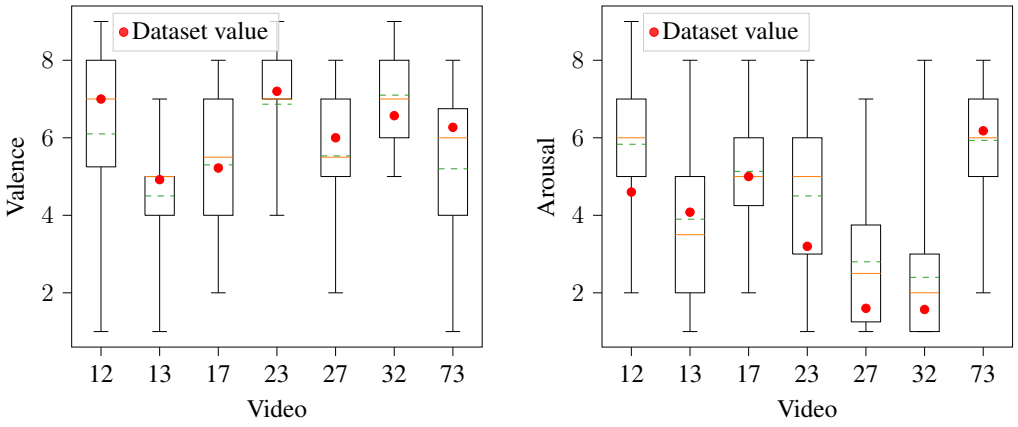


Figure *5.8: Arousal and valence ratings by users for each videos. The green dotted line corresponds to the mean and the orange solid line to the median.*

5.5.1.2 Connecting EDA with graded arousal

We investigate the correspondence between SCR and arousal ratings. We gather the average SCR values $\text{SCR}_{u,v}$ for every pair (u, v) of user u and video v, and corresponding graded arousal $\text{GA}_{u,v}$. First, we average both variables over all users for every video, and obtain seven sample pairs $(\text{GA}_v, \text{SCR}_v)$, shown in Fig. 5.9-left. We verify as did Toet et al. (2020) that the video ranking according to mean graded arousal is similar to the video ranking according to mean SCR. We also compute the Spearman correlation coefficient (CC) between GA_v and SCR_v for all seven videos. The Spearman CC between mean graded arousal and mean SCR is $(0.92, p = 0.003)$. According to (Walline, 2001, appx. 6C, p. 79), such level of correlation is significant ($\alpha = 0.05$, $\beta = 0.2$) from 7 samples (see (UCSF, 2021)).

We then consider the 217 sample pairs $(\text{GA}_{u,v}, \text{SCR}_{u,v})$. It is interesting to observe that the Pearson or Spearman CCs do not show any correlation between these pairs. Looking more closely at the data, we identify that the mean level of SCR per user, $\text{SCR}_u = \mathbb{E}_v[\text{SCR}_{u,v}]$ (averaged over all videos), varies significantly over the users $(M = 6.0e{-}4, SD = 6.2e{-}4)$. With the rationale that the *excitability* of a user is person-dependent and impacts the absolute SCR values, we verify whether the SCR variations relative to this individual's mean are better associated with graded arousal. To do so, we define centered SCR as $\text{cSCR}_{u,v} = \text{SCR}_{u,v} - \text{SCR}_u$, and do the same with graded arousal $\text{cGA}_{u,v} = \text{GA}_{u,v} - \text{GA}_u$. Fig. 5.9-right represents the scatter plot of $\text{cSCR}_{u,v}$ against $\text{cGA}_{u,v}$. The Spearman CC between both is $(0.25, p < 0.001)$. According to (Walline, 2001, appx. 6C, p. 79) implemented in (UCSF, 2021), such level of correlation is significant ($\alpha = 0.05$, $\beta = 0.2$) from 52 samples. There is therefore a moderate significant correlation between centered SCR and centered graded arousal (Akoglu, 2018).

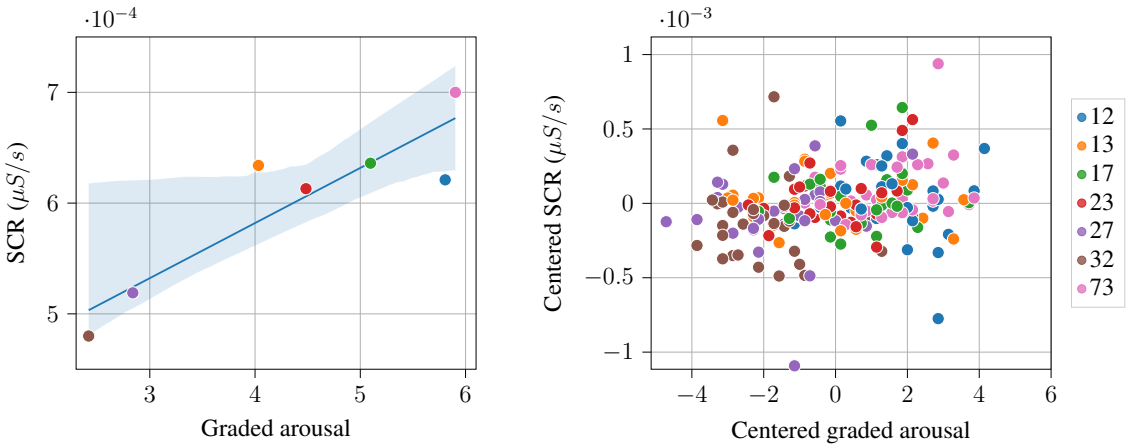


Figure 5.9: *Dots colors code for video ID (legend on the right). Left: Scatter plot of SCR_v against GA_v. The shaded area represents the 95% CI of the linear regressor (solid blue line). Right: Scatter plot of $\text{cSCR}_{u,v}$ against $\text{cGA}_{u,v}$.*

5.5.2 Investigating the link between attention, emotion and content

Our objective is to compare the accuracy of both types of saliency maps, HL and LL, to match the users' fixations over every frame of the 360° video. To do so, we compute the normalized scanpath saliency (NSS), which measures the amount of saliency around fixations (Le Meur & Baccino, 2013). We consider segments of 5 sec. to average the saliency maps of all frames and aggregate the user's fixations in this interval, hence obtaining an NSS value for both saliency types $\text{NSS}^{HL}_{u,v,i}$ and $\text{NSS}^{LL}_{u,v,i}$ for every user u, video v, and interval i. The averages over intervals (resp. users) are denoted by $\text{NSS}_{u,v}$ and NSS_v, respectively. We analyze the association between $\text{NSS}^{HL}_{u,v}$ and $\text{NSS}^{LL}_{u,v}$ with mean centered SCR denoted $\text{cSCR}_{u,v}$ and graded arousal $\text{GA}_{u,v}$. SCR is centered per user with $\text{cSCR}_{u,v} = \text{SCR}_{u,v} - E_v[\text{SCR}_{u,v}]$ because the preliminary analysis has shown that the absolute levels of SCR vary significantly across users, but intra-user variations across videos are consistent with the ordering of each user's arousal ratings.

5.5.2.1 Results

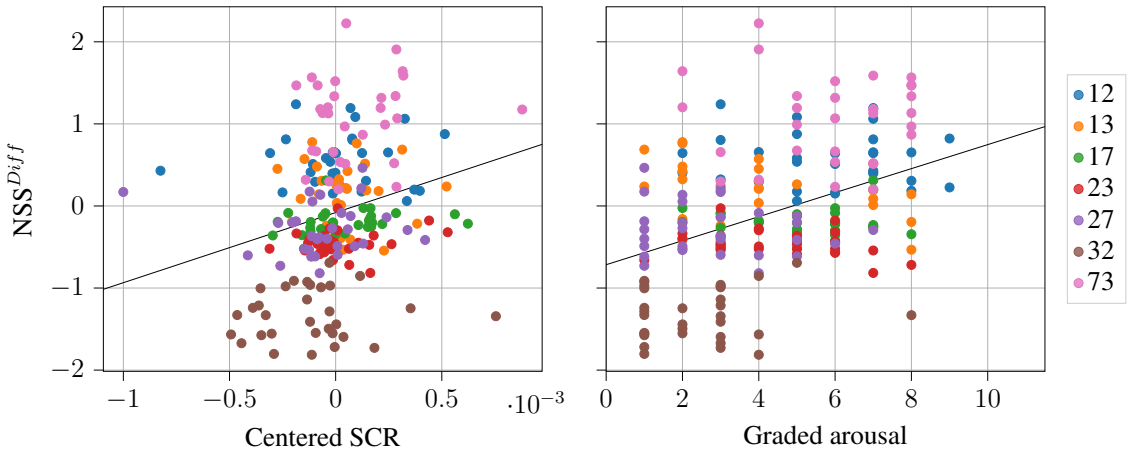


Figure *5.10: $\text{NSS}^{Diff}_{u,v}$ against $\text{cSCR}_{u,v}$ and $\text{GA}_{u,v}$ for every user u and video v. The black line shows a linear regression model fitted on the data.*

To analyze the difference in accuracy of both types of saliency depending on the user's arousal, we consider in Fig. 5.10 the difference $\text{NSS}^{Diff}_{u,v} = \text{NSS}^{HL}_{u,v} - \text{NSS}^{LL}_{u,v}$ plotted against $\text{cSCR}_{u,v}$ (left) and graded arousal $\text{GA}_{u,v}$ (right) for all u, v, the points being colored per video. The major finding is the increasing trend of NSS^{Diff} with EDA and graded arousal. Specifically, the PCC between NSS^{Diff} and EDA cSCR is 0.25 ($p < 10^{-3}$), and the PCC between NSS^{Diff} and graded arousal $\text{GA}_{u,v}$ is 0.41 ($p < 10^{-9}$). These estimates are obtained over 217 (u, v) samples. According to Walline (Walline, 2001, Appendix 6C, page 79), such levels of correlation are significant for 123 and 44 samples, respectively (see (UCSF, 2021)).

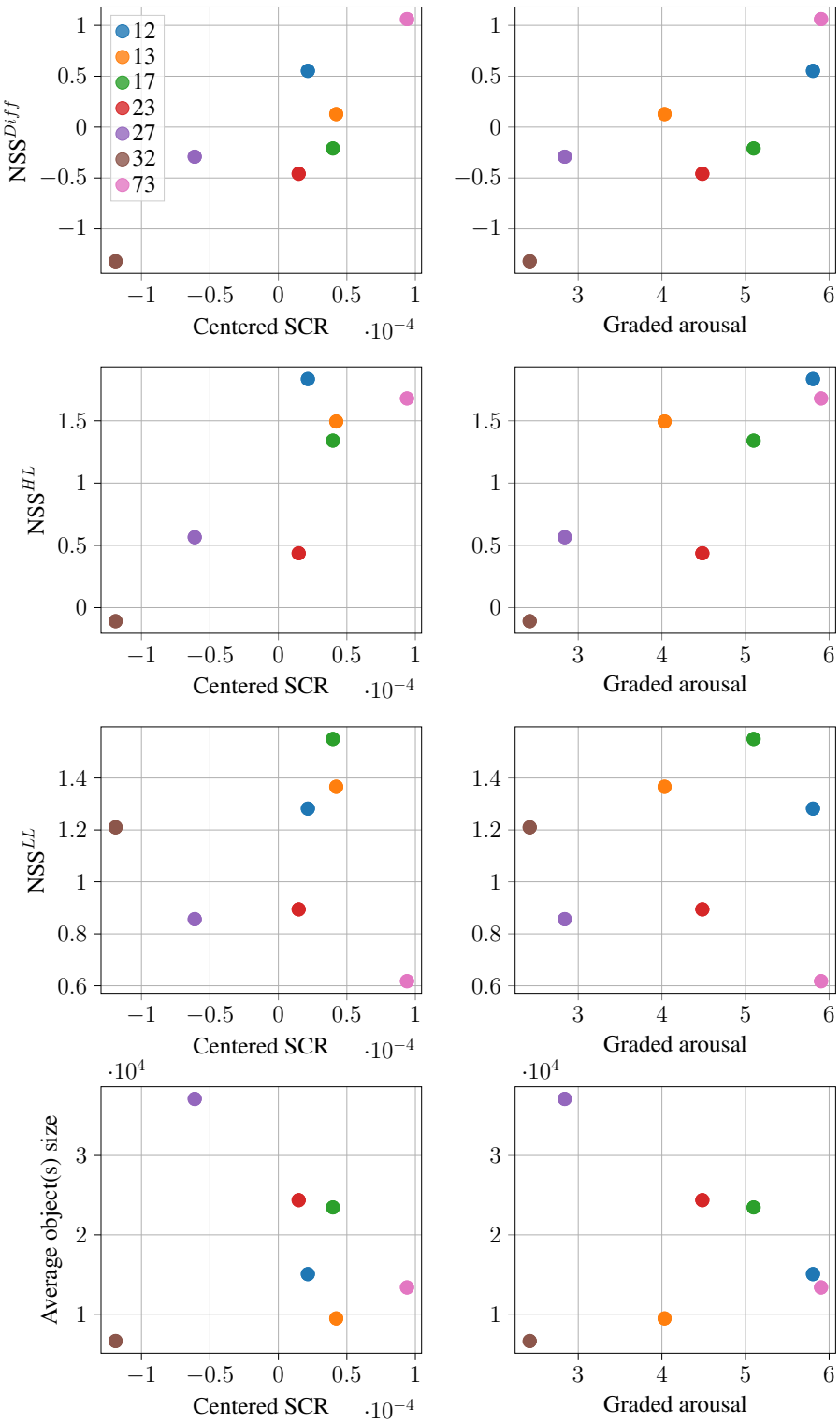


Figure *5.11: From top to bottom:* NSS_v^{Diff}, NSS_v^{HL}, NSS_v^{LL} *and average number of pixels inside objects against* $cSCR_v$ *(left) and* GA_v *(right) for all videos* v.

We then analyze the same associations averaged per video in Fig. 5.11, where the x-axis of the first row is cSCR_v and that of the second row is GA_v, with v in the set of video indices. The columns are numbered from the left. We first confirm from the leftmost column that ordering and appearance of NSS_v^{Diff} against EDA or graded arousal are close. Second, we observe a clear increasing trend confirming the above positive significant correlation results.

To investigate the reasons for this trend, we decompose NSS_v^{Diff} into its individual components NSS_v^{HL} and NSS_v^{LL} depicted in columns 2 and 3. Owing to the similarity of trends against EDA and graded arousal, we conduct the analysis only on the latter. We first observe an increasing trend of NSS_v^{HL}. It could have been even clearer considering that underwater objects in video 12 (brown dot) are often missed by the object detector (large shark), hence under-estimating NSS_{12}^{HL}. We can then question whether this increase is due to users focusing more when more aroused, or to intrinsic features of the videos, where larger objects would appear in higher arousal videos. We verify in the last column that the increase in NSS^{HL} with arousal cannot be entirely attributed to a relatively larger area occupied by objects. Second, column 3 shows no clear trend, NSS^{LL} seems to remains steady over the range of arousal. One exception is the highest-arousal video (73, green dot) where the very low NSS^{LL} is explained by large and dark homogeneous objects (black furred gorilla). The variation in NSS^{LL} does not appear to be related to EDA or graded arousal.

We can therefore conclude that the increasing trend of NSS^{Diff} with arousal is mainly due to higher NSS^{HL} for higher-arousal videos. A first conclusion we may draw is that the relative weight of HL saliency should vary in a saliency model depending on the user's arousal state. Sensing the user's state may hence help predict their attention.

5.5.2.2 Discussion

We cannot claim causation on whether users focus because they are more aroused by the content, or if they are more aroused because they focus on objects. A first question is whether significantly different levels of arousal occur for users on the same video. This would mean that the video content alone is not informative enough to adapt the relative weights of HL and LL saliency to users. On the contrary, if the video content is sufficient, then one can think of leveraging arousal (physiological or subjective) measurements in quality assessment sessions to serve as an auxiliary loss to train (deep) saliency models.

While arousal and valence are major dimensions to describe user emotion of a given content like a video, the richer experience of an immersive and possibly interactive environment is described over various additional dimensions, particularly presence, immersion, agency, engagement, flow, usability, skill or judgement (Tcha-Tokey, Christmann, Loup-Escande, & Richir, 2016). Recently, valence, arousal and agency have been shown to interact in non-trivial ways to produce presence (Jicol et al., 2021). In 6DoF environments, which we are currently investigating for rehabilitation scenarios and where engagement, skills and judgments are major outcomes, it is crucial to adapt the environment's content to provide proper adaptive guidance to the user. This requires an under-

standing and the prediction of the user's attention and intents, which depend on the user's emotional state. This work in 3DoF immersive low-interaction environment hence serves as a baseline for immersive interactive environments.

5.6 Effects of emotions on head motion predictability

In this section, we want to investigate the effects of emotions on head motion predictability in VR. We first define the head motion prediction problem and presents the prediction method we consider in Sec. 5.6.1. We detail the chosen datasets and measures in Sec. 5.6.2. We make hypotheses based on previous works and analyze their validity from the data in Sec. 5.6.3, and we model the effects of user emotions and motion on predictability with a SEM in Sec. 5.6.4. Finally, we discuss limitations and perspectives in Sec. 5.6.5.

5.6.1 Head motion prediction

We first define the problem of head motion prediction in Sec. 5.6.1.1, then describe the chosen method and the motivation behind this choice in Sec. 5.6.1.2.

5.6.1.1 Problem definition

We consider the same head motion prediction problem as in chapter 3. In this section, we use the following description. We consider that a given 360° video v of duration T seconds is being watched by a user u. The head trajectory of the user is denoted $\mathbf{P}^{u,v}_{0:T}$, with $\mathbf{P}$ storing the head coordinates on the unit sphere (as, e.g., Euler angles, Cartesian coordinates or quaternions).

At any time t in $[0, T]$, we want to predict the future trajectory $\mathbf{P}^{u,v}_{t:t+H}$ over a prediction horizon H, assuming only $\mathbf{P}^{u,v}_{0:t}$ and the video content of v are known. That is, we do not assume any knowledge of traces other than u on this video v.

5.6.1.2 Prediction method

In order to predict head motion, many methods have already been proposed. Clustering-based methods, like the one proposed by Petrangeli et al. (2018) or the one proposed by Nasrabadi et al. (2020) need to rely on other user traces of the same video v and cannot be used on new videos. Since we do not assume any knowledge of traces other than the current user u on the current video v, we need to consider other kinds of prediction methods. Recent works like TRACK (Romero Rondón et al., 2021) and VPT360 (Chao et al., 2021) are deep-learning models that have been shown to perform well on various head motion datasets. We choose two main methods presented by Romero Rondón et al. (2021), named *Deep-position-only* and TRACK. We make this choice because (i) these approaches are representative of other prior approaches relying on sequence-to-sequence architectures, (ii) they are very close to our proposed state-of-the-art approach, DVMS

(chapter 3), when it comes to single trajectory prediction, and (iii) the models and entire framework are made publicly available (Romero Rondón et al., 2020).

To conduct our study, we consider both models trained on two different head motion datasets from David et al. (2018) and Y. Xu et al. (2018), and selected the trained models that obtained the best results when testing (without re-training or fine-tuning) on our data described in Sec. 5.6.2. All the trained models were similar in performance on our data, but the models trained on the dataset by Y. Xu et al. (2018), the largest dataset, performed slightly better. Then, we inspected the mutual effects, such as those shown in Fig. 5.16, when the prediction error is obtained with *Deep-position-only* and TRACK. As results were qualitatively similar, for the rest of the chapter, we have chosen to only present results obtained with TRACK.

TRACK is a sequence-to-sequence deep model using separate long short-term memory (LSTM) units to encode both the past positions and the visual saliency. The same kind of LSTM units, combined with fully connected layers are then used to decode the future positions based on the embeddings given by the encoder. The visual saliency is made up of 384x216 saliency maps extracted from the video frames by PanoSalNet. A simplified diagram of the architecture is shown in Fig. 5.12. TRACK was fully re-implemented using PyTorch and trained on multiple head motion datasets as provided in the repository (Romero Rondón et al., 2020). To conduct our study, we chose the trained model that obtained the best results when testing (without re-training or fine-tuning) on our data described in Sec. 5.6.2. All the trained models were similar in performance on our data, but the model trained on the dataset by Y. Xu et al. (2018), the largest dataset, performed slightly better.

5.6.2 Datasets and measures

In this section, we present the datasets considered for our data analysis, and our choice of user-centric and video-centric measures. The effect of these measures on motion predictability in investigated next in Sec. 5.6.3 and 5.6.4.

5.6.2.1 Datasets

We consider the only two datasets available where both user movements and emotions have been collected from immersive viewing of 360° videos.

The first dataset we consider is PEM360, our own dataset collected from user experiments, described in Sec. 5.3 and 5.4.

Our second source of data comes from CEAP-360VR (Xue, Ali, Zhang, et al., 2021). This dataset is made from user experiments with 32 participants each watching 8 videos in a VR headset equipped with an eye-tracker, recording head and eye movements. After each video, the users grade their emotions valence and arousal.

Additionally, emotional ratings are continuously annotated by the users thanks to a controller in their hand, along with physiological measurements with a wristband. We do

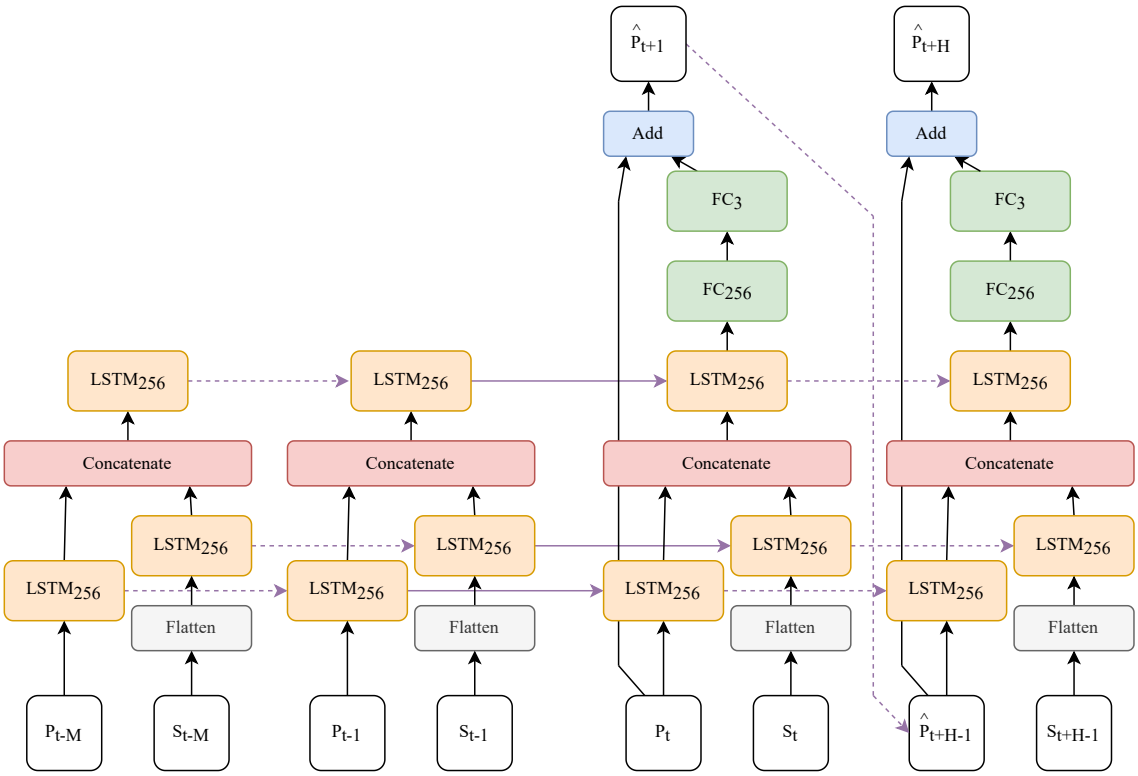


Figure *5.12: Simplified architectural diagram of the prediction method TRACK. Input P denotes positional coordinates and input S denotes visual content in the form of a frame saliency map.*

not use this latter data in this work. Head and eye movement were also recorded using the VR headset with an integrated eye tracker.

In both datasets, users were asked to rate each video using the self-assessment manikin (SAM) (Bradley & Lang, 1994), giving individual ratings of valence and arousal.

The videos shown to users in both datasets come from the same 360° video database (B. J. Li et al., 2017), with average ratings of valence and arousal from 95 participants. We report in Table 5.2 the details of every video from both datasets. The left-most column "ID" refers to the video ID the author used in their dataset. The right-most column "Database ID" indicates the original ID in the 360° video database (B. J. Li et al., 2017). Ratings of valence and arousal given in this table are the original average ratings of the database. We show the video ratings on the valence-arousal plane in Fig. 5.13. In each of the datasets, the videos were trimmed, so the videos are not exactly the same as in the database. The clips trimmed from each video were manually curated to preserve important semantic information of the original video, which preserves the validity of the original valence-arousal ratings. The start offset of each video as well as the duration of the trimmed clip are specified in the table.

As shown by the asterisk*, we note that videos 32 and V6 are different versions of the same video (clipped at different times), which makes a total of 14 distinct videos experienced in VR by 63 different users.

Table 5.2: Details of selected videos, combining the two datasets. The ID refers to the original database (B. J. Li et al., 2017). Ratings of valence and arousal are between 1 and 9.

ID	Valence	Arousal	Start (s)	Duration (s)	Database ID
12	7.00	4.60	5	98	12
13	4.92	4.08	4	127	13
17	5.22	5.00	5	64	17
23	7.20	3.20	8	135	23
27	6.00	1.60	60	120	27
32⋆	6.57	1.57	40	90	32⋆
73	6.27	6.18	9	61	73
V1	7.47	5.35	0	60	50
V2	6.13	1.80	10	60	38
V3	3.20	5.60	65	59	21
V4	2.53	3.82	3	60	14
V5	6.75	7.42	0	60	52
V6⋆	6.57	1.57	0	60	32⋆
V7	4.40	6.70	127	60	68
V8	2.73	3.80	41	60	19

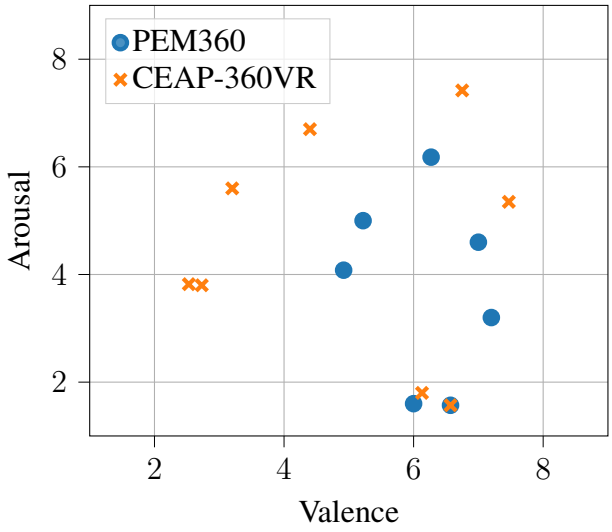


Figure *5.13: Average ratings of valence and arousal for all the considered videos.*

5.6.2.2 Measures

Outcome measure The objective of this analysis is to evaluate the influence of various factors on the prediction error of head movements. We define the prediction error as the average displacement error (ADE) between the predicted head positions and the actual future head positions over a prediction horizon H. We set $H = 5$ seconds, the standard prediction horizon in recent deep prediction methods (Romero Rondón et al., 2021; Chao

et al., 2021), which we also used in chapter 3 covers both user inertia and content saliency (Romero Rondón et al., 2021).

We define the displacement error between two head positions (x_1, y_1, z_1) and (x_2, y_2, z_2) as the great circle distance between these two points. Since (x, y, z) are the Cartesian coordinates of a point on the unit sphere, we can easily compute the great circle distance $\Delta\sigma$ from the Euclidean distance d between these two positions as $\Delta\sigma = 2 \cdot \arcsin \frac{d}{2}$.

User-centric measures We consider two types of user-centric measures: those related to emotions, and those related to motion. The measures of emotions are considered as the subjective ratings of valence and arousal made by each user after experiencing each 360° video, as detailed in Sec. 5.6.2.1.

The user motion can be characterized by various metrics, such as mean values of the head or gaze yaw and pitch angles, or the standard deviations of these positional components (B. J. Li et al., 2017; Xue, Ali, Zhang, et al., 2021). Here, we choose to combine these elements and consider the angular speed of the head movements. Specifically, to compute head speed, we first convert the head coordinates collected from the VR headset into Cartesian coordinates, where each recorded head position at time t is a point on the unit sphere of coordinates (x_t, y_t, z_t).

The head motion data in CEAP360-VR is originally in the format $(\psi_t, \theta_t, \phi_t)$, where ψ is the yaw, θ is the pitch, and ϕ is the roll. These coordinates were first transformed to have $\psi \in [0, 2\pi[$ where 0 is the left edge of the equirectangular frame, and $\theta \in [0, \pi[$ where 0 is the top edge of the equirectangular frame. Cartesian (x_t, y_t, z_t) coordinates are then obtained as projections of these angles using this set of equations:

$$\begin{cases} x_t = \cos\psi_t \cdot \sin\theta_t \\ y_t = \sin\psi_t \cdot \sin\theta_t \\ z_t = \cos\theta_t \end{cases}$$

We define the instantaneous head speed at time t as the total angular speed, noted ω_t, computed from the great-circle distance between two consecutive positions divided by the sampling rate of the recordings. The average head speed is then taken as the mean of all instantaneous head speeds for a given user on a given video.

Video-centric measures As for user motion, several characterizations of 360° video content are possible. For example, Almquist et al. (2018) proposed a taxonomy in four categories depending on the location of the regions of interest. Romero Rondón et al. (2021) built on this taxonomy and categorized videos based on the entropy of the head location heat maps. David et al. (2018) and Xue, Ali, Zhang, et al. (2021) considered spatial information and temporal information to characterize the 360° videos. In the preliminary study presented in this chapter, we consider legacy spatial and temporal information, and show their relevance characterizing the effects of emotions on motion predictability.

Spatial information and temporal information are scene-specific metrics defined in ITU-T Recommendation P.910 (ITU-T P.910, 2021). According to the ITU-T recommendation, SI and TI are "critical parameters" playing "a crucial role in determining the amount of video compression that is possible".

Spatial information (SI) or spatial perceptual information is "a measure that generally indicates the amount of spatial detail in a picture. (...) It is usually higher for more spatially complex scenes." "The SI is based on the Sobel filter. Each video frame (luminance plane) is first filtered with the Sobel filter. The standard deviation over the pixels in each Sobel-filtered frame is then computed", resulting in the SI for a single frame. We consider SI_v, the average SI for all the frames of video v.

Temporal information (TI) or temporal perceptual information is "a measure that generally indicates the amount of temporal changes of a video sequence. (...) It is usually higher for high motion sequences." "TI is based upon the motion difference feature, that is the difference between the pixel values (of the luminance plane) at the same location in space but at successive frames." The standard deviation over the pixels of all the differences between successive frames is then computed to give the TI for two consecutive frames. We consider TI_v, the average TI for all the frames of video v. "More motion in adjacent frames will result in higher values of TI."

5.6.3 Hypothesis testing

Based on previous works (B. J. Li et al., 2017; Xue, Ali, Ding, & Cesar, 2021), we make the following *a priori* hypotheses:

(H1) Prediction error is lower for higher user arousal.

(H2) Prediction error is higher for higher user valence.

(H3) Head speed mediates the effect of valence on error.

To analyze the validity of the above hypotheses, we first binarize some variables and perform analysis of variance (ANOVA) testing, shown in Table 5.3. The analysis of linear correlations on continuous data is incorporated into the structural equation modeling in Sec. 5.6.4.

*Table 5.3: F-scores of one-way ANOVA. The significance of group difference is denoted with * for $p < 10^{-2}$ and ** for $p < 10^{-3}$.*

	SI_{bin}	TI_{bin}	$Arousal_{bin}$	$Valence_{bin}$
Prediction error	70.89**	79.09**	15.15**	7.67*
Head speed	17.77**	19.94**	2.76	15.78**
Arousal	51.90**	55.50**	(1253**)	0.37
Valence	30.60**	2.69	0.42	(1266**)

The binarization is performed on *SI*, *TI*, *Arousal* and *Valence* (denoting continuous variables) to obtain SI_{bin}, TI_{bin}, $Arousal_{bin}$ and $Valence_{bin}$. For *SI* and *TI* of every video v, binarization thresholds are chosen so that approximately half of the videos are in each

partition: SI_{bin} = -1 (resp. 1) for $SI_v \leq 45$ (resp. > 45), and TI_{bin} = 0 (resp. 1) for $TI_v \leq 3$ (resp. > 3). In Fig. 5.16, SI_{bin} is denoted "Low SI" or "High SI" with the same threshold. For *Arousal* and *Valence* of every user-video pair (u, v), $Arousal_{bin}$ = 0 (resp. = 1) for $Arousal_{u,v} \leq 5$ (resp. > 5), and the same to obtain $Valence_{bin}$. This threshold was chosen because the ratings are between 1 and 9, and 5 is usually considered to be neutral. In Fig. 5.14, $Arousal_{bin}$ (resp. $Valence_{bin}$) is also referred to as "LA" for low *Arousal* (resp. "LV" for low *Valence*) and "HA" for high *Arousal* (resp. "HV" for high *Valence*), with the same thresholds as defined above.

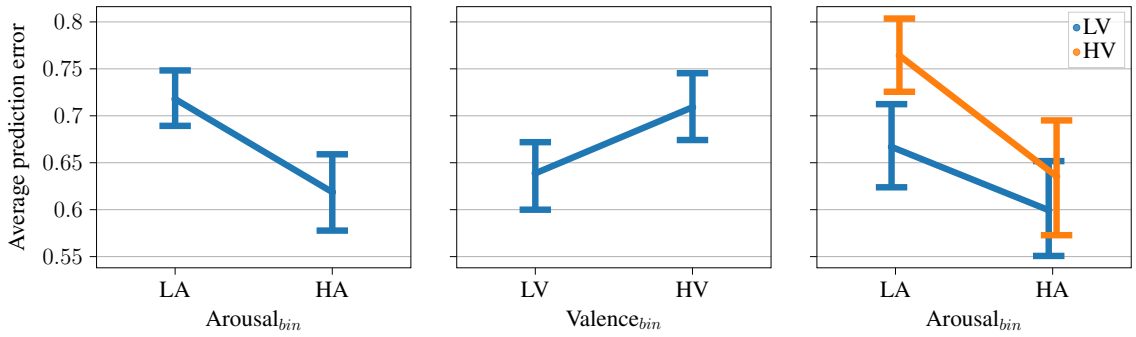


Figure *5.14: Prediction error against binarized $Arousal_{bin}$ (left) and $Valence_{bin}$ (center). Right: Difference in variation of Prediction error against $Arousal_{bin}$ depending on $Valence_{bin}$.*

We first observe from Table 5.3 that SI_{bin} significantly impacts all variables (*Prediction error*, *Head speed*, *Valence* and *Arousal*), while TI_{bin} does not significantly impact *Valence*. The relations mentioned in (H1) and (H2) are significant. Fig. 5.14-left and 5.14-center show the direction of the association with 95% confidence intervals. We can therefore accept (H1) and (H2).

To investigate the sizes of the significant effects, Fig. 5.15 shows scatter plots of error as a function of *Arousal* and *Head speed*, as well as how *Head speed* varies with graded *Arousal* and *Valence*. We first observe that there is a strong correlation between *Prediction error* and *Head speed*. We also observe that, as hinted in preliminary results from B. J. Li et al. (2017) and Xue, Ali, Ding, and Cesar (2021), *Prediction error* tends to decrease with *Arousal*. While *Head speed* does not seem to significantly vary with *Arousal*, as confirmed by the ANOVA result in Table 5.3, the scatter plot of *Head Speed* versus *Valence* shows that the significant association between both, shown by the corresponding ANOVA result in Table 5.3, is an increasing function. This is in line with (H3), which will be validated in the next section.

As Fig. 5.15-top-right shows the strong association of *Prediction error* with *Head speed* and Table 5.3 shows significant associations of *Prediction error*, *Head speed*, *Arousal* and *Valence* with SI_{bin}, we analyze whether some variables interact with *Arousal* and *Valence* in their effect on *Head speed* and *Prediction error*. Fig. 5.16 shows that there is a possible interaction between the video feature SI_{bin} and *Arousal*, and SI_{bin} and *Valence*, in their effect on *Prediction error* and *Head speed*. This can be seen in the different

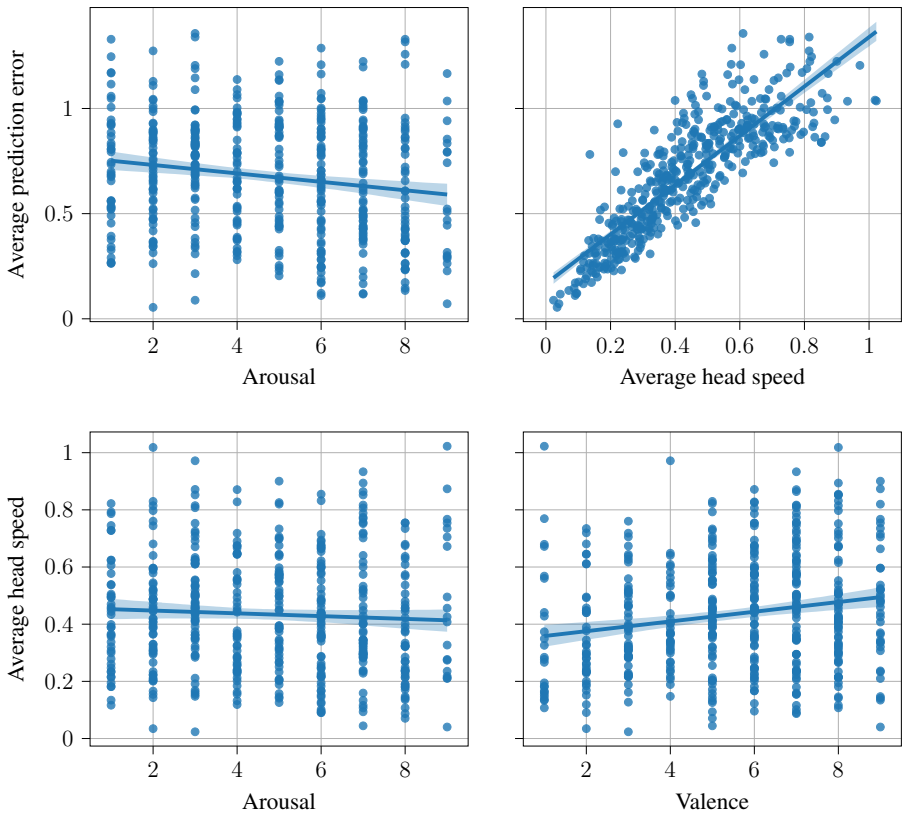


Figure *5.15: Scatter plots of Prediction error against Arousal and Head speed (top row), and Head speed against Arousal and Valence (bottom row). Straight lines are linear regression models fitted on the data. Shaded areas represent 95% confidence intervals.*

slopes of linear model fitting the cloud of points, for each set of (u, v) points, for all users $u \in \mathcal{U}$ and videos v such that $SI_{bin}(v)$ = -1, or 1.

Also, it is interesting to observe in Fig. 5.14-right that *Prediction error* does not decrease in the same way with increased *Arousal*, depending on whether *Valence* is graded high or low. Indeed, *Prediction error* decreases more when *Arousal* increases when *Valence* is high. We may assume that the user tends to move more when they enjoy the video, and higher *Arousal* means more involvement/attentional capture, and hence synchronization between motion and the content's salient regions, facilitating the prediction. This corresponds partly to (H3) and is investigated in the next section.

5.6.4 Modeling the effect of emotions and video characteristics on motion predictability

We now construct a structural equation model of the data. SEM is established as a methodological approach to represent how different variables affect each other (Hoyle, 1995). It allows to build a network of causal relations, and to investigate direct and indirect effects with mediating variables and external moderators interacting on the effect. A SEM

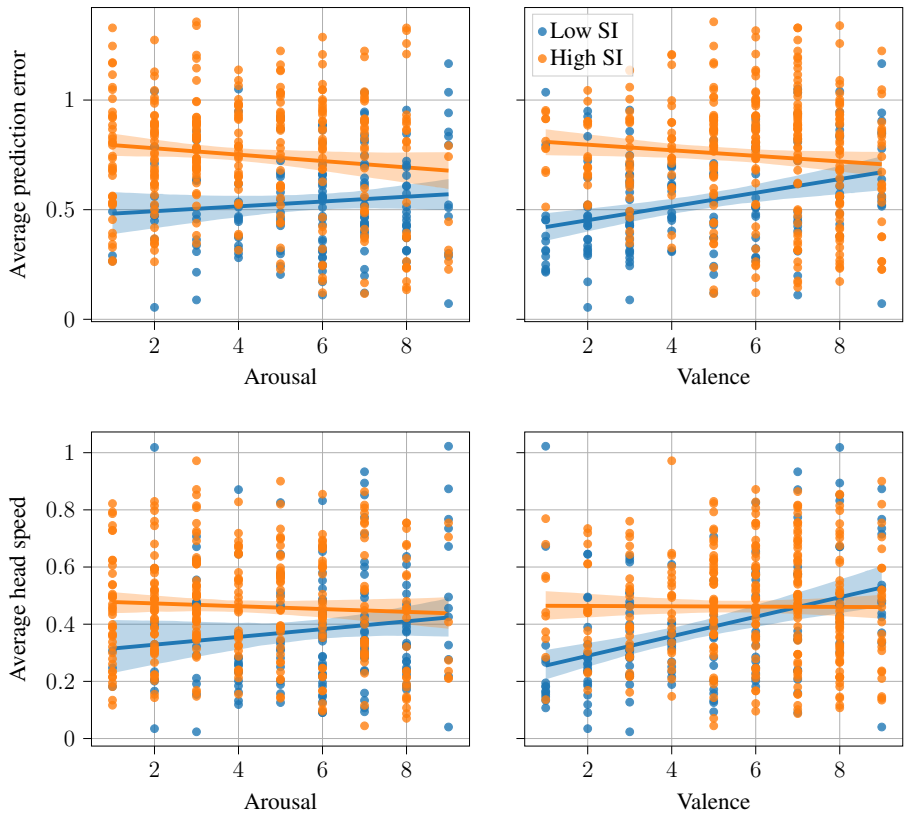


Figure *5.16: Scatter plots of Prediction error and Head speed against Arousal and Valence, disaggregated over SI_{bin}. Straight lines are linear regression models fitted on the data. Shaded areas represent 95% confidence intervals.*

therefore gathers significant linear relations, enabling to both incorporate the correlation coefficient and measure the size of the effect. We construct a SEM based on accepted (H1) and (H2), and incorporating the possible interaction of SI_{bin} with *Arousal* and *Valence*. An interaction effect is modeled as the product of two variables, one of which is binary. Owing to the above analysis of Fig. 5.16, we define interaction variables *Arousal* $\times$ SI_{bin} and *Valence* $\times$ SI_{bin}. We then consider possible causal relationships from *Arousal*, *Valence*, *Arousal* $\times$ SI_{bin} and *Valence* $\times$ SI_{bin} to both *Head speed* and *Prediction error*, as well as relationship from *Head speed* to *Prediction error*.

We use the Python toolkit Semopy (Igolkina & Meshcheryakov, 2020; Meshcheryakov, Igolkina, & Samsonova, 2021), using the Wishart log-likelihood objective function. The resulting SEM is shown in Fig. 5.17, where only edges with regression coefficients significantly different from 0 have been kept (with $p \leq 0.01$). Every edge is tagged with the unstandardized coefficient of the linear relationship between both participating variables, and with the corresponding standardized coefficient in parenthesis. The unstandardized coefficient is impacted by the difference in the relative scale of the variables, while the standardized coefficient is independent of the scale and represents by how many

standard deviations the end variable varies when the regressor variable increases by one standard deviation.

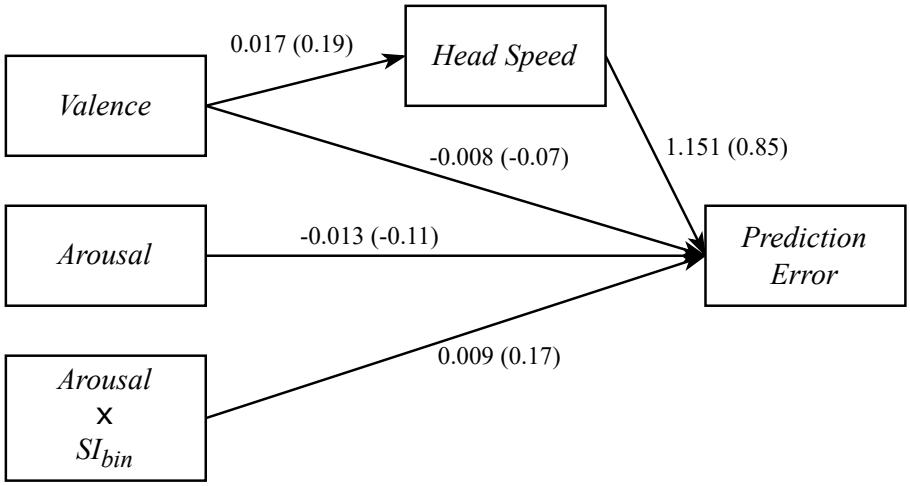


Figure *5.17: Structural equation model (SEM) describing the direct and indirect effects of user Valence and Arousal onto Prediction error, mediated by Head speed and moderated by video measure* SI_{bin}.

First, the model shows that the major impact of *Valence* on *Prediction error* is mediated by *Head speed*. Indeed, the standardized indirect effect of *Valence* on *Prediction error* is 0.19 $\times$ 0.85 = 0.16, while the direct effect is only -0.07. (H3) is therefore validated. This suggests that users rating the video with higher *Valence* tend to move more. This connects with the results obtained by B. J. Li et al. (2017). Second, the model confirms that the prediction error varies inversely with *Arousal*, with a standardized linear coefficient of -0.11. Third, the interaction effect is significant. Indeed, for a high *SI* video with SI_{bin} = 1, the total effect of *Arousal* on *Prediction error* is -0.11 + 0.17 $\times$ 1 = 0.06. In this case, the effect of *Arousal* on *Prediction error* is low and not significantly negative. However, for a low *SI* video with SI_{bin} = -1, the total effect of *Arousal* on *Prediction error* is -0.11 + 0.17 $\times$ -1 = -0.28. SI_{bin} is therefore a strong moderator of the effect of *Arousal* on *Prediction error*. To interpret this result, one may investigate how SI_{bin} connects with video categories, such as those proposed by Almquist et al. (2018). We may think that a high SI_{bin} describes videos with numerous salient areas in frames, hence yielding more exploratory head movements difficult to predict, even though the person rates their arousal/involvement in the video as high. Finally, this last result means that the video feature SI is a strong confounding factor which must be taken into consideration when one chooses 360° videos to investigate the impact of user emotion on motion prediction.

5.6.5 Discussion

The results we just presented open the path to promising directions to understanding the human motion process in immersive environments, as detailed in Sec. 5.7. Let us mention here some limits and perspectives of the presented data analysis.

First, obtaining results on more than 14 videos will be important for generalization, and to avoid possible spurious correlation between arousal and valence ratings that may impact our findings. Second, the main outcome variable considered here being the prediction error, the results may depend on the type of predictor considered. While, for the reasons described in Sec. 5.6.1.2, we verified that the results were similar between both methods taken from Romero Rondón et al. (2021), other families of approaches might lead to different effects of emotion on predictability. Third, we have considered head motion in this work, but it would be important to identify how the effects of emotions differ when predicting eye motion. More generally, while we have focused on only three types of user-centric measures (arousal, valence and head motion speed) and two types of video-centric measures (SI and TI), it will be most interesting to generalize this approach to more user-centric measures such as electrodermal activity and gaze, and video-centric measures such as video categories (focus or exploration (Almquist et al., 2018), fear or happiness (Jicol et al., 2021), possibly relating SI and TI to these).

5.7 Conclusion

In this chapter, we have presented three contributions stemming from user experiments aiming at better understanding the link between immersive content, attention, emotion, and movements in virtual reality.

We have presented the new PEM360 dataset of 360° videos with continuous physiological measurements, subjective emotional ratings and user motion traces. The stimuli are selected to enable investigating the spatiotemporal connection between user attention, user emotions and visual content. We have described the data collection process, the preprocessing workflow of the different data modalities, and exemplified some possible novel types of analyses to demonstrate the potential insights that can be drawn from PEM360. The artifacts are made available in a reproducible framework based on notebooks.

We have presented a first analysis on the impact of emotions on the accuracy of saliency estimators. We have measured the effect of user arousal (both physiologically and subjectively measured) on two types of saliency maps, high-level (HL) saliency and low-level (LL) saliency. We have showed that the accuracy of HL saliency increases when user arousal increases, while the accuracy of LL saliency is not affected.

We have also presented a first investigation into the effect of emotion on head motion predictability. We considered two datasets totalling 14 videos and 63 users, providing head motion traces and arousal and valence subjective ratings. Through hypothesis testing and structural equation modeling, we have shown that the predictability of head motion increases with arousal but decreases with valence, that the effect of valence on pre-

dictability is mediated by head speed, and that video SI interacts in the effect of arousal on predictability, a high SI moderating the effect.

This work opens the way to better understand factors impacting the human motion and their effect on the performance of head motion predictors, and how such knowledge can be leveraged to improve prediction. This can be done by augmenting the datasets with videos with specific emotional and visual features where head motion prediction is harder, or by designing ancillary training losses where a deep neural model would have to learn how to predict the user emotional state from the video content and the user's past motion. We explore this kind emotion-aware viewport prediction approach in chapter 6. An important outcome of this work is also to estimate the motion predictability from user emotional state. Such an estimation of the confidence of head motion prediction can readily be leveraged in the optimization of a 360° streaming system, even more so if the user emotional state is estimated with lightweight non-invasive device such as finger straps to measure electrodermal activity.

CHAPTER 6

Learning from emotions to improve viewport prediction

Predicting the viewport in virtual environments improves the quality of experience but remains an open challenge, as many factors can influence human attention and movements in VR.
With recent datasets collecting more diverse data modalities beyond the immersive content, such as emotional data, deep learning algorithms might be able to learn from these new modalities to improve viewport prediction.
In this chapter, we present a work in progress to build a modular multimodal deep architecture for viewport prediction, with the objective to learn transferable representations of these diverse modalities. We propose a general architecture and provide leads for upcoming developments.
Preliminary results without additional modalities show that the proposed architecture outperforms its competitors by up to 21%, with 150x fewer parameters.

Contents

6.1 Introduction

Being able to predict human attention and movements in virtual environments is key to maximizing the quality of experience, as not everything can be downloaded or rendered in high quality, as seen in the previous chapters. For this reason, many approaches to viewport prediction have been developed (see Sec. 2.2.1), including our own, DVMS, presented in chapter 3. Recent approaches to viewport prediction (Chao et al., 2021; Y. Xu, Zhang, & Gao, 2022) typically use deep learning models and learn to predict future head trajectories from past known user head positions as well as information from the immersive content itself (such as saliency maps or optical flow).

However, as seen in Sec. 5.6, emotions can significantly influence human behavior and predictability. Understanding factors that influence human attention, and thus movements within virtual environments is still an open challenge. This has led to the collection of new multimodal datasets: beyond navigational data, more dimensions of the VR experience are captured such as emotions and physiological signals. Examples include CEAP-360VR (Xue, Ali, Zhang, et al., 2021) and PEM360, the dataset we collected in described in Sec. 5.4. To our knowledge, deep learning models have not yet considered these additional data modalities for viewport prediction.

In this chapter, we investigate the following question: *How can we learn from the additional modalities available in limited datasets, such as emotions, and take advantage of this knowledge when deploying in environments where we do not have access to these modalities?*

Following recent trends in multimodal and cross-modal representation learning (Radford et al., 2021; Singh et al., 2022), we present a work in progress with the first steps towards building a modular multimodal deep architecture for viewport prediction, able to learn transferable cross-modal representations from jointly training on multiple modalities, and that can be easily incremented with additional modalities. Our contributions are:

- A new modular, efficient, multimodal architecture for viewport prediction that can be incremented to learn from more modalities.
- Early results comparing to the current state-of-the-art methods for online viewport prediction in 360° videos,
- A glance at the upcoming developments to integrate emotions and enchance our architecture.

Part of the work and ideas presented in this chapter are the outcome of a 3-month research stay at Centrum Wiskunde & Informatica (CWI, national research institute for mathematics and computer science in the Netherlands), carried out in collaboration with Dr. Silvia Rossi, Dr Irene Viola, and Prof. Pablo Cesar. Part of the work presented in this chapter was the object of a short paper submission at the 2nd computer vision for Metaverse workshop (CV4Metaverse) 2023, co-located with the 2023 International

Conference on Computer Vision (ICCV), and was not accepted. This work is still ongoing and will be the object of future submissions at other venues.

We first provide a background on cross-modal learning in Sec. 6.2. We then detail our motivations and introduce our proposed architecture in Sec. 6.3. We show the results achieved by our model in Sec. 6.4. We cast a glance at the upcoming developments and improvements of this work in Sec. 6.5. Finally, we conclude the chapter in Sec. 6.6.

6.2 Background on cross-modal learning

Benefiting from additional modalities to improve a unimodal task falls into the scope of cross-modal learning (Xing, Rostamzadeh, Oreshkin, & O. Pinheiro, 2019; Mu, Liang, & Goodman, 2020), a subfield of multimodal machine learning. Several approaches to cross-modal learning have been proposed in which we distinguish two families discussed in this section.

6.2.1 Cross-modal knowledge distillation

In the realm of supervised learning, knowledge distillation (KD) aims to transfer expertise from a large, proficient teacher network to a small, efficient, less robust student network. KD can be broadly categorized into three types: response-based (Hinton, Vinyals, & Dean, 2015; G. Chen, Choi, Yu, Han, & Chandraker, 2017; Furlanello, Lipton, Tschannen, Itti, & Anandkumar, 2018; Guo et al., 2020; Mirzadeh et al., 2020; Beyer et al., 2022; B. Zhao, Cui, Song, Qiu, & Liang, 2022), representation-based (Romero et al., 2015; Z. Huang & Wang, 2017; Zagoruyko & Komodakis, 2017; J. Kim, Park, & Kwak, 2018; Heo et al., 2019; T. Wang, Yuan, Zhang, & Feng, 2019; Passban, Wu, Rezagholizadeh, & Liu, 2021; Shu, Liu, Gao, Yan, & Shen, 2021; Baevski et al., 2022), and relation-based (Passalis & Tefas, 2018; Y. Liu et al., 2019; W. Park, Kim, Lu, & Cho, 2019; Tung & Mori, 2019). Response-based methods focus on matching softened logits, encouraging the student to produce similar predictions to those of the teacher. Representation-based methods focus on aligning features in the latent space. Relation-based methods focus on matching inter-sample relationships. Each method addresses different aspects of the knowledge transfer process, catering to the nuances of the data and the learning task.

With cross-modal knowledge distillation, the objective is to transfer knowledge from a superior modality (teacher) to an inferior modality (student). Here, the teacher network has access to a modality that is more potent or comprehensive compared to what the student possesses (Stroud, Ross, Sun, Deng, & Sukthankar, 2020; M. Hu et al., 2020; L. Zhao, Peng, Chen, Kapadia, & Metaxas, 2020; Ren, Du, Lv, Han, & He, 2021; X. Li, Lei, Sun, & Kuang, 2022). Several noteworthy examples of this approach include contrastive representation distillation (CRD) (Y. Tian, Krishnan, & Isola, 2020), compositional contrastive learning (CCL) (Y. Chen, Xian, Koepke, Shan, & Akata, 2021), and complementary relation contrastive distillation (CRCD) (Zhu et al., 2021). These methods exemplify the principle of leveraging the teacher's expertise in a superior modality

to guide and enhance the learning of the student in a different modality, showcasing the versatility and applicability of cross-modal knowledge transfer techniques in various domains, such as video action recognition (Stroud et al., 2020), brain tumor segmentation (M. Hu et al., 2020), 3D hand pose estimation (L. Zhao et al., 2020), lip reading (Ren et al., 2021), and land cover classification (X. Li et al., 2022).

6.2.2 Multimodal learning with missing modalities

In the context of multimodal learning, inputs from various modalities such as like images, texts, or audio are harnessed conjointly to convey a shared concept. In recent years, multimodal transformers have emerged as general-purpose models capable of processing inputs from diverse modalities. These models fuse multimodal inputs through token concatenation, eliminating the need for modality-specific feature extractors. They have found widespread applications across various multimodal tasks (Botach, Zheltonozhskii, & Baskin, 2022; Gabeur, Sun, Alahari, & Schmid, 2020; W. Kim, Son, & Kim, 2021; J. Li et al., 2021). However, their effectiveness is compromised when a modality is absent, leading to inaccurate predictions (M. Ma, Ren, Zhao, Testuggine, & Peng, 2022). Addressing this issue, recent research endeavors (M. Ma et al., 2021; J. Zhao, Li, & Jin, 2021; M. Ma et al., 2022; J. Zeng, Liu, & Zhou, 2022) have focused on developing robust multimodal models capable of accommodating missing modalities, ensuring the integrity of multimodal fusion and enhancing prediction accuracy.

Recently, Wei, Luo, and Luo (2023) proposed to categorize incomplete multimodal learning methods in two types: customized methods and unified methods. Customized methods need specific models to recover missing modalities within each incomplete modality combination. These methods are further categorized into sample-based customized methods, which reconstruct missing modalities at the input space using GANs (Cai, Wang, Gao, Shen, & Ji, 2018; Jue et al., 2019; Pan, Liu, Lian, Xia, & Shen, 2020; A. Liu et al., 2021; Y. Wang et al., 2021) and customized representation-based customized methods, which reconstruct missing modalities through VAEs knowledge distillation (Hoffman, Gupta, & Darrell, 2016; Garcia, Morerio, & Murino, 2018; Stroud et al., 2020) or matrix completion (Lin et al., 2021; J. Liu et al., 2021). While these methods have demonstrated promising results, they necessitate training and deploying a distinct model for each subset of missing modalities, significantly increasing complexity in practical applications. To decrease the complexity, unified methods, which aim to train one model to deal with different incomplete modality combinations, were developed. The proposed methods reach this objective by extracting modality-invariant features (Havaei, Guizard, Chapados, & Bengio, 2016; Chartsias, Joyce, Giuffrida, & Tsaftaris, 2018; Q. Yin, Wu, & Wang, 2017; van Tulder & de Bruijne, 2019; Dorent, Joutard, Modat, Ourselin, & Vercauteren, 2019; M. Ma et al., 2021; T. Zhou, Canu, Vera, & Ruan, 2020; Ding, Yu, & Yang, 2021; Y. Zhang et al., 2022).

While not explicitly accounting for missing modalities, multimodal foundation models (Radford et al., 2021; Jia et al., 2021; J. Li et al., 2021) learn from large scale image-text data with various multimodal pre-training objectives and can be transferred to a wide

range of downstream tasks, even without explicit supervised fine-tuning. Recently, Singh et al. (2022) showed that we can build modular multimodal foundation models that can handle unimodal, cross-modal as well as multimodal tasks, by combining transformer-based encoders and pre-training them accordingly. More importantly, they showed that pre-training on more than one modality allowed to learn better joint representations and obtain better results on unimodal tasks.

Following the findings of Singh et al. (2022), we propose to build a transformer-based modular multimodal model for viewport prediction, which learns unified representations, without the need of customized models or distillation losses.

6.3 A first proposal of a new modular multimodal architecture for viewport prediction

6.3.1 Problem definition

The problem that we consider is the prediction of the future viewport in 360° videos. This is the same problem as defined in Sec. 3.3.2. We shortly remind the reader of the prediction problem in this section. We define P_t to be the center of the viewport of a user watching a 360° video at time t, that we also call the *head position* at time t. We define M to be the past window of positions that can be used to make a prediction. We also assume that the entire content of the video is available and can be used to make predictions.

When doing an *online* prediction of the viewport, we want to output a prediction $\hat{P}_{t+1:t+H}$ of the future head positions $P_{t+1:t+H}$ between t and $t+H$, where H is the prediction horizon.

The prediction problem can be formulated as a minimization of the distance $d(\hat{P}_{t+1:t+H}, P_{t+1:t+H})$, the distance between the predicted future positions and the ground-truth future positions.

6.3.2 Motivation

The objectives of our architecture are to obtain cross-modal representations that are transferable to domains with fewer modalities, and to be easily incrementable with new modalities. We present a conceptual diagram of this architecture in Fig. 6.1. The blue boxes and arrows correspond to elements that have been implemented and for which we provide results in Sec. 6.4.3. Orange boxes include modalities that may not be available at test time, but from which we may be able to learn useful transferable representations. Additional modalities may include, e.g., interaction logs in interactive virtual environments.

The advantages of such an architecture are numerous:

- We can pre-train single modalities independently, which makes training from noisy samples easier.

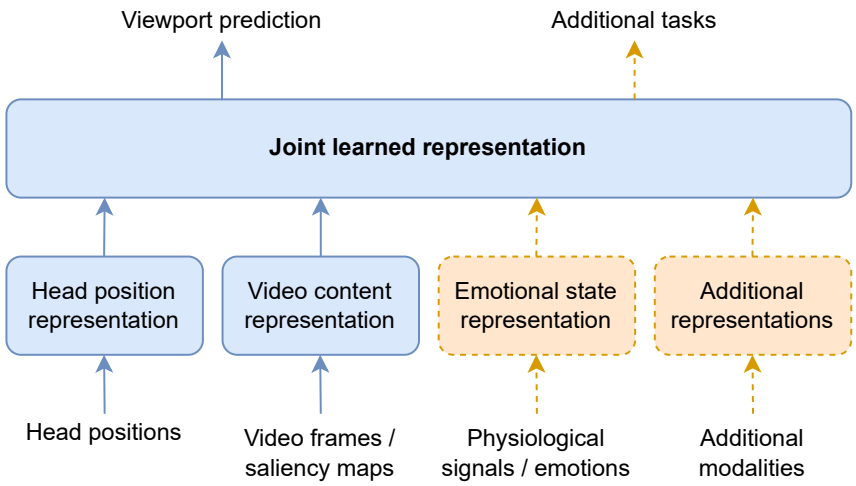


Figure *6.1: Conceptual diagram of a modular multimodal architecture for viewport prediction. Each small box is a learned cross-modal representation of a single modality.*

- Given enough data, We can pre-train on multiple modalities, and the learned cross-modal representations will be transferable to smaller versions of the model with fewer modalities.
- We can pre-train on a variety of tasks other than viewport prediction, such as masked auto-encoding, which has been shown to be a good pre-training objective that generalizes well to a wide range of downstream tasks (K. He et al., 2022; Bachmann, Mizrahi, Atanov, & Zamir, 2022; Tong, Song, Wang, & Wang, 2022).

6.3.3 Proposed Architecture

Several elements of our architecture are inspired from a transformer-based architecture, *Perceiver* (Jaegle et al., 2021). This decision was motivated by the fact that, despite its success, the self-attention mechanism of the transformer model scales quadratically with the size of the inputs, which can lead to model training and inference being very computationally demanding. The attention bottleneck introduced by the cross-attention module of Perceiver allows for more efficient models, eliminating the quadratic scaling problem. Multiple extensions of Perceiver have been shown to work with many kinds of inputs and outputs (Jaegle et al., 2022; Hawthorne et al., 2022; Z. Tang et al., 2023), making it a good choice for a general-purpose modular architecture.

We present an implementation diagram of the proposed architecture in Fig. 6.2. As this is an ongoing work, the version of the architecture that we propose only uses the head positions and video content as multimodal inputs. The architecture is divided in four modules explained below.

Position encoder We choose to represent head positions with 3D Cartesian coordinates of points on the unit sphere, following DVMS in chapter 3 and previous work (Romero Rondón et al., 2021; Chao et al., 2021). While recent deep models that only use

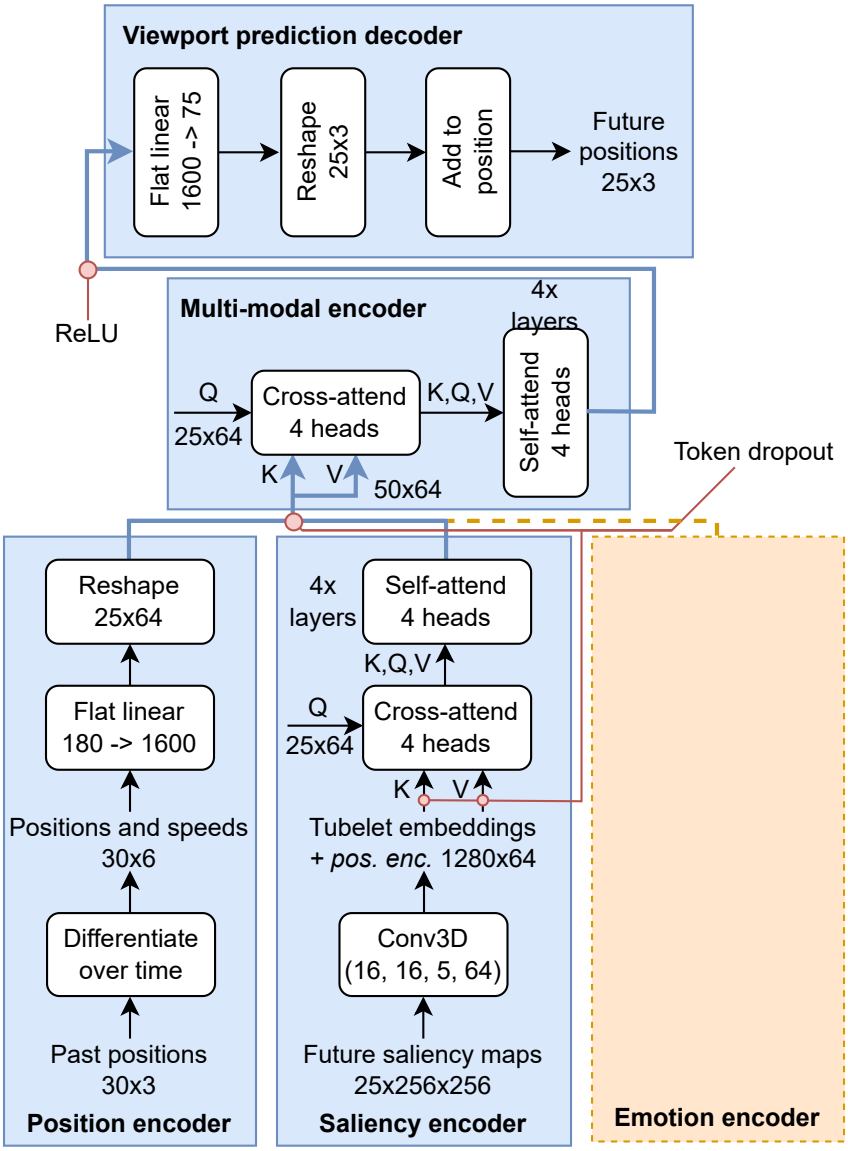


Figure 6.2: *Diagram of the proposed modular multimodal architecture for viewport prediction. We use the same attention blocks as in Perceiver IO (Jaegle et al., 2022). Q denotes fixed queries defined in Sec. 6.3.3. The shape of the arrays are given as an example of prediction setting where we would consider $M = 6$ seconds of past positions to predict the future viewport over $H = 5$ seconds, with a sampling rate of 5 Hz. In this example, 25 future saliency maps of 256 x 256 pixels are given to the saliency encoder.*

the past head positions to predict the future viewport have used RNN-based (DVMS) and transformer-based architectures (Chao et al., 2021), novel work on time series forecasting shows that a simple linear baseline can outperform most transformer models (A. Zeng et al., 2023). Inspired from this work, we have designed a simple multilayer perceptron (MLP) baseline that we detail in Fig. 6.3 and explain in Sec. 6.4.1. This baseline benefits from the direct multi-step (DMS) forecasting strategy described by A. Zeng et al. (2023). From these results, we design a simple linear encoder for the positions described in Fig.

6.2. We first temporally differentiate the positions over the past M seconds to get the speed vectors, and a flattened array containing all the past positions and speeds between $t - M$ and t is fed to a linear layer. The output of this linear layer is then reshaped into tokens that will go into the multimodal encoder. We did not find any gain to using the auto-regressive framework of Perceiver AR (Hawthorne et al., 2022) for our problem.

Saliency encoder For this model, we decide to work with pre-extracted saliency maps from PanoSalNet (Nguyen et al., 2018), following TRACK (Romero Rondón et al., 2021). Following recent work to adapt the vision transformer (ViT (Dosovitskiy et al., 2021)) to videos (Arnab et al., 2021; Tong et al., 2022), we use a shallow 3D convolutional layer to obtain tubelet embeddings, corresponding to spatio-temporal patches of the sequence of saliency maps. Specifically, we use temporally dilated tubelets, where each embedding containing information about evenly-spaced, non-consecutive saliency maps. A sinusoidal position encoding is added to each embedding before being projected into a smaller latent space by the Perceiver-inspired cross-attention module, followed by several self-attention layers. The size of this latent space is determined by the size of the latent query Q, that we initialize with sinusoidal position encodings. The cross-attention module is followed by several layers of multi-head self-attention, which does not scale quadratically with the inputs, but with the size of the initial latent query, thanks to the upstream cross-attention module. During training, we randomly replace 50% of the tokens with zeros, labeled as "token dropout" in Fig. 6.2. This greatly reduced overfitting in our case.

Multimodal encoder To obtain a joint representation of the encoded modalities, we take full advantage of the cross-attention and iterative latent self-attention modules of Perceiver. The tokens originating from the unimodal encoding modules are concatenated and projected into a smaller latent space by another Perceiver-inspired cross-attention module. The cross-attention module is again followed by several layers of multi-head self-attention to obtain a final latent joint representation of our multimodal inputs. We found that using token dropout on the tokens before the cross-attention module also helped to reduce overfitting.

Viewport prediction decoder The design of the viewport prediction decoder is also motivated by our findings with the MLP baseline model (see Sec. 6.4.1). The decoder is detailed in Fig. 6.2. The tokens of the final latent joint representation of the multimodal inputs are flattened into a one-dimensional array and go through a ReLU activation before being fed to a linear layer. The output of the linear layer is reshaped into a $S \cdot H$ x 3 array, with S being the sampling rate of the head motion traces, in Hertz, and H the prediction horizon, in seconds. A residual connection not shown in the diagram adds this array to the last known position, the model therefore only predicting the displacement.

Additional modality encoders, such as the emotion encoder, will be the subject of upcoming work, briefly discussed in Sec. 6.5.

6.4 Comparison of viewport prediction methods

The results in this section are obtained from our experiments, except for VPT360, because the code was not available at the time of writing. We follow the online viewport prediction setting described in Sec. 6.3.1, and we set H to 5 seconds.

6.4.1 Compared models

***Trivial-static* baseline** is a trivial baseline already shown to outperform previous viewport prediction architectures by Romero Rondón et al. (2021). The predicted head positions are equal to the last known head position.

VPT360 (Chao et al., 2021) is a transformer-based deep learning model which only uses the past positions for viewport prediction. As the code is not available, we are only able to report results on the MMSys18 dataset, which are reported in their paper. They set $M = 1$ second, which was reported to be the optimal value in their case.

DVMS-1 is our implementation of a DVMS-based architecture, discussed in chapter 3. It is a GRU-based sequence-to-sequence deep learning model which only uses the past positions to predict K multiple trajectories of head positions. It is here used with $K = 1$ to predict one trajectory. We set $M = 5$ seconds. We found this value to be optimal, but the differences between $M = 1$ and $M = 5$ were marginal (less than 0.5% difference in error).

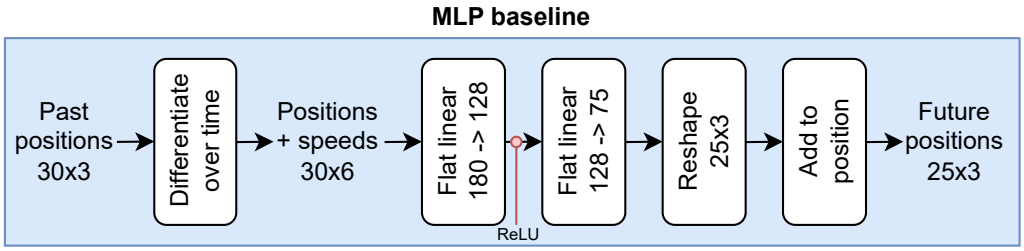


Figure *6.3: Diagram of the MLP baseline model. The shapes of the arrays follow the same prediction setting as in Fig. 6.2.*

MLP baseline is a new baseline that we propose. We provide a diagram in Fig. 6.3. This baseline benefits from the direct multi-step (DMS) forecasting strategy described by A. Zeng et al. (2023). We first temporally differentiate the positions to get the speed vectors, and a flattened array containing all the past positions and speeds between $t - M$ and t is fed to a linear layer with 128 units. After a ReLU activation, this flat representation is fed to another linear layer with 75 units, which is then reshaped into an array of shape 25x3. These values are added to the last known position to directly predict the next 25 positions. We set M to 6 seconds, which we found to be optimal, even though the difference in error between $M = 1, 2, 3, 4, 5, 6$ was marginal (less than 0.5%).

TRACK (Romero Rondón et al., 2021) is an LSTM-based sequence-to-sequence architecture that simultaneously processes head positions and saliency maps with separate

LSTM cells, before fusing them with a shared LSTM cell, at each time step. Two versions were proposed: using content-based saliency maps extracted from the content with PanoSalNet (Nguyen et al., 2018), and using ground-truth maps generated from the head motion traces. We use $M = 1$ second, which was reported to be the optimal value in their case.

The **proposed architecture**, explained in Sec. 6.3.3, is described in detail in Fig. 6.2. We use the same PanoSalNet-extracted (Nguyen et al., 2018) saliency maps as TRACK, for fairness. We set M to 6 seconds, following our findings with the MLP baseline.

6.4.2 Experimental settings

Datasets

We test all the models on six datasets containing head motion traces of people watching 360° videos. Four of these datasets, CVPR18 (Y. Xu et al., 2018), MMSys18 (David et al., 2018), PAMI18 (M. Xu, Song, et al., 2019), and MM18 (Nguyen et al., 2018) were taken from the reproducible framework of Romero Rondón et al. (2020), and we use the same train and test sets as they did.

We also test the considered methods on two recent datasets that collected additional modalities, emotions and physiological signals, and have not yet been used in the context of viewport prediction, CEAP-360VR (Xue, Ali, Zhang, et al., 2021) and PEM360, our collected dataset discussed in chapter 5. The train and test splits for these datasets were chosen with the objective to have a balanced training and testing sets in terms of content metrics (i.e., spatial information and temporal information (ITU-T P.910, 2021)) and emotional ratings (valence and arousal). Information about the train/test split will be made available along with the code. Future increments of our architecture will benefit from the additional modalities collected in these datasets.

Training settings All models, apart from VPT360, were trained end-to-end with the task to predict 5 seconds of future head positions on the train set of our largest dataset, CVPR18, and tested with the same task on the test set of every dataset without fine-tuning. Our experiments have shown that it yields better performance than training on individual datasets. All the training settings and hyper-parameters will be made available along with the code.

Metric We use the great-circle distance (gcd) to measure the distance between two head positions. The great-circle distance can be defined as a function of the Euclidean distance of two points P^1 and P^2 on the unit sphere, as shown in Eq. 6.1.

$$\text{gcd}(P^1, P^2) = 2 \cdot \arcsin\left(\frac{\|P^1 - P^2\|_2}{2}\right) \tag{6.1}$$

We report the average displacement error (ADE) between generated trajectories and ground-truth trajectories. The ADE between two head motion trajectories P^1 and P^2 of length T is defined in Eq. 6.2, where gcd is the great-circle distance.

$$\text{ADE}(T^1, T^2) = \frac{1}{L} \sum_{t \in 1:L} \text{gcd}(T_t^1, T_t^2) \tag{6.2}$$

6.4.3 Results

Results are presented in Table 6.1.

*Table 6.1: Average displacement error (great-circle distance, lower is better) for all models divided in three categories depending on their input modalities, between $t + 0.2$ and $t + H$, with H in seconds. Best results for each modality are in **bold** only if they improve over models that use fewer modalities.*

			CVPR18		MMSys18		PAMI18		MM18		PEM360		CEAP-360VR	
	Model	Params	$H=1$	$H=5$	$H=1$	$H=5$	$H=1$	$H=5$	$H=1$	$H=5$	$H=1$	$H=5$	$H=1$	$H=5$
	Trivial-static	0	0.263	0.672	0.322	0.883	0.169	0.439	0.190	**0.530**	0.313	0.889	0.177	0.548
Only past	VPT360	6.3M	-	-	0.239	0.809	-	-	-	-	-	-	-	-
positions	DVMS-**1**	110k	0.199	0.613	**0.222**	**0.773**	0.133	0.386	0.152	0.561	0.231	**0.850**	0.119	0.471
	MLP	33k	**0.197**	**0.612**	**0.222**	0.783	**0.131**	**0.384**	**0.147**	0.556	**0.228**	0.858	**0.117**	**0.470**
Content-based	TRACK	172M	0.197	0.621	0.223	0.801	0.132	0.401	0.152	0.654	0.231	0.884	0.119	0.477
saliency + pos	Proposed	1.1M	0.201	0.621	0.227	0.815	0.133	0.386	0.149	0.545	0.232	0.876	0.122	0.470
Ground-truth	TRACK	136M	0.211	0.564	0.238	0.700	0.160	0.396	0.182	0.499	0.249	0.801	0.144	0.500
saliency + pos	Proposed	991K	**0.192**	**0.548**	**0.216**	**0.693**	**0.129**	**0.367**	**0.144**	**0.451**	**0.224**	**0.752**	0.118	**0.459**

When only using the past positions, we observe that DVMS and the MLP baseline have very close performance, with the MLP baseline slightly outperforming DVMS in most cases, with 3x fewer parameters. The MLP baseline matches or outperforms all competitors, except on long prediction on the MM18 dataset, where the *trivial-static* baseline performs best.

When integrating the content-based saliency maps into the model, we can see that the performance is always degraded on average. However, the proposed model is usually on par with TRACK (1.3% better on average) with more than 150x fewer parameters.

When integrating the ground-truth saliency maps into the model, the proposed model consistently outperforms TRACK (by 10.1% on average, up to 20.9%) and also improves over the baselines that only use past positions. Long-term (5 seconds) prediction is improved on all the datasets, and short-term (1 second) prediction is never degraded, and even slightly improved on 5 of the 6 datasets.

6.4.4 Interpretations and discussion

It was shown that integrating content-based saliency maps can lead to viewport prediction improvements on certain types of video, but it led to a degradation of performance on average in our case. Romero Rondón et al. (2021) have shown with TRACK that it can be very challenging to use the video content to make predictions of the viewport, especially because we can build strong baselines only using the positions. First, the performance of the models that use content-based saliency is also dependent on the accuracy of saliency extractors, and results should improve when using a more accurate saliency extractor. Second, the attention of the user is not always synchronized with the video content, depending, among other things, on their emotional state. Adding information about the user

emotional state during training might help solving the problem of content synchronization and improve the predictions, and this will be the subject of our future work to increment this architecture.

6.5 Upcoming developments

While the preliminary results we just presented are encouraging, this work is still ongoing. In this section, we provide some elements about upcoming developments.

6.5.1 Potential improvements

We list some ideas that can be explored to improve the model:

- Saliency maps: content-based saliency maps degrade the performance, while ground-truth saliency maps significantly improve over the "position-only" setting. A first idea could be to use better saliency, as we are using the same PanoSalNet (Nguyen et al., 2018) saliency maps as TRACK (Romero Rondón et al., 2021). This idea was partially explored by using saliency maps from PAVER (Yun et al., 2022), but the results were nearly identical. A second idea could be to facilitate training by progressively replacing ground-truth saliency maps with content-based saliency maps, using a process like morphing (Vallez, Bueno, Deniz, & Blanco, 2022).
- Pre-training tasks: the only task on which the model was trained and tested was viewport prediction. Masked modeling of the head trajectories could be explored. These ideas are currently being investigated for trajectory prediction (H. Chen et al., 2023; P. Wu et al., 2023).
- Positional encodings: in this first version, sinusoidal positional encodings are added to data modalities, but also used as cross-modal queries. Instead of using fixed positional encodings, learned embeddings are a possibility (Carion et al., 2020).

6.5.2 Integrating emotional data

As stated in Sec. 6.1, our objective was to "learn from the additional modalities available in limited datasets and take advantage of this knowledge when deploying in environments where we do not have access to these modalities". Specifically, this would mean using emotional data during training, but not in testing. For the model to be able to take advantage of the additional training knowledge that comes with emotional data, it needs to learn how to better exploit the data from the other modalities. This is what happens in foundation models like FLAVA (Singh et al., 2022), where the multimodal pre-training objective allows to learn a better representation from multiple input modalities, which is conserved in unimodal downstream tasks.

This is only possible if there are exploitable correlations between the modalities that would not be exploited otherwise. For this reason, we need *rich* representations of the

input modalities. We argue that content-based saliency maps do not fit this definition of *rich* representation. For this reason, simply having emotional data as an additional input could improve the performance in training, but would likely degrade in testing. In order to add emotional data, the next step will be to integrate *rich* representations of the video content instead of saliency maps. Features extracted from the video by a pre-trained foundation model are an example of *rich* representation of the content that we could use. Ideas for the integration of emotional data include an additional emotion encoder that could be inspired from the position encoder, because the emotional data is present in the form of time series in the datasets we consider.

6.6 Conclusion

In this chapter, we have presented ongoing work about a new modular multimodal deep architecture for viewport prediction, motivated by the need for a model that can learn from additional modalities, such as emotions, and transfer this knowledge in a context where these additional modalities may not be available.

Early results combining past positions and future saliency maps show that our model can outperform the existing state-of-the-art by up to 21%, while dividing the number of parameters by more than 150. Our architecture can easily be incremented with new modalities, which will be the subject of future work.

We have given several leads for future developments and potential improvements of this architecture.

This architecture establishes a first step towards an architecture able to learn transferable cross-modal representations from jointly training on multiple modalities in the context of viewport prediction in VR.

CHAPTER 7

Conclusion and perspectives

7.1 Conclusion

Returning to the initial objectives of the work presented in this manuscript, we contributed to addressing the challenges that arise when predicting the user's viewport in VR. We worked on deep learning approaches for the design of new VR streaming systems that improve the quality of experience and can better adapt to each user.

In chapter 3, to consider the randomness and diversity of human motion when predicting head movements based on past head trajectories, we presented the first method for multiple head motion prediction in 360° videos. Our main contribution is a new learning framework, called DVMS, which builds on deep latent variable models and allows to predict multiple future trajectories from a given past. DVMS provides a training procedure to obtain a flexible and lightweight stochastic prediction model compatible with sequence-to-sequence architectures. We assessed DVMS on 4 datasets and showed that it outperforms competitors adapted from the self-driving domain by up to 41%, on prediction horizons up to 5 seconds. We analyzed the latent space of our model and showed that the stationarity of the prediction error enabled easy likelihood estimation of the trajectories, enabling direct integration in streaming optimization. DVMS paves the way for multiple head motion prediction in 360° videos.

In chapter 4, to evaluate the system gains of DVMS and address the reproducibility issue of viewport-adaptive streaming algorithms, we proposed SMART360, a new trace-driven simulation environment that enables new comparisons different motion prediction and adaptive bitrate strategies with numerous metrics and graphical visualizations. SMART360 overcomes the drawbacks of the few existing alternative tools by providing highly-configurable code, with many inputs and settings, and offers a more realistic streaming behavior. We described the structure of SMART360 and explained how new motion predictors and adaptive bitrate algorithms can be implemented inside the simulation environment to be evaluated and compared. Thanks to SMART360, we were able to deploy an extensive system evaluation of our proposed DVMS framework, considering four different datasets of user, video and network bandwidth traces. We showed that predicting multiple trajectories yields a higher fairness between the traces. We also showed that predicting the ideal number of trajectories led to visual quality gains up to 10% for

20% to 30% of the users. We believe that SMART360 can improve the reproducibility of research regarding 360° video viewport-adaptive streaming algorithms, and make future comparisons of new strategies easier for researchers.

In chapter 5, to investigate the relationship between immersive content, attention, emotion, and movements in virtual reality, we collected PEM360, a new dataset with head motion traces, gaze scanpaths, physiological measurements, and subjective emotional ratings of people watching 360° videos. We made PEM360 publicly available. We presented a first analysis on the impact of emotions on the accuracy of saliency estimators. We measured the effect of user arousal (both physiologically and subjectively measured) on two types of saliency maps, high-level (HL) saliency and low-level (LL) saliency. We showed that the accuracy of HL saliency increases when user arousal increases, while the accuracy of LL saliency is not affected. We also presented a first investigation into the effects of emotions on head motion predictability. Through hypothesis testing and structural equation modeling, we showed that the predictability of head motion increased with arousal but decreased with valence, that the effect of valence on predictability was mediated by head speed, and that video SI interacted in the effect of arousal on predictability. This work opens the way to better understand factors impacting the human motion and their effect on the performance of head motion predictors, and how such knowledge can be leveraged to improve prediction. An important outcome of this work is also to estimate the motion predictability from user emotional state. Such an estimation of the confidence of head motion prediction can readily be leveraged in the optimization of a 360° streaming system.

In chapter 6, we presented ongoing work about a new modular multimodal deep architecture for viewport prediction, motivated by the need for a model that can learn from additional modalities, such as emotional data, and transfer this knowledge in a context where these additional modalities may not be available. Early results combining past positions and future saliency maps showed that our model can outperform the existing state-of-the-art by up to 21%, while dividing the number of parameters by more than 150. Our architecture can easily be incremented with new modalities, which will be the subject of future work. We gave several leads for future developments and potential improvements of this architecture. This architecture establishes a first step towards an architecture able to learn transferable cross-modal representations from jointly training on multiple modalities in the context of viewport prediction in VR.

7.2 Perspectives

Overall, our contributions open new exciting applications and research perspectives.

The DVMS framework for multiple trajectory prediction will easily be extended to other contexts. Transformer-based models (Chao et al., 2021), context-aware architectures that consider more than the past positions (Romero Rondón et al., 2021), short-term gaze prediction models (Mondal et al., 2023), and 6DoF motion prediction models for

interactive virtual environments (Zheng et al., 2022) are the most straightforward applicative extensions of multiple trajectory prediction with DVMS.

Stochastic models for head trajectory prediction and uncertainty quantification approaches, based on the user emotional state or with variational models such as discussed in Sec. 3.7, will enable new uncertainty-aware VR streaming systems with extra levers (Dambra et al., 2018; Sassatelli et al., 2020). Such a system might trigger a snap-change when head movements become unpredicable, for example.

Exciting new research also lies in multimodal learning for VR. We describe ongoing work and give elements about future developments of a new modular multimodal model for viewport prediction in chapter 6. Models with more modalities, such as emotions or spatial audio (Singla et al., 2023; Q. Yang et al., 2023), potentially modulating the importance of modalities over time, will continue to be investigated.

Finally, we reflect on the paradigm shift brought by foundation models (Bommasani et al., 2022), and the impact it will have on prediction models for virtual environments. These new large-scale models are trained on broad data in a self-supervised manner and can be adapted to a wide range of downstream tasks. Their scaling capabilities, their large number of parameters, and the massive amount of data they are trained on make them excellent at transfer learning. Simple adaptations of these models now outperform specialized fine-tuned models in many tasks (Brown et al., 2020; Radford et al., 2021; Singh et al., 2022; Zara et al., 2023), which encourages their use in many applications. While foundation models for trajectory prediction do not exist yet, foundation models for time series are starting to emerge (Garza & Mergenthaler-Canseco, 2023). Nevertheless, existing vision and language foundation models can already be used as feature extractors in directly applicable multimodal prediction contexts. While application of foundation models to VR prediction task may prove beneficial, we believe that it is important to consider the associated risks, as we still lack a clear understanding of how they work, when they fail, and what their biases are.

References

Adeli, V., Ehsanpour, M., Reid, I., Niebles, J. C., Savarese, S., Adeli, E., & Rezatofighi, H. (2021). TRiPOD: Human Trajectory and Pose Dynamics Forecasting in the Wild. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 13370–13380). doi: 10.1109/ICCV48922.2021.01314

Akhshabi, S., Anantakrishnan, L., Begen, A. C., & Dovrolis, C. (2012). What Happens When HTTP Adaptive Streaming Players Compete for Bandwidth? In *Proceedings of the 22nd International Workshop on Network and Operating System Support for Digital Audio and Video* (p. 9–14). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2229087.2229092

Akhshabi, S., Begen, A. C., & Dovrolis, C. (2011). An Experimental Evaluation of Rate-Adaptation Algorithms in Adaptive Streaming over HTTP. In *Proceedings of the Second Annual ACM Conference on Multimedia Systems* (p. 157–168). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/1943552.1943574

Akhtar, Z., Nam, Y. S., Govindan, R., Rao, S., Chen, J., Katz-Bassett, E., ... Zhang, H. (2018). Oboe: Auto-Tuning Video ABR Algorithms to Network Conditions. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication* (p. 44–58). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3230543.3230558

Akoglu, H. (2018). User's guide to correlation coefficients. *Turkish Journal of Emergency Medicine*, *18*(3), 91–93. doi: 10.1016/j.tjem.2018.08.001

Aladagli, A. D., Ekmekcioglu, E., Jarnikov, D., & Kondoz, A. (2017). Predicting head trajectories in 360° virtual reality videos. In *2017 International Conference on 3D Immersion (IC3D)* (pp. 1–6). doi: 10.1109/IC3D.2017.8251913

Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., & Savarese, S. (2016). Social LSTM: Human Trajectory Prediction in Crowded Spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 961–971). doi: 10.1109/CVPR.2016.110

Alemi, A., Poole, B., Fischer, I., Dillon, J., Saurous, R. A., & Murphy, K. (2018, 10–15 Jul). Fixing a Broken ELBO. In J. Dy & A. Krause (Eds.), *Proceedings of the 35th International Conference on Machine Learning* (Vol. 80, pp. 159–168). PMLR.

Alemi, A. A., Fischer, I., & Dillon, J. V. (2018). *Uncertainty in the Variational Information Bottleneck.* Retrieved from https://arxiv.org/abs/1807.00906

Alemi, A. A., Fischer, I., Dillon, J. V., & Murphy, K. (2017). Deep Variational Information Bottleneck. In *International Conference on Learning Representations.*

Almquist, M., Almquist, V., Krishnamoorthi, V., Carlsson, N., & Eager, D. (2018). The Prefetch Aggressiveness Tradeoff in 360° Video Streaming. In *Proceedings of the 9th ACM Multimedia Systems Conference* (p. 258–269). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3204949.3204970

Amirian, J., Hayet, J.-B., & Pettré, J. (2019). Social Ways: Learning Multi-Modal Distributions of Pedestrian Trajectories With GANs. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp. 2964–2972). doi: 10.1109/CVPRW.2019.00359

Anand, D., Togou, M. A., & Muntean, G.-M. (2021). A Machine Learning Solution for Automatic Network Selection to Enhance Quality of Service for Video Delivery. In *2021 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)* (pp. 1–5). doi: 10.1109/BMSB53066.2021.9547176

Ardizzone, L., Lüth, C., Kruse, J., Rother, C., & Köthe, U. (2019). *Guided Image Generation with Conditional Invertible Neural Networks.* Retrieved from https://arxiv.org/abs/1907.02392

Ariyo, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Stock Price Prediction Using the ARIMA Model. In *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation* (pp. 106–112). doi: 10.1109/UKSim.2014.67

Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., & Schmid, C. (2021). ViViT: A Video Vision Transformer. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 6816–6826). doi: 10.1109/ICCV48922.2021.00676

Artaud, A. (1938). *Le théâtre et son double*. Gallimard.

Azevedo, R. G. d. A., Birkbeck, N., De Simone, F., Janatra, I., Adsumilli, B., & Frossard, P. (2020). Visual Distortions in 360° Videos. *IEEE Transactions on Circuits and Systems for Video Technology*, *30*(8), 2524–2537. doi: 10.1109/TCSVT.2019.2927344

Babaeizadeh, M., Finn, C., Erhan, D., Campbell, R. H., & Levine, S. (2018). Stochastic Variational Video Prediction. In *International Conference on Learning Representations.*

Bachmann, R., Mizrahi, D., Atanov, A., & Zamir, A. (2022). MultiMAE: Multi-Modal Multi-Task Masked Autoencoders. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVII* (p. 348–367). Berlin, Heidelberg: Springer-Verlag. doi: 10.1007/978-3-031-19836-6_20

Baevski, A., Hsu, W.-N., Xu, Q., Babu, A., Gu, J., & Auli, M. (2022, 17–23 Jul). data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, & S. Sabato (Eds.), *Proceedings of the 39th International Conference on Machine Learning* (Vol. 162, pp. 1298–1312). PMLR.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). *Neural Machine Translation by Jointly Learning to Align and Translate.* Retrieved from https://arxiv.org/abs/1409.0473

Ban, Y., Xie, L., Xu, Z., Zhang, X., Guo, Z., & Wang, Y. (2018). CUB360: Exploiting Cross-Users Behaviors for Viewport Prediction in 360 Video Adaptive Streaming. In *2018 IEEE International Conference on Multimedia and Expo (ICME)* (pp. 1–6). doi: 10.1109/ICME.2018.8486606

Baños, R. M., Botella, C., Rubió, I., Quero, S., García-Palacios, A., & Alcañiz, M. (2008). Presence and Emotions in Virtual Environments: The Influence of Stereoscopy. *CyberPsychology & Behavior*, *11*(1), 1–8. (PMID: 18275306) doi: 10.1089/cpb.2007.9936

Bao, Y., Wu, H., Zhang, T., Ramli, A. A., & Liu, X. (2016). Shooting a moving target: Motion-prediction-based transmission for 360-degree videos. In *2016 IEEE International Conference on Big Data (Big Data)* (pp. 1161–1170). doi: 10.1109/BigData.2016.7840720

Barreda-Ángeles, M., Aleix-Guillaume, S., & Pereda-Baños, A. (2020). An "Empathy Machine" or a "Just-for-the-Fun-of-It" Machine? Effects of Immersion in Nonfiction 360-Video Stories on Empathy and Enjoyment. *Cyberpsychology, Behavior, and Social Networking*, *23*(10), 683–688. (PMID: 32716643) doi: 10.1089/cyber.2019.0665

Barrett, L. F. (1998). Discrete emotions or dimensions? The role of valence focus and arousal focus. *Cognition and Emotion*, *12*(4), 579–599. (Place: United Kingdom Publisher: Taylor & Francis) doi: 10.1080/026999398379574

Battisti, F., Baldoni, S., Brizzi, M., & Carli, M. (2018). A feature-based approach for saliency estimation of omni-directional images. *Signal Processing: Image Communication*, *69*, 53–59. (Salient360: Visual attention modeling for 360° Images) doi: 10.1016/j.image.2018.03.008

Battisti, F., & Carli, M. (2019). Depth-based saliency estimation for omnidirectional images. *Electronic Imaging*, *31*(11), 271–1–271–1. doi: 10.2352/ISSN.2470-1173.2019.11.IPAS-271

Bayer, J., & Osendorfer, C. (2015). *Learning Stochastic Recurrent Networks.* Retrieved from https://arxiv.org/abs/1411.7610

Bentaleb, A., Taani, B., Begen, A. C., Timmerer, C., & Zimmermann, R. (2019). A Survey on Bitrate Adaptation Schemes for Streaming Media Over HTTP. *IEEE Communications Surveys & Tutorials*, *21*(1), 562–585. doi: 10.1109/COMST.2018.2862938

Ben Yahia, M., Le Louedec, Y., Simon, G., & Nuaymi, L. (2018). HTTP/2-Based Streaming Solutions for Tiled Omnidirectional Videos. In *2018 IEEE International Symposium on Multimedia (ISM)* (pp. 89–96). doi: 10.1109/ISM.2018.00023

Berlincioni, L., Becattini, F., Seidenari, L., & Del Bimbo, A. (2021). Multiple Future Prediction Leveraging Synthetic Trajectories. In *2020 25th International Conference on Pattern Recognition (ICPR)* (pp. 6081–6088). doi: 10.1109/ICPR48806.2021.9412158

Beyer, L., Zhai, X., Royer, A., Markeeva, L., Anil, R., & Kolesnikov, A. (2022). Knowledge distillation: A good teacher is patient and consistent. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 10915–10924). doi: 10.1109/CVPR52688.2022.01065

Bhattacharyya, A., Schiele, B., & Fritz, M. (2018). Accurate and Diverse Sampling of Sequences Based on a "Best of Many" Sample Objective. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 8485–8493). doi: 10.1109/CVPR.2018.00885

Bishop, C. M. (1994). *Mixture density networks* [Technical Report]. Birmingham.

Blanchard, C., Burgess, S., Harvill, Y., Lanier, J., Lasko, A., Oberman, M., & Teitel, M. (1990). Reality Built for Two: A Virtual Reality Tool. In *Proceedings of the 1990 Symposium on Interactive 3D Graphics* (p. 35–36). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/91385.91409

Bolot, J.-C., & Turletti, T. (1994). A rate control mechanism for packet video in the Internet. In *Proceedings of INFOCOM '94 Conference on Computer Communications* (pp. 1216–1223 vol.3). doi: 10.1109/INFCOM.1994.337568

Bolot, J.-C., Turletti, T., & Wakeman, I. (1994). Scalable Feedback Control for Multicast Video Distribution in the Internet. In *Proceedings of the Conference on Communications Architectures, Protocols and Applications* (p. 58–67). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/190314.190320

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., ... Liang, P. (2022). *On the Opportunities and Risks of Foundation Models.* Retrieved from https://arxiv.org/abs/2108.07258

Botach, A., Zheltonozhskii, E., & Baskin, C. (2022). End-to-End Referring Video Object Segmentation with Multimodal Transformers. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4975–4985). doi: 10.1109/CVPR52688.2022.00493

Boucsein, W. (2012). *Electrodermal activity, 2nd ed.* New York, NY, US: Springer Science + Business Media. (Pages: xviii, 618) doi: 10.1007/978-1-4614-1126-0

Box, G., & Jenkins, G. (1970). *Time Series Analysis: Forecasting and Control*. Holden-Day.

Bradley, M. M., & Lang, P. J. (1994). Measuring emotion: The self-assessment manikin and the semantic differential. *Journal of Behavior Therapy and Experimental Psychiatry*, *25*(1), 49–59. doi: 10.1016/0005-7916(94)90063-9

Bradley, M. M., & Lang, P. J. (2000). Measuring Emotion: Behavior, Feeling, and Physiology. In R. D. R. Lane, L. Nadel, G. L. Ahern, J. Allen, & A. W. Kaszniak (Eds.), *Cognitive Neuroscience of Emotion* (pp. 25–49). Oxford University Press.

Braithwaite, J. J., Watson, D. P. Z., Jones, R. S. G., & Rowe, M. A. (2013). Guide for Analysing Electrodermal Activity & Skin Conductance Responses for Psychological Experiments. *CTIT technical reports series*.

Broderick, D. (1982). *The Judas Mandala*. Pocket Books.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... Amodei, D. (2020). Language Models are Few-Shot Learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Systems* (Vol. 33, pp. 1877–1901). Curran Associates, Inc.

Brueck, D. F., & Hurst, M. B. (U.S. Patent US7818444B2, Oct. 2010). *Apparatus, system, and method for multi-bitrate content streaming*.

Brunnström, K., Beker, S. A., de Moor, K., Dooms, A., Egger, S., Garcia, M.-N., ... Zgank, A. (2013, March). *Qualinet White Paper on Definitions of Quality of Experience*. Retrieved from https://hal.science/hal-00977812 (Qualinet White Paper on Definitions of Quality of Experience Output from the fifth Qualinet meeting, Novi Sad, March 12, 2013)

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., & Lerchner, A. (2018). *Understanding disentangling in β-VAE*. Retrieved from https://arxiv.org/abs/1804.03599

Cai, L., Wang, Z., Gao, H., Shen, D., & Ji, S. (2018). Deep Adversarial Learning for Multi-Modality Missing Data Completion. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (p. 1158–1166). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3219819.3219963

Cao, Z., Gao, H., Mangalam, K., Cai, Q.-Z., Vo, M., & Malik, J. (2020). Long-Term Human Motion Prediction with Scene Context. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*

(p. 387–404). Berlin, Heidelberg: Springer-Verlag. doi: 10.1007/978-3-030-58452-8_23

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-End Object Detection with Transformers. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I* (p. 213–229). Berlin, Heidelberg: Springer-Verlag. doi: 10.1007/978-3-030-58452-8_13

Cen, S., Pu, C., Staehli, R., Cowan, C., & Walpole, J. (1995). A distributed real-time MPEG video audio player. In T. D. C. Little & R. Gusella (Eds.), *Network and Operating Systems Support for Digital Audio and Video* (pp. 142–153). Berlin, Heidelberg: Springer Berlin Heidelberg.

Cen, S., Walpole, J., & Pu, C. (1997). Flow and congestion control for Internet media streaming applications. In K. Jeffay, D. D. Kandlur, & T. Roscoe (Eds.), *Multimedia Computing and Networking 1998* (Vol. 3310, pp. 250 – 264). SPIE. doi: 10.1117/12.298426

Cerf, M., Harel, J., Einhaeuser, W., & Koch, C. (2007). Predicting human gaze using low-level saliency combined with face detection. In J. Platt, D. Koller, Y. Singer, & S. Roweis (Eds.), *Advances in Neural Information Processing Systems* (Vol. 20). Curran Associates, Inc.

Chaabouni, S., & Precioso, F. (2019). Impact of Saliency and Gaze Features on Visual Control: Gaze-Saliency Interest Estimator. In *Proceedings of the 27th ACM International Conference on Multimedia* (p. 1367–1374). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3343031.3350964

Chakareski, J., Aksu, R., Corbillon, X., Simon, G., & Swaminathan, V. (2018). Viewport-Driven Rate-Distortion Optimized 360º Video Streaming. In *2018 IEEE International Conference on Communications (ICC)* (pp. 1–7). doi: 10.1109/ICC.2018.8422859

Chang, M.-F., Lambert, J., Sangkloy, P., Singh, J., Bak, S., Hartnett, A., … Hays, J. (2019). Argoverse: 3D Tracking and Forecasting With Rich Maps. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 8740–8749). doi: 10.1109/CVPR.2019.00895

Chao, F.-Y., Battisti, F., Lebreton, P., & Raake, A. (2023). Chapter 5 - Omnidirectional video saliency. In G. Valenzise, M. Alain, E. Zerman, & C. Ozcinar (Eds.), *Immersive Video Technologies* (pp. 123–158). Academic Press. doi: 10.1016/B978-0-32-391755-1.00011-0

Chao, F.-Y., Ozcinar, C., & Smolic, A. (2021). 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)* (pp. 1–6). doi: 10.1109/MMSP53017.2021.9733647

Chao, F.-Y., Ozcinar, C., & Smolic, A. (2022). Privacy-Preserving Viewport Prediction using Federated Learning for 360° Live Video Streaming. In *2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP)* (pp. 1–6). doi: 10.1109/MMSP55362.2022.9950044

Chao, F.-Y., Ozcinar, C., Wang, C., Zerman, E., Zhang, L., Hamidouche, W., ... Smolic, A. (2020). Audio-Visual Perception of Omnidirectional Video for Virtual Reality Applications. In *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)* (pp. 1–6). doi: 10.1109/ICMEW46912.2020.9105956

Chao, F.-Y., Ozcinar, C., Zhang, L., Hamidouche, W., Deforges, O., & Smolic, A. (2020). Towards Audio-Visual Saliency Prediction for Omnidirectional Video with Spatial Audio. In *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)* (pp. 355–358). doi: 10.1109/VCIP49819.2020.9301766

Chartsias, A., Joyce, T., Giuffrida, M. V., & Tsaftaris, S. A. (2018). Multimodal MR Synthesis via Modality-Invariant Latent Representation. *IEEE Transactions on Medical Imaging*, *37*(3), 803–814. doi: 10.1109/TMI.2017.2764326

Chen, G., Choi, W., Yu, X., Han, T., & Chandraker, M. (2017). Learning Efficient Object Detection Models with Knowledge Distillation. In I. Guyon et al. (Eds.), *Advances in Neural Information Processing Systems* (Vol. 30). Curran Associates, Inc.

Chen, H., Wang, J., Shao, K., Liu, F., Hao, J., Guan, C., ... Heng, P.-A. (2023). *Traj-MAE: Masked Autoencoders for Trajectory Prediction.* Retrieved from https://arxiv.org/abs/2303.06697

Chen, J., Luo, X., Hu, M., Wu, D., & Zhou, Y. (2021). Sparkle: User-Aware Viewport Prediction in 360-Degree Video Streaming. *IEEE Transactions on Multimedia*, *23*, 3853–3866. doi: 10.1109/TMM.2020.3033127

Chen, Y., Xian, Y., Koepke, A. S., Shan, Y., & Akata, Z. (2021). Distilling Audio-Visual Knowledge by Compositional Contrastive Learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 7012–7021). doi: 10.1109/CVPR46437.2021.00694

Chen, Z., Li, Y., & Zhang, Y. (2018). Recent advances in omnidirectional video coding for virtual reality: Projection and evaluation. *Signal Processing*, *146*, 66–78. doi: 10.1016/j.sigpro.2018.01.004

Chen, Z., Zou, L., Tao, X., Xu, L., Muntean, G.-M., & Wang, X. (2022). EdgeVR360: Edge-Assisted Multiuser-Oriented Intelligent 360-degree Video Delivery Scheme over Wireless Networks. In L. Fang, D. Povey, G. Zhai, T. Mei, & R. Wang (Eds.), *Artificial Intelligence* (pp. 242–255). Cham: Springer Nature Switzerland.

Cheng, H.-T., Chao, C.-H., Dong, J.-D., Wen, H.-K., Liu, T.-L., & Sun, M. (2018). Cube Padding for Weakly-Supervised Saliency Prediction in 360° Videos. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 1420–1429). doi: 10.1109/CVPR.2018.00154

Chiariotti, F. (2021). A survey on 360-degree video: Coding, quality of experience and streaming. *Computer Communications*, *177*, 133–155. doi: 10.1016/j.comcom.2021.06.029

Chiariotti, F., D'Aronco, S., Toni, L., & Frossard, P. (2016). Online Learning Adaptation Strategy for DASH Clients. In *Proceedings of the 7th International Conference on Multimedia Systems*. New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2910017.2910603

Child, R., Gray, S., Radford, A., & Sutskever, I. (2019). *Generating Long Sequences with Sparse Transformers.* Retrieved from https://arxiv.org/abs/1904.10509

Chopra, L., Chakraborty, S., Mondal, A., & Chakraborty, S. (2021). PARIMA: Viewport Adaptive 360-Degree Video Streaming. In *Proceedings of the Web Conference 2021* (p. 2379–2391). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3442381.3450070

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C., & Bengio, Y. (2015). A Recurrent Latent Variable Model for Sequential Data. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 28). Curran Associates, Inc.

Claeys, M., Latré, S., Famaey, J., & De Turck, F. (2014). Design and Evaluation of a Self-Learning HTTP Adaptive Video Streaming Client. *IEEE Communications Letters*, *18*(4), 716–719. doi: 10.1109/LCOMM.2014.020414.132649

Claeys, M., Latré, S., Famaey, J., Wu, T., Leekwijck, W. V., & Turck, F. D. (2014). Design and optimisation of a (FA)Q-learning-based HTTP adaptive streaming client. *Connection Science*, *26*(1), 25–43. doi: 10.1080/09540091.2014.885273

Cokelek, M., Imamoglu, N., Ozcinar, C., Erdem, E., & Erdem, A. (2023). *Spherical Vision Transformer for 360-degree Video Saliency Prediction.* Retrieved from https://arxiv.org/abs/2308.13004

Conn, C., Lanier, J., Minsky, M., Fisher, S., & Druin, A. (1989). Virtual Environments and Interactivity: Windows to the Future. In *ACM SIGGRAPH 89 Panel Proceedings*

(p. 7–18). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/77276.77278

Corbillon, X., De Simone, F., & Simon, G. (2017). 360-Degree Video Head Movement Dataset. In *Proceedings of the 8th ACM on Multimedia Systems Conference* (p. 199–204). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3083187.3083215

Corbillon, X., De Simone, F., Simon, G., & Frossard, P. (2018). Dynamic Adaptive Streaming for Multi-Viewpoint Omnidirectional Videos. In *Proceedings of the 9th ACM Multimedia Systems Conference* (p. 237–249). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3204949.3204968

Corbillon, X., Devlic, A., Simon, G., & Chakareski, J. (2017). Optimal Set of 360-Degree Videos for Viewport-Adaptive Streaming. In *Proceedings of the 25th ACM International Conference on Multimedia* (p. 943–951). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3123266.3123372

Corbillon, X., Simon, G., Devlic, A., & Chakareski, J. (2017). Viewport-adaptive navigable 360-degree video delivery. In *2017 IEEE International Conference on Communications (ICC)* (pp. 1–7). doi: 10.1109/ICC.2017.7996611

Dahou, Y., Tliba, M., McGuinness, K., & O'Connor, N. (2021). ATSal: An Attention Based Architecture for Saliency Prediction in 360° Videos. In A. Del Bimbo et al. (Eds.), *Pattern Recognition. ICPR International Workshops and Challenges* (pp. 305–320). Cham: Springer International Publishing.

Dambra, S., Samela, G., Sassatelli, L., Pighetti, R., Aparicio-Pardo, R., & Pinna-Déry, A.-M. (2018). Film Editing: New Levers to Improve VR Streaming. In *Proceedings of the 9th ACM Multimedia Systems Conference* (p. 27–39). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3204949.3204962

David, E. J., Gutiérrez, J., Coutrot, A., Da Silva, M. P., & Le Callet, P. (2018). A Dataset of Head and Eye Movements for 360° Videos. In *Proceedings of the 9th ACM Multimedia Systems Conference* (p. 432–437). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3204949.3208139

David, E. J., Lebranchu, P., Perreira Da Silva, M., & Le Callet, P. (2022, 03). What are the visuo-motor tendencies of omnidirectional scene free-viewing in virtual reality? *Journal of Vision*, *22*(4), 12–12. doi: 10.1167/jov.22.4.12

De Abreu, A., Toni, L., Thomos, N., Maugey, T., Pereira, F., & Frossard, P. (2015). Optimal layered representation for adaptive interactive multiview video streaming. *Journal of Visual Communication and Image Representation*, *33*, 255–264. doi: 10.1016/j.jvcir.2015.09.010

De Abreu, A., Ozcinar, C., & Smolic, A. (2017). Look around you: Saliency maps for omnidirectional images in VR applications. In *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)* (pp. 1–6). doi: 10.1109/QoMEX.2017.7965634

De Cicco, L., Caldaralo, V., Palmisano, V., & Mascolo, S. (2013). ELASTIC: A Client-Side Controller for Dynamic Adaptive Streaming over HTTP (DASH). In *2013 20th International Packet Video Workshop* (pp. 1–8). doi: 10.1109/PV.2013.6691442

De Cicco, L., Mascolo, S., & Palmisano, V. (2011). Feedback Control for Adaptive Live Video Streaming. In *Proceedings of the Second Annual ACM Conference on Multimedia Systems* (p. 145–156). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/1943552.1943573

De Divitiis, L., Becattini, F., Baecchi, C., & Del Bimbo, A. (2021). Garment recommendation with memory augmented neural networks. In *Proceedings of the 25th International Conference on Pattern Recognition (ICPR)* (pp. 282–295). Springer. doi: 10.1007/978-3-030-68790-8_23

Dendorfer, P., Ošep, A., & Leal-Taixé, L. (2020). Goal-GAN: Multimodal Trajectory Prediction Based on Goal Position Estimation. In *Computer Vision – ACCV 2020: 15th Asian Conference on Computer Vision, Kyoto, Japan, November 30 – December 4, 2020, Revised Selected Papers, Part II* (p. 405–420). Berlin, Heidelberg: Springer-Verlag. doi: 10.1007/978-3-030-69532-3_25

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In J. Burstein, C. Doran, & T. Solorio (Eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)* (pp. 4171–4186). Association for Computational Linguistics. doi: 10.18653/v1/n19-1423

Ding, Y., Yu, X., & Yang, Y. (2021). RFNet: Region-aware Fusion Network for Incomplete Multi-modal Brain Tumor Segmentation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 3955–3964). doi: 10.1109/ICCV48922.2021.00394

Dorent, R., Joutard, S., Modat, M., Ourselin, S., & Vercauteren, T. (2019). Hetero-Modal Variational Encoder-Decoder for Joint Modality Completion and Segmentation. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part II* (p. 74–82). Berlin, Heidelberg: Springer-Verlag. doi: 10.1007/978-3-030-32245-8_9

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... Houlsby, N. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations.*

Eleftheriadis, A., & Anastassiou, D. (1995). Constrained and general dynamic rate shaping of compressed digital video. In *Proceedings., International Conference on Image Processing* (Vol. 3, pp. 396–399 vol.3). doi: 10.1109/ICIP.1995.537655

Fan, C.-L., Lee, J., Lo, W.-C., Huang, C.-Y., Chen, K.-T., & Hsu, C.-H. (2017). 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* (p. 67–72). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3083165.3083180

Fan, C.-L., Lo, W.-C., Pai, Y.-T., & Hsu, C.-H. (2019, aug). A Survey on 360° Video Streaming: Acquisition, Transmission, and Display. *ACM Comput. Surv.*, *52*(4). doi: 10.1145/3329119

Fan, S., Shen, Z., Jiang, M., Koenig, B. L., Xu, J., Kankanhalli, M. S., & Zhao, Q. (2018). Emotional Attention: A Study of Image Sentiment and Visual Attention. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 7521–7531). doi: 10.1109/CVPR.2018.00785

Fan, X., Cai, Y., Yang, Y., Xu, T., Li, Y., Zhang, S., & Zhang, F. (2021). Detection of scene-irrelevant head movements via eye-head coordination information. *Virtual Reality & Intelligent Hardware*, *3*(6), 501–514. doi: 10.1016/j.vrih.2021.08.007

Fang, Y., Lin, W., Fang, Z., Lei, J., Le Callet, P., & Yuan, F. (2014). Learning visual saliency for stereoscopic images. In *2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)* (pp. 1–6). doi: 10.1109/ICMEW.2014.6890709

Fang, Y., Wang, J., Narwaria, M., Le Callet, P., & Lin, W. (2014). Saliency Detection for Stereoscopic Images. *IEEE Transactions on Image Processing*, *23*(6), 2625–2636. doi: 10.1109/TIP.2014.2305100

Felnhofer, A., Kothgassner, O. D., Schmidt, M., Heinzle, A.-K., Beutl, L., Hlavacs, H., & Kryspin-Exner, I. (2015). Is virtual reality emotionally arousing? Investigating five emotion inducing virtual park scenarios. *International Journal of Human-Computer Studies*, *82*, 48–56. doi: 10.1016/j.ijhcs.2015.05.004

Feng, X., Bao, Z., & Wei, S. (2021). LiveObj: Object Semantics-based Viewport Prediction for Live Mobile Virtual Reality Streaming. *IEEE Transactions on Visualization and Computer Graphics*, *27*(5), 2736–2745. doi: 10.1109/TVCG.2021.3067686

Feng, X., Li, W., & Wei, S. (2021). LiveROI: Region of Interest Analysis for Viewport Prediction in Live Mobile Virtual Reality Streaming. In *Proceedings of the 12th ACM*

Multimedia Systems Conference (p. 132–145). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3458305.3463378

Feng, X., Liu, Y., & Wei, S. (2020). LiveDeep: Online Viewport Prediction for Live Virtual Reality Streaming Using Lifelong Deep Learning. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (pp. 800–808). doi: 10.1109/VR46266.2020.00104

Feng, X., Swaminathan, V., & Wei, S. (2019, jun). Viewport Prediction for Live 360-Degree Mobile Video Streaming Using User-Content Hybrid Motion Tracking. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, *3*(2). doi: 10.1145/3328914

Fort, S., Hu, H., & Lakshminarayanan, B. (2020). *Deep Ensembles: A Loss Landscape Perspective.* Retrieved from https://arxiv.org/abs/1912.02757

Fraccaro, M., Sø nderby, S. r. K., Paquet, U., & Winther, O. (2016). Sequential Neural Models with Stochastic Layers. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 29). Curran Associates, Inc.

Fragkiadaki, K., Levine, S., Felsen, P., & Malik, J. (2015). Recurrent Network Models for Human Dynamics. In *2015 IEEE International Conference on Computer Vision (ICCV)* (pp. 4346–4354). doi: 10.1109/ICCV.2015.494

Furlanello, T., Lipton, Z., Tschannen, M., Itti, L., & Anandkumar, A. (2018, 10–15 Jul). Born Again Neural Networks. In J. Dy & A. Krause (Eds.), *Proceedings of the 35th International Conference on Machine Learning* (Vol. 80, pp. 1607–1616). PMLR.

Gabeur, V., Sun, C., Alahari, K., & Schmid, C. (2020). Multi-Modal Transformer for Video Retrieval. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV* (p. 214–229). Berlin, Heidelberg: Springer-Verlag. doi: 10.1007/978-3-030-58548-8_13

Gal, Y., & Ghahramani, Z. (2016, 20–22 Jun). Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In M. F. Balcan & K. Q. Weinberger (Eds.), *Proceedings of The 33rd International Conference on Machine Learning* (Vol. 48, pp. 1050–1059). New York, New York, USA: PMLR.

Garcia, N. C., Morerio, P., & Murino, V. (2018). Modality Distillation with Multiple Stream Networks for Action Recognition. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VIII* (p. 106–121). Berlin, Heidelberg: Springer-Verlag. doi: 10.1007/978-3-030-01237-3_7

Garza, A., & Mergenthaler-Canseco, M. (2023). *TimeGPT-1.* Retrieved from https://arxiv.org/abs/2310.03589

Girin, L., Leglaive, S., Bie, X., Diard, J., Hueber, T., & Alameda-Pineda, X. (2021, dec). Dynamical Variational Autoencoders: A Comprehensive Review. *Found. Trends Mach. Learn.*, *15*(1–2), 1–175. doi: 10.1561/2200000089

Giuliari, F., Hasan, I., Cristani, M., & Galasso, F. (2021). Transformer Networks for Trajectory Forecasting. In *2020 25th International Conference on Pattern Recognition (ICPR)* (pp. 10335–10342). doi: 10.1109/ICPR48806.2021.9412190

Gomes, P., Rossi, S., & Toni, L. (2021). Spatio-Temporal Graph-RNN for Point Cloud Prediction. In *2021 IEEE International Conference on Image Processing (ICIP)* (pp. 3428–3432). doi: 10.1109/ICIP42928.2021.9506084

Gong, Z., Tang, Y., & Liang, J. (2023). *PatchMixer: A Patch-Mixing Architecture for Long-Term Time Series Forecasting*. Retrieved from https://arxiv.org/abs/2310.00655

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative Adversarial Nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K. Weinberger (Eds.), *Advances in Neural Information Processing Systems* (Vol. 27). Curran Associates, Inc.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2020, oct). Generative Adversarial Networks. *Commun. ACM*, *63*(11), 139–144. doi: 10.1145/3422622

Gu, A., Dao, T., Ermon, S., Rudra, A., & Ré, C. (2020). HiPPO: Recurrent Memory with Optimal Polynomial Projections. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Systems* (Vol. 33, pp. 1474–1487). Curran Associates, Inc.

Gu, A., Goel, K., Gupta, A., & Ré, C. (2022). On the Parameterization and Initialization of Diagonal State Space Models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in Neural Information Processing Systems* (Vol. 35, pp. 35971–35983). Curran Associates, Inc.

Gu, A., Goel, K., & Re, C. (2022). Efficiently Modeling Long Sequences with Structured State Spaces. In *International Conference on Learning Representations.*

Gu, A., Johnson, I., Goel, K., Saab, K., Dao, T., Rudra, A., & Ré, C. (2021). Combining Recurrent, Convolutional, and Continuous-time Models with Linear State Space Layers. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, & J. W. Vaughan (Eds.), *Advances in Neural Information Processing Systems* (Vol. 34, pp. 572–585). Curran Associates, Inc.

Gu, A., Johnson, I., Timalsina, A., Rudra, A., & Re, C. (2023). How to Train your HIPPO: State Space Models with Generalized Orthogonal Basis Projections. In *International Conference on Learning Representations.*

Guan, J., Yuan, Y., Kitani, K. M., & Rhinehart, N. (2020). Generative Hybrid Representations for Activity Forecasting With No-Regret Learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 170–179). doi: 10.1109/CVPR42600.2020.00025

Guo, Q., Wang, X., Wu, Y., Yu, Z., Liang, D., Hu, X., & Luo, P. (2020). Online Knowledge Distillation via Collaborative Learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 11017–11026). doi: 10.1109/CVPR42600.2020.01103

Gupta, A., Gu, A., & Berant, J. (2022). Diagonal State Spaces are as Effective as Structured State Spaces. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in Neural Information Processing Systems* (Vol. 35, pp. 22982–22994). Curran Associates, Inc.

Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., & Alahi, A. (2018). Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2255–2264). doi: 10.1109/CVPR.2018.00240

Gutiérrez, J., David, E., Rai, Y., & Le Callet, P. (2018). Toolbox and dataset for the development of saliency and scanpath models for omnidirectional/360° still images. *Signal Processing: Image Communication*, *69*, 35–42. (Salient360: Visual attention modeling for 360° Images) doi: 10.1016/j.image.2018.05.003

Gutiérrez, J., David, E. J., Coutrot, A., Da Silva, M. P., & Le Callet, P. (2018). Introducing UN Salient360! Benchmark: A platform for evaluating visual attention models for 360° contents. In *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)* (pp. 1–3). doi: 10.1109/QoMEX.2018.8463369

Ha, D., & Schmidhuber, J. (2018). Recurrent World Models Facilitate Policy Evolution. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*. Curran Associates, Inc. doi: 10.5555/3327144.3327171

Haseeb Ul Hassan, S. M., Brennan, A., Muntean, G.-M., & McManis, J. (2023). User Profile-Based Viewport Prediction Using Federated Learning in Real-Time 360-Degree Video Streaming. In *2023 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)* (pp. 1–7). doi: 10.1109/BMSB58369.2023.10211169

Havaei, M., Guizard, N., Chapados, N., & Bengio, Y. (2016). HeMIS: Hetero-Modal Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II* (p. 469–477). Berlin, Heidelberg: Springer-Verlag. doi: 10.1007/978-3-319-46723-8_54

Hawthorne, C., Jaegle, A., Cangea, C., Borgeaud, S., Nash, C., Malinowski, M., … Engel, J. (2022, 17–23 Jul). General-purpose, long-context autoregressive modeling with Perceiver AR. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, & S. Sabato (Eds.), *Proceedings of the 39th International Conference on Machine Learning* (Vol. 162, pp. 8535–8558). PMLR.

He, D., Westphal, C., & Garcia-Luna-Aceves, J. J. (2018). Network Support for AR/VR and Immersive Video Application: A Survey. In *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications - ICETE* (pp. 359–369). SciTePress. doi: 10.5220/0006941705250535

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2022). Masked Autoencoders Are Scalable Vision Learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 15979–15988). doi: 10.1109/CVPR52688.2022.01553

Hedger, N., Garner, M., & Adams, W. J. (2019, June). Do emotional faces capture attention, and does this depend on awareness? Evidence from the visual probe paradigm. *Journal of experimental psychology. Human perception and performance*, *45*(6), 790—802. doi: 10.1037/xhp0000640

Heilig, M. L. (1955). El Cine del Futuro: the cinema of the future. *Espacios Magazine*, *23 / 24*.

Heilig, M. L. (U.S. Patent US2955156A, Oct. 1960). *Stereoscopic-television apparatus for individual use.*

Heilig, M. L. (U.S. Patent US3050870A, Aug. 1962). *Sensorama simulator.*

Heo, B., Kim, J., Yun, S., Park, H., Kwak, N., & Choi, J. Y. (2019). A Comprehensive Overhaul of Feature Distillation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 1921–1930). doi: 10.1109/ICCV.2019.00201

Hinton, G., Vinyals, O., & Dean, J. (2015). *Distilling the Knowledge in a Neural Network.* Retrieved from https://arxiv.org/abs/1503.02531

Hirway, A., Qiao, Y., & Murray, N. (2022). Spatial Audio in 360° Videos: Does It Influence Visual Attention? In *Proceedings of the 13th ACM Multimedia Systems Conference* (p. 39–51). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3524273.3528179

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Hoffman, J., Gupta, S., & Darrell, T. (2016). Learning with Side Information through Modality Hallucination. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 826–834). doi: 10.1109/CVPR.2016.96

Hooft, J. V. d., Vega, M. T., Petrangeli, S., Wauters, T., & Turck, F. D. (2019, dec). Tile-Based Adaptive Streaming for Virtual Reality Video. *ACM Trans. Multimedia Comput. Commun. Appl.*, *15*(4). doi: 10.1145/3362101

Horn, B. K., & Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, *17*(1), 185–203. doi: 10.1016/0004-3702(81)90024-2

Hosseini, M., & Swaminathan, V. (2016a). Adaptive 360 VR Video Streaming Based on MPEG-DASH SRD. In *2016 IEEE International Symposium on Multimedia (ISM)* (pp. 407–408). doi: 10.1109/ISM.2016.0093

Hosseini, M., & Swaminathan, V. (2016b). Adaptive 360 VR Video Streaming: Divide and Conquer. In *2016 IEEE International Symposium on Multimedia (ISM)* (pp. 107–110). doi: 10.1109/ISM.2016.0028

Hosseini, M., & Timmerer, C. (2018). Dynamic Adaptive Point Cloud Streaming. In *Proceedings of the 23rd Packet Video Workshop* (p. 25–30). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3210424.3210429

Hou, X., Dey, S., Zhang, J., & Budagavi, M. (2021). Predictive Adaptive Streaming to Enable Mobile 360-Degree and VR Experiences. *IEEE Transactions on Multimedia*, *23*, 716–731. doi: 10.1109/TMM.2020.2987693

Hoyle, R. H. (Ed.). (1995). *Structural equation modeling: Concepts, issues, and applications.* Thousand Oaks, CA, US: Sage Publications, Inc. (Pages: xxii, 289)

Hoßfeld, T., Skorin-Kapov, L., Heegaard, P. E., & Varela, M. (2017). Definition of QoE Fairness in Shared Systems. *IEEE Communications Letters*, *21*(1), 184–187. doi: 10.1109/LCOMM.2016.2616342

Hristova, H., Simon, G., Corbillon, X., Devlic, A., & Swaminathan, V. (2021). Heterogeneous Spatial Quality for Omnidirectional Video. *IEEE Transactions on Circuits and Systems for Video Technology*, *31*(4), 1411–1424. doi: 10.1109/TCSVT.2020.3007620

Hu, H., Xu, Z., Zhang, X., & Guo, Z. (2019). Optimal Viewport-Adaptive 360-Degree Video Streaming Against Random Head Movement. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)* (pp. 1–6). doi: 10.1109/ICC.2019.8761189

Hu, M., Maillard, M., Zhang, Y., Ciceri, T., La Barbera, G., Bloch, I., & Gori, P. (2020). Knowledge Distillation from Multi-Modal to Mono-Modal Segmentation Networks. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part I* (p. 772–781). Berlin, Heidelberg: Springer-Verlag. doi: 10.1007/978-3-030-59710-8_75

Huang, T., Zhou, C., Zhang, R.-X., Wu, C., & Sun, L. (2022). Learning Tailored Adaptive Bitrate Algorithms to Heterogeneous Network Conditions: A Domain-Specific Priors and Meta-Reinforcement Learning Approach. *IEEE Journal on Selected Areas in Communications*, *40*(8), 2485–2503. doi: 10.1109/JSAC.2022.3180804

Huang, T., Zhou, C., Zhang, R.-X., Wu, C., Yao, X., & Sun, L. (2019). Comyco: Quality-Aware Adaptive Video Streaming via Imitation Learning. In *Proceedings of the 27th ACM International Conference on Multimedia* (p. 429–437). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3343031.3351014

Huang, T.-Y., Handigol, N., Heller, B., McKeown, N., & Johari, R. (2012). Confused, Timid, and Unstable: Picking a Video Streaming Rate is Hard. In *Proceedings of the 2012 Internet Measurement Conference* (p. 225–238). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2398776.2398800

Huang, T.-Y., Johari, R., McKeown, N., Trunnell, M., & Watson, M. (2014). A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service. In *Proceedings of the 2014 ACM Conference on SIGCOMM* (p. 187–198). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2619239.2626296

Huang, Y., Bi, H., Li, Z., Mao, T., & Wang, Z. (2019). STGAT: Modeling Spatial-Temporal Interactions for Human Trajectory Prediction. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 6271–6280). doi: 10.1109/ICCV.2019.00637

Huang, Z., & Wang, N. (2017). *Like What You Like: Knowledge Distill via Neuron Selectivity Transfer*. Retrieved from https://arxiv.org/abs/1707.01219

Hyndman, R. J., Koehler, A. B., Snyder, R. D., & Grose, S. (2002). A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, *18*(3), 439–454. doi: 10.1016/S0169-2070(01)00110-8

Igolkina, A. A., & Meshcheryakov, G. (2020). semopy: A Python Package for Structural Equation Modeling. *Structural Equation Modeling: A Multidisciplinary Journal*, *0*(0), 1–12. doi: 10.1080/10705511.2019.1704289

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

Ivanovic, B., & Pavone, M. (2019). The Trajectron: Probabilistic Multi-Agent Trajectory Modeling With Dynamic Spatiotemporal Graphs. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 2375–2384). doi: 10.1109/ICCV.2019.00246

Jaegle, A., Borgeaud, S., Alayrac, J.-B., Doersch, C., Ionescu, C., Ding, D., ... Carreira, J. (2022). Perceiver IO: A General Architecture for Structured Inputs & Outputs. In *International Conference on Learning Representations.*

Jaegle, A., Gimeno, F., Brock, A., Vinyals, O., Zisserman, A., & Carreira, J. (2021, 18–24 Jul). Perceiver: General Perception with Iterative Attention. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th International Conference on Machine Learning* (Vol. 139, pp. 4651–4664). PMLR.

Jain, A., Zamir, A. R., Savarese, S., & Saxena, A. (2016). Structural-RNN: Deep Learning on Spatio-Temporal Graphs. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 5308–5317). doi: 10.1109/CVPR.2016.573

Jedari, B., Premsankar, G., Illahi, G., Francesco, M. D., Mehrabi, A., & Ylä-Jääski, A. (2021). Video Caching, Analytics, and Delivery at the Wireless Edge: A Survey and Future Directions. *IEEE Communications Surveys & Tutorials*, *23*(1), 431–471. doi: 10.1109/COMST.2020.3035427

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., ... Duerig, T. (2021, 18–24 Jul). Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th International Conference on Machine Learning* (Vol. 139, pp. 4904–4916). PMLR.

Jiang, J., Sekar, V., & Zhang, H. (2012). Improving Fairness, Efficiency, and Stability in HTTP-Based Adaptive Video Streaming with FESTIVE. In *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies* (p. 97–108). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2413176.2413189

Jiang, M., Huang, S., Duan, J., & Zhao, Q. (2015). SALICON: Saliency in Context. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1072–1080). doi: 10.1109/CVPR.2015.7298710

Jiang, X., Chiang, Y.-H., Zhao, Y., & Ji, Y. (2018). Plato: Learning-based Adaptive Streaming of 360-Degree Videos. In *2018 IEEE 43rd Conference on Local Computer Networks (LCN)* (pp. 393–400). doi: 10.1109/LCN.2018.8638092

Jicol, C., Wan, C. H., Doling, B., Illingworth, C. H., Yoon, J., Headey, C., ... O'Neill, E. (2021). Effects of Emotion and Agency on Presence in Virtual Reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems.* New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3411764.3445588

Jin, Y., Liu, J., Wang, F., & Cui, S. (2022). 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* (p. 1025–1034). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3503161.3548200

Jue, J., Jason, H., Neelam, T., Andreas, R., Sean, B. L., Joseph, D. O., & Harini, V. (2019). Integrating Cross-modality Hallucinated MRI with CT to Aid Mediastinal Lung Tumor Segmentation. In D. Shen et al. (Eds.), *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019* (pp. 221–229). Cham: Springer International Publishing.

Juluri, P., Tamarapalli, V., & Medhi, D. (2016). QoE management in DASH systems using the segment aware rate adaptation algorithm. In *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium* (pp. 129–136). doi: 10.1109/NOMS .2016.7502805

Kan, N., Li, C., Yang, C., Dai, W., Zou, J., & Xiong, H. (2021). Uncertainty-Aware Robust Adaptive Video Streaming with Bayesian Neural Network and Model Predictive Control. In *Proceedings of the 31st ACM Workshop on Network and Operating Systems Support for Digital Audio and Video* (p. 17–24). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3458306.3458872

Kanakia, H., Mishra, P. P., & Reibman, A. (1993). An Adaptive Congestion Control Scheme for Real-Time Packet Video Transport. In *Conference Proceedings on Communications Architectures, Protocols and Applications* (p. 20–31). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/166237.166240

Kendall, A., & Gal, Y. (2017). What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In I. Guyon et al. (Eds.), *Advances in Neural Information Processing Systems* (Vol. 30). Curran Associates, Inc.

Kilian, L., & Taylor, M. P. (2003). Why is it so difficult to beat the random walk forecast of exchange rates? *Journal of International Economics*, *60*(1), 85–107. (Emperical Exchange Rate Models) doi: 10.1016/S0022-1996(02)00060-0

Kim, H. S., Nam, S. B., Choi, S. G., Kim, C. H., Sung, T. T. K., & Sohn, C.-B. (2018). HLS-based 360 VR using spatial segmented adaptive streaming. In *2018 IEEE International Conference on Consumer Electronics (ICCE)* (pp. 1–4). doi: 10.1109/ICCE.2018.8326272

Kim, J., Park, S., & Kwak, N. (2018). Paraphrasing Complex Network: Network Compression via Factor Transfer. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 31). Curran Associates, Inc.

Kim, W., Son, B., & Kim, I. (2021, 18–24 Jul). ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th International Conference on Machine Learning* (Vol. 139, pp. 5583–5594). PMLR.

Kimura, A. (2020, September). *pySaliencyMap*. Retrieved from https://github.com/akisatok/pySaliencyMap

Kingma, D. P., & Dhariwal, P. (2018). Glow: Generative Flow with Invertible 1x1 Convolutions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 31). Curran Associates, Inc.

Kingma, D. P., & Welling, M. (2014). Auto-Encoding Variational Bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*. ICLR.

Koo, T. K., & Li, M. Y. (2016, June). A Guideline of Selecting and Reporting Intraclass Correlation Coefficients for Reliability Research. *Journal of chiropractic medicine*, *15*(2), 155–163. (Edition: 2016/03/31 Publisher: Elsevier) doi: 10.1016/j.jcm.2016.02.012

Kosaraju, V., Sadeghian, A., Martín-Martín, R., Reid, I., Rezatofighi, H., & Savarese, S. (2019). Social-BiGAT: Multimodal Trajectory Forecasting using Bicycle-GAN and Graph Attention Networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 32). Curran Associates, Inc.

Krueger, M. (1983). *Artificial Reality*. Addison-Wesley.

Kua, J., Armitage, G., & Branch, P. (2017). A Survey of Rate Adaptation Techniques for Dynamic Adaptive Streaming Over HTTP. *IEEE Communications Surveys & Tutorials*, *19*(3), 1842–1866. doi: 10.1109/COMST.2017.2685630

Lai, G., Chang, W.-C., Yang, Y., & Liu, H. (2018). Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (p. 95–104). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3209978.3210006

Lanier, J. (n.d.). *A Vintage Virtual Reality Interview*.

The Lawnmower Man. (1992). United States: New Line Cinema.

Lederer, S., Müller, C., & Timmerer, C. (2012). Dynamic Adaptive Streaming over HTTP Dataset. In *Proceedings of the 3rd Multimedia Systems Conference* (p. 89–94). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2155555.2155570

Lee, N., Choi, W., Vernaza, P., Choy, C. B., Torr, P. H. S., & Chandraker, M. (2017). DESIRE: Distant Future Prediction in Dynamic Scenes with Interacting Agents. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2165–2174). doi: 10.1109/CVPR.2017.233

Le Meur, O., & Baccino, T. (2013, March). Methods for comparing scanpaths and saliency maps: strengths and weaknesses. *Behavior Research Methods*, *45*(1), 251–266. doi: 10.3758/s13428-012-0226-9

Li, B. J., Bailenson, J. N., Pines, A., Greenleaf, W. J., & Williams, L. M. (2017). A Public Database of Immersive VR Videos with Corresponding Ratings of Arousal, Valence, and Correlations between Head Movements and Self Report Measures. *Frontiers in Psychology*, *8*. doi: 10.3389/fpsyg.2017.02116

Li, C., Toni, L., Zou, J., Xiong, H., & Frossard, P. (2018). QoE-Driven Mobile Edge Caching Placement for Adaptive Video Streaming. *IEEE Transactions on Multimedia*, *20*(4), 965–984. doi: 10.1109/TMM.2017.2757761

Li, C., Zhang, W., Liu, Y., & Wang, Y. (2019). Very Long Term Field of View Prediction for 360-Degree Video Streaming. In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)* (pp. 297–302). doi: 10.1109/MIPR.2019.00060

Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., & Hoi, S. C. H. (2021). Align before Fuse: Vision and Language Representation Learning with Momentum Distillation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, & J. W. Vaughan (Eds.), *Advances in Neural Information Processing Systems* (Vol. 34, pp. 9694–9705). Curran Associates, Inc.

Li, J., Zhai, G., Zhu, Y., Zhou, J., & Zhang, X.-P. (2022). How Sound Affects Visual Attention in Omnidirectional Videos. In *2022 IEEE International Conference on Image Processing (ICIP)* (pp. 3066–3070). doi: 10.1109/ICIP46576.2022.9897737

Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.-X., & Yan, X. (2019). Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 32). Curran Associates, Inc.

Li, X., Lei, L., Sun, Y., & Kuang, G. (2022). Dynamic-Hierarchical Attention Distillation With Synergetic Instance Selection for Land Cover Classification Using Missing Heterogeneity Images. *IEEE Transactions on Geoscience and Remote Sensing*, *60*, 1–16. doi: 10.1109/TGRS.2020.3048332

Li, Y., Xu, Y., Xie, S., Ma, L., & Sun, J. (2019). Two-Layer FoV Prediction Model for Viewport Dependent Streaming of 360-Degree Videos. In X. Liu, D. Cheng, & L. Jinfeng (Eds.), *Communications and Networking* (pp. 501–509). Cham: Springer International Publishing.

Li, Z., Zhu, X., Gahm, J., Pan, R., Hu, H., Begen, A. C., & Oran, D. (2014). Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale. *IEEE Journal on Selected Areas in Communications*, *32*(4), 719–733. doi: 10.1109/JSAC.2014.140405

Lim, B., Arik, S., Loeff, N., & Pfister, T. (2021). Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, *37*(4), 1748–1764. doi: 10.1016/j.ijforecast.2021.03.012

Lin, Y., Gou, Y., Liu, Z., Li, B., Lv, J., & Peng, X. (2021). COMPLETER: Incomplete Multi-view Clustering via Contrastive Prediction. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 11169–11178). doi: 10.1109/CVPR46437.2021.01102

Liu, A., Tan, Z., Wan, J., Liang, Y., Lei, Z., Guo, G., & Li, S. Z. (2021). Face Anti-Spoofing via Adversarial Cross-Modality Translation. *IEEE Transactions on Information Forensics and Security*, *16*, 2759–2772. doi: 10.1109/TIFS.2021.3065495

Liu, J., Liu, X., Zhang, Y., Zhang, P., Tu, W., Wang, S., ... Yang, Y. (2021). Self-Representation Subspace Clustering for Incomplete Multi-View Data. In *Proceedings of the 29th ACM International Conference on Multimedia* (p. 2726–2734). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3474085.3475379

Liu, J., Simon, G., Corbillon, X., Chakareski, J., & Yang, Q. (2020). Delivering Viewport-Adaptive 360-Degree Videos in Cache-Aided MEC Networks. In *2020 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)* (pp. 1–6). doi: 10.1109/BMSB49480.2020.9379613

Liu, S., Yu, H., Liao, C., Li, J., Lin, W., Liu, A. X., & Dustdar, S. (2022). Pyraformer: Low-Complexity Pyramidal Attention for Long-Range Time Series Modeling and Forecasting. In *International Conference on Learning Representations.*

Liu, Y., Cao, J., Li, B., Yuan, C., Hu, W., Li, Y., & Duan, Y. (2019). Knowledge Distillation via Instance Relationship Graph. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 7089–7097). doi: 10.1109/CVPR.2019.00726

Liu, Y., Zhang, J., Fang, L., Jiang, Q., & Zhou, B. (2021). Multimodal Motion Prediction with Stacked Transformers. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 7573–7582). doi: 10.1109/CVPR46437.2021.00749

Liu, Z., Su, P., Wu, S., Shen, X., Chen, H., Hao, Y., & Wang, M. (2021). Motion Prediction using Trajectory Cues. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 13279–13288). doi: 10.1109/ICCV48922.2021.01305

Lo, W.-C., Fan, C.-L., Lee, J., Huang, C.-Y., Chen, K.-T., & Hsu, C.-H. (2017). 360° Video Viewing Dataset in Head-Mounted Virtual Reality. In *Proceedings of the 8th ACM on Multimedia Systems Conference* (p. 211–216). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3083187.3083219

Lyu, Z., Aminian, G., & Rodrigues, M. R. D. (2021). Toward Minimal-Sufficiency in Regression Tasks: An Approach Based on a Variational Estimation Bottleneck. In *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)* (pp. 1–6). doi: 10.1109/MLSP52302.2021.9596368

Ma, M., Ren, J., Zhao, L., Testuggine, D., & Peng, X. (2022). Are Multimodal Transformers Robust to Missing Modality? In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 18156–18165). doi: 10.1109/CVPR52688.2022.01764

Ma, M., Ren, J., Zhao, L., Tulyakov, S., Wu, C., & Peng, X. (2021, May). SMIL: Multimodal Learning with Severely Missing Modality. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(3), 2302–2310. doi: 10.1609/aaai.v35i3.16330

Ma, X., Li, Q., Jiang, Y., Muntean, G.-M., & Zou, L. (2022). Learning-Based Joint QoE Optimization for Adaptive Video Streaming Based on Smart Edge. *IEEE Transactions on Network and Service Management*, *19*(2), 1789–1806. doi: 10.1109/TNSM.2022.3145619

Ma, X., Li, Q., Zou, L., Peng, J., Zhou, J., Chai, J., ... Muntean, G.-M. (2022). QAVA: QoE-Aware Adaptive Video Bitrate Aggregation for HTTP Live Streaming Based on Smart Edge Computing. *IEEE Transactions on Broadcasting*, *68*(3), 661–676. doi: 10.1109/TBC.2022.3171131

Makansi, O., Ilg, E., Cicek, O., & Brox, T. (2019). Overcoming Limitations of Mixture Density Networks: A Sampling and Fitting Framework for Multimodal Future Prediction. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 7137–7146). doi: 10.1109/CVPR.2019.00731

Makowski, D., Pham, T., Lau, Z. J., Brammer, J. C., Lespinasse, F., Pham, H., ... Chen, S. H. A. (2021, feb). NeuroKit2: A Python toolbox for neurophysiological signal processing. *Behavior Research Methods*, *53*(4), 1689–1696. doi: 10.3758/s13428-020-01516-y

Mangalam, K., Girase, H., Agarwal, S., Lee, K.-H., Adeli, E., Malik, J., & Gaidon, A. (2020). It Is Not the Journey But the Destination: Endpoint Conditioned Trajectory Prediction. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II* (p. 759–776). Berlin, Heidelberg: Springer-Verlag. doi: 10.1007/978-3-030-58536-5_45

Mao, H., Netravali, R., & Alizadeh, M. (2017). Neural Adaptive Video Streaming with Pensieve. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication* (p. 197–210). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3098822.3098843

Mao, Y., Sun, L., Liu, Y., & Wang, Y. (2020). Low-Latency FoV-Adaptive Coding and Streaming for Interactive 360° Video Streaming. In *Proceedings of the 28th ACM International Conference on Multimedia* (p. 3696–3704). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3394171.3413751

Marchetti, F., Becattini, F., Seidenari, L., & Del Bimbo, A. (2020a). MANTRA: Memory Augmented Networks for Multiple Trajectory Prediction. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 7141–7150). doi: 10.1109/CVPR42600.2020.00717

Marchetti, F., Becattini, F., Seidenari, L., & Del Bimbo, A. (2020b). Multiple Trajectory Prediction of Moving Agents With Memory Augmented Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(6), 6688–6702. doi: 10.1109/TPAMI.2020.3008558

Martinez, J., Black, M. J., & Romero, J. (2017). On Human Motion Prediction Using Recurrent Neural Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4674–4683). doi: 10.1109/CVPR.2017.497

Mazumdar, P., Arru, G., Carli, M., & Battisti, F. (2019). Face-aware Saliency Estimation Model for 360° Images. In *2019 27th European Signal Processing Conference (EUSIPCO)* (pp. 1–5). doi: 10.23919/EUSIPCO.2019.8902556

Mazumdar, P., Arru, G., Carli, M., & Battisti, F. (2021). Analysis of the influence of human faces for the estimation of salience in omnidirectional images. In *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP)* (pp. 1–5). doi: 10.1109/MMSP53017.2021.9733697

Mazumdar, P., & Battisti, F. (2019). A Content-Based Approach for Saliency Estimation in 360 Images. In *2019 IEEE International Conference on Image Processing (ICIP)* (pp. 3197–3201). doi: 10.1109/ICIP.2019.8803296

Mazumdar, P., Lamichhane, K., Carli, M., & Battisti, F. (2019). A Feature Integrated Saliency Estimation Model for Omnidirectional Immersive Images. *Electronics*, *8*(12). doi: 10.3390/electronics8121538

Mehrabian, A., & Russell, J. (1974). *An Approach to Environmental Psychology*. M.I.T. Press.

Merriam-Webster. (2023). virtual reality. In *Merriam-Webster.com dictionary*. Retrieved 2023-10-04, from https://www.merriam-webster.com/dictionary/virtual%20reality

Meshcheryakov, G., Igolkina, A. A., & Samsonova, M. G. (2021). *semopy 2: A Structural Equation Modeling Package with Random Effects in Python.* Retrieved from https://arxiv.org/abs/2106.01140

Mirzadeh, S. I., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., & Ghasemzadeh, H. (2020, Apr.). Improved Knowledge Distillation via Teacher Assistant. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(04), 5191–5198. doi: 10.1609/aaai.v34i04.5963

Mohamed, S., & Lakshminarayanan, B. (2017). *Learning in Implicit Generative Models.* Retrieved from https://arxiv.org/abs/1610.03483

Mondal, S., Yang, Z., Ahn, S., Samaras, D., Zelinsky, G., & Hoai, M. (2023, June). Gazeformer: Scalable, Effective and Fast Prediction of Goal-Directed Human Attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1441–1450).

Monks, J., Olaru, A., & Muntean, G.-M. (2019). Buffer-Aware Dynamic Adaptive Streaming over Content Centric Networks. In *2019 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)* (pp. 1–6). doi: 10.1109/BMSB47279.2019.8971891

Monroy, R., Lutz, S., Chalasani, T., & Smolic, A. (2018). SalNet360: Saliency maps for omni-directional images with CNN. *Signal Processing: Image Communication*, *69*, 26–34. (Salient360: Visual attention modeling for 360° Images) doi: 10.1016/j.image.2018.05.005

Mu, J., Liang, P., & Goodman, N. (2020, July). Shaping Visual Representations with Language for Few-Shot Classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 4823–4830). Online: Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.436

Mueller, C., Lederer, S., Timmerer, C., & Hellwagner, H. (2013). Dynamic Adaptive Streaming over HTTP/2.0. In *2013 IEEE International Conference on Multimedia and Expo (ICME)* (pp. 1–6). doi: 10.1109/ICME.2013.6607498

Mutha, N. (2017, September). *Equirectangular-toolbox.* Retrieved from https://github.com/NitishMutha/equirectangular-toolbox

Nasoz, F., Alvarez, K., Lisetti, C. L., & Finkelstein, N. (2004, Feb 01). Emotion recognition from physiological signals using wireless sensors for presence technologies. *Cognition, Technology & Work*.

Nasrabadi, A. T., Mahzari, A., Beshay, J. D., & Prakash, R. (2017a). Adaptive 360-degree video streaming using layered video coding. In *2017 IEEE Virtual Reality (VR)* (pp. 347–348). doi: 10.1109/VR.2017.7892319

Nasrabadi, A. T., Mahzari, A., Beshay, J. D., & Prakash, R. (2017b). Adaptive 360-Degree Video Streaming Using Scalable Video Coding. In *Proceedings of the 25th ACM International Conference on Multimedia* (p. 1689–1697). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3123266.3123414

Nasrabadi, A. T., Samiei, A., Mahzari, A., McMahan, R. P., Prakash, R., Farias, M. C. Q., & Carvalho, M. M. (2019). A Taxonomy and Dataset for 360° Videos. In *Proceedings of the 10th ACM Multimedia Systems Conference* (p. 273–278). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3304109.3325812

Nasrabadi, A. T., Samiei, A., & Prakash, R. (2020). Viewport Prediction for 360° Videos: A Clustering Approach. In *Proceedings of the 30th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video* (p. 34–39). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3386290.3396934

Nayakanti, N., Al-Rfou, R., Zhou, A., Goel, K., Refaat, K. S., & Sapp, B. (2023). Wayformer: Motion Forecasting via Simple & Efficient Attention Networks. In *2023 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 2980–2987). doi: 10.1109/ICRA48891.2023.10160609

Neal, R. M. (2012). *Bayesian learning for neural networks* (Vol. 118). Springer. doi: 10.1007/978-1-4612-0745-0

Ngampruetikorn, V., & Schwab, D. J. (2022). Information bottleneck theory of high-dimensional regression: relevancy, efficiency and optimality. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in Neural Information Processing Systems* (Vol. 35, pp. 9784–9796). Curran Associates, Inc.

Nguyen, A., Yan, Z., & Nahrstedt, K. (2018). 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* (p. 1190–1198). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3240508.3240669

Nie, Y., Nguyen, N. H., Sinthong, P., & Kalagnanam, J. (2023). A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *The Eleventh International Conference on Learning Representations.*

Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2020). N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations.*

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., ... Snoek, J. (2019). Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 32). Curran Associates, Inc.

Ozcinar, C., Cabrera, J., & Smolic, A. (2018a). Omnidirectional Video Streaming Using Visual Attention-Driven Dynamic Tiling for VR. In *2018 IEEE Visual Communications and Image Processing (VCIP)* (pp. 1–4). doi: 10.1109/VCIP.2018.8698638

Ozcinar, C., Cabrera, J., & Smolic, A. (2018b). Viewport-Aware Omnidirectional Video Streaming Using Visual Attention and Dynamic Tiles. In *2018 7th European Workshop on Visual Information Processing (EUVIP)* (pp. 1–6). doi: 10.1109/EUVIP.2018.8611777

Ozcinar, C., Cabrera, J., & Smolic, A. (2019). Visual Attention-Aware Omnidirectional Video Streaming Using Optimal Tiles for Virtual Reality. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, *9*(1), 217–230. doi: 10.1109/JETCAS.2019.2895096

Ozcinar, C., De Abreu, A., Knorr, S., & Smolic, A. (2017). Estimation of Optimal Encoding Ladders for Tiled 360° VR Video in Adaptive Streaming Systems. In *2017 IEEE International Symposium on Multimedia (ISM)* (pp. 45–52). doi: 10.1109/ISM.2017.17

Ozcinar, C., De Abreu, A., & Smolic, A. (2017). Viewport-aware adaptive 360° video streaming using tiles for virtual reality. In *2017 IEEE International Conference on Image Processing (ICIP)* (pp. 2174–2178). doi: 10.1109/ICIP.2017.8296667

Ozcinar, C., İmamoğlu, N., Wang, W., & Smolic, A. (2021, Apr 01). Delivery of omnidirectional video using saliency prediction and optimal bitrate allocation. *Signal, Image and Video Processing*, *15*(3), 493–500. doi: 10.1007/s11760-020-01769-2

Ozcinar, C., & Smolic, A. (2018). Visual Attention in Omnidirectional Video for Virtual Reality Applications. In *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)* (pp. 1–6). doi: 10.1109/QoMEX.2018.8463418

Pallavicini, F., Pepe, A., & Minissi, M. E. (2019). Gaming in Virtual Reality: What Changes in Terms of Usability, Emotional Response and Sense of Presence Compared to Non-Immersive Video Games? *Simulation & Gaming*, *50*(2), 136–159. doi: 10.1177/1046878119831420

Pan, Y., Liu, M., Lian, C., Xia, Y., & Shen, D. (2020). Spatially-Constrained Fisher Representation for Brain Disease Identification With Incomplete Multi-Modal Neuroimages. *IEEE Transactions on Medical Imaging*, *39*(9), 2965–2975. doi: 10.1109/TMI.2020.2983085

Park, J., Chou, P. A., & Hwang, J.-N. (2019). Rate-Utility Optimized Streaming of Volumetric Media for Augmented Reality. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, *9*(1), 149–162. doi: 10.1109/JETCAS.2019.2898622

Park, J., & Nahrstedt, K. (2019). Navigation Graph for Tiled Media Streaming. In *Proceedings of the 27th ACM International Conference on Multimedia* (p. 447–455). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3343031.3351021

Park, S., Bhattacharya, A., Yang, Z., Das, S. R., & Samaras, D. (2021). Mosaic: Advancing User Quality of Experience in 360-Degree Video Streaming With Machine Learning. *IEEE Transactions on Network and Service Management*, *18*(1), 1000–1015. doi: 10.1109/TNSM.2021.3053183

Park, S., Hoai, M., Bhattacharya, A., & Das, S. R. (2021, January). Adaptive Streaming of 360-Degree Videos With Reinforcement Learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (pp. 1839–1848). doi: 10.1109/WACV48630.2021.00188

Park, W., Kim, D., Lu, Y., & Cho, M. (2019). Relational Knowledge Distillation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3962–3971). doi: 10.1109/CVPR.2019.00409

Parsaeifard, B., Saadatnejad, S., Liu, Y., Mordan, T., & Alahi, A. (2021). Learning Decoupled Representations for Human Pose Forecasting. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)* (pp. 2294–2303). doi: 10.1109/ICCVW54120.2021.00259

Passalis, N., & Tefas, A. (2018). Learning Deep Representations with Probabilistic Knowledge Transfer. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XI* (p. 283–299). Berlin, Heidelberg: Springer-Verlag. doi: 10.1007/978-3-030-01252-6_17

Passban, P., Wu, Y., Rezagholizadeh, M., & Liu, Q. (2021, May). ALP-KD: Attention-Based Layer Projection for Knowledge Distillation. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(15), 13657–13665. doi: 10.1609/aaai.v35i15.17610

Perkis, A., Timmerer, C., Baraković, S., Husić, J. B., Bech, S., Bosse, S., ... Zadtootaghaj, S. (2020). *QUALINET White Paper on Definitions of Immersive Media Experience (IMEx).* Retrieved from https://arxiv.org/abs/2007.07032

Petrangeli, S., Famaey, J., Claeys, M., Latré, S., & De Turck, F. (2015, oct). QoE-Driven Rate Adaptation Heuristic for Fair Adaptive Video Streaming. *ACM Trans. Multimedia Comput. Commun. Appl.*, *12*(2). doi: 10.1145/2818361

Petrangeli, S., Simon, G., & Swaminathan, V. (2018). Trajectory-Based Viewport Prediction for 360-Degree Virtual Reality Videos. In *2018 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)* (pp. 157–160). doi: 10.1109/AIVR.2018.00033

Petrangeli, S., Simon, G., Wang, H., & Swaminathan, V. (2019). Dynamic Adaptive Streaming for Augmented Reality Applications. In *2019 IEEE International Symposium on Multimedia (ISM)* (pp. 56–567). doi: 10.1109/ISM46123.2019.00017

Petrangeli, S., Swaminathan, V., Hosseini, M., & De Turck, F. (2017). An HTTP/2-Based Adaptive Streaming Framework for 360° Virtual Reality Videos. In *Proceedings of the 25th ACM International Conference on Multimedia* (p. 306–314). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3123266.3123453

Polakovič, A., Rozinaj, G., & Muntean, G.-M. (2022). User Gaze-Driven Adaptation of Omnidirectional Video Delivery Using Spatial Tiling and Scalable Video Encoding. *IEEE Transactions on Broadcasting*, *68*(3), 609–619. doi: 10.1109/TBC.2022.3157470

Postels, J., Blum, H., Strümpler, Y., Cadena, C., Siegwart, R., Gool, L. V., & Tombari, F. (2021). *The Hidden Uncertainty in a Neural Network's Activations.* Retrieved from https://arxiv.org/abs/2012.03082

Postels, J., Ferroni, F., Coskun, H., Navab, N., & Tombari, F. (2019). Sampling-Free Epistemic Uncertainty Estimation Using Approximated Variance Propagation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 2931–2940). doi: 10.1109/ICCV.2019.00302

Qian, F., Han, B., Xiao, Q., & Gopalakrishnan, V. (2018). Flare: Practical Viewport-Adaptive 360-Degree Video Streaming for Mobile Devices. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking* (p. 99–114). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3241539.3241565

Qian, F., Ji, L., Han, B., & Gopalakrishnan, V. (2016). Optimizing 360 Video Delivery over Cellular Networks. In *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges* (p. 1–6). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2980055.2980056

Qiao, M., Xu, M., Wang, Z., & Borji, A. (2021). Viewport-Dependent Saliency Prediction in 360° Video. *IEEE Transactions on Multimedia*, *23*, 748–760. doi: 10.1109/TMM.2020.2987682

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... Sutskever, I. (2021, 18–24 Jul). Learning Transferable Visual Models From Natural Language Supervision. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th International Conference on Machine Learning* (Vol. 139, pp. 8748–8763). PMLR.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... Liu, P. J. (2020, jan). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.*, *21*(1).

Rai, Y., Gutiérrez, J., & Le Callet, P. (2017). A Dataset of Head and Eye Movements for 360 Degree Images. In *Proceedings of the 8th ACM on Multimedia Systems Conference* (p. 205–210). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3083187.3083218

Rai, Y., Le Callet, P., & Cheung, G. (2016). Quantifying the relation between perceived interest and visual salience during free viewing using trellis based optimization. In *2016 IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)* (pp. 1–5). doi: 10.1109/IVMSPW.2016.7528228

Rai, Y., Le Callet, P., & Guillotel, P. (2017). Which saliency weighting for omni directional image quality assessment? In *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)* (pp. 1–6). doi: 10.1109/QoMEX.2017.7965659

Ramanujan, R., Newhouse, J., Kaddoura, M., Ahamad, A., Chartier, E., & Thurber, K. (1997). Adaptive streaming of MPEG video over IP networks. In *Proceedings of 22nd Annual Conference on Local Computer Networks* (pp. 398–409). doi: 10.1109/LCN.1997.631009

Ren, S., Du, Y., Lv, J., Han, G., & He, S. (2021). Learning from the Master: Distilling Cross-modal Advanced Knowledge for Lip Reading. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 13320–13328). doi: 10.1109/CVPR46437.2021.01312

Rezende, D. J., Mohamed, S., & Wierstra, D. (2014). Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of the 31st International Conference on Machine Learning (ICML)* (p. II–1278–II–1286). JMLR.org. doi: 10.5555/3044805.3045035

Ribezzo, G., De Cicco, L., Palmisano, V., & Mascolo, S. (2020). TAPAS-360° : A Tool for the Design and Experimental Evaluation of 360° Video Streaming Systems. In *Proceedings of the 28th ACM International Conference on Multimedia* (p. 4477–4480). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3394171.3414541

Robitza, W. (2019). *CRF Guide (Constant Rate Factor in x264, x265 and libvpx).* Retrieved from https://slhck.info/video/2017/02/24/crf-guide.html

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., & Bengio, Y. (2015). *FitNets: Hints for Thin Deep Nets.* Retrieved from https://arxiv.org/abs/1412.6550

Romero Rondón, M. F., Sassatelli, L., Aparicio-Pardo, R., & Precioso, F. (2020). A Unified Evaluation Framework for Head Motion Prediction Methods in 360° Videos. In *Proceedings of the 11th ACM Multimedia Systems Conference* (p. 279–284). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3339825.3394934

Romero Rondon, M. F., Zanca, D., Melacci, S., Gori, M., & Sassatelli, L. (2021). HeMoG: A White-Box Model to Unveil the Connection between Saliency Information and Human Head Motion in Virtual Reality. In *2021 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)* (pp. 10–18). doi: 10.1109/AIVR52153.2021.00012

Romero Rondón, M. F., Sassatelli, L., Aparicio-Pardo, R., & Precioso, F. (2021). TRACK: A New Method From a Re-Examination of Deep Architectures for Head Motion Prediction in 360° Videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(9), 5681–5699. doi: 10.1109/TPAMI.2021.3070520

Rossi, S., De Simone, F., Frossard, P., & Toni, L. (2019). Spherical Clustering of Users Navigating 360° Content. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4020–4024). doi: 10.1109/ICASSP.2019.8683854

Rossi, S., Guedes, A., & Toni, L. (2023). Chapter 3 - Streaming and user behavior in omnidirectional videos. In G. Valenzise, M. Alain, E. Zerman, & C. Ozcinar (Eds.), *Immersive Video Technologies* (pp. 49–83). Academic Press. doi: 10.1016/B978-0-32-391755-1.00009-2

Rossi, S., Ozcinar, C., Smolic, A., & Toni, L. (2020, may). Do Users Behave Similarly in VR? Investigation of the User Influence on the System Design. *ACM Trans. Multimedia Comput. Commun. Appl.*, *16*(2). doi: 10.1145/3381846

Rossi, S., & Toni, L. (2017). Navigation-aware adaptive streaming strategies for omnidirectional video. In *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)* (pp. 1–6). doi: 10.1109/MMSP.2017.8122230

Rossi, S., & Toni, L. (2020). 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* (p. 19–24). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3386293.3397115

Rossi, S., Viola, I., Jansen, J., Subramanyam, S., Toni, L., & Cesar, P. (2021). Influence of Narrative Elements on User Behaviour in Photorealistic Social VR. In *Proceedings of the International Workshop on Immersive Mixed and Virtual Environment Systems (MMVE '21)* (p. 1–7). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3458307.3463371

Rossi, S., Viola, I., Toni, L., & Cesar, P. (2021). A New Challenge: Behavioural Analysis Of 6-DOF User When Consuming Immersive Media. In *2021 IEEE International Conference on Image Processing (ICIP)* (pp. 3423–3427). doi: 10.1109/ICIP42928.2021.9506525

Rossi, S., Viola, I., Toni, L., & Cesar, P. (2023). Extending 3-DoF Metrics to Model User Behaviour Similarity in 6-DoF Immersive Applications. In *Proceedings of the 14th Conference on ACM Multimedia Systems* (p. 39–50). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3587819.3590976

Ruan, J., & Xie, D. (2021). Networked VR: State of the Art, Solutions, and Challenges. *Electronics*, *10*(2). doi: 10.3390/electronics10020166

Rupprecht, C., Laina, I., DiPietro, R., Baust, M., Tombari, F., Navab, N., & Hager, G. D. (2017). Learning in an Uncertain World: Representing Ambiguity Through Multiple Hypotheses. In *2017 IEEE International Conference on Computer Vision (ICCV)* (pp. 3611–3620). doi: 10.1109/ICCV.2017.388

Russell, J. A. (1980, June). A circumplex model of affect. *Journal of Personality and Social Psychology*, *39*(6), 1161–1178. doi: 10.1037/h0077714

Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Rezatofighi, H., & Savarese, S. (2019). SoPhie: An Attentive GAN for Predicting Paths Compliant to Social and Physical Constraints. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1349–1358). doi: 10.1109/CVPR.2019.00144

Sani, Y., Mauthe, A., & Edwards, C. (2017). Adaptive Bitrate Selection: A Survey. *IEEE Communications Surveys & Tutorials*, *19*(4), 2985–3014. doi: 10.1109/COMST.2017.2725241

Sassatelli, L., Winckler, M., Fisichella, T., Aparicio, R., & Pinna-Déry, A.-M. (2019). A New Adaptation Lever in 360° Video Streaming. In *Proceedings of the 29th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video* (pp. 37–42). New York, NY, USA: ACM. doi: 10.1145/3304112.3325610

Sassatelli, L., Winckler, M., Fisichella, T., Dezarnaud, A., Lemaire, J., Aparicio-Pardo, R., & Trevisan, D. (2020). New interactive strategies for virtual reality streaming in degraded context of use. *Computers & Graphics*, *86*, 27–41. doi: 10.1016/j.cag.2019.10.005

Schupp, H. T., Stockburger, J., Codispoti, M., Junghöfer, M., Weike, A. I., & Hamm, A. O. (2007). Selective Visual Attention to Emotion. *Journal of Neuroscience*, *27*(5), 1082–1089. doi: 10.1523/JNEUROSCI.3223-06.2007

Shafi, R., Shuai, W., & Younus, M. U. (2020, Sep). 360-Degree Video Streaming: A Survey of the State of the Art. *Symmetry*, *12*(9), 1491. doi: 10.3390/sym12091491

Shi, S., Jiang, L., Dai, D., & Schiele, B. (2022). Motion Transformer with Global Intention Localization and Local Movement Refinement. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in Neural Information Processing Systems* (Vol. 35, pp. 6531–6543). Curran Associates, Inc.

Shi, W., Li, Q., Zhang, R., Shen, G., Jiang, Y., Yuan, Z., & Muntean, G.-M. (2021). QoE Ready to Respond: A QoE-Aware MEC Selection Scheme for DASH-Based Adaptive Video Streaming to Mobile Users. In *Proceedings of the 29th ACM International Conference on Multimedia* (p. 4016–4024). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3474085.3475325

Shu, C., Liu, Y., Gao, J., Yan, Z., & Shen, C. (2021). Channel-wise Knowledge Distillation for Dense Prediction. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 5291–5300). doi: 10.1109/ICCV48922.2021.00526

Singh, A., Hu, R., Goswami, V., Couairon, G., Galuba, W., Rohrbach, M., & Kiela, D. (2022). FLAVA: A Foundational Language And Vision Alignment Model. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 15617–15629). doi: 10.1109/CVPR52688.2022.01519

Singla, A., Robotham, T., Bhattacharya, A., Menz, W., P. Habets, E. A., & Raake, A. (2023). Saliency of Omnidirectional Videos with Different Audio Presentations: Analyses and Dataset. In *2023 15th International Conference on Quality of Multimedia Experience (QoMEX)* (pp. 264–269). doi: 10.1109/QoMEX58391.2023.10178588

Sinha, S., Bharadhwaj, H., Goyal, A., Larochelle, H., Garg, A., & Shkurti, F. (2021). DIBS: Diversity Inducing Information Bottleneck in Model Ensembles. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021* (pp. 9666–9674). AAAI Press. doi: 10.1609/AAAI.V35I11.17163

Sitzmann, V., Serrano, A., Pavel, A., Agrawala, M., Gutierrez, D., Masia, B., & Wetzstein, G. (2018). Saliency in VR: How Do People Explore Virtual Environments? *IEEE Transactions on Visualization and Computer Graphics*, *24*(4), 1633–1642. doi: 10.1109/TVCG.2018.2793599

Smith, J. T., Warrington, A., & Linderman, S. (2023). Simplified State Space Layers for Sequence Modeling. In *The Eleventh International Conference on Learning Representations.*

Sofianos, T., Sampieri, A., Franco, L., & Galasso, F. (2021). Space-Time-Separable Graph Convolutional Network for Pose Forecasting. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 11189–11198). doi: 10.1109/ICCV48922.2021.01102

Sohn, K., Lee, H., & Yan, X. (2015). Learning Structured Output Representation using Deep Conditional Generative Models. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 28 (NIPS).* Curran Associates, Inc. doi: 10.5555/2969442.2969628

Spiteri, K. (2021). *Video Adaptation for High-Quality Content Delivery* (Doctoral dissertation). doi: 10.7275/20604181

Spiteri, K., Sitaraman, R., & Sparacio, D. (2018). From Theory to Practice: Improving Bitrate Adaptation in the DASH Reference Player. In *Proceedings of the 9th ACM Multimedia Systems Conference* (p. 123–137). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3204949.3204953

Spiteri, K., Sitaraman, R., & Sparacio, D. (2019, jul). From Theory to Practice: Improving Bitrate Adaptation in the DASH Reference Player. *ACM Trans. Multimedia Comput. Commun. Appl.*, *15*(2s). doi: 10.1145/3336497

Spiteri, K., Urgaonkar, R., & Sitaraman, R. K. (2016). BOLA: Near-optimal bitrate adaptation for online videos. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications* (pp. 1–9). doi: 10.1109/INFOCOM.2016.7524428

Spiteri, K., Urgaonkar, R., & Sitaraman, R. K. (2020). BOLA: Near-Optimal Bitrate Adaptation for Online Videos. *IEEE/ACM Transactions on Networking*, *28*(4), 1698–1711. doi: 10.1109/TNET.2020.2996964

Srikanth, S., Ansari, J. A., Ram, R. K., Sharma, S., Murthy, J. K., & Krishna, K. M. (2019). INFER: INtermediate representations for FuturE pRediction. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 942–949). doi: 10.1109/IROS40897.2019.8968553

Statista Market Insights. (2023). *AR & VR - Worldwide | Statista Market Forecast*. Retrieved 2023-10-05, from https://www.statista.com/outlook/amo/ar-vr/worldwide

Stockhammer, T. (2011). Dynamic Adaptive Streaming over HTTP –: Standards and Design Principles. In *Proceedings of the Second Annual ACM Conference on Multimedia Systems* (p. 133–144). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/1943552.1943572

Stroud, J. C., Ross, D. A., Sun, C., Deng, J., & Sukthankar, R. (2020). D3D: Distilled 3D Networks for Video Action Recognition. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)* (pp. 614–623). doi: 10.1109/WACV45572.2020.9093274

Subjective video quality assessment methods for multimedia applications (Recommendations). (2021). International Telecommunication Union: Telecommunication Standardization Sector.

Sun, L., Mao, Y., Zong, T., Liu, Y., & Wang, Y. (2020). Flocking-Based Live Streaming of 360-Degree Video. In *Proceedings of the 11th ACM Multimedia Systems Conference* (p. 26–37). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3339825.3391856

Sun, Y., Yin, X., Jiang, J., Sekar, V., Lin, F., Wang, N., ... Sinopoli, B. (2016). CS2P: Improving Video Bitrate Selection and Adaptation with Data-Driven Throughput Prediction. In *Proceedings of the 2016 ACM SIGCOMM Conference* (p. 272–285). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2934872.2934898

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K. Weinberger (Eds.), *Advances in Neural Information Processing Systems* (Vol. 27). Curran Associates, Inc.

Tang, W., Wu, S., Vigier, T., & Da Silva, M. P. (2020). Influence of Emotions on Eye Behavior in Omnidirectional Content. In *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)* (pp. 1–6). doi: 10.1109/QoMEX48832.2020.9123126

Tang, Z., Cho, J., Lei, J., & Bansal, M. (2023). PERCEIVER-VL: Efficient Vision-and-Language Modeling with Iterative Latent Attention. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (pp. 4399–4409). doi: 10.1109/WACV56688.2023.00439

Taylor, S. J., & Letham, B. (2018). Forecasting at Scale. *The American Statistician*, *72*(1), 37–45. doi: 10.1080/00031305.2017.1380080

Tcha-Tokey, K., Christmann, O., Loup-Escande, E., & Richir, S. (2016, January). Proposition and Validation of a Questionnaire to Measure the User Experience in Immersive Virtual Environments. *International Journal of Virtual Reality*, *16*(1), 33–48. doi: 10.20870/IJVR.2016.16.1.2880

TensorfFlow. (2021, September). *TensorFlow 2 YOLOv4*. Retrieved from https://wiki.loliot.net/docs/lang/python/libraries/yolov4/python-yolov4-about/

Thiede, L., & Brahma, P. (2019). Analyzing the Variety Loss in the Context of Probabilistic Trajectory Prediction. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 9953–9962). doi: 10.1109/ICCV.2019.01005

Tian, G., & Liu, Y. (2012). Towards Agile and Smooth Video Adaptation in Dynamic HTTP Streaming. In *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies* (p. 109–120). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2413176.2413190

Tian, Y., Krishnan, D., & Isola, P. (2020). Contrastive Representation Distillation. In *International Conference on Learning Representations.*

Tishby, N., & Zaslavsky, N. (2015). Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)* (pp. 1–5). doi: 10.1109/ITW.2015.7133169

Toet, A., Heijn, F., Brouwer, A.-M., Mioch, T., & van Erp, J. B. F. (2020). An Immersive Self-Report Tool for the Affective Appraisal of 360° VR Videos. *Frontiers in Virtual Reality*, *1*. doi: 10.3389/frvir.2020.552587

Togou, M. A., & Muntean, G.-M. (2022). An Elastic DASH-based Bitrate Adaptation Scheme for Smooth On-Demand Video Streaming. In *2022 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)* (pp. 1–6). doi: 10.1109/BMSB55706.2022.9828754

Tong, Z., Song, Y., Wang, J., & Wang, L. (2022). VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in Neural Information Processing Systems* (Vol. 35, pp. 10078–10093). Curran Associates, Inc.

Toni, L., Aparicio-Pardo, R., Pires, K., Simon, G., Blanc, A., & Frossard, P. (2015, feb). Optimal Selection of Adaptive Streaming Representations. *ACM Trans. Multimedia Comput. Commun. Appl.*, *11*(2s). doi: 10.1145/2700294

Toni, L., Aparicio-Pardo, R., Simon, G., Blanc, A., & Frossard, P. (2014). Optimal Set of Video Representations in Adaptive Streaming. In *Proceedings of the 5th ACM Multimedia Systems Conference* (p. 271–282). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2557642.2557652

Toni, L., & Frossard, P. (2017). Optimal Representations for Adaptive Streaming in Interactive Multiview Video Systems. *IEEE Transactions on Multimedia*, *19*(12), 2775–2787. doi: 10.1109/TMM.2017.2713644

Toni, L., Thomos, N., & Frossard, P. (2013). Interactive free viewpoint video streaming using prioritized network coding. In *2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)* (pp. 446–451). doi: 10.1109/MMSP.2013.6659330

Tung, F., & Mori, G. (2019). Similarity-Preserving Knowledge Distillation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 1365–1374). doi: 10.1109/ICCV.2019.00145

UCSF. (2021). *Sample Size Calculators for designing clinical research.* Retrieved from https://sample-size.net/correlation-sample-size/

Vallez, N., Bueno, G., Deniz, O., & Blanco, S. (2022). Diffeomorphic transforms for data augmentation of highly variable shape and texture objects. *Computer Methods and Programs in Biomedicine*, *219*, 106775. doi: 10.1016/j.cmpb.2022.106775

van Amersfoort, J., Smith, L., Teh, Y. W., & Gal, Y. (2020, 13–18 Jul). Uncertainty Estimation Using a Single Deep Deterministic Neural Network. In H. D. III & A. Singh (Eds.), *Proceedings of the 37th International Conference on Machine Learning* (Vol. 119, pp. 9690–9700). PMLR.

van der Hooft, J., Petrangeli, S., Claeys, M., Famaey, J., & De Turck, F. (2015). A learning-based algorithm for improved bandwidth-awareness of adaptive streaming clients. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)* (pp. 131–138). doi: 10.1109/INM.2015.7140285

van der Hooft, J., Petrangeli, S., Wauters, T., Huysegems, R., Alface, P. R., Bostoen, T., & De Turck, F. (2016). HTTP/2-Based Adaptive Streaming of HEVC Video Over 4G/LTE Networks. *IEEE Communications Letters*, *20*(11), 2177–2180. doi: 10.1109/LCOMM.2016.2601087

van der Hooft, J., Wauters, T., De Turck, F., Timmerer, C., & Hellwagner, H. (2019). Towards 6DoF HTTP Adaptive Streaming Through Point Cloud Compression. In *Proceedings of the 27th ACM International Conference on Multimedia* (p. 2405–2413). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3343031.3350917

van der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, *9*(86), 2579–2605.

van Tulder, G., & de Bruijne, M. (2019). Learning Cross-Modality Representations From Multi-Modal Images. *IEEE Transactions on Medical Imaging*, *38*(2), 638–648. doi: 10.1109/TMI.2018.2868977

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is All you Need. In I. Guyon et al. (Eds.), *Advances in Neural Information Processing Systems* (Vol. 30). Curran Associates, Inc.

Voigt-Antons, J.-N., Lehtonen, E., Palacios, A. P., Ali, D., Kojic, T., & Möller, S. (2020). Comparing Emotional States Induced by 360° Videos Via Head-Mounted Display and Computer Screen. In *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)* (pp. 1–6). doi: 10.1109/QoMEX48832.2020.9123125

Walker, J., Doersch, C., Gupta, A., & Hebert, M. (2016). An uncertain future: Forecasting from static images using variational autoencoders. In *Proceedings of the 14th European Conference on Computer Vision (ECCV)* (pp. 835–851). Springer. doi: 10.1007/978-3-319-46478-7_51

Walline, J. J. (2001). Designing Clinical Research: an Epidemiologic Approach, 2nd Ed. *Optometry and Vision Science*, *78*(8).

Wang, C., Rizk, A., & Zink, M. (2016). SQUAD: A Spectrum-Based Quality Adaptation for Dynamic Adaptive Streaming over HTTP. In *Proceedings of the 7th International Conference on Multimedia Systems*. New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2910017.2910593

Wang, J., Da Silva, M. P., Le Callet, P., & Ricordel, V. (2013). Computational Model of Stereoscopic 3D Visual Saliency. *IEEE Transactions on Image Processing*, *22*(6), 2151–2165. doi: 10.1109/TIP.2013.2246176

Wang, S., Li, B. Z., Khabsa, M., Fang, H., & Ma, H. (2020). *Linformer: Self-Attention with Linear Complexity*. Retrieved from https://arxiv.org/abs/2006.04768

Wang, T., Yuan, L., Zhang, X., & Feng, J. (2019). Distilling Object Detectors With Fine-Grained Feature Imitation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4928–4937). doi: 10.1109/CVPR.2019.00507

Wang, Y., Zhang, Y., Liu, Y., Lin, Z., Tian, J., Zhong, C., ... He, Z. (2021). ACN: Adversarial Co-Training Network for Brain Tumor Segmentation with Missing Modalities. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021: 24th International Conference, Strasbourg, France, September 27 – October 1, 2021, Proceedings, Part VII* (p. 410–420). Berlin, Heidelberg: Springer-Verlag. doi: 10.1007/978-3-030-87234-2_39

Wei, S., Luo, C., & Luo, Y. (2023, June). MMANet: Margin-Aware Distillation and Modality-Aware Regularization for Incomplete Multimodal Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 20039–20049).

Whittle, P. (1951). *Hypothesis Testing in Time Series Analysis*. Almqvist & Wiksells boktr.

Wu, C., Tan, Z., Wang, Z., & Yang, S. (2017). A Dataset for Exploring User Behaviors in VR Spherical Video Streaming. In *Proceedings of the 8th ACM on Multimedia Systems Conference* (p. 193–198). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3083187.3083210

Wu, C., Wang, Z., & Sun, L. (2021). PAAS: A Preference-Aware Deep Reinforcement Learning Approach for 360° Video Streaming. In *Proceedings of the 31st ACM Workshop on Network and Operating Systems Support for Digital Audio and Video* (p. 34–41). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3458306.3460995

Wu, H., Xu, J., Wang, J., & Long, M. (2021). Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, & J. W. Vaughan (Eds.), *Advances in Neural Information Processing Systems* (Vol. 34, pp. 22419–22430). Curran Associates, Inc.

Wu, P., Majumdar, A., Stone, K., Lin, Y., Mordatch, I., Abbeel, P., & Rajeswaran, A. (2023). *Masked Trajectory Models for Prediction, Representation, and Control.* Retrieved from https://arxiv.org/abs/2305.02968

Xiao, H., Xu, C., Feng, Z., Ding, R., Yang, S., Zhong, L., ... Muntean, G.-M. (2022). A Transcoding-Enabled 360° VR Video Caching and Delivery Framework for Edge-Enhanced Next-Generation Wireless Networks. *IEEE Journal on Selected Areas in Communications*, *40*(5), 1615–1631. doi: 10.1109/JSAC.2022.3145813

Xing, C., Rostamzadeh, N., Oreshkin, B., & O. Pinheiro, P. O. (2019). Adaptive Cross-Modal Few-shot Learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 32). Curran Associates, Inc.

Xu, M., Li, C., Chen, Z., Wang, Z., & Guan, Z. (2019). Assessing Visual Quality of Omnidirectional Videos. *IEEE Transactions on Circuits and Systems for Video Technology*, *29*(12), 3516–3530. doi: 10.1109/TCSVT.2018.2886277

Xu, M., Li, C., Zhang, S., & Le Callet, P. (2020). State-of-the-Art in 360° Video/Image Processing: Perception, Assessment and Compression. *IEEE Journal of Selected Topics in Signal Processing*, *14*(1), 5–26. doi: 10.1109/JSTSP.2020.2966864

Xu, M., Song, Y., Wang, J., Qiao, M., Huo, L., & Wang, Z. (2019). Predicting Head Movement in Panoramic Video: A Deep Reinforcement Learning Approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(11), 2693–2708. doi: 10.1109/TPAMI.2018.2858783

Xu, Y., Dong, Y., Wu, J., Sun, Z., Shi, Z., Yu, J., & Gao, S. (2018). Gaze Prediction in Dynamic 360° Immersive Videos. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5333–5342). doi: 10.1109/CVPR.2018.00559

Xu, Y., Zhang, Z., & Gao, S. (2022). Spherical DNNs and Their Applications in 360° Images and Videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(10), 7235–7252. doi: 10.1109/TPAMI.2021.3100259

Xue, T., Ali, A. E., Ding, G., & Cesar, P. (2021). Investigating the Relationship between Momentary Emotion Self-Reports and Head and Eye Movements in HMD-Based 360° VR Video Watching. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems.* New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3411763.3451627

Xue, T., Ali, A. E., Zhang, T., Ding, G., & Cesar, P. (2021). CEAP-360VR: A Continuous Physiological and Behavioral Emotion Annotation Dataset for 360° VR Videos. *IEEE Transactions on Multimedia*, *25*, 243–255. doi: 10.1109/TMM.2021.3124080

Xue, T., El Ali, A., Zhang, T., Ding, G., & Cesar, P. (2021). RCEA-360VR: Real-Time, Continuous Emotion Annotation in 360° VR Videos for Collecting Precise Viewport-Dependent Ground Truth Labels. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems.* New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3411764.3445487

Yan, F. Y., Ayers, H., Zhu, C., Fouladi, S., Hong, J., Zhang, K., ... Winstein, K. (2020, February). Learning in situ: a randomized experiment in video streaming. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)* (pp. 495–511). Santa Clara, CA: USENIX Association.

Yang, L., Xu, M., Guo, Y., Deng, X., Gao, F., & Guan, Z. (2022). Hierarchical Bayesian LSTM for Head Trajectory Prediction on Omnidirectional Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(11), 7563–7580. doi: 10.1109/TPAMI.2021.3117019

Yang, Q., Li, Y., Li, C., Wang, H., Yan, S., Wei, L., ... Frossard, P. (2023). SVGC-AVA: 360-Degree Video Saliency Prediction with Spherical Vector-Based Graph Convolution and Audio-Visual Attention. *IEEE Transactions on Multimedia*, 1–16. doi: 10.1109/TMM.2023.3306596

Yang, Y., Geng, S., Zhang, B., Zhang, J., Wang, Z., Zhang, Y., & Doermann, D. (2023, Jun 06). Long term 5G network traffic forecasting via modeling non-stationarity with deep learning. *Communications Engineering*, *2*(1), 33. doi: 10.1038/s44172-023-00081-4

Yaqoob, A., Bi, T., & Muntean, G.-M. (2019). A DASH-based Efficient Throughput and Buffer Occupancy-based Adaptation Algorithm for Smooth Multimedia Streaming. In *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)* (pp. 643–649). doi: 10.1109/IWCMC.2019.8766648

Yaqoob, A., Bi, T., & Muntean, G.-M. (2020). A Survey on Adaptive 360° Video Streaming: Solutions, Challenges and Opportunities. *IEEE Communications Surveys & Tutorials*, *22*(4), 2801–2838. doi: 10.1109/COMST.2020.3006999

Yaqoob, A., & Muntean, G.-M. (2020). A Weighted Tile-based Approach for Viewport Adaptive 360° Video Streaming. In *2020 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)* (pp. 1–7). doi: 10.1109/BMSB49480.2020.9379517

Yaqoob, A., & Muntean, G.-M. (2021). A Combined Field-of-View Prediction-Assisted Viewport Adaptive Delivery Scheme for 360° Videos. *IEEE Transactions on Broadcasting*, *67*(3), 746–760. doi: 10.1109/TBC.2021.3105022

Yaqoob, A., & Muntean, G.-M. (2023, aug). Advanced Predictive Tile Selection Using Dynamic Tiling for Prioritized 360° Video VR Streaming. *ACM Trans. Multimedia Comput. Commun. Appl.*, *20*(1). doi: 10.1145/3603146

Yaqoob, A., Togou, M. A., & Muntean, G.-M. (2022). Dynamic Viewport Selection-Based Prioritized Bitrate Adaptation for Tile-Based 360° Video Streaming. *IEEE Access*, *10*, 29377–29392. doi: 10.1109/ACCESS.2022.3157339

Ye, Z., Li, Q., Ma, X., Zhao, D., Jiang, Y., Ma, L., ... Muntean, G.-M. (2023). VRCT: A Viewport Reconstruction-Based 360° Video Caching Solution for Tile-Adaptive Streaming. *IEEE Transactions on Broadcasting*, *69*(3), 691–703. doi: 10.1109/TBC.2023.3274350

Yin, Q., Wu, S., & Wang, L. (2017, jul). Unified Subspace Learning for Incomplete and Unlabeled Multi-View Data. *Pattern Recogn.*, *67*(C), 313–327. doi: 10.1016/j.patcog .2017.01.035

Yin, X., Jindal, A., Sekar, V., & Sinopoli, B. (2015). A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication* (p. 325–338). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2785956.2787486

Yin, X., Sekar, V., & Sinopoli, B. (2014). Toward a Principled Framework to Design Dynamic Adaptive Streaming Algorithms over HTTP. In *Proceedings of the 13th ACM Workshop on Hot Topics in Networks* (p. 1–7). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2670518.2673877

Yu, J., & Liu, Y. (2019). Field-of-View Prediction in 360-Degree Videos with Attention-Based Neural Encoder-Decoder Networks. In *Proceedings of the 11th ACM Workshop on Immersive Mixed and Virtual Environment Systems* (p. 37–42). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3304113.3326118

Yuan, Y., Weng, X., Ou, Y., & Kitani, K. (2021). AgentFormer: Agent-Aware Transformers for Socio-Temporal Multi-Agent Forecasting. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 9793–9803). doi: 10.1109/ ICCV48922.2021.00967

Yule, G. U. (1921). On the Time-Correlation Problem, with Especial Reference to the Variate-Difference Correlation Method. *Journal of the Royal Statistical Society*, *84*(4), 497–526. doi: 10.1111/j.2397-2335.1921.tb00740.x

Yule, G. U. (1927). On a Method of Investigating Periodicities in Disturbed Series, with Special Reference to Wolfer's Sunspot Numbers. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, *226*, 267–298.

Yun, H., Lee, S., & Kim, G. (2022). Panoramic Vision Transformer for Saliency Detection in 360° Videos. In S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, & T. Hassner (Eds.), *Computer Vision – ECCV 2022* (pp. 422–439). Cham: Springer Nature Switzerland.

Zagoruyko, S., & Komodakis, N. (2017). Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. In *International Conference on Learning Representations.*

Zara, G., Conti, A., Roy, S., Lathuilière, S., Rota, P., & Ricci, E. (2023, October). The Unreasonable Effectiveness of Large Language-Vision Models for Source-Free Video Domain Adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (p. 10307-10317).

Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2023, Jun.). Are Transformers Effective for Time Series Forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence*, *37*(9), 11121–11128. doi: 10.1609/aaai.v37i9.26317

Zeng, J., Liu, T., & Zhou, J. (2022). Tag-Assisted Multimodal Sentiment Analysis under Uncertain Missing Modalities. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (p. 1545–1554). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3477495.3532064

Zerman, E., Kulkarni, R., & Smolic, A. (2021). User Behaviour Analysis of Volumetric Video in Augmented Reality. In *2021 13th International Conference on Quality of Multimedia Experience (QoMEX)* (pp. 129–132). doi: 10.1109/QoMEX51781.2021.9465456

Zhang, A., Li, Q., Chen, Y., Ma, X., Zou, L., Jiang, Y., ... Muntean, G.-M. (2021). Video Super-Resolution and Caching—An Edge-Assisted Adaptive Video Streaming Solution. *IEEE Transactions on Broadcasting*, *67*(4), 799–812. doi: 10.1109/TBC.2021.3071010

Zhang, R., Liu, J., Liu, F., Huang, T., Tang, Q., Wang, S., & Yu, F. R. (2021). Buffer-Aware Virtual Reality Video Streaming With Personalized and Private Viewport Prediction. *IEEE Journal on Selected Areas in Communications*, *40*(2), 694–709. doi: 10.1109/JSAC.2021.3119144

Zhang, X., Cheung, G., Le Callet, P., & Tan, J. Z. G. (2020). Sparse Directed Graph Learning for Head Movement Prediction in 360 Video Streaming. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2678–2682). doi: 10.1109/ICASSP40776.2020.9053598

Zhang, X., Toni, L., Frossard, P., Zhao, Y., & Lin, C. (2019). Adaptive Streaming in Interactive Multiview Video Systems. *IEEE Transactions on Circuits and Systems for Video Technology*, *29*(4), 1130–1144. doi: 10.1109/TCSVT.2018.2819804

Zhang, Y., He, N., Yang, J., Li, Y., Wei, D., Huang, Y., ... Zheng, Y. (2022). mmFormer: Multimodal Medical Transformer For Incomplete Multimodal Learning Of Brain Tumor Segmentation. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part V* (p. 107–117). Berlin, Heidelberg: Springer-Verlag. doi: 10.1007/978-3-031-16443-9_11

Zhang, Z., Xu, Y., Yu, J., & Gao, S. (2018). Saliency Detection in 360° Videos. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part VII* (p. 504–520). Berlin, Heidelberg: Springer-Verlag. doi: 10.1007/978-3-030-01234-2_30

Zhao, B., Cui, Q., Song, R., Qiu, Y., & Liang, J. (2022). Decoupled Knowledge Distillation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 11943–11952). doi: 10.1109/CVPR52688.2022.01165

Zhao, H., Gao, J., Lan, T., Sun, C., Sapp, B., Varadarajan, B., ... Anguelov, D. (2021, 16–18 Nov). TNT: Target-driven Trajectory Prediction. In J. Kober, F. Ramos, & C. Tomlin (Eds.), *Proceedings of the 2020 Conference on Robot Learning* (Vol. 155, pp. 895–904). PMLR.

Zhao, H., & Wildes, R. P. (2021). Where are you heading? Dynamic Trajectory Prediction with Expert Goal Examples. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 7609–7618). doi: 10.1109/ICCV48922.2021.00753

Zhao, J., Li, R., & Jin, Q. (2021, August). Missing Modality Imagination Network for Emotion Recognition with Uncertain Missing Modalities. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 2608–2618). Online: Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.203

Zhao, L., Peng, X., Chen, Y., Kapadia, M., & Metaxas, D. N. (2020). Knowledge As Priors: Cross-Modal Knowledge Generalization for Datasets Without Superior Knowledge. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 6527–6536). doi: 10.1109/CVPR42600.2020.00656

Zheng, Y., Yang, Y., Mo, K., Li, J., Yu, T., Liu, Y., ... Guibas, L. J. (2022). GIMO: Gaze-Informed Human Motion Prediction In Context. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIII* (p. 676–694). Berlin, Heidelberg: Springer-Verlag. doi: 10.1007/978-3-031-19778-9_39

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021, May). Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(12), 11106–11115. doi: 10.1609/aaai.v35i12.17325

Zhou, T., Canu, S., Vera, P., & Ruan, S. (2020). Brain Tumor Segmentation with Missing Modalities via Latent Multi-Source Correlation Representation. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part IV* (p. 533–541). Berlin, Heidelberg: Springer-Verlag. doi: 10.1007/978-3-030-59719-1_52

Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., & Jin, R. (2022, 17–23 Jul). FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, & S. Sabato (Eds.),

Proceedings of the 39th International Conference on Machine Learning (Vol. 162, pp. 27268–27286). PMLR.

Zhu, J., Tang, S., Chen, D., Yu, S., Liu, Y., Rong, M., ... Wang, X. (2021). Complementary Relation Contrastive Distillation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 9256–9265). doi: 10.1109/CVPR46437.2021.00914

Zimmerman, T. G., Lanier, J., Blanchard, C., Bryson, S., & Harvill, Y. (1986). A Hand Gesture Interface Device. In *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface* (p. 189–192). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/29933.275628

Zink, M., Sitaraman, R., & Nahrstedt, K. (2019). Scalable 360° Video Stream Delivery: Challenges, Solutions, and Opportunities. *Proceedings of the IEEE*, *107*(4), 639–650. doi: 10.1109/JPROC.2019.2894817

Zou, X. K., Erman, J., Gopalakrishnan, V., Halepovic, E., Jana, R., Jin, X., ... Sinha, R. K. (2015). Can Accurate Predictions Improve Video Streaming in Cellular Networks? In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications* (p. 57–62). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2699343.2699359

List of Figures

List of Tables

List of Algorithms

Deep learning pour le streaming adaptatif de vidéos à 360° en réalité virtuelle

Quentin GUIMARD

Résumé

La réalité virtuelle (VR) a évolué de manière significative ces dernières années. Les casques immersifs devenant de plus en plus abordables et populaires, de nombreuses applications sont à l'horizon, des vidéos à 360° aux formations interactives en passant par les environnements virtuels collaboratifs. Cependant, pour atteindre des niveaux élevés de qualité perçue, la bande passante du réseau et les ressources de calcul nécessaires peuvent être supérieures de plusieurs ordres de grandeur à celles requises pour un contenu 2D traditionnel. Pour pallier ce problème, des stratégies de streaming qui adaptent le débit vidéo aux conditions du réseau et à l'orientation de la tête de la personne ont été mises en œuvre afin d'améliorer la qualité d'expérience. Étant donné que la plupart des algorithmes de débit adaptatif reposent sur l'utilisation d'une mémoire tampon vidéo suffisamment grande pour compenser les fluctuations de la bande passante, l'algorithme doit savoir où la personne regardera quelques secondes avant la lecture pour adapter correctement la qualité. La qualité d'expérience pour le streaming 360° dépend donc de la prédiction des mouvements de la tête en VR. Malheureusement, il s'agit d'un problème difficile en raison (i) du caractère aléatoire des mouvements humains, (ii) de la diversité des trajectoires de tête des personnes qui regardent des vidéos à 360° ce qui entraîne une ambiguïté entre les trajectoires passées, et (iii) des nombreux facteurs qui influencent le comportement, l'attention et les mouvements de la personne en VR. Afin de concevoir des systèmes de streaming VR qui s'adaptent mieux à chaque personne, il est important de comprendre les différents facteurs, leurs interactions et leurs effets sur le comportement humain. La collecte et l'exploitation de nouvelles données relatives à ces facteurs pourraient aider à désambiguïser les trajectoires la tête et à améliorer leur prédiction. Ce travail est divisé en quatre contributions principales. Premièrement, nous avons proposé un nouveau framework de deep learning variationnel pour prédire de multiples trajectoires possibles de mouvements de tête afin de mieux prendre en compte la diversité des trajectoires. Nous avons montré que notre modèle surpasse les performances de concurrents adaptés du domaine de la conduite autonome, réduisant l'erreur jusqu'à 41 % sur quatre datasets. Nous avons ensuite proposé un nouveau simulateur de streaming 360° afin de mesurer les gains système de notre framework et de permettre de comparer facilement les stratégies de streaming adaptatif. Nous avons montré que la prédiction de trajectoires multiples conduit à une plus grande équité entre les usagers, avec des gains de qualité atteignant jusqu'à 10 % pour 20 à 30 % des personnes. En parallèle, nous avons mené des expériences avec des personnes et des analyses statistiques pour mieux comprendre l'interaction entre le contenu immersif, l'attention et les émotions. Nous avons observé que le degré d'activation physiologique de la personne était corrélé à l'attention portée aux objets, et nous avons quantifié les effets des émotions sur la prédictibilité des mouvements de la tête. Enfin, nous avons voulu tirer parti des données liées aux émotions afin d'apprendre de meilleures représentations et d'améliorer la prédiction des mouvements de la tête. Inspirés par les travaux récents sur la distillation cross-modale et les modèles de fondation multimodaux, nous avons commencé à travailler sur une nouvelle architecture de deep learning multimodale capable d'apprendre des représentations transférables de modalités qui ne sont disponibles qu'au moment de l'apprentissage. Nous avons obtenu des résultats préliminaires qui surpassent de 21 % l'état de l'art existant tout en réduisant considérablement le nombre de paramètres.

Mots-clés : Apprentissage profond, Réseaux de neurones artificiels, Réalité virtuelle, Streaming, Régression, Multimedia

Deep learning for adaptive 360° video streaming in virtual reality

Quentin GUIMARD

Abstract

Virtual reality (VR) has evolved significantly in recent years. As head-mounted displays become more affordable and popular, new opportunities for high-quality immersive experiences are opening up. A variety of exciting applications are on the horizon, from 360° videos to interactive training simulations and collaborative virtual environments. However, to achieve high levels of perceptual quality, the required network bandwidth and GPU computing resources can be orders of magnitude higher than those required for traditional 2D content. To mitigate this, adaptive streaming strategies have been implemented to improve the quality of experience (QoE) for people watching 360° videos over the Internet. This is done by adapting the video quality to the network conditions and the user's head orientation. Since most adaptive bitrate algorithms rely on using a large enough video buffer to compensate for bandwidth fluctuations, the algorithm needs to know where the person will be looking a few seconds before playback to make the appropriate quality decisions. Improving the QoE for 360° video streaming therefore depends on accurately predicting the user's viewport in VR. Unfortunately, viewport prediction is a challenging problem due to (i) the inherent randomness of human motion, (ii) the diversity of head trajectories among people watching 360° video, which leads to ambiguity between similar past trajectories, and (iii) the many factors that influence user behavior, attention, and movement in VR. In order to design VR streaming systems that can better adapt to each user, it is important to understand the different factors, their interactions, and their effects on human behavior. Collecting and exploiting additional data modalities related to these factors could help disambiguate head trajectories and improve viewport prediction. The work covered in this manuscript touches on many areas, including the design of various multimodal deep learning architectures applied to regression, dynamic optimization problems, time series forecasting, and user experiments along with associated statistical analyses. This work is divided into four main contributions. First, we studied the similarity between head motion trajectories and proposed a new variational deep learning framework for predicting multiple possible head motion trajectories to better account for trajectory diversity. While our framework is compatible with any sequence-to-sequence architecture, we implemented a flexible and lightweight stochastic prediction model and showed that it outperformed competitors adapted from the self-driving domain by up to 41% on four datasets. We then proposed a new trace-driven 360° video streaming simulator to measure the system gains of our framework and provide a way to easily compare adaptive streaming strategies. We showed that predicting multiple trajectories leads to higher fairness among simulated users, with gains for 20% to 30% of users reaching up to 10% in visual quality. In parallel, we conducted user experiments and statistical analyses to better understand the interaction between immersive content, attention, and emotions, as well as the effects of emotions on user motion. We observed that user arousal correlated with the accuracy of high-level saliency. We also quantified the effects of valence and arousal on the predictability of head movements and their interaction with spatial information. Finally, we wanted to take advantage of additional emotion-related data modalities to learn better representations and improve viewport prediction. Motivated by recent work on cross-modal knowledge distillation and multimodal foundation models, we initiated work on a new multimodal deep architecture able to learn transferable representations of modalities that are only available at training time. We obtained early results outperforming the existing state-of-the-art by up to 21% while greatly reducing the number of parameters.

Keywords: Deep learning, Artificial neural networks, Virtual reality, Streaming, Regression, Multimedia