



Politecnico
di Bari

Repository Istituzionale dei Prodotti della Ricerca del Politecnico di Bari

Design of ai-based viewport-adaptive immersive video streaming systems

This is a PhD Thesis

Original Citation:

Design of ai-based viewport-adaptive immersive video streaming systems / Farooq, M.. - ELETTRONICO. - (2026).

Availability:

This version is available at <http://hdl.handle.net/11589/308401> since: 2026-10-14

Published version

DOI:

Publisher: Politecnico di Bari

Terms of use:

(Article begins on next page)



Politecnico
di Bari

Design of AI-Based Viewport-Adaptive Immersive Video Streaming Systems

Muhammad Farooq

October 13, 2026

Coordinator: Prof. Mario Carpentieri

Supervisor: Prof. Saverio Mascolo

Co-Supervisor: Prof. Luca De Cicco

38th cycle, Department of Electrical and Information Engineering

Politecnico di Bari, Italy

Acknowledgments

I would like to express my sincere gratitude to my supervisors, Prof. Luca De Cicco and Prof. Saverio Mascolo, for their continuous guidance and valuable feedback throughout my PhD research. Their support and encouragement have been essential to the development of this work.

I am grateful to Dr. Gioacchino Manfredi for his guidance and helpful discussions during my research. His assistance and willingness to share his knowledge greatly contributed to the progress of my work.

I would also like to thank all members of the C3LAB for creating a supportive and collaborative research environment. Working with such a dedicated and inspiring group has been a valuable and enriching experience during my PhD journey.

Finally, I would like to thank Prof. Maria Martini for her supervision and support during my research period abroad at Kingston University, London, UK. Her guidance and suggestions were very valuable to the progress of my research and development of this thesis.

Abstract

Immersive Virtual Reality (VR) applications have been widely developed in recent years. However, streaming of 360° video content is a key issue in multimedia networking. Compared to traditional videos, 360° video content captures the whole spherical view, while the user only watches a small portion of the video called Field of View (FoV) at any time. Conventional systems stream the entire projected panoramic video frame at the same quality, which results in bandwidth loss as a user's FoV contains only a limited part of the full video. This thesis presents the experimental design and analysis of AI-based viewport-adaptive streaming systems, which aim at overcoming this inefficient bandwidth utilization by integrating a Machine Learning (ML)-based viewport prediction model with multi-bitrate tile-based video delivery using the Quick UDP Internet Connections (QUIC) transport protocol.

This work is classified into two parts where the first part focuses on comparing viewport prediction results obtained by various ML models. Furthermore, a comparison between Long Short-Term Memory (LSTM) model results and state-of-the-art approaches is also carried out. Systematic comparison includes six ML models—LSTM, Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), Convolutional Neural Network (CNN), Linear Regression (LR) and Multi-layer Perceptron (MLP)—to predict user FoV based on Head-Mounted Display (HMD) motion traces with horizons starting from next-frame up to five seconds ahead considering a frame rate of 30fps. Results indicate that the CNN model provides the best performance in terms of short-term horizon (next-frame viewport prediction), whereas GRU and LSTM models

are more performative in the long-term horizon (5-seconds) as these models have the ability to capture temporal dynamics in the head motion. Comparing viewport prediction results of LSTM model with a state-of-the-art approach called Multi-modal Fusion Transformer (MFTR) shows that the LSTM model outperforms the other in terms of the Average Prediction (AP) accuracy across all considered horizons. This shows that recurrent models with appropriate parameter configuration perform better with less computational effort compared to much more complex architectures.

Second part of thesis presents an end-to-end streaming architecture that combines viewport prediction using an LSTM model with adaptive tile-based bitrate allocation over QUIC. The corresponding 2D projection of the 360° video frame is divided into a 3×3 tile grid, where the predicted FoV tiles are assigned a higher bitrate of 1.0 Mbps, with neighboring and non-visible tiles are allocated lower bitrates. Multiplexed streams in QUIC perform parallel transmission of tiles with two priorities (high and low) without Head-of-Line (HoL) blocking, which reduces the latency of FoV content delivery. Experimental setup contains real user HMD traces which are used to predict the users' FoVs. The prediction results are integrated into an adaptive bitrate decision mechanism that accounts for network bandwidth constraints. The system maintains a stable playback buffer by ensuring that the total bitrate assigned to all tiles remains within the available bandwidth. Moreover, the FoV missing ratio is compared with overall missing-tile ratio across the streaming session, showing that the average FoV missing ratio remains lower than the missing ratio computed over all tiles.

Although the results prove the effectiveness of streaming system, there are still some limitations that should be considered. Viewport prediction accuracy is high for next-frame prediction; however, it gradually decreases as prediction horizon increases. Future work will consider the following steps: integration of attention-based and hybrid neural network structures, joint bitrate and tile-prioritization decisions based on Deep Reinforcement Learning (DRL), evaluation using different network traces such as mobile 5G and edge computing platform, and real-world deployment.

Abstract

(Italian)

Negli ultimi anni sono state ampiamente sviluppate applicazioni di VR immersiva. Tuttavia, lo streaming di contenuti video a 360° rappresenta una sfida cruciale nelle reti multimediali. Rispetto ai video tradizionali, i contenuti a 360° catturano l'intera visuale sferica, mentre l'utente ne osserva solo una porzione ridotta — definita FoV — in un dato momento. I sistemi convenzionali trasmettono l'intero fotogramma video panoramico proiettato con la medesima qualità, comportando uno spreco di larghezza di banda, dato che il FoV dell'utente comprende solo una parte limitata dell'intero video. Questa tesi presenta la progettazione sperimentale e l'analisi di sistemi di streaming adattivo alla viewport basati su intelligenza artificiale, i quali mirano a superare l'inefficiente utilizzo della larghezza di banda integrando un modello di previsione della viewport basato su machine learning con la distribuzione video a più bitrate basata su tile, utilizzando il protocollo di trasporto QUIC.

Il presente lavoro è suddiviso in due parti: la prima si concentra sul confronto dei risultati di previsione della viewport ottenuti tramite diversi modelli di ML. Viene inoltre effettuato un confronto tra i risultati del modello LSTM e quelli di approcci allo stato dell'arte. L'analisi comparativa sistematica include sei modelli di ML (LSTM, RNN, GRU, CNN, LR e MLP) per la previsione del FoV dell'utente basata sulle tracce di movimento dell'HMD, considerando orizzonti temporali che vanno dal fotogramma successivo fino a cinque secondi nel futuro (con un frame rate di 30 fps). I risultati indicano che il modello CNN offre le prestazioni migliori per l'orizzonte a

breve termine (previsione della viewport al fotogramma successivo), mentre i modelli GRU e LSTM risultano più performanti sull'orizzonte a lungo termine (5 secondi), grazie alla loro capacità di catturare le dinamiche temporali del movimento della testa. Il confronto tra i risultati di previsione della viewport del modello LSTM e un approccio allo stato dell'arte denominato MFTR evidenzia che il modello LSTM supera l'altro in termini di accuratezza AP per tutti gli orizzonti temporali considerati. Ciò dimostra che i modelli ricorrenti, con un'adequata configurazione dei parametri, offrono prestazioni migliori a fronte di un minore sforzo computazionale rispetto ad architetture molto più complesse.

La seconda parte della tesi presenta un'architettura di streaming end-to-end che combina la previsione della viewport, basata su un modello LSTM, con un'allocazione adattiva del bitrate a livello di tile su protocollo QUIC. La proiezione 2D del fotogramma video a 360° viene suddivisa in una griglia di tile 3×3 : alle tile previste all'interno del campo visivo (FoV) viene assegnato un bitrate più elevato (1,0 Mbps), mentre alle tile adiacenti e a quelle non visibili vengono assegnati bitrate inferiori. I flussi multiplati su QUIC consentono la trasmissione parallela delle tile secondo due livelli di priorità (alta e bassa) senza incorrere nel blocco HOL (Head-of-Line), riducendo così la latenza di consegna dei contenuti del FoV. La configurazione sperimentale utilizza tracce reali di HMD degli utenti per prevedere i loro FoV. I risultati della previsione vengono integrati in un meccanismo di decisione del bitrate adattivo che tiene conto dei vincoli di larghezza di banda della rete. Il sistema mantiene stabile il buffer di riproduzione garantendo che il bitrate totale assegnato a tutte le tile rimanga entro i limiti della larghezza di banda disponibile. Inoltre, il tasso di mancata copertura del FoV viene confrontato con il tasso complessivo di tile mancanti nell'intera sessione di streaming, evidenziando come il tasso medio di mancata copertura del FoV rimanga inferiore al tasso calcolato sull'insieme di tutte le tile.

Sebbene i risultati dimostrino l'efficacia del sistema di streaming, permangono alcune limitazioni da considerare. La precisione nella previsione della viewport è elevata

per il fotogramma successivo; tuttavia, diminuisce progressivamente all'aumentare dell'orizzonte di previsione. Gli sviluppi futuri riguarderanno: l'integrazione di strutture di reti neurali ibride e basate su meccanismi di attenzione; decisioni congiunte su bitrate e prioritizzazione dei tile basate su DRL; la valutazione mediante diverse tracce di rete (come reti mobili 5G e piattaforme di edge computing); e l'implementazione in scenari reali.

Table of Contents

List of Figures	i
List of Tables	iii
List of Abbreviations	iii
Scientific Contributions	vii
1 Introduction	1
1.1 Capturing 360° videos	1
1.2 Delivery of 360° videos over the Internet	2
1.3 Projecting 360° video	3
1.4 Low latency and bandwidth efficiency in 360° video streaming	4
1.5 Visibility of 360° videos	6
1.6 360° video streaming approaches	6
1.7 Input for viewport estimation	8
1.8 Transport of 360° video streaming	9
1.9 Thesis Outline	11
1.10 Summary of chapter	12
2 Background	13
2.1 Video Representation and Projection Formats	14
2.2 Tile-Based Adaptive Streaming	16

TABLE OF CONTENTS

2.3	Viewport Prediction Techniques	18
2.4	Bitrate Allocation and QoE Optimization	19
2.5	Transport Protocols for Immersive Streaming	21
2.6	Research Gaps and Challenges	23
2.7	Summary of chapter	24

TABLE OF CONTENTS

I	Learning-Based Viewport Prediction	26
3	Viewport Prediction in Immersive Videos	27
3.1	Introduction	27
3.2	Dataset description	28
3.3	Dataset preprocessing	32
3.4	ML models for viewport prediction	35
3.4.1	LSTM model	36
3.4.2	RNN model	37
3.4.3	GRU model	38
3.4.4	CNN model	39
3.4.5	LR model	39
3.4.6	MLP model	40
3.5	Results and Discussion	41
3.6	Cross-Video Validation and Ranking Stability	47
3.7	State-of-the-art Comparison	49
3.7.1	Dataset description	49
3.7.2	Data preprocessing	50
3.7.3	Viewport Prediction Framework	52
3.8	Summary of chapter	61
II	Adaptive 360° Video Streaming Systems	63
4	360° Video Streaming Using QUIC	64
4.1	Introduction	64
4.2	System Architecture	65
4.3	Video Encoding and Tiling	67
4.4	Viewport prediction	69
4.5	QUIC-Based Streaming Framework	72

TABLE OF CONTENTS

4.6	Adaptive Streaming Strategy	75
4.7	Experimental Setup	79
4.8	Performance Metrics	82
4.9	Results and Discussion	85
4.10	Summary of chapter	91
5	Conclusions	96

List of Figures

1.2.1 End-to-end adaptive 360° video streaming architecture.	2
1.3.1 Commonly used projection formats in 360° video streaming	4
2.1.1 3×3 tile grid of 360° video ERP frame	14
2.2.1 Comparison of coarse and fine tiling strategies in viewport-adaptive 360° video streaming	18
3.2.1 Temporal variation of raw pitch angle showing vertical head movement	31
3.2.2 Temporal variation of raw yaw angle showing horizontal head movement.	31
3.2.3 Head movement trajectories	32
3.3.1 Viewport representation on a 3 × 3 ERP tile grid.	34
3.4.1 Workflow for Viewport Prediction Using ML Models.	36
3.5.1 RMSE CDF obtained by the considered models.	44
3.5.2 MSE CDF obtained by the considered models.	44
3.5.3 MAE CDF obtained by the considered models.	45
3.5.4 Comparison of tile selection accuracy across models.	46
3.7.1 Training and validation loss curve of LSTM model.	53
3.7.2 Training and validation loss curve of LSTM-CNN model.	56
3.7.3 AP accuracy across different time horizons using only gaze data. . . .	58
3.7.4 Average Overlap (AO) across different time horizons using only gaze data.	59
3.7.5 AP across different time horizons using gaze and saliency data.	60

3.7.6 AO across different time horizons using only gaze and saliency data. . .	60
4.2.1 Overview of viewport-adaptive 360° video streaming framework . . .	66
4.3.1 3×3 tile partitioning of an Equirectangular Projection (ERP) 360° video frame.	68
4.4.1 Architecture of LSTM-based viewport prediction model.	71
4.5.1 Client-server interaction in QUIC-based viewport-adaptive streaming framework.	74
4.6.1 Estimated available bandwidth for each segment based on actual seg- ment size and download time.	76
4.6.2 Requested aggregate bitrate for each segment under the adaptive tile-based streaming policy.	79
4.8.1 Segment-wise download time measured during QUIC-based delivery. .	84
4.9.1 Segment-wise overall missing ratio across the streaming session. . . .	86
4.9.2 Temporal evolution of playback buffer level during the streaming session.	86
4.9.3 Segment-wise FoV missing ratio across the streaming session.	87
4.9.4 Comparison of all tiles and FoV tiles missing ratio for all segments. .	88
4.9.5 Comparison between estimated available bandwidth and requested aggregate bitrate for each segment.	89
4.9.6 Bandwidth margin per segment, difference between estimated available bandwidth and requested aggregate bitrate.	89
4.9.7 Heatmap of tile-level bitrate allocation across segments.	90

List of Tables

2.1	Comparison of projection formats in 360° video streaming systems. . .	15
2.2	Comparison of viewport prediction techniques for 360° video streaming.	20
3.1	Complexity and Real-Time Performance of Different Models	40
3.2	Cross-video and cross-user performance dispersion and model-ranking stability for next-frame viewport prediction.	48
3.3	University of Padua dataset used in this research.	51
3.4	Statistics summary of dataset.	51
3.5	Comparison of LSTM and MFTR using gaze data.	57
3.6	Comparison of LSTM–CNN and MFTR using gaze and saliency data.	57
4.1	Client tile request and system behavior per segment	75
4.2	Adaptive bitrate allocation across tile categories	80
4.3	Experimental setup for viewport-adaptive QUIC streaming framework.	93
4.4	Segment-wise missing ratio and FoV reliability analysis	94
4.5	Playback buffer evolution during streaming	94
4.6	Segment-wise bandwidth estimation and bitrate allocation behavior .	95

List of Abbreviations

ABR Adaptive Bitrate

AO Average Overlap

AP Average Prediction

AR Augmented Reality

AVC Advanced Video Coding

BCE Binary Cross Entropy

CDF Cumulative Distribution Function

CMP Cube-map Projection

CNN Convolutional Neural Network

DASH Dynamic Adaptive Streaming over HTTP

DL Deep Learning

DRL Deep Reinforcement Learning

ERP Equirectangular Projection

FoV Field of View

GRU Gated Recurrent Unit

HEVC High-Efficiency Video Coding

HMD Head-Mounted Display

HoL Head-of-Line

HTTP Hypertext Transfer Protocol

LR Linear Regression

LSTM Long Short-Term Memory

MAE Mean Absolute Error

MFTR Multi-modal Fusion Transformer

ML Machine Learning

MLP Multi-layer Perceptron

MSE Mean Squared Error

PSNR Peak Signal-to-Noise Ratio

QoE Quality of Experience

QUIC Quick UDP Internet Connections

ReLU Rectified Linear Unit

RMSE Root Mean Squared Error

RNN Recurrent Neural Network

RTP Real-Time Transport Protocol

RTSP Real-Time Streaming Protocol

SSIM Structural Similarity Index

TCP Transmission Control Protocol

UDP User Datagram Protocol

VR Virtual Reality

VVC Versatile Video Coding

WebRTC Web Real-Time Communication

Scientific Contributions

International Conferences

Farooq Muhammad, Gioacchino Manfredi, Maria Martini, Saverio Mascolo and Luca De Cicco. “A Comparison of AI Models for Viewport Prediction in Immersive 360° Video Streaming.” 12th International Conference on Control, Decision and Information Technologies (CoDIT 2026). DOI: 10.1109/CoDIT70676.2026.11630935

International Journals

Farooq Muhammad, Gioacchino Manfredi, Luca De Cicco and Saverio Mascolo. “Adaptive 360° Video Streaming: Prediction, Tiling and Transport Trade-Offs.” *Networks* (MDPI), 2026. DOI: 10.3390/network6030066

Chapter 1

Introduction

In recent years, immersive multimedia technologies have experienced significant growth due to advancement in Virtual Reality (VR) systems and rapid development of high-performance Head-Mounted Display (HMD) devices [1–4]. Modern immersive applications enable users to experience digital environments in a highly interactive manner by allowing them to freely explore virtual scenes through head and gaze movements [5, 6]. As a result, immersive media has attracted considerable attention both in research and industry, with applications in virtual tourism, remote collaboration, remote healthcare, education, gaming and live sports broadcasting [3, 7]. Among the different immersive media formats, 360° or omnidirectional video has emerged as a key technology enabling users to observe a scene in all directions and experience content in a way that closely resembles real-world perception.

1.1 Capturing 360° videos

360° videos provide user with a complete and uninterrupted visualization of the surrounding environment. Comparing with conventional videos that stream complete video over a uniform quality, omnidirectional videos record the entire spherical scene around the camera, however user only watch limited part of 360° video. Specialized

camera rigs composed of multiple synchronized lenses are typically used to capture video streams from different angles. These streams are then processed using stitching algorithms that combine the captured footage into a seamless panoramic representation. Through this process, a continuous spherical video is generated, allowing users to dynamically change their viewing direction and explore the scene interactively. Such videos are widely used in applications including sports broadcasting, entertainment, surveillance, tourism and environmental visualization where the ability to observe the environment from multiple perspectives significantly enhances the user experience [8].

1.2 Delivery of 360° videos over the Internet

Delivering immersive content over the Internet requires an end-to-end streaming pipeline that coordinates with viewport prediction results, tile-based segments, network delivery and client-side rendering. The complete streaming architecture of adaptive 360° video is given in Fig. 1.2.1.

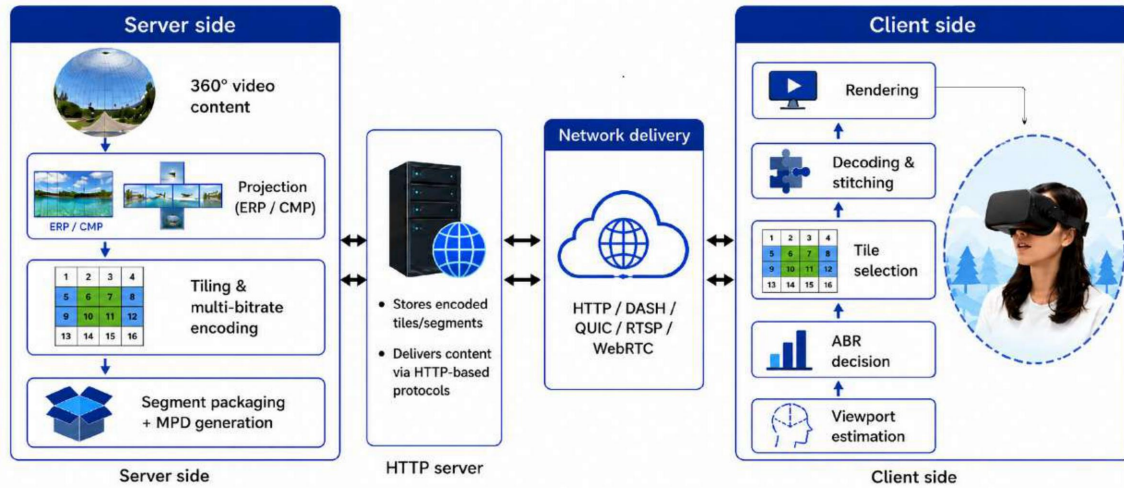


Figure 1.2.1: End-to-end adaptive 360° video streaming architecture.

Fig. 1.2.1 represents the overall streaming pipeline of 360° videos. It consists of two main components: server and client side. On the server side, a 360° video is first projected onto a 2D format, ERP or Cube-map Projection (CMP). The projected

video frame is spatially divided into a number of tiles and encoded at multi-bitrate levels. These encoded segments are packaged together with the media presentation description (MPD) file and made available for delivery through the streaming server.

On the client side, the playback device constantly estimates the user Field of View (FoV) and monitors varying network conditions. Adaptive Bitrate (ABR) module uses this information to select the relevant tiles with appropriate quality levels to request from the server. The requested tiles are processed and rendered to produce the viewport in the HMD. This interaction between projection, tiling, adaptive bitrate selection and viewport rendering is used to achieve the goal of the streaming system for efficient use of network resources and ensure high perceptual quality to the user FoV.

1.3 Projecting 360° video

For efficient storage and processing of omnidirectional video, the spherical representation of the captured content is mapped onto a 2D plane. Several projection techniques have been proposed for this purpose, including Equirectangular Projection (ERP), pyramid projection and CMP [2]. These projection formats are shown in Fig. 1.3.1.

Among the above-mentioned projections, ERP is mostly preference due to its simplicity and compatibility with existing video compression standards. However, ERP introduces spatial distortion near the poles of the projection, which can reduce coding efficiency and visual quality. On the other hand, projection schemes such as CMP attempt to minimize this issue by mapping the spherical surface onto the faces of a cube, in order to provide more uniform sampling of the visual content.

After projecting spherical content in to planar format, traditional video compression standards are applied to reduce the required storage and transmission bandwidth. The omnidirectional videos are encoded using commonly adopted video compression standards such as H.264 Advanced Video Coding (AVC) and H.265 High-Efficiency

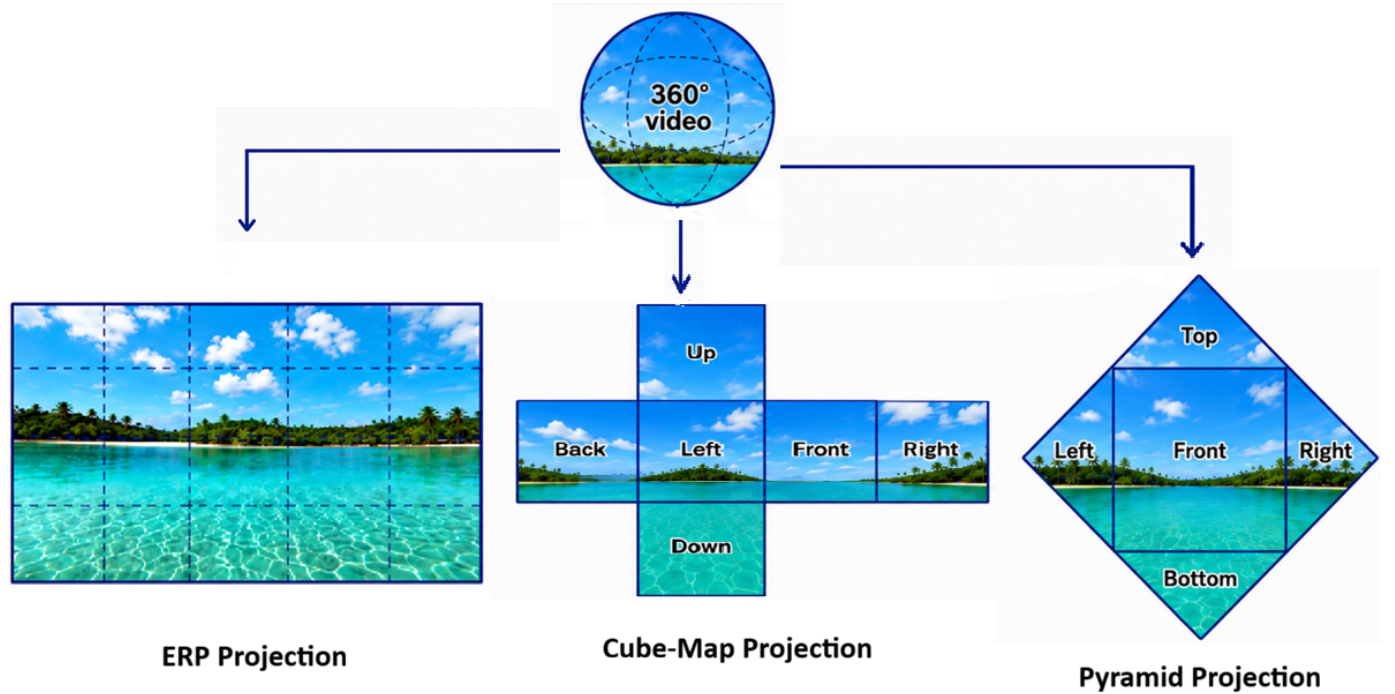


Figure 1.3.1: Commonly used projection formats in 360° video streaming

Video Coding (HEVC). These codecs enable efficient compression by exploiting spatial and temporal redundancy in the video sequence while maintaining high visual quality.

Recent study on emerging codecs such as Versatile Video Coding (VVC) have been proposed to further improve compression efficiency for high-resolution immersive media. With the continuous improvement of hardware acceleration and network infrastructure, 360° videos can be viewed seamlessly on a variety of platforms including smartphones, desktop computers and immersive VR headsets [9].

1.4 Low latency and bandwidth efficiency in 360° video streaming

Despite these technological advancements, streaming omnidirectional video over the Internet introduces several challenges. Compared with traditional videos, 360° videos contain significantly large amount of visual information because the entire scene is

captured and transmitted even though the user observes only a small portion of the spherical content at any given time. As a result, streaming the complete panoramic video leads to extremely high bandwidth consumption and inefficient utilization of network resources. In particular, These limitations are important for mobile networks and wireless environments where bandwidth availability and latency constraints can considerably affect the quality of the immersive experience.

Immersive multimedia streaming systems need to meet strict latency requirements in addition to ensuring bandwidth efficiency. In contrast to conventional video services where buffering delays of several seconds may be tolerated, immersive virtual reality applications require rapid response to user head movements in order to avoid motion-to-photon latency and visual discomfort. When a user rotates their head while wearing a head-mounted display, the visible viewport changes immediately and the streaming system must quickly deliver the corresponding video content. If the required video tiles are not available in time, the user may experience temporary quality degradation or viewport blank regions, which negatively affects the perceived Quality of Experience (QoE). Therefore, minimizing latency and ensuring timely delivery of viewport-relevant video segments is a critical requirement for immersive streaming systems [10, 11].

Another challenge is the highly dynamic nature of user viewing behavior. Compared with traditional video playback, where the viewing area is fixed, users in immersive environments can freely explore the entire panoramic scene. This means that the visible viewport may change rapidly depending on head movement and gaze direction. As a result, streaming systems need to adapt to unpredictable user behavior while maintaining stable video quality. Predicting future viewing directions has become an important research topic in immersive multimedia systems. Accurate prediction helps the streaming system to proactively transmit the tiles most likely to be viewed by the user [12, 13].

1.5 Visibility of 360° videos

In the context of 360° video streaming, the term *viewport* or FoV refers to the specific portion of the panoramic video that is visible to the user through the HMD. Because the viewport typically covers only a small fraction of the entire video sphere, transmitting the full panoramic video at the same quality level is highly inefficient. Consequently, research in immersive video streaming has focused on developing techniques that prioritize the delivery of video data corresponding to the user's viewport while reducing the quality of non-visible regions.

1.6 360° video streaming approaches

Two main approaches have been proposed in the literature for streaming omnidirectional videos. The first approach, referred to as *viewport-independent streaming*, transmits the entire 360° video to the client at uniform quality. Although this approach ensures that any part of the video can be immediately rendered when the user changes their viewing direction, it requires extremely high bandwidth and is therefore not suitable for large-scale deployment. The second approach, known as *viewport-adaptive streaming*, focuses on delivering high-resolution content only for the portion of the video that falls within the predicted viewport while transmitting lower quality versions of the remaining regions. This strategy significantly reduces the required bandwidth while maintaining high perceptual quality for the user.

Tile-based streaming has emerged as one of the most effective solutions for implementing viewport-adaptive delivery of immersive video. In this approach, the panoramic video is spatially divided into multiple independent regions known as tiles. Each tile can be encoded and transmitted at different quality levels depending on its relevance to the user's viewport. When users watch a specific region of the video, the tiles corresponding to that region are delivered at higher resolution while the remaining tiles are transmitted at lower bitrates. By dynamically adjusting the

quality of individual tiles, tile-based streaming enables efficient bandwidth utilization while preserving high visual quality in the viewport.

The design of tile-based streaming systems involves several important trade-offs. Increasing the number of tiles allows the system to more precisely match high-quality video regions with the user’s viewport. This fine-grained adaptation can significantly improve bandwidth efficiency by reducing the transmission of unnecessary high-resolution content outside the viewport. However, a large number of tiles also increases encoding complexity, storage requirements and signaling overhead, since each tile has to be encoded, stored and transmitted independently. Furthermore, decoding multiple tiles simultaneously may increase computational load on the client device [11, 14].

Another important consideration in tiled streaming systems is the synchronization and stitching of independently decoded tiles. Because tiles are encoded separately, differences in bitrate or encoding parameters between neighboring tiles may introduce visible seams or quality discontinuities at tile boundaries. To mitigate these issues, several studies have investigated optimized tile partitioning strategies and coordinated bitrate allocation mechanisms that maintain visual consistency across adjacent tiles [10, 15]. These design considerations highlight the importance of carefully balancing tile granularity, encoding efficiency and playback quality in immersive streaming architectures.

However, tile-based streaming introduces new challenges related to latency and prediction accuracy. When a user changes their viewing direction, the system needs to rapidly deliver the corresponding tiles to avoid visible quality degradation or playback interruptions. To address this challenge, many recent studies have focused on predicting future user viewports using Machine Learning (ML) techniques. Accurate viewport prediction allows the streaming system to proactively deliver the tiles that are likely to be viewed in future, thereby reducing latency and improving the user QoE.

1.7 Input for viewport estimation

Various ML and Deep Reinforcement Learning (DRL) approaches have been proposed for viewport prediction [16–18]. These approaches analyze historical head movement patterns, gaze behavior along with video saliency information to estimate the future viewport of the user. Sequential deep learning models such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU) and Recurrent Neural Network (RNN) have been widely used to capture modeling temporal dependencies in head movement trajectories [19–21]. Additionally, Convolutional Neural Networks (CNNs) have been applied to extract spatial features from video frames to improve prediction accuracy [22].

Viewport prediction models typically rely on multiple sources of information to estimate future viewing directions. One of the most widely used inputs is the historical head movement trajectory of the user, which captures temporal patterns in viewing behavior over time. By analyzing sequences of yaw and pitch angles recorded using HMD devices, ML models can learn motion patterns and anticipate future viewing directions. Literature has reported that RNN models such as LSTM and GRU are more effective for modeling these temporal dependencies in HMD data [12].

In addition to head movement trajectories, visual saliency information extracted from video frames can provide valuable indicators regarding which regions of the scene are likely to attract user attention. Saliency-based approaches analyze visual features such as motion and object presence to identify areas that are more likely to be viewed by users. Combining head movement information with visual saliency features has been shown to improve prediction performance by incorporating both user behavior and content characteristics into the prediction model [23, 24]. These hybrid approaches have therefore become increasingly popular in recent viewport prediction research.

1.8 Transport of 360° video streaming

Beside viewport prediction, the underlying transport protocol also plays an important role in immersive video delivery. Traditional adaptive streaming systems rely on Hypertext Transfer Protocol (HTTP)-based delivery over Transmission Control Protocol (TCP). However, TCP introduces Head-of-Line (HoL) blocking and latency overhead, which may negatively affect interactive immersive applications. Recent studies focus on Quick UDP Internet Connections (QUIC) transport protocol which has gained attention as a promising alternative for multimedia delivery. QUIC operates over User Datagram Protocol (UDP) and integrates transport and security mechanisms, enabling faster connection establishment, improved congestion control and reduced latency. These characteristics make QUIC more suitable for adaptive streaming of immersive media.

Compared with traditional TCP-based transport, QUIC introduces several architectural improvements that make it suitable for immersive media delivery. One of the key advantages of QUIC is the elimination of head-of-line blocking between independent streams. In TCP-based HTTP streaming, packet loss in a single stream can delay the delivery of subsequent packets, even if they belong to unrelated video segments. QUIC avoids this limitation by allowing multiple independent streams to be transmitted over a single connection, improving robustness and reducing latency for adaptive video streaming applications [25].

Another important feature of QUIC is its reduced connection establishment latency. Traditional TCP connections require multiple handshake steps before encrypted data transmission can start, whereas QUIC integrates transport and security handshakes into a single process. This design significantly reduces startup delays and allows streaming systems to adapt more quickly to variable network conditions. As a result, QUIC has gained increasing attention as a promising transport protocol for low-latency multimedia streaming systems [14, 26].

Objective of this research is to investigate design and performance of AI-based

viewport-adaptive immersive video streaming systems. In particular, this work investigates on how ML techniques can be used to predict user viewports in 360° video environments and how these predictions can be integrated with adaptive streaming mechanisms to improve bandwidth efficiency and overall user experience. The proposed system combines viewport prediction with tile-based video delivery, enabling the streaming platform to prioritize the transmission of video regions that are most likely to be viewed by the user. By focusing network resources on the predicted viewport, the system can significantly reduce unnecessary bandwidth consumption while maintaining high visual quality in the user’s FoV.

To analyze the effectiveness of viewport prediction, various ML models are evaluated using a publicly available dataset of user head-movement traces in immersive video environments. This study includes comparison of prediction performance of models used across different prediction horizons, containing next-frame prediction as well as longer-term forecasts of up to five seconds. In addition to prediction accuracy, some other factors as computational complexity, inference latency and real-time feasibility are also considered, as these aspects play an important role in the practical deployment of prediction models within live streaming systems.

Beyond prediction components, this work also investigates the integration of viewport prediction techniques into a complete streaming pipeline for immersive media delivery. In particular, the thesis explores tile-based streaming approaches that dynamically allocate higher video quality to the tiles corresponding to the predicted viewport while delivering lower quality for the remaining regions of the 360° video. Furthermore, the role of modern transport protocols is examined, with a focus on QUIC for low-latency delivery of immersive video content. Combined effect of ML-based viewport prediction with adaptive tile-based streaming and efficient transport mechanisms, the proposed framework aims to improve the scalability and performance of immersive video streaming systems.

Various ML and Deep Learning (DL) models have been explored for viewport

prediction in immersive video streaming systems. These include both sequence-based models, such as LSTM, RNN and GRU, as well as non-sequential approaches like Linear Regression (LR), Multi-layer Perceptron (MLP) and Convolutional Neural Network (CNN). Each of the above mentioned models offers different advantages in terms of capturing temporal dependencies, computational complexity and suitability for real-time deployment. Moreover, selection of a suitable model involves inevitable trade-offs between prediction accuracy, computational efficiency and practical feasibility. Understanding these trade-offs is essential for the design of effective viewport-adaptive streaming architectures.

1.9 Thesis Outline

This thesis is organized into five chapters, each addressing a different component of viewport-adaptive immersive video streaming systems.

Chapter 2 presents the background and related work on immersive multimedia systems and 360° video streaming. It introduces the fundamental concepts of omnidirectional video, projection formats, tile-based streaming, and existing approaches for viewport prediction.

Chapter 3 focuses on viewport prediction using machine learning techniques. Different models are investigated and evaluated using a public dataset, and their performance is analyzed in terms of prediction accuracy, computational complexity and suitability for real-time deployment.

Chapter 4 investigates adaptive streaming mechanisms for 360° videos. In particular, tile-based streaming strategies and the use of the QUIC protocol are explored to improve bandwidth efficiency and reduce latency in immersive media delivery.

Finally, Chapter 5 summarizes the key contributions of this thesis and discusses possible directions for future research in AI-driven immersive video streaming systems.

1.10 Summary of chapter

This chapter provides a general introduction of 360° videos and explains their acquisition, encoding and preprocessing steps. It introduces several projection formats and focuses on the commonly used ERP projection in adaptive viewport prediction streaming systems. It also discusses on complete streaming pipeline of 360° videos with server and client connection and emerging protocols used in streaming process. Based on these concepts, the research objective of this thesis is to design and evaluate AI-based viewport-adaptive streaming systems that combine ML-based viewport prediction, tiling, adaptive bitrate decision and selection of transport protocol to improve bandwidth efficiency and user QoE.

Chapter 2

Background

The rapid growth in immersive multimedia technologies has led to an increasing interest in 360° video streaming systems. As compared to conventional video streaming, where users watch the entire frame, immersive video environments allow users to interactively explore a spherical scene through head movements using a HMD. As a result, only a limited portion of the panoramic video, known as FoV, is visible to the user at any given time. Studies show that the instantaneous viewport typically covers only 15–25% of the entire spherical frame, while the remaining regions are outside the user’s FoV [5]. Despite this localized viewing behavior, naïve streaming approaches often transmit the entire 360° frame at uniform quality, leading to significant bandwidth inefficiency and unnecessary network resource consumption. Empirical studies indicate that up to 70–80% of transmitted pixels may not be viewed by the user in such scenarios [27, 28]. Therefore, designing efficient viewport-adaptive streaming systems has become a major research focus in immersive multimedia delivery.

Adaptive 360° video streaming systems typically consist of several tightly coupled components, including spherical video representation, viewport modeling, spatial tiling, ABR decision mechanisms, viewport prediction algorithms and transport-layer protocols. The interaction between these components determines the overall streaming performance and directly influences the perceived QoE of users. Recent research

emphasizes that these components should not be optimized independently, however they are considered as part of a unified system-level architecture [5].

2.1 Video Representation and Projection Formats

Several challenges are observed in complete streaming process of 360° videos, one of the fundamental challenge is the representation of spherical video content. Since conventional video codecs operate on two-dimensional frames, the spherical video must first be projected onto a planar surface before encoding. Several projection techniques have been proposed in the literature, including ERP, CMP, octahedron and pyramid projection [29–31]. Figure 2.1.1 represents the ERP of a spherical video content into a 2D representation and its partitioning into spatial tiles for viewport-adaptive streaming.

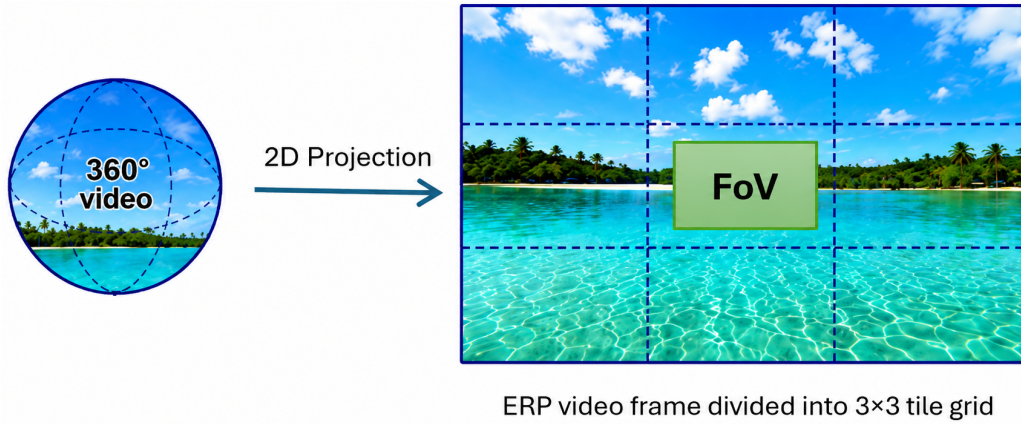


Figure 2.1.1: 3×3 tile grid of 360° video ERP frame

A comparison of commonly used spherical projection formats is summarized in Table 2.1. These projection methods differ in terms of geometric distortion, coding efficiency and compatibility with existing video compression pipelines. Each projection format provides a different trade-off between representation simplicity and sampling uniformity on the spherical surface. ERP offers straightforward mapping and codec compatibility [32], whereas CMP improves spatial uniformity at the cost

Table 2.1: Comparison of projection formats in 360° video streaming systems.

Reference	Projection	Advantages	Limitations	Application	Key Observations
[32]	ERP	Simple mapping and direct compatibility with conventional video codecs	Severe distortion near the poles due to non-uniform sampling density	Most widely used representation in streaming pipelines	Adopted in MPEG omnidirectional media format standards
[33]	CMP	More uniform spatial sampling compared to ERP	Discontinuities between cube faces and boundary artifacts	Viewport-adaptive streaming and tile-based systems	Improves coding efficiency by reducing polar redundancy
[34]	Pyramid	Reduced pole distortion and improved sampling efficiency	Increased mapping complexity and encoding overhead	Experimental immersive streaming architectures	Used in early viewport-aware streaming research
[35]	Octahedron	Balanced distortion distributed across projection surface	Higher computational complexity for mapping operations	Research prototypes and advanced spherical processing	Considered for improved geometric consistency in spherical video

of additional processing complexity [33]. Pyramid and octahedron projections have also been investigated in the literature as alternative mappings that aim to balance distortion distribution across the spherical surface [34,35]. As mentioned in Table 2.1, each projection format exhibits specific advantages and limitations that influence its suitability for immersive streaming applications. ERP is preferred because of its easy mapping and compatibility with conventional video encoding standards such as H.264 and H.265. However, the non-uniform sampling density introduced near the poles results in spatial distortion and reduced coding efficiency. CMP addresses this limitation by distributing the spherical content across six cube faces, which improves spatial uniformity and makes it well suited for viewport-adaptive streaming systems. Alternative formats such as pyramid and octahedron projections attempt to further balance distortion across the spherical surface, although their higher mapping complexity has limited their adoption in practical streaming pipelines.

Among these approaches, ERP remains the most widely adopted format due to its simplicity and compatibility with existing video compression pipelines. In ERP, the spherical surface is mapped onto a rectangular image by converting spherical coordinates into planar coordinates. However, this projection introduces non-uniform sampling density, particularly near the poles, which reduces compression efficiency and increases redundancy in non-viewed regions [5]. In ERP, spherical coordinates defined by longitude (θ) and latitude (ϕ) are mapped onto a two-dimensional rectangular

image through a direct linear transformation. Although this mapping is simple and compatible with conventional video encoding pipelines, it introduces non-uniform sampling density across the image. In particular, pixels located near the poles correspond to a much smaller area on the spherical surface compared with pixels near the equator. As a result, polar regions become oversampled while equatorial regions remain comparatively underrepresented, leading to inefficient bitrate allocation during compression. This distortion characteristic reduces coding efficiency and increases redundancy in regions that are rarely viewed by users in immersive viewing scenarios [36]. To mitigate these distortions, CMP projection has been proposed as an alternative mapping method, where the spherical surface is projected onto the six faces of a cube. This approach improves spatial uniformity and better aligns with viewport-based streaming architectures.

Although projection methods simplify encoding and transmission, they also introduce distortion and spatial redundancy that must be carefully considered when designing adaptive streaming systems. Consequently, many modern approaches integrate projection design with tiling strategies and bitrate allocation algorithms to improve streaming efficiency.

2.2 Tile-Based Adaptive Streaming

To address the inefficiency of transmitting the entire panoramic frame, tile-based streaming architectures have been proposed. With such approaches, the projected video frame is partitioned into multiple tiles that can be independently encoded and transmitted. This allows the streaming system to allocate high bitrate levels to tiles located within predicted FoV while transmitting peripheral tiles at low quality levels.

Tile-based adaptive streaming has become a widely adopted technique in immersive video delivery systems, with representative implementations including 360ProbDASH [37] and viewport-adaptive DASH architectures [38]. In these systems, the panoramic

video frame is spatially partitioned into multiple tiles that can be independently encoded and transmitted according to their relevance to the user’s viewport. This spatial adaptation enables the streaming system to allocate higher quality to tiles that fall within the visible region while reducing the bitrate of tiles outside the user’s FoV. As a result, spatially adaptive streaming significantly improves bandwidth utilization while maintaining high visual quality in the regions that are most important for the viewer. Experimental studies demonstrate that such approaches can substantially improve streaming efficiency, reporting improvements in viewport Peak Signal-to-Noise Ratio (PSNR) of up to 39% when probabilistic tile selection strategies are applied [5].

Despite these advantages, tile-based streaming introduces several important design challenges. One of the key design parameters is the granularity of tiling used to partition the projected 360° video frame. Fine-grained tiling improves viewport precision because smaller tiles allow the streaming system to more accurately match the transmitted video quality with the user’s FoV. In this approach, higher bitrate is assigned to viewport tiles, while low bitrate is assigned to tiles outside the viewport to use available bandwidth efficiently. However, fine tiling, partitions the 2D image into a large number of tiles, which increases encoding complexity, storage requirements and number of client requests.

In contrast, coarse tiling divides video frame into a smaller number of larger tiles, which reduces encoding overhead and simplifies storage and delivery management. Nevertheless, this approach limits spatial adaptation because a large tile may partially overlap with the user’s FoV, forcing the system to transmit high-quality data for regions that are not actually viewed. It results in degrading FoV quality and bandwidth loss. Studies show that increasing tile granularity can increase encoding overhead by up to four times due to inter-tile prediction loss and boundary artifacts [5]. Therefore, selecting an appropriate tiling configuration requires balancing FoV prediction accuracy, encoding efficiency and network overhead.

Figure 2.2.1 indicates the difference between coarse and fine tiling configurations. In the coarse tiling example, panoramic frame is divided into a small number of large tiles, which simplifies encoding but reduces the ability to precisely adapt video quality to the user’s viewport. In contrast, the fine tiling configuration divides the frame into a larger number of smaller tiles, allowing the system to better match the transmitted quality to the viewport region. While this improves spatial adaptation and reduces unnecessary data transmission outside the FoV, it also increases system complexity and encoding overhead.

To address these limitations, several studies explore variable-sized tiling strategies that dynamically adapt tile boundaries based on viewport geometry and predicted user behavior [39] [40] [41] [42] and [43]. These approaches are used to improve coding efficiency and reduce redundant transmission by placing tile boundaries closer to the regions where users are most likely to watch.

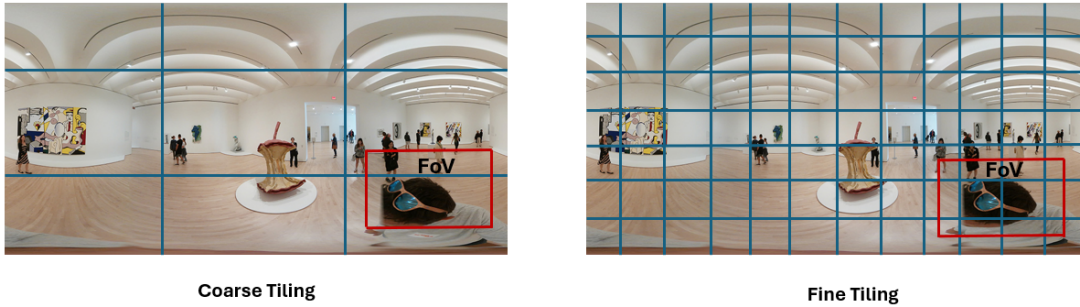


Figure 2.2.1: Comparison of coarse and fine tiling strategies in viewport-adaptive 360° video streaming

2.3 Viewport Prediction Techniques

Viewport prediction plays a critical role in enabling viewport-adaptive streaming systems. The objective of viewport prediction is to estimate the future viewing direction of the user so that the streaming system can proactively deliver the corresponding

video tiles before playback deadlines.

Early viewport prediction methods relied on simple trajectory extrapolation techniques using LR or velocity-based head movement models. However, these approaches exhibit limited accuracy for prediction horizons beyond one second due to the unpredictable nature of human head movements. To address this limitation, ML and DL techniques have been mostly adopted for viewport prediction. Sequential models such as LSTM, GRU and RNN have reported strong performance in modeling temporal dependencies in user head movement trajectories. Studies such as GLVP [44] show that RNN architectures can improve prediction accuracy by up to 20% compared to traditional statistical methods.

Content-based prediction approaches have also been proposed to incorporate visual features extracted from video frames. These methods analyze saliency maps, object motion and scene dynamics to estimate regions that are likely to attract user attention. By combining trajectory information with visual content features, hybrid models can achieve higher prediction accuracy in complex viewing scenarios.

Recent studies also explore transformer-based architectures [19] and multi-modal learning frameworks [45], which combine head motion, visual saliency and cross-user behavior to improve long-horizon prediction. Table 2.2 compares representative viewport prediction techniques for 360° video streaming. These methods can be classified into trajectory-based, content-based and hybrid approaches.

2.4 Bitrate Allocation and QoE Optimization

Adaptive bitrate allocation selects the quality level of each tile according to the available network bandwidth and user’s predicted FoV. In immersive systems, ABR gives higher bitrate priority to tiles inside the predicted FoV, while non-FoV tiles are assigned low priority. Therefore, adaptive streaming is formulated as a QoE optimization problem in the literature, where QoE models evaluate the viewing

Table 2.2: Comparison of viewport prediction techniques for 360° video streaming.

Reference	Input Data	Model	Advantage	Limitation
[19]	Viewport scanpath trajectories	Transformer (self-attention model)	High accuracy for long-term viewport prediction and ability to capture temporal dependencies	High computational complexity and large model size
[20]	Head trajectory + saliency maps from video frames	Hybrid model (LSTM + CNN + Fully Connected layers)	Combines trajectory prediction with visual attention information for improved accuracy	Requires video content processing and higher computational cost
[22]	Video frame features	CNN	Effective for live VR streaming and content-driven prediction	Requires real-time training and video processing
[45]	User head trajectory + video content + cross-user behavior	Graph Neural Network (GNN)	Exploits multi-feature fusion and relational modeling for improved QoE	Requires complex feature extraction and graph construction
[46]	Head movement trajectories	LR / ANN / CNN / LSTM	Simple implementation and interpretable prediction models	Limited long-term prediction accuracy and sensitivity to user behavior variability
[47]	User viewing trajectories	ML (window-based supervised learning)	Content-agnostic prediction with lower computational complexity	Lower accuracy compared to content-aware deep learning models
[48]	Historical head movement and user viewing patterns	LSTM	Captures temporal dynamics of user viewing behavior	Prediction accuracy degrades for longer prediction horizons
[49]	Historical trajectory + video saliency information	Transformer-based multi-modal fusion model	State-of-the-art long-term prediction accuracy using trajectory-content fusion	Complex architecture and high computational cost
[50]	Viewport center coordinates and head motion data	Statistical and ML comparison framework	Provides systematic evaluation of prediction methods and metrics	Performance drops significantly for prediction windows longer than 1 second

experience under changing bitrate and playback conditions. Buffer-aware adaptation algorithms and DRL-based controllers have been proposed to dynamically optimize tile bitrate selection under fluctuating network conditions. Viewport prediction-aware bitrate allocation can significantly improve streaming performance. However, prediction errors can lead to visible quality degradation when incorrect tiles are delivered. Therefore, probabilistic prediction models and uncertainty-aware bitrate allocation strategies are used to minimize the viewport prediction errors [51, 52]

2.5 Transport Protocols for Immersive Streaming

Transport protocols play an important role in the performance of immersive video streaming systems. Adaptive streaming architectures rely on HTTP-based delivery over TCP, as implemented in Dynamic Adaptive Streaming over HTTP (DASH). In HTTP adaptive streaming, video content is segmented and encoded at multi bitrate levels that allow the client to dynamically select appropriate segments depending on network conditions and buffer status [53, 54]. While TCP provides reliable and ordered data delivery, its strict reliability mechanisms introduce limitations for latency-sensitive immersive media. In particular, TCP suffers from HoL blocking, where the loss of a single packet can delay the delivery of subsequent packets, leading to increased latency and potential playback interruptions during streaming sessions [55].

To address these limitations, alternative transport mechanisms, such as HTTP/2, QUIC-based HTTP/3 and real-time media transport protocols such as Real-Time Transport Protocol (RTP)/Real-Time Streaming Protocol (RTSP) and Web Real-Time Communication (WebRTC), have been explored. HTTP/2 enables multiplexed delivery, enabling multiple segments or tiles to be transferred more efficiently over a single connection.. Server push allows the server to proactively transmit multiple video segments without waiting for explicit client requests, thereby reducing round-trip latency and improving bandwidth utilization [56]. These features are particularly

beneficial for tiled immersive video streaming, where multiple tiles corresponding to the user’s viewport must be delivered within strict deadlines [57,58]. Experimental studies demonstrate that HTTP/2-based frameworks can significantly increase throughput and reduce playback freezes in high-latency networks [57].

Over the past few years, QUIC has gained significant attention as a promising transport protocol for immersive media delivery. Operating over UDP, QUIC combines reliability of TCP with lower latency and improved stream multiplexing. As compared to TCP, QUIC eliminates transport-layer HoL blocking by allowing independent delivery of multiplexed streams, enabling more efficient transmission of viewport-dependent tiles [55]. Moreover, QUIC supports faster connection establishment and flexible congestion control mechanisms, which are advantageous for interactive immersive applications. Experimental work on tiled DASH streaming over QUIC shows improvement in video quality and reduction in missing tiles ratio by prioritizing urgent viewport tiles within multiplexed streams [59].

Recent work also explores advanced architectures such as multi-path QUIC, which distributes tile requests across multiple network paths to further reduce download times and rebuffering events [60]. In addition, customized QUIC-based transmission protocols have been proposed for tile-based VR streaming, where video tiles are mapped to individual transport streams with different priorities and reliability requirements [61]. Such designs enable more efficient handling of dynamic viewport changes and network fluctuations in immersive streaming scenarios.

The adoption of new transport protocols faces several challenges. Some key factors, such as congestion control behavior, compatibility with existing content delivery infrastructures and deployment complexity, should be carefully considered when integrating protocols such as HTTP/3 and QUIC into large-scale immersive streaming systems. Ensuring smooth playback with current network architectures and maintaining stable performance under network fluctuating behavior remain important considerations for practical deployment.

2.6 Research Gaps and Challenges

Although significant improvement has been done in viewport prediction and adaptive streaming technologies, still several challenges persist.

First, many existing studies evaluate predicting user FoV algorithms independently without considering their interaction with bitrate allocation strategies and tiling based adaptive streaming. However, prediction accuracy alone does not guarantee to improve performance of streaming systems.

Second, tile granularity and encoding efficiency introduce important trade-offs that remain insufficiently explored in existing literature. Fine-grained tiling improves viewport precision, however it increases encoding overhead and storage requirements.

Third, transport-layer behavior is often ignored in viewport prediction studies, despite its significant influence on tile delivery latency and buffer stability.

In addition, cross-layer optimization between viewport prediction, adaptive bitrate allocation, tiling strategies, and transport protocol design remains an open research problem. Therefore, designing efficient immersive video streaming systems requires a comprehensive approach that jointly considers prediction accuracy, encoding efficiency, as well as network transport mechanisms.

The work presented in this thesis addresses these challenges by investigating ML-based approaches for viewport prediction and their integration within viewport-adaptive immersive video streaming systems. In particular, this thesis evaluates several ML and DL models to predict the user's future viewport using historical head movement information. These prediction models are analyzed across different time horizons, ranging from next-frame prediction to longer-term forecasts of up to five seconds to assess their prediction accuracy and suitability for real-time immersive streaming applications.

Beyond viewport prediction, the thesis also examines how prediction results can be incorporated into practical adaptive streaming architectures for 360° video delivery. In this context, tile-based streaming strategies are considered to efficiently allocate

higher quality video data to the regions corresponding to the predicted viewport while reducing the bitrate of non-visible areas.

To evaluate the impact of modern transport protocols on immersive video delivery, the thesis further investigates the use of QUIC protocol for streaming 360° videos. The implementation is based on the open-source *aioquic-360-video-streaming* framework¹, which provides a QUIC-based client-server architecture for immersive media transmission. Using this framework, the thesis analyzes the behavior of QUIC-based streaming mechanisms and explores their potential advantages over traditional HTTP/TCP-based approaches in terms of latency, bandwidth efficiency and scalability. By combining viewport prediction with tile-based adaptive streaming and QUIC-based delivery, this work aims to contribute toward the design of more efficient and scalable immersive video streaming systems.

2.7 Summary of chapter

This chapter presents a comprehensive literature review of recent studies related to adaptive 360° video streaming systems. The structure of the chapter covers spherical 360° video representation and projection formats adopted in recent studies, an overview of tile-based viewport-adaptive systems for media delivery with a focus on the trade-offs between key aspects such as tile partitioning and viewport prediction, and the use of three categories of input data, defined as user HMD orientation data (pitch and yaw angles), saliency maps and a combination of both HMD and saliency information for analyzing viewport estimation performance.

Furthermore, this chapter discusses bitrate allocation and transport protocols for immersive streaming systems. QoE-based bitrate adaptation is examined with a focus on giving high bitrate to predicted FoV tiles and low bitrate to non-FoV tiles under bandwidth constraints. The chapter also reviews the characteristics and limitations

¹<https://github.com/gufernandez/aioquic-360-video-streaming>

of transport protocols such as TCP and QUIC with particular attention to latency and stream multiplexing in immersive multimedia delivery..

Part I

Learning-Based Viewport
Prediction

Chapter 3

Viewport Prediction in Immersive Videos

3.1 Introduction

Viewport prediction plays a fundamental role in enabling efficient delivery of immersive 360° video streaming systems. As compared to conventional video streaming where the entire frame is visible to users, immersive videos are projected onto a spherical surface and consumed through HMD. At any given moment, the user only observes a limited FoV that covers a small portion of the full panoramic scene. As a result, transmitting the complete high-resolution 360° frame leads to significant bandwidth inefficiency, since a large fraction of the transmitted pixels remains outside the user’s instantaneous viewport. To address this inefficiency, viewport-adaptive streaming systems selectively allocate higher bitrate to the region that the user is expected to observe, while reducing the quality of areas outside the viewport.

Viewport prediction techniques aim to forecast the future viewing direction of the user by analyzing historical head movement patterns and behavioral viewing dynamics. By accurately predicting the user future viewport, streaming systems can proactively deliver high-quality tiles corresponding to the predicted viewing region, reducing

bandwidth consumption while maintaining perceptual video quality. This predictive capability is important in latency-sensitive immersive applications, where delays in delivering the correct viewport content may lead to visible quality degradation or blank viewports. Therefore, accurate viewport prediction enables more efficient bandwidth utilization and smoother immersive playback, making it a key component of modern adaptive 360° video streaming architectures. In this chapter, we focus on the practical implementation of viewport prediction using user HMD data obtained from publicly available datasets. The following section introduces the dataset used in this study and also describes the representation of user head orientation traces that serve as the input for the viewport prediction framework.

3.2 Dataset description

A publicly available dataset ¹ was considered for this framework that contains 88 omnidirectional (360°) videos along with the corresponding user viewing histories collected from an average of 45 users per video. These viewing histories capture the temporal evolution of user head movements while interacting with immersive video content through HMDs or equivalent viewing devices. Such datasets are widely used in the research community to analyze user viewing behavior and to develop viewport prediction and adaptive streaming algorithms for immersive multimedia systems.

In this dataset, user head motion is represented through orientation parameters that describe the direction in which the user is looking at each instant of time. Two different mathematical representations are employed in the dataset: Euler angles and unit quaternions. In Euler representation, head orientation is described using pitch and yaw angles, while the roll component is set to zero, as roll rotations are negligible in viewport prediction tasks for immersive video streaming. Alternatively, some portions of the dataset represent head orientation using unit quaternions, which

¹<https://github.com/360VidStr/A-large-dataset-of-360-video-user-behaviour/blob/main/README.md>

provide a compact and numerically stable way to represent three-dimensional rotations without suffering from issues such as gimbal lock that may occur with Euler angle representations.

The complete dataset is composed of six independent sub-datasets where each one belongs to different experimental studies conducted by various research groups. These studies are reported in the corresponding articles [28] [62] [27] [63] [64] and [65]. Each sub-dataset was collected under different experimental conditions, including variations in video content, number of participants, viewing devices and experimental protocols. In addition, the aggregated dataset contains head orientation trajectories for 3791 independent viewings of 88 different 360° videos, with an average of approximately 45 viewings per video. The original datasets also differ in terms of sampling rates, video durations and HMD devices used during data collection such as Oculus DK2, HTC Vive and Oculus Go. Among these six sub-datasets, datasets 2, 3 and 5 represent the head orientation of users using Euler angles (pitch, yaw). Remaining datasets store head orientation information using quaternion representations, where each orientation sample is described by a four-dimensional unit quaternion vector.

Although the aggregated dataset contains 88 videos distributed across multiple sub-datasets, the complete set of corresponding traces is not publicly available from the referenced repository. Among the traces represented using pitch and yaw angles, traces 7–32 and 42–59 are identified in the dataset description. At the time of this study, only traces 7–16 and 42–59 were publicly accessible, while traces 17–32 were unavailable. Therefore, 28 usable pitch–yaw traces were available for the present viewport-prediction framework. The trace 55 was used as the development trace, while the remaining 27 traces were used for cross-video validation experiment.

Each data file in the dataset corresponds to a specific omnidirectional video. It contains the head orientation traces of multiple users who watched that particular video. Head-movement traces capture how users explore the 360° scene over time, providing important information about attention patterns and motion history. These

traces form the fundamental input for ML models that are used to predict FoV directions and enable viewport-adaptive video streaming. The dataset used in this work is sampled at a frequency of 10 Hz, which means that HMD data is recorded every 0.1 seconds throughout the video duration. This sampling rate provides sufficient temporal granularity to capture the continuous head movements of users while watching immersive content, while still maintaining a manageable data size for analysis and model training. Accordingly, the sequence of sampling instants can be represented as $t = [0, 0.1, 0.2, \dots]$ where each value in the sequence corresponds to a discrete time step at which the user’s head orientation is recorded. For any timestamp t , the dataset stores the corresponding orientation parameters that describe the viewing direction of each user. These time-indexed head orientation traces serve as the primary input signals for the viewport prediction framework developed in this thesis. The availability of such detailed user viewing histories enables the design and evaluation of AI-based viewport prediction algorithms which can learn temporal patterns in user head motion and anticipate future viewing directions. These predictions are essential for viewport-adaptive streaming systems, where video tiles inside the predicted viewport are transmitted at higher quality, while tiles outside the viewport are delivered at lower quality to optimize bandwidth usage.

When multiple users watch the same immersive video content, their head movements can be analyzed using the pitch angle over time, which represents the vertical component of their viewing direction. The pitch angle captures the up–down head movement of users while interacting with immersive video content. Figure 3.2.1 shows the temporal evolution of the raw pitch angles obtained from the head motion traces of users viewing the same video sequence. As shown in the figure, the pitch signal exhibits temporal variations and occasional abrupt changes which reflects natural vertical head movements during immersive video exploration.

Similarly, the yaw angle represents the horizontal component of the user’s viewing direction that describes the left–right head rotation during video playback. Figure 3.2.2

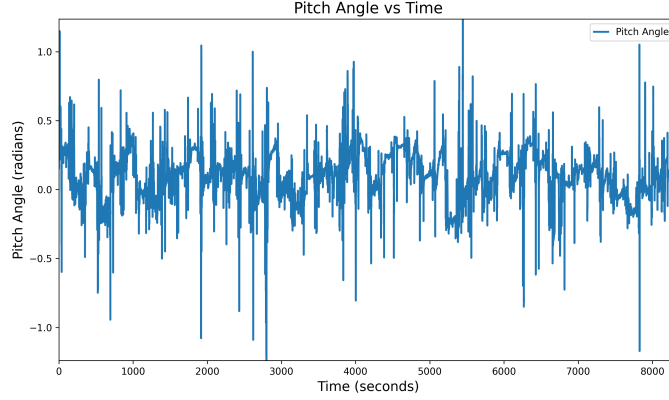


Figure 3.2.1: Temporal variation of raw pitch angle showing vertical head movement

shows the temporal variation of the raw yaw angle extracted from head motion traces of users watching the same omnidirectional video. Compared to pitch movements, the yaw angles exhibit larger fluctuations, indicating that users tend to explore the scene more extensively in the horizontal direction than in the vertical direction. These head orientation traces serve as the primary input signals for the viewport prediction models, enabling the learning algorithms to capture temporal viewing patterns and predict future user viewports.

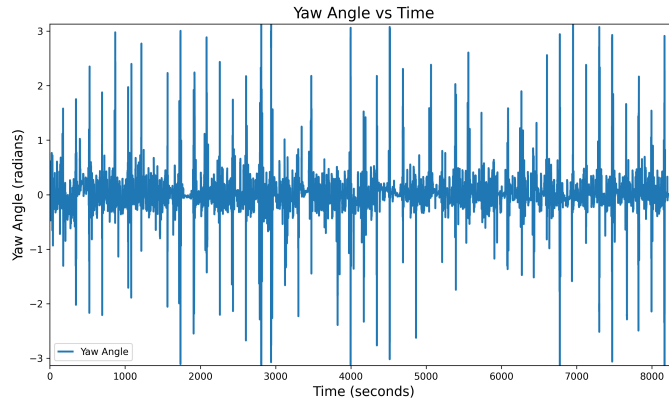


Figure 3.2.2: Temporal variation of raw yaw angle showing horizontal head movement.

To demonstrate user viewing behavior, Figure 3.2.3 shows the head movement trajectories associated with three users in the considered dataset while watching the same spherical video content. These trajectories represent the temporal evolution

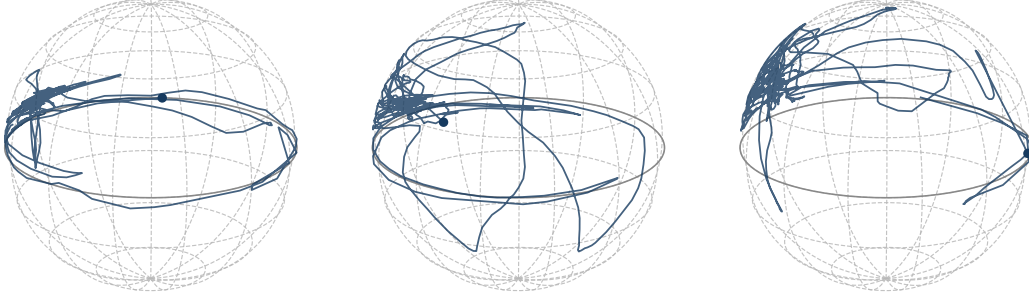


Figure 3.2.3: Head movement trajectories

of the users viewing directions as they explore different regions of the immersive scene during playback. For processing purposes, as explained further in the following section, such trajectories, together with the corresponding video frames, are projected onto 2D plane using several projection techniques which enable efficient analysis and processing of viewport information.

3.3 Dataset preprocessing

As discussed above, the first line of each data file contains timestamps that indicate the time instances at which the head orientation measurements are recorded during video playback. Because the sampling time, i.e. the timestamps, is the same for all users, the timestamp line is removed during the preprocessing stage. Therefore, a fixed time step of $\Delta t = 0.1$ seconds is assumed for all users of each video, which corresponds to the sampling frequency of the dataset. This assumption simplifies the processing of the head motion traces and allows the temporal evolution of the viewing direction to be analyzed using a uniform time interval across all users.

To process the viewport data in a planar domain, an ERP projection is adopted, which maps the spherical 360° video onto a 2D rectangular frame. Video files considered in this study are provided in 8K resolution, and the viewport resolution is

set to 1920×1080 (width $\times$ height), corresponding to the region observed by the user through HMD device.

Pitch and yaw angles representing the user’s viewing direction are converted into pixel coordinates $[p_x, p_y]$, which correspond to the horizontal and vertical positions of the viewport center on the projected 2D video frame. In other words, p_x represents the x-coordinate along the width of the ERP frame, while p_y represents the y-coordinate along the height of the frame. These coordinates indicate the exact location of the viewport center within the panoramic video frame and allow the user’s viewing trajectory to be represented in the 2D domain. With the positions p_x and p_y available over time, the horizontal and vertical viewport velocities $[v_x, v_y]$ can be computed as:

$$v_x = \frac{\Delta p_x}{\Delta t}, \quad v_y = \frac{\Delta p_y}{\Delta t}$$

where Δp_x and Δp_y represent the changes in position along the x and y axes, respectively, and Δt is the sampling time. These velocity components capture the temporal dynamics of the user’s head movement, providing additional information about how quickly the viewport shifts across the panoramic scene. Thus, the features considered as input to the ML models are p_x , p_y , v_x , and v_y , which jointly describe the spatial position and motion of the viewport over time.

For tile-based video streaming, each 360° video frame is divided into different number of tiles; however, in this study we considered nine tiles arranged in a 3×3 tile grid (three rows and three columns) in the ERP frame of immersive media delivery. This spatial partitioning enables selective transmission of different regions of the panoramic video according to the user’s viewport. The height and width of each tile are computed as $H_{tile} = H_{video}/3$ and $W_{tile} = W_{video}/3$, respectively. Since the ERP video resolution used in this study is 7680×3840 (8K), each tile has a width of $7680/3 = 2560$ pixels and a height of $3840/3 = 1280$ pixels. This tile-based representation allows the streaming system to allocate higher quality to tiles inside the user’s viewport while reducing the bitrate of tiles outside the visible region.

The user's FoV may contain 1, 2 or up to 4 tiles depending on the head orientation while watching the 360° video. As the user changes their viewing direction, the viewport may shift across tile boundaries, causing multiple tiles to fall within the visible region. In these cases, all tiles that overlap with the FoV must be transmitted and rendered to ensure a seamless viewing experience. If the FoV covers even a small portion of a tile, that tile must be encoded and delivered at high resolution to avoid visible quality degradation within the user's viewport.

This mechanism ensures that the user always receives high-quality content in the region they are currently observing while maintaining efficient bandwidth utilization for the remaining parts of the panoramic video. Figure 3.3.1 shows viewport representation on the 3×3 tile grid. The blue point denotes the viewport center (p_x, p_y) at any time instant, which is derived from the user's head orientation.

The red rectangle represents the viewport resolution of 1920×1080 pixels (width $\times$ height) within the projected ERP frame². Since the viewport can overlap with neighboring tiles, multiple tiles may fall inside the FoV and must therefore be transmitted to fully reconstruct the visible region.

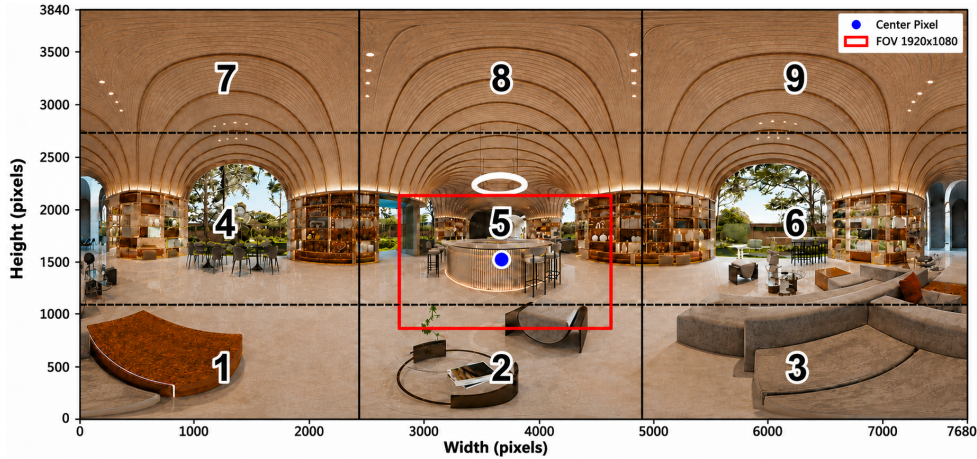


Figure 3.3.1: Viewport representation on a 3×3 ERP tile grid.

In tile based representation, adaptive bitrate decision allocates a high bitrate to

²<https://github.com/bytedance/E3PO/blob/main/e3po/e3po.yml>

the tiles within the user FoV, while neighboring or peripheral tiles get a relatively lower bitrate. The remaining non-visible tiles are assigned with the lowest bitrate. This technique is used for efficient delivery of immersive media content and helps to save bandwidth loss. In this study, pitch and yaw angles are transformed into pixel coordinates and plotted on projected video frame. These pixel coordinates represent user viewing position and FoV is constructed around them. For this experimental setup, FoV resolution is set to 1920×1080 pixels (width $\times$ height). The ML models use input features, containing pixel coordinates and velocity coordinates, to improve viewport prediction accuracy. For next-frame prediction, each model outputs 9 probability values, corresponding to the 9 tiles in the next frame. For 1-second prediction at 30 fps, the model predicts a sequence of 30 outputs, each containing 9 probability values. The same process is applied for 3-second viewport prediction.

3.4 ML models for viewport prediction

Various ML models are considered in this experiment to evaluate their performance in predicting user viewports for immersive 360° video streaming. The models are assessed for both short- and long-term prediction accuracy, with prediction horizons ranging from the next frame up to 5 seconds. The main objective is to analyze how different model architectures capture temporal patterns in user head movements and predict the user’s future viewing direction. In immersive video environments, accurately predicting the viewport is essential for enabling viewport-adaptive streaming, where high-quality video segments are delivered only to the regions that the user is expected to observe.

As explained in the workflow shown in Figure 3.4.1, the user’s head orientation (pitch and yaw sequences) are first converted into pixel coordinates (p_x, p_y) on the ERP-projected video frame. These pixel coordinates represent the viewport center over time. From these coordinates, additional motion features such as the horizontal

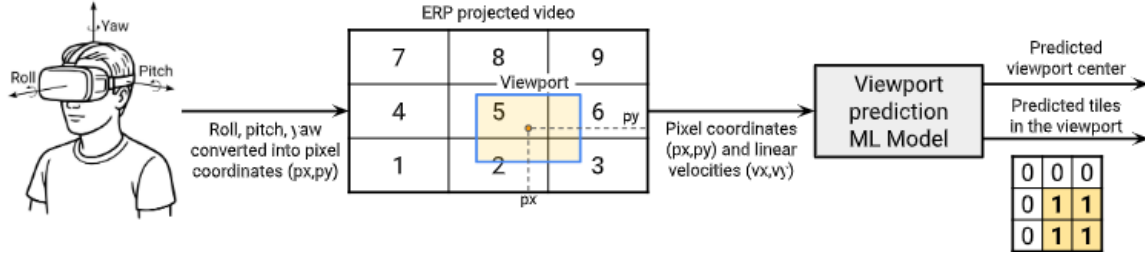


Figure 3.4.1: Workflow for Viewport Prediction Using ML Models.

and vertical viewport velocities (v_x, v_y) are computed to capture the dynamics of the user's head movement. These spatial and temporal features are used as inputs to the ML prediction models in exam. The trained models estimate the future viewport center, which is subsequently mapped onto the corresponding tiles within the 3×3 grid representation of the video frame. Two distinct evaluation representations are considered: coordinate-based viewport-centre prediction and tile-based multi-label classification over the 3×3 tile grid.

3.4.1 LSTM model

The LSTM model is employed in this study due to its ability of capturing temporal dependencies in sequential data to make it highly effective for predicting user viewing behavior over time. By leveraging its memory cells and gating mechanisms, the model can capture both short-term variations and longer-term motion trends in user head movements. Input sequence length for the LSTM model is set to 30 frames for both next-frame and 1-second FoV prediction, which provides sufficient recent context while maintaining computational efficiency. This configuration allows the model to learn the temporal relationship between recent head movement patterns and the immediate future viewport position with reasonable precision.

For long-term prediction horizons, the sequence length is increased to capture a broader history of user motion and behavioral patterns. In addition, for 3-second and 5-second FoV predictions, the sequence length is extended to 90 frames ($3 \times$

30) and 150 frames (5×30), respectively. This adjustment ensures that the model incorporates more comprehensive temporal information that enables it to better understand motion continuity and gradual directional changes. As a result, the model can produce more reliable and stable predictions of future viewport positions over extended time intervals, where short-term context alone would be insufficient.

The LSTM model used for this experiment contains three stacked layers with 200, 100 and 50 neurons, respectively, which allows it to progressively extract higher-level temporal features from the input sequence. This layered structure helps the model to capture both complex motion dynamics and subtle variations in user behavior over time. For next-frame or 1-second FoV prediction, the input consists of four features, namely the pixel coordinates (p_x, p_y) and velocity components (v_x, v_y) . Therefore, with a sequence length of 30, the input shape provided to the model is defined as (sequence length, number of features), i.e., (30, 4), which ensures that both spatial and motion-related information are effectively utilized. To improve generalization and reduce the risk of overfitting, a dropout layer with a rate of 0.5 is incorporated along with a batch-normalization layer to stabilize and accelerate the training process. These regularization techniques help the model maintain robustness across different user motion patterns. Moreover, a sigmoid activation function is applied in the output layer to produce normalized predictions suitable for viewport estimation tasks.

3.4.2 RNN model

The RNN model is employed using the same experimental setup as the LSTM model, including the same sequence length of 30 samples of pixel and velocity coordinates and the same training parameters. It contains three hidden layers of 200, 100 and 50 neurons, respectively, enabling the network to learn hierarchical temporal representations from sequential input data. This multi-layered structure allows the model to capture underlying patterns in user motion, although with less memory capability compared to more advanced recurrent architectures. A learning rate of

0.0001 is used during training to ensure stable and gradual convergence of the model parameters that reduce the likelihood of overshooting optimal solutions. A sigmoid activation function is applied at the output layer, which allows the model to generate normalized predictions suitable for viewport estimation tasks. The model is trained for a maximum of 150 epochs; however, to prevent overfitting and unnecessary computation, an early stopping mechanism is implemented with a patience value of 10 epochs. This ensures that training halts when no significant improvement is observed over consecutive iterations. Furthermore, a batch size of 32 is used to balance computational efficiency and gradient stability during training. Binary Cross Entropy (BCE) loss function is employed to optimize the model parameters to make it well-suited for learning prediction boundaries in this context.

3.4.3 GRU model

The GRU model is implemented using the same hyper-parameter configuration adopted for the LSTM and RNN models, ensuring consistency across all experimental settings. In particular, key training parameters such as the learning rate, batch size, number of training epochs and early stopping patience level are kept identical to those used in the previous models. This uniform configuration helps maintain a controlled environment for evaluation and avoids introducing variability due to differing training conditions. By standardizing these hyper-parameters, the comparison among the different recurrent architectures becomes more reliable and meaningful, as any observed differences in performance can be attributed primarily to the structural characteristics of the models themselves. This approach allows for a clearer assessment of how GRU architecture with its simplified gating mechanism performs in comparison to LSTM and RNN models in predicting user viewport behavior over time.

3.4.4 CNN model

The first layer of CNN model consists of one-dimensional (1D) convolutional filters with 16 channels followed by a Rectified Linear Unit (ReLU) activation function to introduce non-linearity into the network. This enables the model to effectively learn local patterns and feature representations from the input data. The input to the CNN is structured as (batch-size, 1, features), where the middle dimension represents a 1D signal rather than a traditional image. This formulation allows the convolutional layers to operate along the feature dimension while preserving the sequential characteristics of the data. To reduce the dimensionality of the resulting feature maps and retain the most relevant information, a max-pooling layer is applied after the first convolutional layer. The second CNN layer employs 32 filters which are followed by a ReLU activation function to further enhance the model's ability to learn complex feature interactions. An adaptive average pooling layer is used to produce a fixed-size output, ensuring compatibility with the subsequent Fully Connected (FC) layer, which performs the final prediction. To maintain consistency and ensure a fair comparison, the learning rate, number of training epochs, loss function, batch size and early stopping patience level are kept the same as those used in the LSTM, RNN and GRU models.

3.4.5 LR model

The LR model is used in state-of-the-art research for short-term viewport prediction [21,66]. In this work, it is implemented as a linear baseline model to enable comparison across various prediction horizons. It works as a reference point as this model provides insight into how well simple linear relationships can capture user viewing behavior when compared to more complex deep learning architectures. It uses a linear layer that learns a direct linear mapping between the input features and the predicted viewport without explicitly modeling temporal dependencies. To enhance its representational capacity while maintaining its role as a baseline, LR model is configured with three

hidden layers containing 1024, 512 and 256 units, respectively. This setup allows the model to better capture feature interactions while still relying on fundamentally linear transformations. The model is trained using the Adam optimizer along with BCE loss function that ensures stable and efficient optimization. Training of this model is performed for up to 150 epochs with a batch size of 32 and an early stopping mechanism with a patience value of 10 is applied to prevent overfitting and reduce unnecessary training time.

3.4.6 MLP model

The MLP model used in this study consists of three FC (linear) layers with 200, 100 and 50 neurons, respectively which enables the network to learn hierarchical feature representations from the input data. This architecture allows the model to capture non-linear relationships between the input features and the predicted viewport despite not explicitly modeling temporal dependencies as in recurrent models. A sigmoid activation function is applied at the output layer to perform normalized predictions that are suitable for viewport estimation tasks. The model is trained for up to 150 epochs with an early stopping mechanism where the patience level is set to 5 epochs to prevent overfitting and reduce unnecessary training time. BCE loss function is used during training to effectively optimize the prediction performance.

Table 3.1: Complexity and Real-Time Performance of Different Models

Model	Params (#)	Size (MB)	Latency (ms)	Throughput (seq/s)	Peak GPU Memory (MB)	Real-Time @30fps
LSTM	317,312	1.22	8.763	3,711.6	72.8	Yes
RNN	80,312	0.32	3.950	13,412.6	71.7	Yes
LR	55	0.05	0.654	116,594.0	68.9	Yes
MLP	26,711	0.11	1.298	39,190.0	69.4	Yes
CNN	76,562	0.30	2.798	21,164.7	70.3	Yes
GRU	238,312	0.92	7.052	8,416.9	81.1	Yes

The results observed in Table 3.1 provide a comparative analysis of the computational complexity and real-time performance of the evaluated models. In terms

of model size and parameter count, LSTM and GRU models are the most complex, containing 317,312 and 238,312 parameters, respectively, while the LR model has a negligible number of parameters, making it the most lightweight model. With respect to latency, the LR model achieves the lowest inference time (0.654 ms), followed by the MLP and CNN models, indicating their suitability for fast predictions. In contrast, the LSTM and GRU models show higher latency due to their recurrent nature and sequential processing requirements. A similar trend is observed in throughput where LR model achieves the highest throughput (116,594 seq/s) while LSTM model records the lowest throughput.

In terms of GPU memory consumption, all models demonstrate comparable usage, with values ranging between 68.9 MB and 81.1 MB where GRU model requires slightly higher memory resources. Despite these differences in computational complexity and efficiency, all models are capable of achieving real-time performance at 30 frames per second as indicated in the last column. Overall, these results highlight a trade-off between model complexity and computational efficiency. Non-sequential models such as LR and MLP offer faster inference and higher throughput whereas sequential models like LSTM and GRU provide better temporal modeling capabilities at the cost of increased computational overhead.

3.5 Results and Discussion

Experimental results show that CNN model achieves the best performance for next-frame viewport prediction which benefits its ability to capture local spatial patterns. In contrast, the GRU model demonstrates superior performance for longer prediction horizons of up to 5 seconds as it more effectively captures temporal dependencies over longer time intervals as discussed in mentioned in results. Accuracy is measured by comparing the tiles actually viewed by the user with those predicted by using ML model, providing a quantitative assessment of the model’s prediction performance.

It evaluates how many of the predicted tiles correctly match with the user's true FoV. Accuracy metric is defined as the ratio of correctly predicted tiles to the total number of predicted tiles as expressed in Equation 3.1. This measure offers a simple and interpretable way to assess the effectiveness of different models in capturing user viewing behavior.

Tile-selection accuracy is evaluated over the 3×3 tile grid. For each sample, the predicted and ground-truth viewports are represented by 9-dimensional binary tile vectors. Therefore, each of the nine tile labels is treated as a prediction, indicating whether the corresponding tile belongs to the FoV or not. The sigmoid output probability of each tile is converted into a binary label using a threshold of 0.5, where a probability greater than or equal to 0.5 is assigned a value of 1 (FoV tile), while a probability below 0.5 is assigned a value of 0 (non-FoV tile). The tile-classification accuracy is defined as

$$\text{Acc}_{\text{tile}} = \frac{N_{\text{correct,tile}}}{N_{\text{total,tile}}} = \frac{N_{\text{correct,tile}}}{9T}, \quad (3.1)$$

where $N_{\text{correct,tile}}$ is the number of correctly predicted tile labels, $N_{\text{total,tile}} = 9T$ is the total number of tile predictions, and T is the number of evaluated samples. Thus, Acc_{tile} evaluates tile prediction and does not measure the error in the predicted viewport-centre coordinates.

Among the available motion traces in the dataset, videos with indices 10, 14, 55 and 58 demonstrated higher viewport prediction accuracy compared with the remaining traces during the initial evaluation using LSTM model. These motion traces showed more consistent head-movement patterns, making them more suitable for reliable viewport-prediction analysis. In this study, video index 55 was selected as the primary dataset for further experimentation as it contains 48 users viewing history, which provides a sufficiently large number of samples for model training and evaluation. Using this motion trace, CNN model achieved the best viewport prediction accuracy: 93.93% for next-frame prediction, indicating its strong capability

in capturing short-term user viewing behavior.

Separately from the tile-based evaluation, viewport-centre prediction error is evaluated using RMSE, MSE and MAE. These regression metrics are calculated from the predicted viewport-centre coordinates $(\hat{x}_j, \hat{y}_j)$ and the corresponding ground-truth coordinates (x_j, y_j) .

Cumulative Distribution Function (CDF) of distortion provides a clear and more comprehensive view of distribution of prediction errors across all test samples. It enables the analysis of how prediction errors are distributed rather than relying only on average metrics. For next-frame viewport prediction, CDF is computed using total test samples (16,685), where the distortion values are derived from the viewport center error vectors. These error vectors quantify the deviation between predicted and ground-truth viewport center positions in terms of their horizontal and vertical coordinates. To calculate this deviation, Root Mean Squared Error (RMSE) is employed which captures the Euclidean distance between the predicted and actual viewport centers. The RMSE for each test sample is defined as:

$$\text{RMSE}_j = \sqrt{\frac{1}{2} [(x_j - \hat{x}_j)^2 + (y_j - \hat{y}_j)^2]} \quad (3.2)$$

where (x_j, y_j) represents ground-truth viewport center coordinates, $(\hat{x}_j, \hat{y}_j)$ denotes predicted coordinates and j is index corresponding to each test sample. This formulation ensures that both horizontal and vertical prediction errors are equally considered in the overall distortion measure.

$$\text{MAE}_j = \frac{1}{2} (|x_j - \hat{x}_j| + |y_j - \hat{y}_j|) \quad (3.3)$$

$$\text{MSE}_j = \frac{1}{2} [(x_j - \hat{x}_j)^2 + (y_j - \hat{y}_j)^2] \quad (3.4)$$

Mean Absolute Error (MAE) captures average absolute deviation between predicted and ground-truth viewport centers, while Mean Squared Error (MSE) empha-

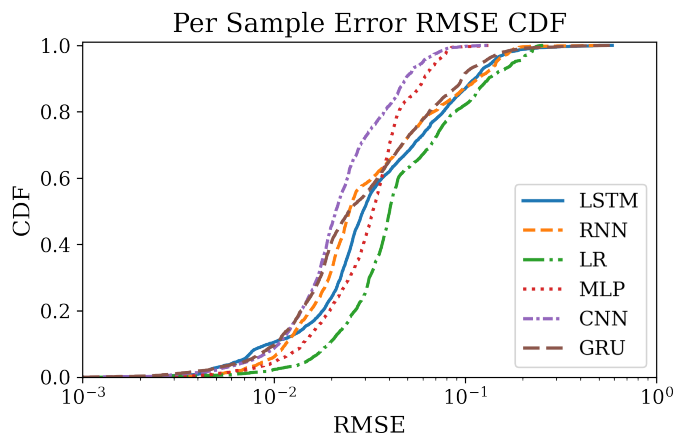


Figure 3.5.1: RMSE CDF obtained by the considered models.

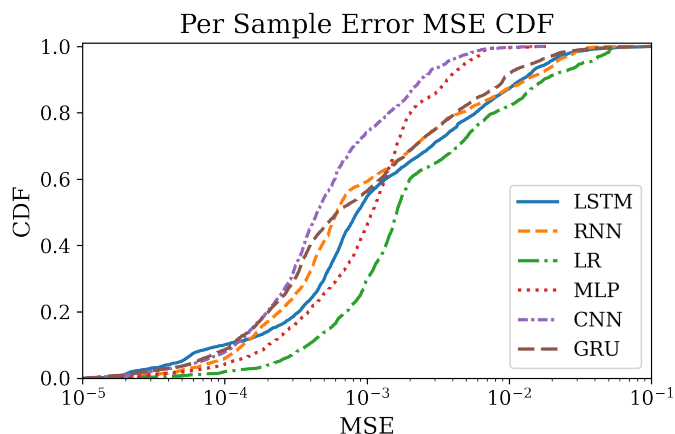


Figure 3.5.2: MSE CDF obtained by the considered models.

sizes larger errors due to the squaring operation. RMSE being square root of MSE provides an interpretable error measure in the same unit as the original coordinates.

Figure 3.5.1 shows CDF plot of RMSE for considered ML models in the case of next-frame viewport prediction. The curve that lies further to the left indicates that the model that produces smaller prediction errors more frequently reflects better regression accuracy. From figure 3.5.1, it can be observed that CNN model consistently outperforms the other models as its curve is shifted towards lower RMSE values. This indicates that CNN model achieves more precise predictions of viewport center as compared to other approaches.

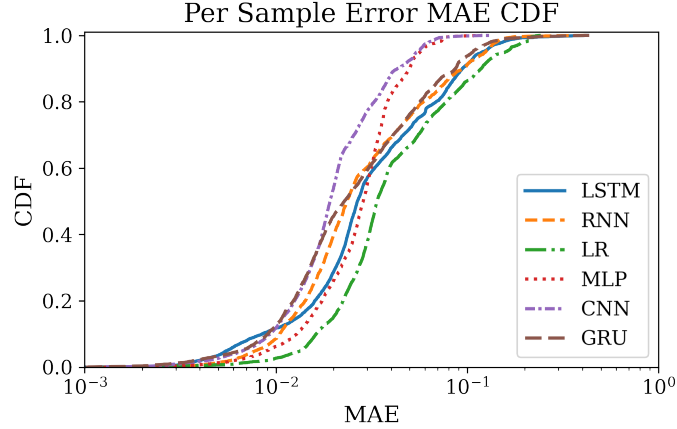


Figure 3.5.3: MAE CDF obtained by the considered models.

Figure 3.5.2 shows CDF plot of MSE for the evaluated models. Similar to RMSE, the curves that are positioned further to the left, correspond to lower error values and improved prediction performance. It can be observed that CNN model again demonstrates better performance as it maintains a higher cumulative probability at lower MSE values. This suggests that CNN model is more effective in minimizing larger prediction errors which are more penalized by squared error formulation. Other models such as LSTM and GRU show comparatively higher error distributions.

Figure 3.5.3 shows CDF plot of MAE for the considered models. MAE metric reflects the average magnitude of prediction errors without emphasizing large deviations. From previous figures, a leftward shift of curve indicates better performance. These results show that CNN model achieves lowest MAE values across most samples that confirms its effectiveness in producing consistently accurate predictions. While models such as RNN and GRU perform reasonably well as they exhibit slightly higher error distributions compared to the CNN that indicate less precise estimation of viewport positions.

CDF analysis across RMSE, MSE and MAE demonstrates that CNN achieves the best regression performance for short-term viewport prediction because it effectively captures spatial patterns in the input features.

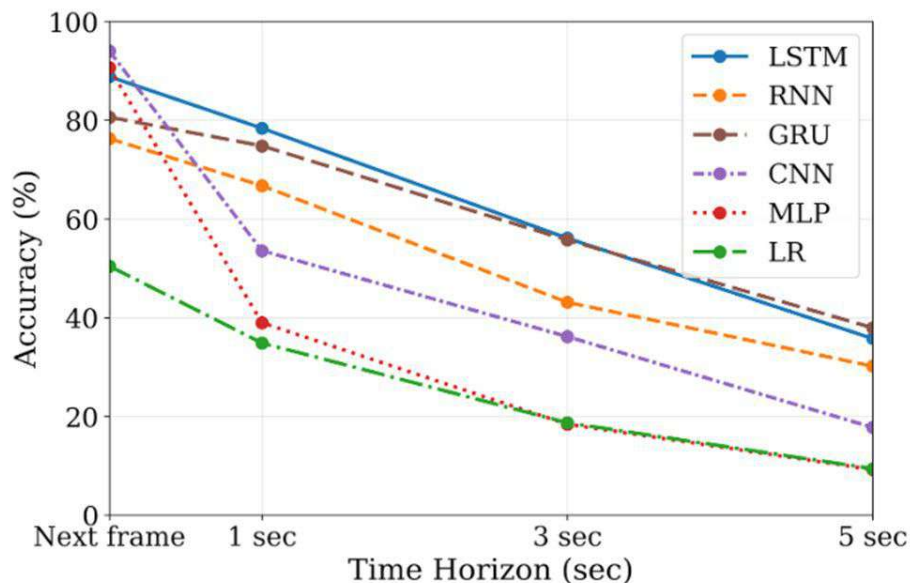


Figure 3.5.4: Comparison of tile selection accuracy across models.

Figure 3.5.4 represents accuracy in tile selection as a consequence of viewport prediction for various ML models across different prediction horizons, from next-frame up-to 5-second intervals. This figure shows how viewport prediction accuracy of the examined ML models varies as the forecasting horizon extends from next-frame prediction up to 5 seconds. It can be observed that LSTM model demonstrates a generally stable and satisfactory trend, highlighting its capability to effectively capture both short-term and long-term temporal dependencies. This makes it a reliable choice across varying prediction intervals.

For next-frame prediction, CNN model outperforms all other approaches which indicates its strong ability to capture immediate spatial patterns. Despite a better performance for next frame viewport estimation, CNN model shows a noticeable decline in accuracy as the prediction horizon increases. This suggests limitations in modeling long-term temporal dependencies. GRU achieves the best performance at the 5-second prediction horizon, with an accuracy of 37.93%. This indicates that GRU model is more effective in handling longer temporal sequences and maintaining

prediction reliability over extended time intervals.

3.6 Cross-Video Validation and Ranking Stability

To assess the generalization of the next-frame model comparison beyond a single trace, an additional cross-video validation experiment was performed. The same six model architectures and training configuration were considered. As before, a single trace was used for model development using 70% training, 10% validation and 20% testing split. Now all 27 remaining available traces were used for external evaluation. The evaluation comprised 1314 user-video trajectories and 7884 user-video-model evaluations. Performance dispersion was quantified across videos and user-video trajectories using the mean and standard deviation. Model-ranking stability was also assessed using the mean rank and rank standard deviation.

For this evaluation, viewport-center accuracy was defined as the percentage of predictions for which the normalized Euclidean distance between the predicted and ground-truth viewport centers was less than or equal to 0.05. Exact tile-set accuracy was used as a tile-based metric, where a prediction was considered correct only when all nine binary tile labels of the predicted FoV matched the corresponding ground-truth tile vector. Table 3.2 summarizes the performance dispersion across 27 external traces and 1314 user-video trajectories, together with the stability of the model ranking.

The cross-video validation results show that the model ranking remains consistent across the evaluated traces. CNN achieves the highest mean exact tile-set accuracy of $81.64 \pm 3.62\%$ across the 27 external traces and a mean tile rank of 1.00 ± 0.00 . In particular, CNN ranks first in exact tile-set accuracy on all 27 external traces. MLP is the second-ranked model, achieving a mean exact tile-set accuracy of $76.71 \pm 5.19\%$ and a mean tile rank of 2.07 ± 0.38 . LSTM and RNN show similar intermediate performance, while GRU and LR obtain lower exact tile-set accuracy. These results

Table 3.2: Cross-video and cross-user performance dispersion and model-ranking stability for next-frame viewport prediction.

(a) Cross-video performance and ranking stability

Model	Center Accuracy (%) Mean $\pm$ SD	Exact Tile-Set Accuracy (%) Mean $\pm$ SD	Center Rank Mean $\pm$ SD	Tile Rank Mean $\pm$ SD
CNN	74.15 $\pm$ 14.84	81.64 $\pm$ 3.62	1.33 $\pm$ 0.83	1.00 $\pm$ 0.00
MLP	64.92 $\pm$ 16.28	76.71 $\pm$ 5.19	2.00 $\pm$ 0.83	2.07 $\pm$ 0.38
RNN	36.18 $\pm$ 8.71	64.83 $\pm$ 6.26	3.93 $\pm$ 0.68	3.48 $\pm$ 0.58
LSTM	32.26 $\pm$ 9.88	65.22 $\pm$ 6.56	4.48 $\pm$ 1.19	3.44 $\pm$ 0.51
GRU	29.22 $\pm$ 14.59	55.47 $\pm$ 6.66	5.19 $\pm$ 1.27	5.00 $\pm$ 0.00
LR	40.16 $\pm$ 16.15	10.66 $\pm$ 8.88	4.07 $\pm$ 1.27	6.00 $\pm$ 0.00

(b) Performance dispersion across 1314 user-video trajectories

Model	Center Accuracy (%) Mean $\pm$ SD	Exact Tile-Set Accuracy (%) Mean $\pm$ SD
CNN	74.16 $\pm$ 19.04	81.63 $\pm$ 8.03
MLP	64.95 $\pm$ 21.66	76.69 $\pm$ 10.47
RNN	36.14 $\pm$ 16.67	64.80 $\pm$ 11.87
LSTM	32.18 $\pm$ 18.86	65.19 $\pm$ 11.83
GRU	29.15 $\pm$ 19.15	55.46 $\pm$ 13.06
LR	40.11 $\pm$ 25.05	10.62 $\pm$ 12.98

indicate that the superior next-frame performance of CNN is not restricted to a single selected video trace.

The user-level analysis shows greater variability than the video-level averages. Across the 1,314 user-video trajectories, CNN achieves the highest mean exact tile-set accuracy of $81.63 \pm 8.03\%$, followed by MLP with $76.69 \pm 10.47\%$. The larger standard deviations at user level indicate variations in individual viewing behavior. Nevertheless, CNN maintains the highest overall performance at both video and user levels. For viewport-center prediction, CNN also achieves the best mean rank of 1.33 ± 0.83 , providing further evidence of stable short-term prediction performance

across different videos and users.

3.7 State-of-the-art Comparison

Since viewport prediction is a time-series problem, the LSTM model is selected due to its well-known capability to learn long-term temporal dependencies in sequential data [48, 67–69]. This LSTM-based approach has been compared against the latest state-of-the-art approach named Multi-modal Fusion Transformer (MFTR) [70] to evaluate its effectiveness in viewport prediction. The comparison is conducted using two commonly adopted evaluation metrics Average Prediction (AP) and Average Overlap (AO) across different prediction horizons that includes next-frame, 1-second and 3-second intervals.

3.7.1 Dataset description

In [70], two datasets were employed to produce the reported results: PVS-HM dataset [71] and Xu-Gaze dataset [24]. The PVS-HM dataset is publicly available³ and contains head movement and eye movement data collected from 58 users across 76 panoramic (360°) videos. The Xu-Gaze dataset is similarly accessible online⁴ and contains head movement and eye-tracking data gathered from multiple users across 208 panoramic (360°) videos in dynamic scenes with each video viewed by at least 31 users. Results claimed in [70] were produced using these datasets.

However, in this study a different dataset is used for both training the LSTM model and reproducing results of the state-of-the-art MFTR approach under the same conditions to ensure a fair comparison. This dataset is publicly accessible online⁵ and was recorded by a PhD student at the University of Padua, Italy. The characteristics of the dataset employed in this research are summarized in Table 3.3,

³<https://github.com/YuhangSong/dhp>

⁴<https://github.com/xuyanyu-shh/VR-EyeTracking>

⁵<https://drive.google.com/file/d/1gldMkuX2ztzpPLhBCif8dLqB0Appk0Cq/view>

whereas a statistical summary of the selected data is provided in Table 3.4. 24 videos were used in the dataset selected where each video consists of gaze tracking data from multiple users. For each video, the number of frames varies significantly that ranges from 343 to 1,801 frames. Moreover, the number of participants differs across videos, ranging from 19 to 23 (number of users), which provides sufficient variation for training and evaluating the LSTM model for viewport prediction. The dataset contains gaze coordinates (e_x, e_y) and saliency maps. The gaze coordinates (e_x, e_y) are used to derive the velocity coordinates (v_x, v_y) .

3.7.2 Data preprocessing

In this experimental setup, the new dataset mentioned above was used for both LSTM model and for reproducing the results of the state-of-the-art MFTR model [70]. In the first step, raw dataset was preprocessed and converted into `.pk1` files, which are used as the required input format to reproduce MFTR model results. The MFTR implementation was obtained from the publicly available source code⁶ to ensure consistency in the evaluation.

The original MFTR study [70] evaluated the model using a sampling rate of 1 fps. In this work, the publicly available MFTR implementation was re-evaluated at 30 fps, using the same temporal resolution adopted for the LSTM model. This deviation from the original MFTR configuration provides denser temporal samples and may affect the prediction performance because consecutive observations are more closely spaced in time. Therefore, the MFTR results reported in this thesis should be interpreted as a re-evaluation under the 30-fps experimental setting rather than a direct reproduction of the results reported in [70]. Nevertheless, using the same 30-fps setting for both LSTM and MFTR ensures a consistent comparison between the two models within the experimental framework of this thesis. In MFTR model, first step was to preprocess gaze data and then results were recorded for different

⁶<https://github.com/alanzhangcs/MFTR>

Table 3.3: University of Padua dataset used in this research.

Video ID	Number of Frames	Number of Users
218	408	19
220	852	19
226	1247	19
239	721	19
243	537	19
244	460	19
246	629	19
248	591	19
250	811	19
255	977	23
261	1079	23
264	474	23
272	372	23
277	801	23
291	1290	20
299	343	20
300	609	20
301	940	20
303	1801	20
304	459	19
306	1069	19
313	792	19
314	991	19
316	721	19

Table 3.4: Statistics summary of dataset.

Metric	Value
Original dataset videos	104
Videos examined	24
Minimum frames for a single video	343
Maximum frames for a single video	1801
Users number variation for each video	19 to 23

prediction horizons containing next-frame, 1-second and 3-seconds. Evaluation of this approach was performed using the same metrics defined in [70] as AP and AO to ensure a fair and consistent comparison.

3.7.3 Viewport Prediction Framework

Two different input configurations were considered for comparison. In the first configuration, only gaze data was considered which contained four features as gaze position (e_x, e_y) and velocity coordinates as (v_x, v_y) . In second configuration, saliency information was also incorporated along with gaze data to analyze the combined effect of visual attention and user gaze on prediction performance.

In the first configuration, the input to the LSTM model consisted of the following four features: (e_x, e_y) and (v_x, v_y) , whereas the output was defined as a binary classification over a 3×3 ERP projection tile grid, resulting in an output vector of size 9 (one value per tile for each ERP projected video frame). Each element of this binary vector represents whether a particular tile belongs to predicted viewport, where one or more tiles (either 1, 2 or 4) can be active depending on the viewport size. The viewport size was defined as (960, 540) pixels (*width, height*), representing the spatial region of interest used for evaluation.

The architecture of this model is composed of three stacked LSTM layers with number of neurons as 256, 128 and 64 respectively. Batch normalization and dropout regularization layers are used after each LSTM layer to improve generalization and to prevent overfitting. The last layer of this model is passed through an FC layer to make a vector output of 9 corresponding to a 3×3 tile-based viewport representation. The output of this model gives nine probabilistic values. A threshold of 0.5 is applied to these values, which means that if the probability value of any tile is equal to or greater than 0.5, it is considered a viewport tile; otherwise, it is considered a non-viewport tile. A sigmoid activation function is used to produce binary predictions. A weighted BCE loss is used to mitigate class-imbalance issue. Training of the model is done with

150 epochs with batch size of 32 and use of Adam optimizer with starting learning rate of 10^{-4} along with a learning rate scheduler to adaptively reduce the learning rate based on validation performance. Early stopping is applied with patience level of 10 to prevent overfitting. Evaluation of the model is done using two metrics such as AP which measures exact match accuracy and AO which quantifies the overlap between predicted and ground-truth viewport tiles. The framework used for the first configuration is identical to the one used in the previous comparative analysis of different models. Head movement coordinates (pitch, yaw angles) are replaced with gaze coordinates as (e_x, e_y) and only one model is considered as LSTM model. Compared with approaches that predict the exact viewport center point (i.e., the precise gaze position of the user), this configuration predicts which tiles are active within the viewport without considering the exact point the user is looking at. Results of AP and AO were recorded using first configuration which consists of gaze data along with velocity coordinates for both MFTR and proposed LSTM model.

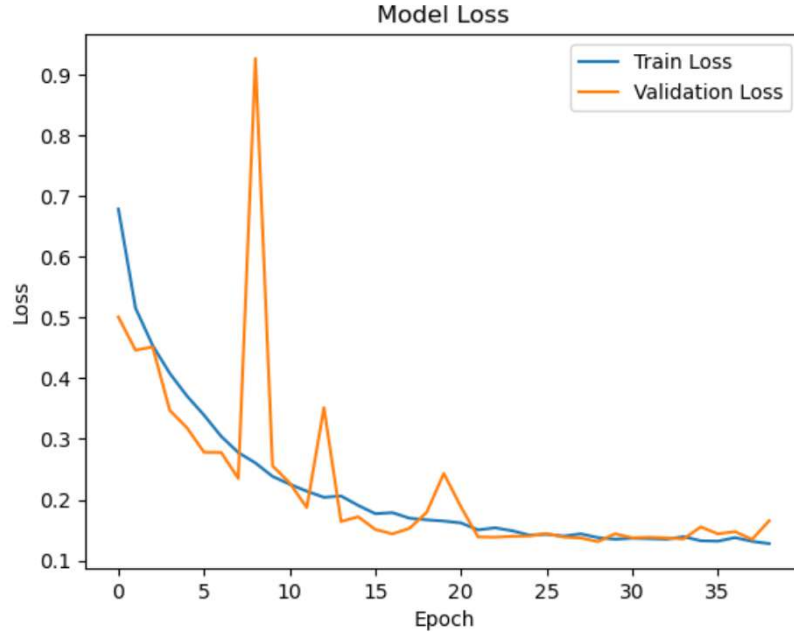


Figure 3.7.1: Training and validation loss curve of LSTM model.

Figure 3.7.1 represents both training and validation loss curves of LSTM model

over multiple epochs. It shows that training loss decreases steadily which indicates effective learning of temporal patterns from input features. Similar decreasing trend is also observed for validation loss curve although it performs some fluctuations when training starts, especially around epoch 8 to 10 where a high spike is observed. This behavior represents model instability at initial learning phase and sensitivity to validation samples. After some epochs, both training and validation losses stabilize and converge to similar values which indicates that the model generalizes well without prominent overfitting behavior. A small gap is observed between training and validation loss that confirms the robustness of the model. In general, loss trends from both curves indicate that the proposed LSTM architecture achieves stable convergence and maintains good generalization performance on unseen data.

For the second configuration, gaze coordinates (e_x, e_y) along with saliency maps are considered to incorporate both user eye tracking and visual content information to improve viewport prediction performance. In this case, gaze coordinates represent user viewing direction while saliency maps provide spatial distribution of visually important regions in the video frame. The input to the LSTM model consists of gaze features (e_x, e_y) combined with saliency-based information extracted from corresponding frames. Similar to first configuration, output is defined as a multi-label binary classification over a 3×3 ERP projection tile grid, resulting in an output vector of size 9. Each element of this vector indicates whether a particular tile belongs to the predicted viewport, where multiple tiles (1, 2, or 4) can be active depending on the viewport size. The same viewport size of (960, 540) is used to maintain consistency in evaluation.

The architecture of this configuration is kept similar to the first configuration, consisting of three stacked LSTM layers with 256, 128 and 64 neurons respectively. Batch normalization and dropout layers are applied after each LSTM layer to enhance generalization and reduce overfitting. The final output is obtained through a FC layer that maps learned temporal features to output vector of size 9 which belongs to

the tile-based representation. The same multi-label classification framework is used where each output tile indicates the probability of being within the user's FoV. A sigmoid activation function is applied at the output layer to obtain binary predictions. Weighted BCE loss is again used to address class imbalance in viewport tiles. Training setup remains similar to previous configuration as 150 epochs, batch size of 32 and Adam optimizer with an initial learning rate of 10^{-4} . Learning rate scheduler is also employed to adaptively reduce the learning rate depending on validation loss and early stopping with a patience level of 10 to avoid overfitting.

The main difference in this configuration is the incorporation of saliency information with gaze coordinates. This provides additional contextual information about visually important regions for the user [70]. Saliency maps are used to extract visual-attention information from video frames, which helps the model better align predicted viewport tiles with visually important regions. The evaluation of this configuration is done using the same metrics used as AP and AO to make a fair comparison with the first one. The results indicate that including saliency information slightly reduces the overlap performance (AO) in this configuration. However, these results can be improved by fine tuning of model hyper-parameters and giving more information as input features such as head, gaze and saliency images along with velocity coordinates.

Figure 3.7.2 shows the behavior of both training and validation loss curves for second configuration. From the figure, training and validation loss curves decrease rapidly at the start of model training which indicates effective learning from both gaze features and saliency information. After some epochs, these curves begin to stabilize and gradually converge to a lower value. Some fluctuations can also be observed in validation loss, however the overall trend remains consistent with training loss. A small gap between the two curves represents good generalization capability without significant overfitting. Furthermore, the loss behavior shows stable convergence of the hybrid LSTM-CNN model.

For the tile-based comparison with MFTR, AP and AO are used as tile-based

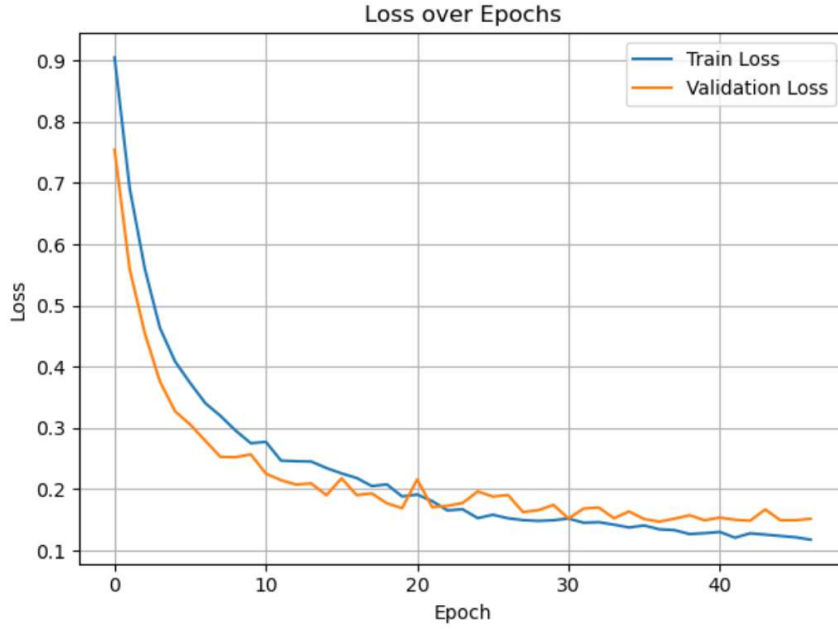


Figure 3.7.2: Training and validation loss curve of LSTM-CNN model.

evaluation metrics, distinct from the viewport-centre regression metrics RMSE, MSE and MAE defined in Section 3.5. AP measures the exact-match accuracy between predicted and actual viewport tiles (ground truth viewport tiles) over a sequence of length T . As defined in Eq. (3.5), It is computed as the average number of correct predictions across all time steps, where $\hat{p}_i$ and p_i represent the predicted and ground-truth viewport tiles at time step i , respectively. The term $\mathbf{1}(\hat{p}_i = p_i)$ denotes an indicator function that equals 1 when the predicted and ground-truth tiles exactly match and 0 otherwise. AO quantifies the spatial overlap between the predicted and actual viewport regions. As defined in Eq. (3.6), it is calculated as the fraction of the ground-truth viewport area covered by the predicted viewport. The function $S(\cdot)$ in Eq. (3.6) denotes the viewport region defined by its width W and height H . Compared with AP, AO is a more flexible evaluation metric because it accounts for partial overlap between the predicted and ground-truth viewport regions. In addition, AP evaluates strict prediction accuracy, whereas AO measures the spatial alignment quality between the predicted and actual viewports.

$$AP = \frac{1}{T} \sum_{i=1}^T \mathbf{1}(\hat{p}_i = p_i) \quad (3.5)$$

$$AO = \frac{1}{T} \sum_{i=1}^T \frac{|S(\hat{p}_i, W, H) \cap S(p_i, W, H)|}{|S(p_i, W, H)|} \quad (3.6)$$

Table 3.5: Comparison of LSTM and MFTR using gaze data.

Prediction Horizon	LSTM (gaze)		MFTR (gaze)	
	AO	AP	AO	AP
Next frame	98.47	92.47	96.92	87.09
1 sec	91.91	70.18	92.09	48.30
3 sec	79.91	44.20	82.87	27.11

Table 3.5 presents a comparison between the LSTM model and MFTR when both gaze and velocity coordinates are used as input features. From table 3.5, the LSTM model outperforms MFTR in terms of AP, achieving higher AP values across all prediction horizons. For next frame prediction, LSTM provides higher AO and AP values, showing better performance in terms of accurate and consistent viewport prediction. As prediction horizon is increased up to 3 seconds, performance degradation can be observed for both models; however, LSTM keeps higher AP values that demonstrate better temporal modeling capability.

Table 3.6: Comparison of LSTM–CNN and MFTR using gaze and saliency data.

Prediction Horizon	LSTM–CNN (gaze+sal)		MFTR (gaze+sal)	
	AO	AP	AO	AP
Next frame	98.39	93.22	98.55	87.90
1 sec	86.90	63.41	90.80	48.59
3 sec	75.86	39.65	84.49	30.64

Table 3.6 presents comparison for the second configuration, where both gaze and saliency information are used together. It shows that AP performance of proposed LSTM-CNN model is higher as compared to MFTR model across all prediction

horizons. For next frame prediction, both models obtain almost the same comparable AO values, however LSTM-CNN indicates higher AP that shows better exact tile prediction. For longer horizons, AO values decrease for both models. However, the LSTM model still maintains better AP performance, which shows the effectiveness of combining gaze features $((e_x, e_y))$ with saliency information. Moreover, these results indicate that LSTM-based approach outperform MFTR in terms of AP accuracy while maintaining competitive overlap performance AO especially for short-term predictions.

Figure 3.7.3 shows AP accuracy using only gaze data of the proposed LSTM model compared with MFTR across different prediction time horizons. It indicates that LSTM consistently outperforms MFTR for all horizons. For next frame prediction, both LSTM and MFTR models achieve high accuracy, while LSTM provides slightly improved AP accuracy. When prediction horizon is increased from 1 second to 3 seconds, AP accuracy decreases for both models because of uncertainty; however, LSTM is still able to maintain a better accuracy as compared to MFTR. This demonstrates the effectiveness of LSTM to capture temporal dependencies using gaze data for short and long-time horizons of viewport prediction.

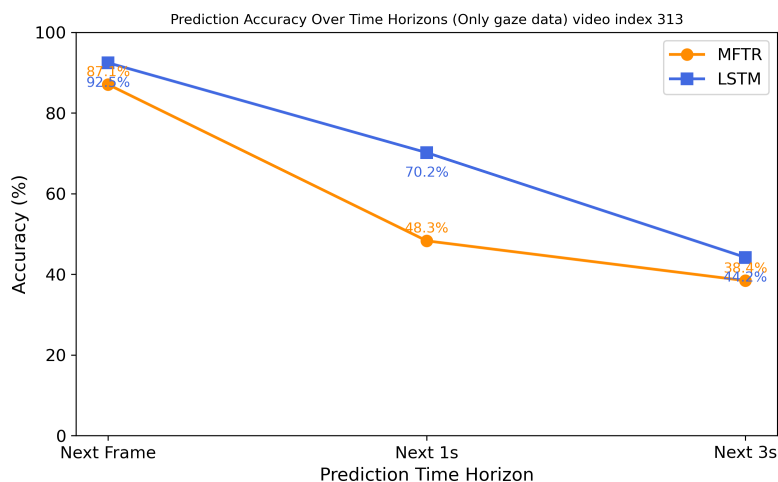


Figure 3.7.3: AP accuracy across different time horizons using only gaze data.

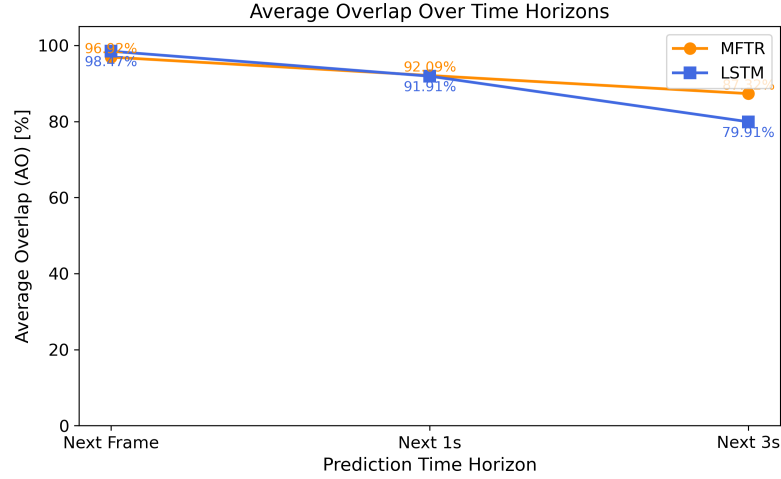


Figure 3.7.4: AO across different time horizons using only gaze data.

Figure 3.7.4 shows AO comparison using only gaze data between LSTM and MFTR across various prediction horizons. It indicates that both models achieve high AO values for next frame prediction while LSTM is able to slightly outperform MFTR. Comparing for 1 second AO, both models provide comparable performance with small difference in AO. However, when prediction horizon is increased to 3 seconds, MFTR achieves better AO values as compared to LSTM. It shows that while LSTM is better performing for short time interval predictions, MFTR maintains relatively better spatial overlap for long-term prediction horizons.

Figures 3.7.5 and 3.7.6 show the AP and AO performance comparison of the LSTM-CNN model with the state-of-the-art MFTR approach using both gaze and saliency information across different prediction horizons. Figure 3.7.5 shows that the proposed model achieves higher AP compared to MFTR for all prediction horizons. For next frame prediction, both models exhibit high AP values, with the proposed model demonstrating improved performance. As the prediction horizon increases from 1 second to 3 seconds, AP decreases for both models due to increased prediction uncertainty; however, the LSTM model consistently maintains higher AP values, indicating its ability to effectively capture both temporal and spatial information.

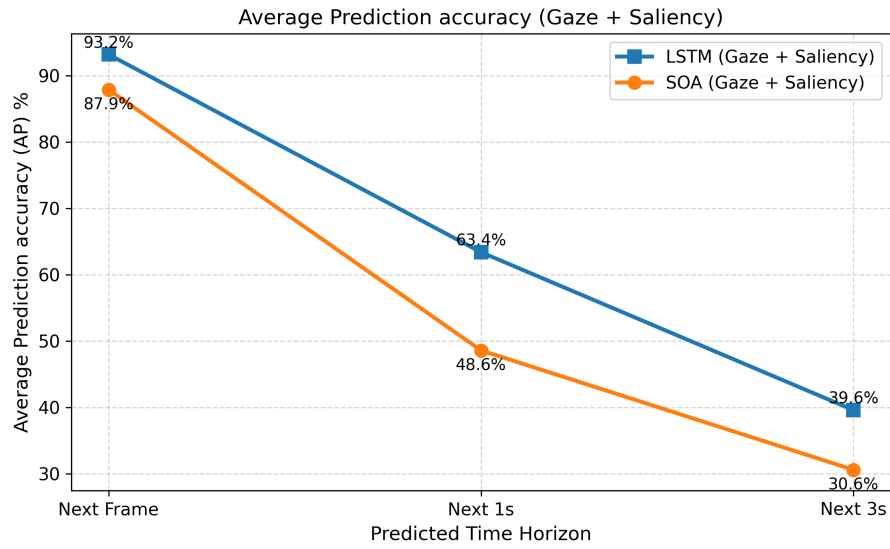


Figure 3.7.5: AP across different time horizons using gaze and saliency data.

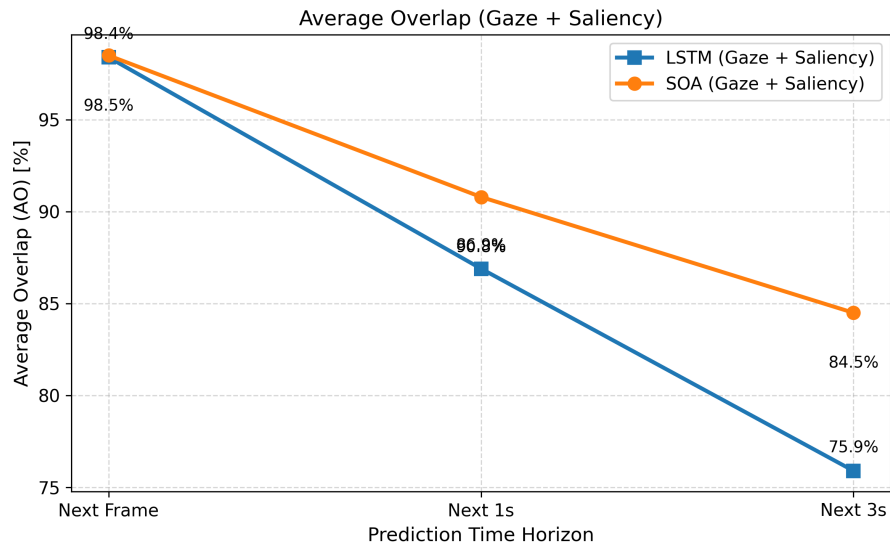


Figure 3.7.6: AO across different time horizons using only gaze and saliency data.

Figure 3.7.6 shows a comparison of AO at different prediction horizons. It can be observed that both models achieve very similar AO values for next frame prediction. For longer prediction horizons, MFTR slightly outperforms the LSTM model in terms of AO. This indicates that while the LSTM-CNN model provides more accurate tile-based predictions (higher AP), the MFTR approach maintains slightly better spatial overlap for longer horizons. In general, the results show that use of saliency information improves prediction accuracy and the hybrid LSTM-CNN model outperforms the MFTR approach in terms of AP while maintaining competitive AO performance across all prediction horizons.

3.8 Summary of chapter

This chapter is classified into two parts. The first part presents a comparison of various ML models, while the second part compares the LSTM model with a state-of-the-art approach (MFTR). The first part covers the complete viewport prediction pipeline, from preprocessing of user viewing histories based on pitch and yaw angles to comparing with accuracy of all considered ML models and to check their suitability for real-time deployment. A publicly available dataset from E3PO framework, whose link is provided in this chapter, is used for this purpose. From pitch and yaw angle data, pixel coordinates are derived and then used to compute velocity components, which together form the input feature set for the ML models. The performance of these examined ML models, i.e. LSTM, RNN, GRU, CNN, LR and MLP, is analyzed across multiple prediction horizons from next frame up-to 5 seconds prediction horizon. For next-frame viewport prediction, the CNN model outperforms all other models, while for longer prediction horizons of up to 5 seconds, the GRU and LSTM models demonstrate better performance from other models considered in this study. CNN performance decreases rapidly as the prediction horizon increases because its ability to capture long-term temporal dependencies is limited. In contrast, LSTM and GRU

perform better at longer prediction horizons because they are designed to model temporal dependencies.

To investigate the generalization of next-frame model comparison, external validation was performed using the remaining 27 available traces. The analysis covered 1314 user-video trajectories and quantified performance dispersion across both videos and individual users. CNN retained the most stable short-term performance, achieving a mean exact tile-set accuracy of $81.64 \pm 3.62\%$ across videos and a mean tile rank of 1.00 ± 0.00 , ranking first on all 27 external traces. The evaluation demonstrates that the principal short-term model ranking is not restricted to a single motion trace.

The second part of this chapter presents a comparison of LSTM model with state-of-the-art MFTR approach, whose source code is publicly available and linked above in this chapter. From its consistent performance across long-term prediction horizons observed in first part, the LSTM model was selected for this comparison. Two evaluation metrics are considered for a fair comparison as AP and AO which are defined in MFTR study and their equations are provided in this chapter. Two different experimental configurations were used: one using only gaze data and the other combining gaze data with saliency maps. These comparisons were conducted across three prediction horizons defined as next-frame, 1-second and 3-second viewport prediction. In both configurations, the LSTM model achieved better AP performance compared to MFTR for accurate tile-based viewport prediction. However, the AO performance of the LSTM model was slightly lower in some cases, which could be improved by tuning the hyperparameter or by replacing the model with GRU model.

Part II

Adaptive 360° Video Streaming Systems

Chapter 4

360° Video Streaming Using QUIC

4.1 Introduction

Immersive multimedia streaming has gained high demand in recent years due to application of 360° video streaming in AR/VR interactive multimedia systems. Comparing with traditional video streaming, 360° video provides a full panoramic view of the environment that allows users to explore different regions of the 360° view. However, this flexible behavior causes inefficiency, as at any given moment, users only observe a limited portion of the entire spherical video named as FoV while the remaining content remains unseen.

As previously explained, streaming the complete 360° video results in excessive bandwidth consumption, which creates a significant challenge for real-time delivery over constrained networks. To address this limitation, tile-based streaming approaches have been widely adopted in state-of-the-art approaches. A key challenge in tile-based streaming lies in accurate predictions of user's FoV. Since network latency prevents instantaneous delivery, predicting the user's motion enables proactive transmission of relevant tiles, thereby improving QoE. Recent advances in ML-based networks such as LSTM and GRU showed better performance to capture user long-term dependencies for better viewport prediction results [12, 68].

In parallel, QUIC protocol which operates over UDP and forms the foundation of HTTP/3, offers significant advantages over traditional TCP-based streaming. It includes low latency in media content delivery, improved congestion control and reducing HoL blocking. This chapter integrates LSTM-based viewport prediction, tile-based multi-bitrate encoding and QUIC-based adaptive streaming. The system is evaluated using motion traces from the E3PO dataset ¹ and its performance is analyzed in terms of missing ratio, bitrate efficiency and streaming latency.

4.2 System Architecture

The system architecture integrates video pre-processing, multi-bitrate encoding, ML-based viewport prediction, adaptive bitrate decision and QUIC-based delivery into a combined end-to-end pipeline. The system is designed to efficiently deliver 360° video content while considering bandwidth constraints and to improve user QoE. Each part of the architecture operates independently and combined with other components that enables flexible behavior for future enhancements. The system workflow starts with user motion traces (pitch, yaw angles) that captures user's head orientation while watching 360° video content. The video index 55 is preprocessed and used for viewport prediction based on LSTM network, which learns temporal dependencies in user behavior and predicts future viewport positions. These prediction results are incorporated with adaptive bitrate decision for network latency and content delivery.

Fig. 4.2.1 represents the complete end-to-end pipeline of the system that contains viewport prediction, adaptive tile selection and QUIC-based delivery. In this framework, the system dynamically selects user's FoV tiles in high priority and non-FoV tiles on low priority based on results from predicted viewport. The tiles are assigned bitrate levels according to classification of tiles in various qualities as high, medium and low that makes sure viewport region is delivered at higher quality while

¹<https://bytedance.larkoffice.com/drive/folder/QQgJfhxs7lor3xdb0WGcTYMsnPb>

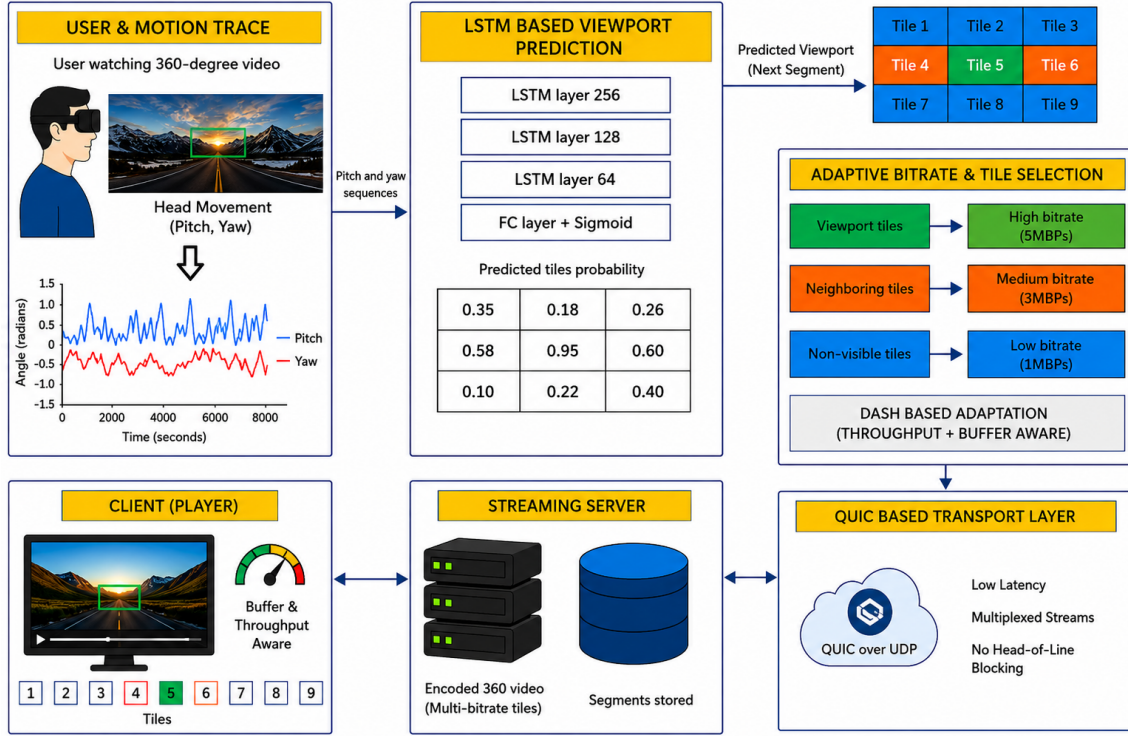


Figure 4.2.1: Overview of viewport-adaptive 360° video streaming framework

neighboring or non-visible regions are transmitted at lower bitrates. The selective allocation of resources significantly improves bandwidth efficiency without changing video quality. Video content is delivered using the QUIC protocol, which provides a low-latency and multiplexed parallel streams over UDP. The system leverages QUIC that supports multiple parallel streams that classifies these streams into priority based as high-priority (viewport tiles), medium or low-priority to neighboring tiles of viewport and low-priority (non-viewport tiles) for transmission. This prioritization confirms that tiles are delivered at their selected bitrate or condition mentioned in algorithm even under variable network conditions. If network bandwidth drops, FoV tiles bitrate changes from 1Mbps to 0.5Mbps and in worst case as 0.3Mbps to all tiles.

The system operates in a closed-loop pattern, where client requests are updating continuously based on both predicted viewport information and real-time network feedback, such as variation in buffer occupancy and throughput. The server responds

to these requests by scheduling tile transmissions using priority-based queuing mechanisms such as Strict Priority (SP) or Weighted Fair Queuing (WFQ). Dynamic interaction between client and server enables the system to adapt changing network conditions and maintain stable playback performance. Integration of viewport prediction, adaptive bitrate selection and QUIC-based transport makes a complete framework that optimizes both network resource utilization and user experience. The architecture has possibility to adapt any advancement for future integration of latest prediction models, adaptive bitrate algorithms and transport optimizations.

4.3 Video Encoding and Tiling

The first step of viewport-adaptive immersive streaming framework is the preprocessing of raw 360° video content which is used later for viewport prediction and media streaming using the QUIC protocol. The panoramic video is encoded into a number of segments and corresponding files are stored into a MPD file for immersive media delivery [40,72]. This step involves transforming the input video into a number of tiles where each tile is encoded at different quality levels and transmitted based on user demand. Implementation of the framework follows a structured pipeline consisting of video acquisition, selection of projection, encoding, tiling and segmentation. The 360° video is downloaded from an online source (link is mentioned at start of this chapter) and trimmed to a fixed duration to ensure consistency across experiments. This step also reduces computational overhead and allows controlled evaluation of the streaming system. The video is converted into YUV format used in video encoding.

The encoding process is performed using Kvazaar encoder, which supports advanced video coding features and contains tile-based partitioning. The aio-quic project code initially uses division of any video content into the grid of 10×20 tiles which is transformed into a grid of 3×3 tiles that results into a total of 9 tiles instead of 200 tiles per frame. This tiling process enables precise spatial control over video quality

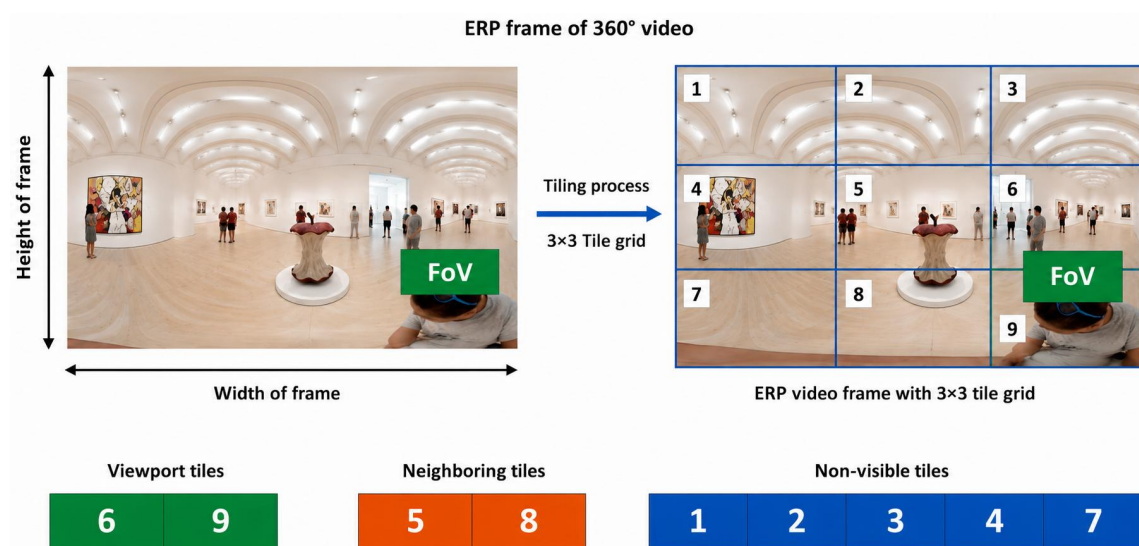


Figure 4.3.1: 3×3 tile partitioning of an ERP 360° video frame.

that allows different regions of the frame to be encoded and transmitted at multiple bitrate levels. The tile partitioning is important in 360° video streaming, where only a subset of tiles contains user FoV at any given time. To enable adaptive streaming, the simulator considers three nominal per-tile bitrate levels of 1 Mbps, 0.5 Mbps and 0.3 Mbps.

These bitrate representations provide flexibility in selecting the appropriate quality level for each tile based on its importance and network conditions. The use of multiple bitrate levels confirms that the system can adapt to bandwidth fluctuations while maintaining visual quality.

Fig. 4.3.1 shows the 3×3 tile layout used to classify regions into viewport, neighboring and non-visible tiles. Following the encoding process, the tiled video streams are saved into MP4 containers and segmented into fixed chunks using MP4Box and considering DASH standard. Each segment belongs to a short time interval of 1 second that enables better temporal adaptation during streaming. The segmentation process is useful for independent retrieval of video segments which is essential for low-latency adaptive streaming. This multi-bitrate tiled representation forms the foundation of the viewport-adaptive streaming system. Combining tiles of different

bitrates, the system acts more flexibly for adaptive content delivery that allows efficient use of network resources based on predicted user behavior. As a result, pre-processing step makes its contribution to reduce bandwidth consumption while maintaining high-quality viewing experience within the user's FoV.

4.4 Viewport prediction

Accurate prediction of user's FoV is an important part of this framework as it directly influences the efficiency of tile-based streaming and the user QoE. In immersive 360° video streaming systems, user interacts with dynamic scenes along with changing head orientation to watch the visible region of the scene. Therefore, predicting future viewport positions enable the system to fetch relevant tiles, solving low latency and minimizing unnecessary bandwidth consumption. In this work, the user's motion trace consists of head orientation parameters such as pitch and yaw angles that are considered as the primary input for the prediction model. These user HMD data inherently exhibit temporal dependencies as the user tends to follow smooth trajectories over time. To capture these temporal user patterns, a LSTM network is used with the same hyper-parameters, loss function, training setting and final sigmoid layer as previously used in viewport prediction chapter. The LSTM networks belong to class of recurrent neural networks which are designed to model sequential data by maintaining both short-term and long-term memory through gated mechanisms, capturing temporal sequences and perform better for long-term viewport prediction tasks. The model processes a sequence of past motion data (named as sequence length) to predict the user viewport for the next frame. The input to the prediction model is a temporal sequence of pixel coordinates and velocity coordinates over a sliding window of length T , expressed as:

$$X_t = \{x_{t-T+1}, x_{t-T+2}, \dots, x_t\} \quad (4.1)$$

where $x_t = [\text{yaw}_t, \text{pitch}_t]$ shows the head orientation of the user at time step t . This sequence is used as input to the LSTM network to capture temporal dependencies in user motion. Temporal evolution of the hidden state is defined as:

$$h_t = \text{LSTM}(x_t, h_{t-1}) \quad (4.2)$$

where x_t denotes the input motion feature vector at time step t , h_{t-1} is the previous hidden state and h_t is updated hidden state. This recursive formulation allows the network to capture temporal information from previous user movements and process it for future viewport prediction. The predicted viewport for next time step is defined as:

$$\hat{y}_{t+1} = f(h_t) \quad (4.3)$$

where $\hat{y}_{t+1}$ is the predicted viewport for the next time step and $f(\cdot)$ represents the fully connected (FC) output layer. In this system, output is mapped to a 9-dimensional tile representation corresponding to the 3×3 tiled video layout. Architecture of the model contains three stacked LSTM layers with 256, 128 and 64 neurons, respectively. This structure allows the model to recursively learn complex temporal features that start from low HMD movement to slightly changing position to explore different parts of the video. To improve generalization and prevent overfitting, batch normalization and dropout layers are employed after each LSTM layer. These regularization techniques stabilize the training process and reduce the model's sensitivity to noise in the input data. Final output layer of network is obtained through a FC layer that maps the temporal features to an output vector of size 9 that corresponds to a 3×3 tile-based representation of the viewport. A multi-label classification framework is used where each output neuron represents the probability of a tile being within the user's FoV. A sigmoid activation function is applied at the output layer to get probability scores for each tile. Probability of each tile whether it belongs to the predicted viewport or not is computed as:

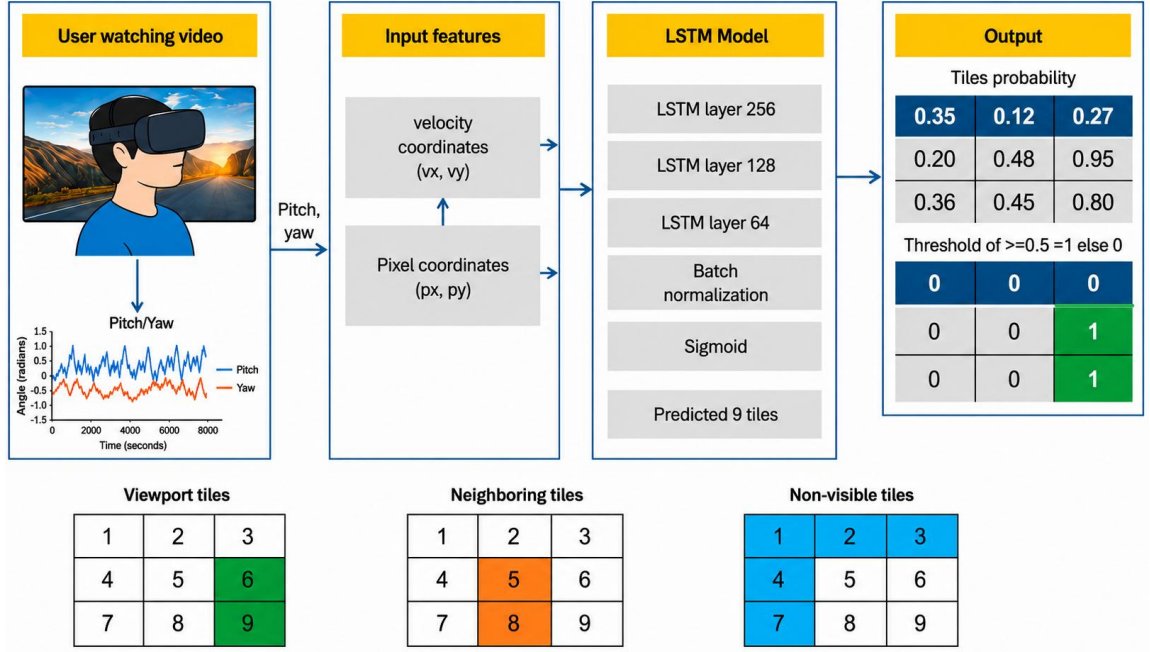


Figure 4.4.1: Architecture of LSTM-based viewport prediction model.

$$p_i = \sigma(z_i) = \frac{1}{1 + e^{-z_i}}, \quad i = 1, 2, \dots, 9 \quad (4.4)$$

where z_i is the logit produced by the output layer for tile i , and p_i represents the predicted probability that tile i belongs to the user's viewport. These probabilities are given threshold of 0.5 to determine which tiles should be prioritized for transmission as FoV and which tiles are not in user FoV. A binary decision for tile selection is obtained using a threshold τ :

$$\hat{v}_i = \begin{cases} 1, & \text{if } p_i \geq \tau \\ 0, & \text{otherwise} \end{cases} \quad (4.5)$$

where $\hat{v}_i = 1$ shows that tile i is predicted to be inside FoV. This decision is used by the adaptive streaming module to classify tiles into viewport, neighboring and non-visible categories.

Fig. 4.4.1 presents the architecture of the LSTM-based viewport prediction model

used in this framework. In addition to the model architecture, a key challenge in viewport prediction is class imbalance where only a subset of tiles belongs to user FoV while the majority do not lie in FoV, a weighted BCE loss function is used. The weighted BCE loss is defined as:

$$\mathcal{L}_{\text{WBCE}} = -\frac{1}{N} \sum_{i=1}^N [w_1 y_i \log(\hat{y}_i) + w_0 (1 - y_i) \log(1 - \hat{y}_i)] \quad (4.6)$$

where y_i is the ground-truth label for tile i , $\hat{y}_i$ is the predicted probability w_1 and w_0 are the weights assigned to positive and negative classes respectively, and N is number of output tiles which is 9 in this case. Loss function increases penalty for poor prediction of viewport tiles and helps the model to learn minority FoV classes effectively. Weighting mechanism assigns higher importance to positive samples (tiles within FoV) which confirms that model does not show biased behavior towards predicting non-FoV tiles. The model is trained for 150 epochs using a batch size of 32 and Adam optimizer with an initial learning rate of 10^{-4} . To enhance training efficiency, a learning rate scheduler is used in adaptive way to reduce the learning rate based on validation loss.

In addition, early stopping with a patience level of 10 epochs is used to prevent overfitting and to make sure that the model retains its generalization capability. Using above mentioned hyper-parameters to capture temporal correlations in user HMD movement and making it a robust deep learning architecture, the LSTM-based model performs accurate and reliable viewport predictions. These predictions are subsequently used for proactive tile selection to improve streaming efficiency and reducing latency in the system.

4.5 QUIC-Based Streaming Framework

The system is built on the transport layer of QUIC protocol and implements the *aiokuic* framework. QUIC is a modern transport protocol that operates over UDP and

forms the foundation of HTTP/3. Comparing with traditional TCP-based streaming approaches, QUIC offers various advantages that includes reduced connection establishment latency through 0-RTT handshakes, better congestion control mechanisms and multiplexed streams without HoL blocking. These features make QUIC a better choice for low latency applications such as viewport-adaptive 360° video streaming. In this framework, the server is responsible for handling tile requests from the client and delivering the corresponding video segments. After receiving a request from client, the server looks at the required segment, tile index and bitrate level, and retrieves the requested encoded tile from the storage. The server transmits the requested data over QUIC streams in a chunk-based way that enables efficient and continuous data delivery across all segments. Transmission time for a tiled segment can be approximated as:

$$T_{i,t}^{\text{tx}} = \frac{S_{i,t}}{B_t} \quad (4.7)$$

where $S_{i,t}$ is size of tile i in a segment t and B_t is the available bandwidth at that time. This equation provides a basic estimation of delivering latency and important to make decision whether a tile can be delivered before its playback deadline or not. This design confirms that large video files are streamed correctly that reduces memory overhead and allow smooth playback at the client side. Interaction between the client and server, together with the priority-based tile delivery process is shown in Fig. 4.5.1.

Table 4.1 summarizes the relationship between client tile requests and server transmission behavior across all segments. Efficient way to manage multiple incoming tile requests is that the server uses two queuing mechanisms: SP and WFQ. In the SP, tile requests are requested based on their assigned priority levels, which makes sure that high-priority tiles (FoV) are transmitted before lower-priority (neighboring and non-FoV) ones. This approach reduces delay for FoV content, however it may lead to poor transmission of low-priority tiles under heavy load conditions. Also, WFQ provides a balanced allocation of resources by classifying bandwidth among

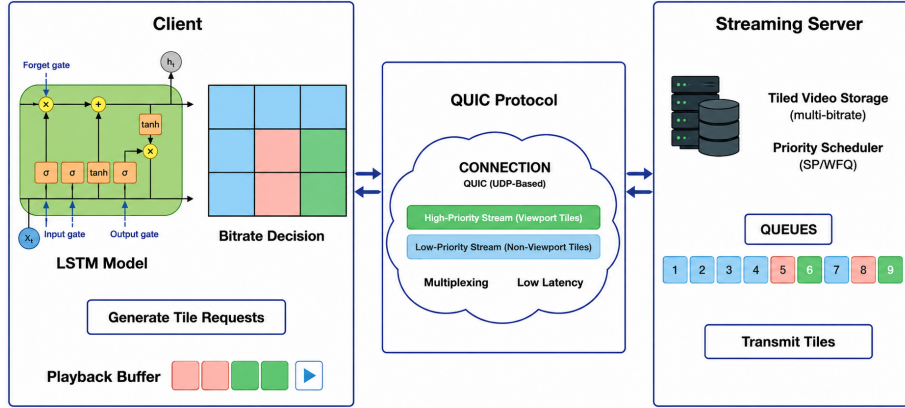


Figure 4.5.1: Client-server interaction in QUIC-based viewport-adaptive streaming framework.

different priority levels according to predefined weights. This helps the system to keep a smooth playback. Client side contains tile requests that are produced dynamically based on the predicted viewport results from LSTM model, also depending on current network and buffer conditions. In priority-based transmission, each requested tile is associated with a priority score:

$$P_{i,t} = \beta_1 V_{i,t} + \beta_2 N_{i,t} + \beta_3 Q_t \quad (4.8)$$

where $V_{i,t}$ shows whether tile i belongs to the predicted viewport, $N_{i,t}$ shows whether it is a neighboring tile and Q_t is current buffer occupancy. Coefficients β_1 , β_2 , and β_3 control the relative importance of viewport relevance and to start playback. This score is used by the server to transmit tile under SP priority which is used in this chapter for experiments. The client classifies tiles into different priority levels depending on their relevance to the user's FoV. Tiles which are within predicted FoV are assigned high priority and transmitted over QUIC stream, while neighboring and non-FoV tiles are assigned lower priority and transmitted over low-priority stream. This difference of traffic across multiple streams represents parallel transmission of data and prevents low-priority data to stop delivery of FoV tiles.

Furthermore, the client maintains a playback buffer and checks if it is filled

Table 4.1: Client tile request and system behavior per segment

Segment	BWE (Mbps)	Req. Bitrate (Mbps)	Tiles	Behavior / Explanation
1	6.19	9.00	9	Initial buffering; bandwidth lower than demand
2	10.55	9.00	9	Stable buffering; sufficient bandwidth
3	10.51	4.50	9	Bitrate reduced; stable transmission
4	10.66	4.50	9	Stable adaptive streaming
5	12.22	5.90	9	Increased bitrate; good network condition
6	10.92	5.90	9	Stable performance maintained
7	10.74	4.50	9	Reduced bitrate; conservative adaptation
8	10.34	5.90	9	Adaptive allocation high at FoV
9	10.53	4.50	9	Stable adaptive streaming
10	10.89	4.50	9	Last segment; stable condition

to confirm smooth video playback. Initial buffering prefetches a number of video segments before playback starts, reducing the chance of playback interruptions. In this experiment, initial buffer size of 2 segments is selected which means first two segments data is prefetched before playback starts. In streaming process, if buffer level drops below a specific threshold, client triggers to send additional requests to refill the buffer and to maintain continuous playback. Integration of QUIC with viewport-adaptive streaming provides efficient and robust transport media delivery. Using QUIC’s multiplexed streams for adaptive tile selection in two priorities, the system obtains low latency delivery and efficient use of bandwidth. This interaction of transport-layer and application-layer intelligence plays a significant role to enable high-quality immersive video streaming across the streaming session.

4.6 Adaptive Streaming Strategy

The adaptive streaming strategy in the framework plays an important role to maintain video quality and bandwidth efficiency. By integrating viewport prediction results with dynamic bitrate tiles selection, the system confirms that network resources

are effectively utilized while maintaining a high QoE for the user. Comparing with conventional streaming systems that uniformly deliver video content, this approach captures spatial and temporal information to allocate bitrate in adaptive way across all regions of the video. For every video segment, the client calculates the bitrate using a DASH-based adaptation algorithm that considers both network throughput and playback buffer status. The estimated available throughput for segment t is defined as:

$$\hat{B}_t = \frac{S_t}{D_t} \quad (4.9)$$

where S_t is the size of the segment data (downloaded segments) in bits and D_t is the corresponding download time in seconds. This estimated throughput is used by the client to check a safe bitrate level for subsequent tile requests.

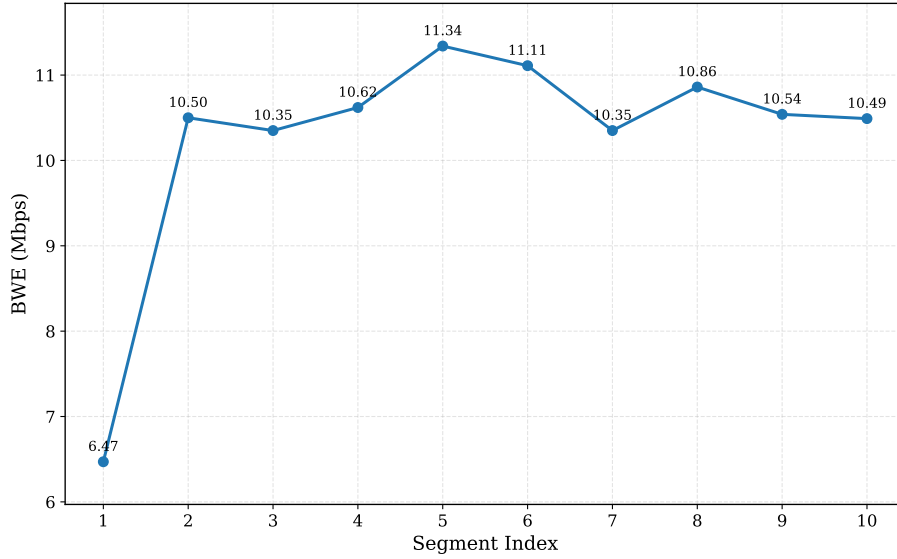


Figure 4.6.1: Estimated available bandwidth for each segment based on actual segment size and download time.

Fig. 4.6.1 shows bandwidth estimation for all segments where the client uses the available transmission capacity from observed segment delivery behavior. The throughput provides an estimation of the available network capacity, while the buffer

level shows the system's resilience to playback interruptions. The playback buffer over time can be calculated as:

$$Q_{t+1} = \max(0, Q_t + \Delta_t - P_t) \quad (4.10)$$

where Q_t represents the buffer occupancy at time t , Δ_t is the duration of latest downloaded playable content and P_t is the playback consumption during the same interval. This equation is used to observe the client buffer and leads the adaptation policy toward conservative bitrate selection when the buffer level becomes very low. Considering these parameter limitations, the client may select a suitable bitrate that prevents buffer underflow (leading to rebuffering) and high bandwidth usage. Another important part of this system is classification of tiles based on their relevance to the predicted FoV. These tiles are classified into three categories:

- **Viewport tiles (high importance):** These tiles fall within user predicted FoV and considered as high priority tiles to the user's visual experience.
- **Neighboring tiles (medium importance):** These tiles are located adjacent to the viewport acting as buffer against prediction errors. In this study, these are given as low priority and bitrate allocation of 0.5Mbps.
- **Non-visible tiles (low importance):** These tiles lie outside user FoV and are unlikely to be observed by the user.

Across all segments, tile set is partitioned into viewport, neighboring and non-visible regions which forms the basis for adaptive bitrate selection. Each category is assigned to a corresponding bitrate level such that viewport tiles are transmitted at highest bitrate (1 Mbps), neighboring tiles at medium bitrate (0.5 Mbps) and non-visible tiles at lowest bitrate (0.3 Mbps). The bitrate allocation confirms that the FoV regions of the video are delivered with maximum quality if network condition is

good, while less important regions as medium or low quality bitrate. The adaptive bitrate selection is defined as:

$$R_{i,t} = \begin{cases} R_{\text{high}}, & \text{if tile } i \in \text{FoV} \\ R_{\text{medium}}, & \text{if tile } i \in \text{Neighbor} \\ R_{\text{low}}, & \text{otherwise} \end{cases} \quad (4.11)$$

where $R_{i,t}$ shows the bitrate assigned to tile i at segment t and R_{high} , R_{medium} and R_{low} indicates the available bitrate levels for FoV, neighboring and non-FoV tiles, respectively. Total requested bitrate for a specific segment t is given by:

$$R_t^{\text{req}} = \sum_{i=1}^{N_t} R_{i,t} \quad (4.12)$$

where N_t shows the number of requested tiles for segment t . This aggregate bitrate is used to check the transmission demand of each segment under the adaptive streaming policy. Combining with tiles assignment category, it represents spatial bitrate allocation behavior of the framework. In addition to spatial adaptation, the system also incorporates temporal adaptation through constantly updating of bitrate decisions across all segments.

Fig. 4.6.2 shows how requested bitrate varies across all segments according to the selected tile set and bitrate selective decision. When network conditions change, the DASH-based algorithm dynamically adjusts the selected bitrate levels to make sure robustness against bandwidth peaks. Furthermore, viewport prediction allows selected FoV tiles to reduce the impact of network latency and improve smooth content delivery. Table 4.2 summarizes the adaptive bitrate allocation across viewport, neighboring and non-visible tile categories for each segment.

An important advantage of this strategy is its ability to deal with prediction errors. By allocating medium bitrate to neighboring tiles, the system provides a safety margin that reduces the effect of prediction errors, thereby lowering visible quality.

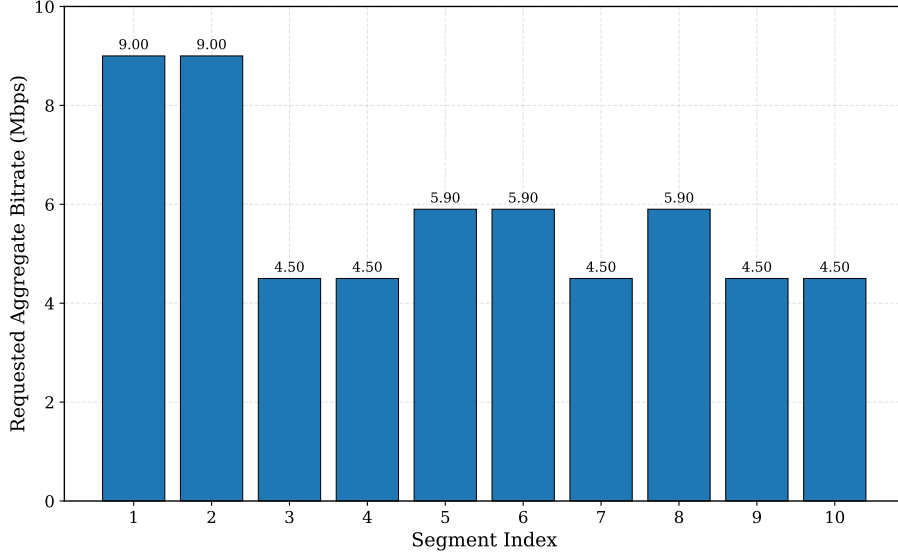


Figure 4.6.2: Requested aggregate bitrate for each segment under the adaptive tile-based streaming policy.

This design choice improves the robustness of the system in real-time environments where user behavior may be unpredictable. The proposed adaptive streaming strategy is efficient as it combines prediction, prioritization and bitrate adaptation to obtain an optimal trade-off between visual quality and bandwidth consumption. By dealing with optimized way to both spatial and temporal features to remove their redundancy in 360° video streaming systems, the system performance improves streaming efficiency while maintaining a smooth playback to meet the user experience.

4.7 Experimental Setup

E3PO dataset is used in this experiment to run the complete pipeline of this streaming system. The dataset contains head movement information captured from real users interaction with immersive video content which makes it suitable for viewport prediction results evaluation and adaptive streaming strategies. In this study, the video sequence *v1_s1* is selected to do all these experiments mentioned in this chapter, along with its corresponding motion trace file containing user HMD data in terms of

Table 4.2: Adaptive bitrate allocation across tile categories

Segment	FoV (Mbps)	Neighbor (Mbps)	Non-visible (Mbps)	Total Bitrate	Strategy	Explanation
1	1.0	0.5	0.3	9.00	Aggressive	Full allocation; BWE insufficient ($<$ demand)
2	1.0	0.5	0.3	9.00	Aggressive	Full allocation; stable (BWE $\geq$ demand)
3	1.0	0.5	0.3	4.50	Reduced	Lower bitrate due to buffer drop
4	1.0	0.5	0.3	4.50	Stable	Conservative stable transmission
5	1.0	0.5	0.3	5.90	Adaptive	Increased FoV emphasis (good BWE)
6	1.0	0.5	0.3	5.90	Adaptive	Stable adaptive allocation
7	1.0	0.5	0.3	4.50	Reduced	Conservative bitrate control
8	1.0	0.5	0.3	5.90	Adaptive	Higher FoV quality maintained
9	1.0	0.5	0.3	4.50	Stable	Balanced conservative mode
10	1.0	0.5	0.3	4.50	Stable	Final stable allocation

pitch and yaw angles.

The pixel coordinates which come from pitch and yaw angles are plotted on 2D video frame to make tile-based viewport representation, which acts as input for the LSTM-based prediction model as well as the adaptive streaming module. Each frame of the video is mapped to a 9 tiles grid that allows the system to find out which tiles belong to the user's FoV at any given time. Also, tile occupancy vector for any frame t can be expressed as:

$$\mathbf{v}_t = [v_{1,t}, v_{2,t}, \dots, v_{9,t}] \quad (4.13)$$

where $v_{i,t} \in \{0, 1\}$ shows whether tile i is inside the user's FoV at time t . This representation is used for prediction target during model training and basis for tile request generation during streaming process. This mapping enables a direct link between user behavior and tile selection while streaming immersive media content.

The simulation environment is configured with a fine-grained spatial partitioning

of the video into 9 tiles per segment which are arranged in a 3×3 grid. This level of granularity provides better control over spatial quality allocation and allows the system to observe effective difference between viewport and non-viewport regions.

To support adaptive bitrate streaming, three nominal bitrate levels are considered in the simulator: 0.3 Mbps, 0.5 Mbps and 1 Mbps, corresponding to low, medium, and high quality, respectively. These values are used for simulation purposes and are not derived from the 7680×3840 video resolution. The streaming framework is implemented in Python using the *aiokuic* library, which gives support for QUIC-based communication. The system is composed of two main components as: a server and a client. The server is responsible for storing encoded video segments and giving response to tile requests, while the client observes user behavior by generating requests based on motion trace data and predicted viewport positions.

The communication between the client and server is carried out over QUIC streams that enables low-latency and multiplexed streams for data transfer. Several experiments are conducted by executing both server and client modules in a controlled simulation environment. The present evaluation focuses on throughput and buffer-driven adaptation under a controlled network setting; time-varying propagation delay and explicit packet-loss profiles are not emulated. The transport evaluation is limited to ten one-second video segments and a single-user setting. In addition, the implemented bitrate controller is a rule-based throughput and buffer-aware adaptation strategy and is not benchmarked against formal ABR controllers such as Model Predictive Control (MPC) or DRL. These aspects are outside the scope of the current experiments and are considered as future work.

The client reads user motion traces from a CSV file and generates tile requests at each segment based on the predicted viewport and adaptive bitrate selection strategy. The server processes these requests using priority-aware queuing mechanisms and transmits the corresponding video tiles. During the simulation, various performance metrics such as missing ratio and bitrate distribution are recorded for analysis.

To ensure stable playback and real-time streaming conditions, an initial buffering phase is introduced, during which a predefined number of segments are pre-loaded before playback begins. This helps to reduce the impact of network variation and reduces errors in smooth playback. In addition, the system constantly observes buffer level and dynamically adjusts tile requests to maintain smooth streaming. All experiments are repeated under consistent configuration settings to ensure reproducibility of results.

Table 4.3 summarizes the key parameters of the experimental setup, including video resolution, tile configuration, bitrate levels and model training settings. These parameters form the basis for results evaluation and to check the performance of the viewport-adaptive QUIC streaming framework.

4.8 Performance Metrics

To evaluate performance of viewport-adaptive streaming system, a set of quantitative metrics is considered. These metrics are designed to capture different aspects of system behavior that includes reliable streaming process, viewport quality, bandwidth efficiency and low latency. By analyzing these indicators jointly, a better understanding of the system's effectiveness can be obtained.

One of the primary metrics used in this study is the *missing ratio*, which measures the percentage of tiles that are not successfully received before their playback deadline. This metric reflects the reliability of the streaming system and its ability to deliver content on time. A lower missing ratio indicates better performance as it shows that most of the required tiles are available for rendering during playback. The missing ratio is defined as:

$$\text{Missing Ratio} = \frac{N_{\text{missed tiles}}}{N_{\text{total tiles}}} \times 100$$

where $N_{\text{missed tiles}}$ shows the number of tiles that were not received in time and

$N_{\text{total tiles}}$ shows the total number of requested tiles. Table 4.4 represents the segment-wise missing-ratio and FoV missing-ratio statistics used for the analysis.

While the missing ratio for all tiles across all segments provide a general indication of system performance, yet it does not fully capture the user's perceived quality, as not all tiles contribute equally to the viewing experience. To address this limitation, the *FoV missing ratio* is used, which evaluates the performance of the system by looking at percentage of missing tiles within the user's FoV. Since the FoV region corresponds to the portion of the video that is actually visible to the user, this metric gives more accurate representation of QoE. It is computed as:

$$\text{FoV missing ratio} = \frac{N_{\text{missed FoV tiles}}}{N_{\text{total FoV tiles}}} \times 100 \quad (4.14)$$

where $N_{\text{missed FoV tiles}}$ represents number of missing tiles inside the user's FoV and $N_{\text{total FoV tiles}}$ is the total number of viewport tiles requested. This metric is more relatable to user-perceived quality as it focuses only on visible content. A low FoV missing ratio indicates that the system successfully prioritizes and delivers viewport content even under constrained network conditions.

Experimental results show that missing events within the FoV are lower than the overall missing ratio that reflects better user-perceived quality in viewport tiles. Total missing ratio is 19.44% and total fov missing ratio is 9.05%. In addition to reliability metrics, bandwidth efficiency is evaluated through the *average bitrate*. This metric represents the mean bitrate delivered across all segments and tiles during the streaming session and provides insightful results that how effectively the system utilizes available network resources. It is defined as:

$$\bar{R} = \frac{1}{T} \sum_{t=1}^T R_t^{\text{req}} \quad (4.15)$$

where R_t^{req} is total requested bitrate for segment t and T is the total number of streamed segments. This metric shows the average network load generated by the

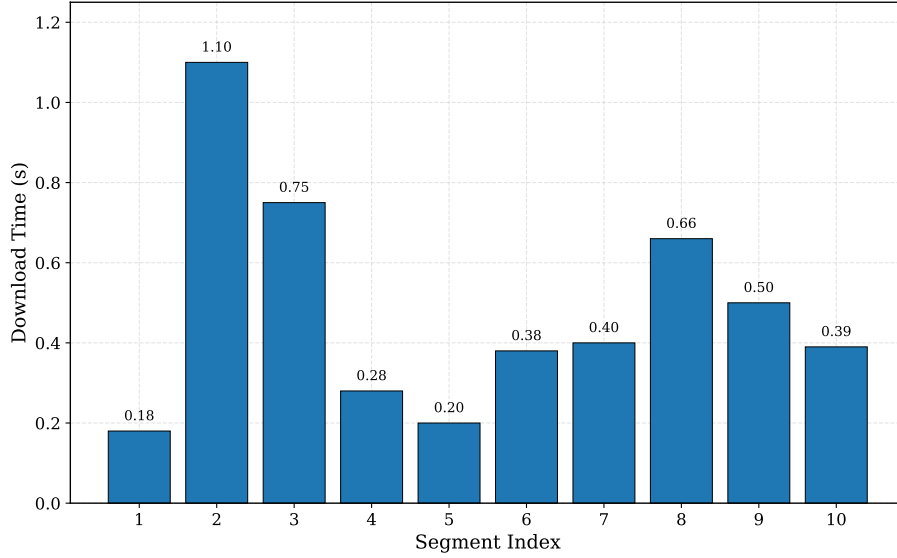


Figure 4.8.1: Segment-wise download time measured during QUIC-based delivery.

adaptive streaming strategy. A well-designed adaptive streaming system aims to maximize perceived quality while minimizing the average bitrate and to achieve an efficient trade-off between quality and resource consumption. Another important metric is the *download time*, which measures the time required to fetch each video segment from the server. The download time for segment t is defined as:

$$D_t = T_t^{\text{end}} - T_t^{\text{start}} \quad (4.16)$$

where T_t^{start} and T_t^{end} represents the segment request initial time and completion time, respectively. This metric is used to analyze the latency performance of the QUIC-based delivery mechanism. This metric is directly related to network latency and transmission efficiency. Segment-wise download time is measured during the streaming process and used to analyze the system response under varying network conditions. Lower download times indicate faster content delivery which contributes to smoother playback.

Fig. 4.8.1 presents the segment download time across video segments, which is used to analyze bandwidth estimation and playback stability. Furthermore, segment-level

statistics such as missing ratio per segment, FoV missing ratio per segment and bitrate distribution across segments are also analyzed to provide a detailed understanding of temporal performance variations. These metrics help to identify specific segments where performance degradation occurs, such as during sudden changes in user motion or network conditions.

By combining performance of all parameters, these performance metrics provide a detailed results evaluation of the streaming system. In addition, combined effect of reliability, quality, efficiency and latency indicators, the analysis captures both system-level performance and user-centric QoE that enables a complete assessment of the effectiveness of the viewport-adaptive QUIC-based streaming approach.

4.9 Results and Discussion

Experimental results show the effectiveness of the viewport-adaptive 360° video streaming over QUIC. The evaluation focuses on multiple aspects of system performance which includes systems reliability, viewport quality, bandwidth utilization and low latency behavior. By analyzing these metrics across all segments and tile categories, a detailed understanding of the system’s performance characteristics is obtained.

Fig. 4.9.1 shows that the missing ratio is not uniformly distributed across segments, with some missing-tile events in a few intervals while the remaining segments maintain relatively stable delivery performance after the initial buffering phase.

The temporal evolution of the playback buffer is shown in Fig. 4.9.2, where short peaks in buffer occupancy are observed, however it maintains threshold level of size 2 in many segments which shows its effectiveness in smooth playback . Table 4.5 summarizes the corresponding playback states in terms of stable intervals, recovery periods and possible stall events.

For first few segments, the system experiences relatively higher missing ratios

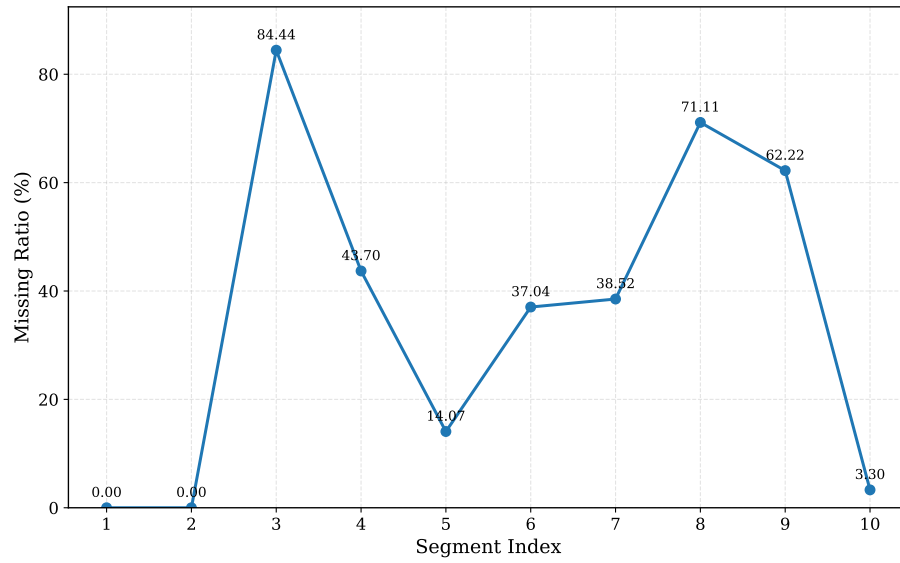


Figure 4.9.1: Segment-wise overall missing ratio across the streaming session.

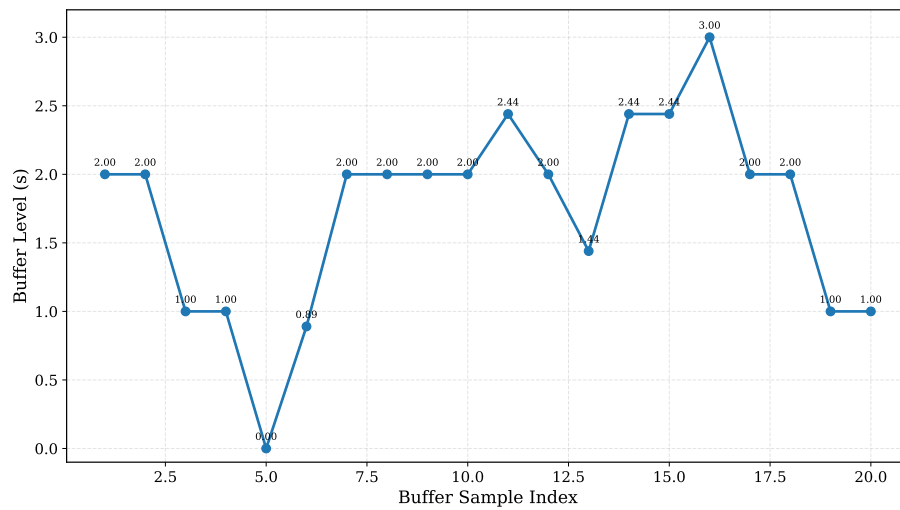


Figure 4.9.2: Temporal evolution of playback buffer level during the streaming session.

due to buffer initialization and the absence of sufficient historical data for accurate prediction. When the streaming session progresses, the LSTM model adapts to user motion patterns that leads to accurate viewport predictions and improved tile selection decisions. This results in a significant reduction in missing tiles which shows improved stability and reliability of the streaming process over time. The FoV missing ratio remains consistently lower than the overall missing ratio which demonstrate the effectiveness of the priority-based scheduling mechanism. Since viewport tiles are transmitted over high-priority QUIC streams, they are delivered more reliably compared to non-FoV tiles. This selective prioritization ensures that the content relevant to the user FoV is available during playback, hence, it significantly improves the perceived QoE even in the presence of variable network conditions.

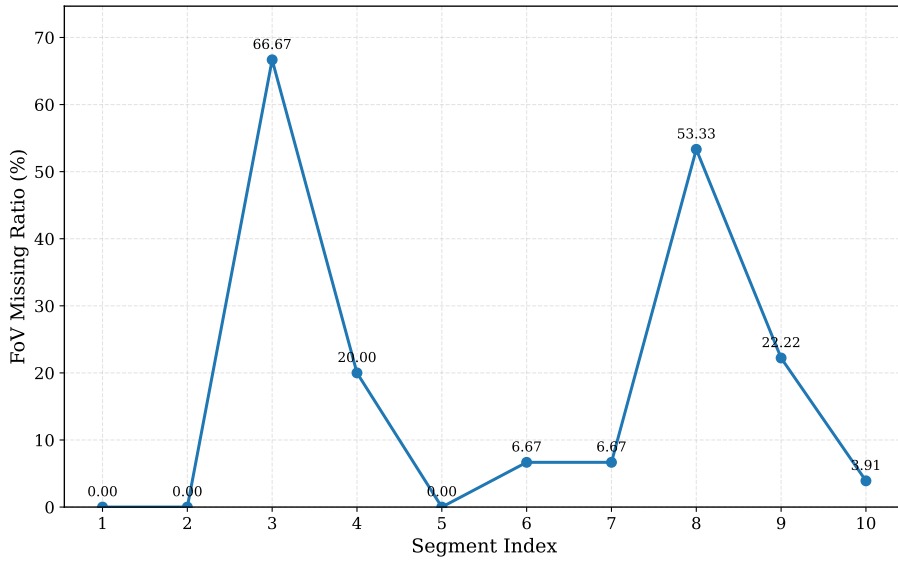


Figure 4.9.3: Segment-wise FoV missing ratio across the streaming session.

Fig. 4.9.3 shows missing FoV tiles across segments containing few peaks which indicate viewport-sensitive tiles were more difficult to deliver within the playback deadline. FoV missing ratio is higher at segment 3 and segment 8. The download time is higher at segment 3 because it is the first segment after the initial buffer stage. At this time, the system requires time to adapt to the sequence of user FoV tiles

across frames. Similarly, segment 8 shows a higher download time because it contains different FoV tiles due to a change in the user’s viewing direction. Due to changing position of user head orientation, some frames have 2 FoV tiles and remaining frames have 4 FoV tiles. In such cases, if more than half frames in a segment have 2 tiles in FoV and some frames have 4 tiles in FoV, for that segment, 2 tiles are considered in FoV for complete segment. It results higher FoV missing tiles in that segment.

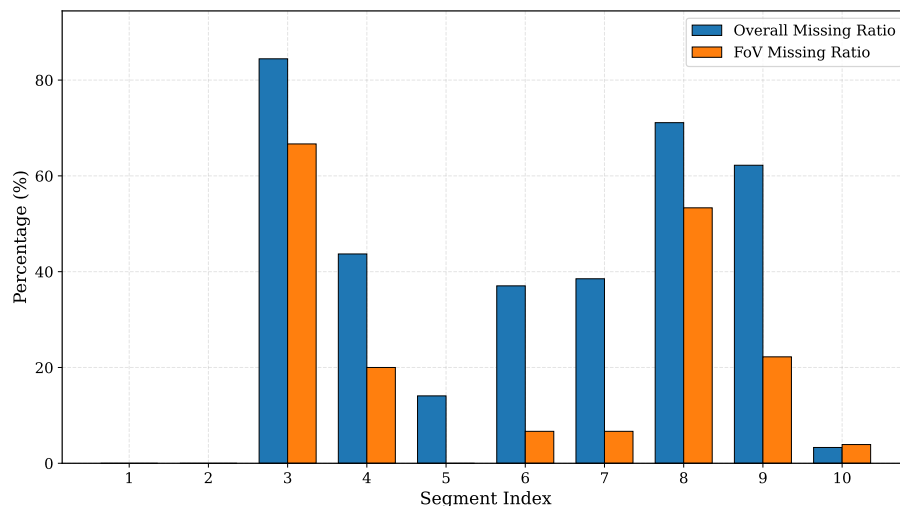


Figure 4.9.4: Comparison of all tiles and FoV tiles missing ratio for all segments.

Fig. 4.9.4 compares all tiles including FoV tiles to summarize missing behavior across all segments that shows viewport-related losses are lower than other tiles losses. The relationship between estimated available bandwidth and requested aggregate bitrate is further summarized by Table 4.6 and Fig. 4.9.5.

Fig. 4.9.5 shows that the requested aggregate bitrate remains below the estimated available bandwidth in most segments, indicating that the adaptation policy remains within the available network budget. Fig. 4.9.6 shows the operating margin of the adaptive scheme which represents whether the requested bitrate remains safely below the estimated bandwidth budget in each segment.

Compared to traditional TCP-based approaches, QUIC’s multiplexed streaming is preferred due to absence of HoL blocking and to parallel transmission of multiple tiles

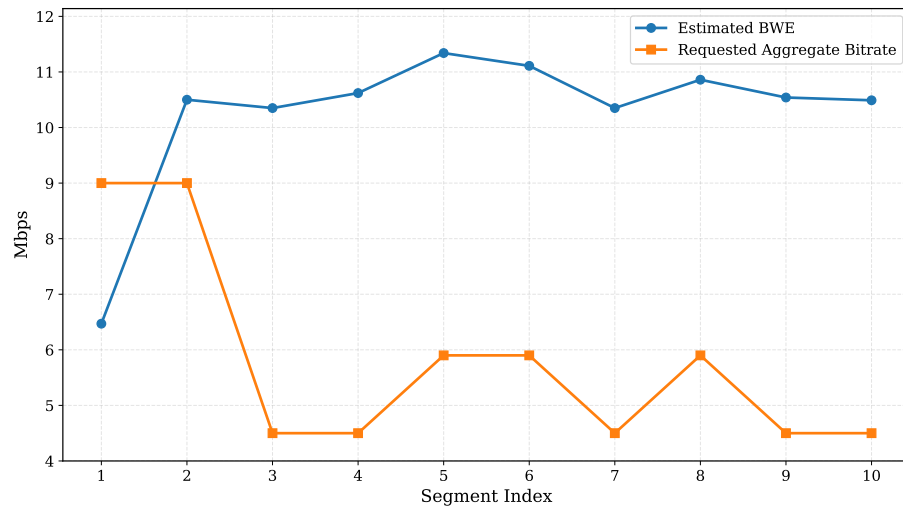


Figure 4.9.5: Comparison between estimated available bandwidth and requested aggregate bitrate for each segment.

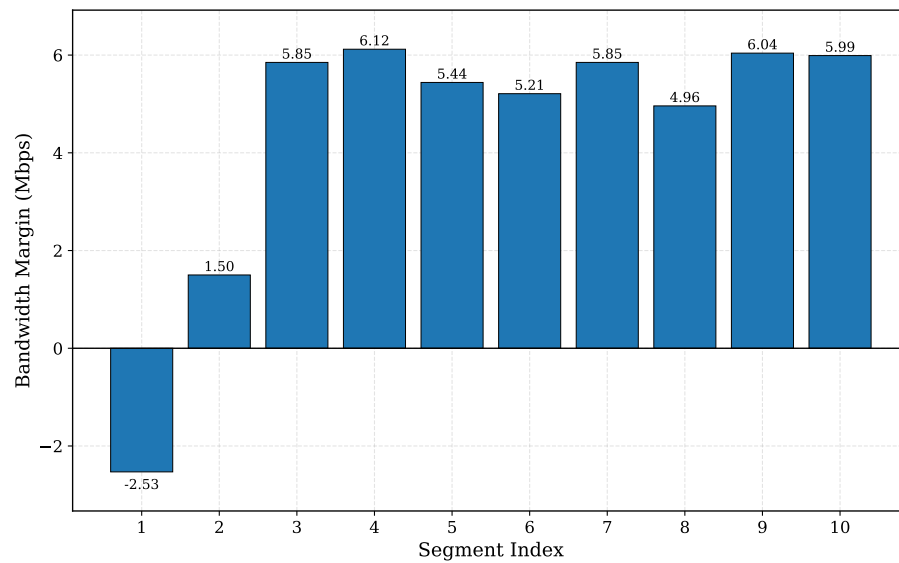


Figure 4.9.6: Bandwidth margin per segment, difference between estimated available bandwidth and requested aggregate bitrate.

without interference. As a result, segment download times remain relatively stable across the streaming session, even when network conditions varies. Buffer level remains relatively stable after initial buffering phase, with small variations observed during playback. This indicates that the adaptive streaming strategy effectively balances data consumption and delivery, avoiding buffer underflow and minimize rebuffering events. The presence of a controlled buffering mechanism confirms that temporary network variations do not significantly impact on the user viewing experience.

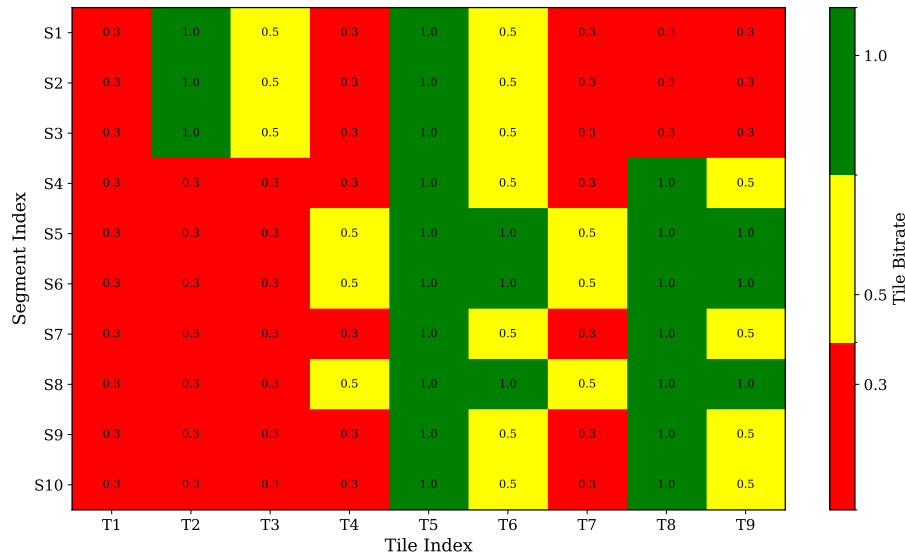


Figure 4.9.7: Heatmap of tile-level bitrate allocation across segments.

Fig. 4.9.7 shows allocation of higher bandwidth to viewport tiles, which are streamed at maximum defined bitrate level of 1.0Mbps shown in green color while yellow color represents neighboring tiles of bitrate 0.5Mbps and red color tiles show non-visible tiles of bitrate 0.3 Mbps (300kbps). Segment-wise analysis of performance metrics shows the impact of prediction accuracy on streaming efficiency. Segments with higher prediction accuracy exhibit lower FoV missing ratios and more efficient bitrate allocation, while segments where viewport prediction accuracy drops result in the transmission of irrelevant tiles or delayed delivery of viewport tiles. This highlights the importance of LSTM model in complete streaming system performance.

Moreover, the results confirm that integration of LSTM-based viewport prediction with QUIC-based adaptive streaming significantly improves the efficiency of 360° video delivery. The combination of predictive tile selection, priority-aware transmission and dynamic bitrate adaptation shows the system performance to deliver high-quality content within the user’s viewport while minimizing bandwidth usage. The use of QUIC further improves the system efficiency by providing low-latency and reliable transport which is essential for real-time immersive applications.

Despite these advantages, certain limitations are persistent. Prediction errors can lead to incorrect tile selection which results in either missing viewport tiles or unnecessary transmission of non-visible tiles. This not only affects QoE; however, it also reduces bandwidth efficiency. In addition, the initial buffering phase causes a startup delay, which may impact user experience, especially in low-latency applications.

The transport evaluation is limited to 10 segments under controlled throughput conditions. Dynamic delay and packet loss are not considered. The rule-based bitrate controller is also not compared with MPC or DRL-based ABR methods. These limitations are considered in future work. Future work will extend the evaluation to longer streaming sessions and multiple users, and will consider time-varying mobile-network profiles with explicit bandwidth variation, propagation delay and packet loss. The current rule-based bitrate controller can also be benchmarked against other ABR algorithms, including MPC and DRL-based controllers, to quantify the benefit of more advanced adaptation strategies.

4.10 Summary of chapter

This chapter presents a viewport-adaptive 360° video streaming framework over QUIC that addresses the challenges of bandwidth constraints and low-latency immersive media delivery. The system combines several key components that include tile-based video multi-bitrate encoding, LSTM-based viewport prediction, adaptive bitrate

selection and QUIC-based transport into a combined efficient streaming pipeline. The results show that combining ML-based prediction with adaptive streaming improves system performance. With the viewport prediction results, the system performance improves to deliver FoV tiles at higher quality while bitrate allocated to non-visible tiles is reduced. The adaptive tiles transmission strategy results in efficient use of network resources.

Furthermore, the QUIC protocol improves system performance by using low-latency communication and supporting more reliable content delivery. Combining priority-based tiles with QUIC streams confirms that FoV tiles are delivered with high bitrate and low latency that is important to maintain smooth playback in interactive environments. Despite these promising results, certain limitations are still persistent. The performance of the system is dependent on the accuracy of viewport prediction model and prediction errors may lead to poor selection of tiles which reduce efficiency. In addition, initial buffer introduces early latency which may affect user experience in real-time applications. To address the limitations, selection of optimal model for viewport prediction is important and also combination of various models such as transformers, attention based mechanism and latest reliable neural network models can be used for better viewport prediction results and adaptive strategies.

Future work will focus on developing a more robust and intelligent system. Integration of DRL to optimize bitrate decision and priorities to FoV and non-FoV tiles shows a promising direction. Moreover, combining user pitch and yaw angles with saliency maps and semantic information can improve viewport prediction accuracy and enhance streaming efficiency. In addition, the framework represents that combining ML models, adaptive multi-bitrate streaming decisions and use of next-generation transport protocols lead to efficient and scalable 360° video delivery. Methodology along with results analysis presented in this chapter contribute to the advancement of immersive multimedia systems and motivates to explore future research directions in viewport-adaptive streaming over emerging network architectures.

Table 4.3: Experimental setup for viewport-adaptive QUIC streaming framework.

Category	Parameter	Description
Video and Tiling	Video format	360° tiled video for viewport-adaptive QUIC based delivery.
	Video resolution	$[7680 \times 3840]$.
	Tile structure	3×3 grid with 9 tiles per segment.
	Tile indexing	Tiles 1–9; QUIC tracks 2–10 mapped to these tiles.
Bitrate Setup	Bitrate ladder	Nominal simulator values: $\{0.3, 0.5, 1.0\}$ Mbps per tile.
	Bitrate assignment	FoV / Neighbor / Non-visible tiles use 1.0 / 0.5 / 0.3 Mbps.
	Minimum aggregate bitrate	2.7 Mbps when all 9 tiles are transmitted at 0.3 Mbps.
	Adaptation goal	Maximize viewport quality while keeping requested bitrate within estimated bandwidth.
Streaming and Transport	Protocol and implementation	QUIC-based client-server adaptive streaming using <code>aioquic</code> .
	Stream classes	Two priority classes: high priority and low priority.
	Stream usage	High priority for FoV tiles; low priority for neighbor and non-visible tiles.
	Number of segments	10 segments, each of 1 second.
Viewport-adaptive Allocation	Tile classes	FoV, neighboring and non-visible tiles.
	Tile counts	2 FoV tiles, 2 neighboring tiles, remaining tiles non-visible.
	Prioritization	FoV > Neighbor > Non-visible.
	Update granularity	Segment-level adaptive tile selection.
Bandwidth and Playback	Bandwidth estimation	Computed from downloaded segment size divided by segment download time.
	BWE expression	$\text{BWE} = \frac{\text{segment size (bits)}}{\text{download time (s)}}.$
	Key variables	Buffer level, download time, requested bitrate, BWE, bandwidth margin.
	Evaluation metrics	Missing ratio, FoV missing ratio, download time, BWE, bandwidth margin, buffer evolution.
Viewport Prediction Model	Prediction input / output	User HMD data (pitch, yaw) to predict viewport tiles.
	Framework	PyTorch-based implementation.
	Training settings	[LSTM model, BCE loss, Adam optimizer, $\text{lr} = 1\text{e-}4$, 150 epochs].

Table 4.4: Segment-wise missing ratio and FoV reliability analysis

Segment	Total MR (%)	FoV MR (%)	Delivery (%)	FoV Reliability (%)	Observation
1	0.00	0.00	100.00	100.00	Initial buffering; full delivery
2	0.00	0.00	100.00	100.00	Stable condition
3	88.15	73.33	11.85	26.67	Severe loss under aggressive bi-rate
4	53.33	33.33	46.67	66.67	Partial recovery
5	16.30	0.00	83.70	100.00	FoV fully protected
6	40.00	6.67	60.00	93.33	Moderate loss; FoV mostly stable
7	38.52	13.33	61.48	86.67	Stabilizing network
8	77.04	60.00	22.96	40.00	High loss; FoV degradation
9	67.41	29.63	32.59	70.37	Partial FoV recovery
10	3.43	3.91	96.57	96.09	Final recovery

Table 4.5: Playback buffer evolution during streaming

Segment	Buffer (s)	Played (s)	Downloaded (s)	State	Observation
1	2.00	0.00	2.00	Good	Initial buffering phase
2	1.00	1.00	2.00	Moderate	Buffer decreases during playback
3	1.00	2.00	3.00	Stable	Continuous playback without interruption
4	1.00	3.00	4.00	Stable	Low but stable buffer
5	2.00	3.00	5.00	Good	Buffer recovery due to fast download
6	2.00	4.00	6.00	Good	Stable playback with sufficient buffer
7	2.00	5.00	7.00	Good	Consistent streaming
8	3.00	5.00	8.00	Excellent	Buffer increases (high BWE)
9	3.00	6.00	9.00	Excellent	Strong buffer stability
10	3.00	7.00	10.00	Excellent	Final stable playback

Table 4.6: Segment-wise bandwidth estimation and bitrate allocation behavior

Segment	Estimated BWE (Mbps)	Required Bitrate (Mbps)	Safety Margin (Mbps)	Tile Selection Mode	Explanation
1	6.19	9.00	-2.81	All tiles (multi-bitrate)	BWE $\uparrow$ demand; aggressive start
2	10.55	9.00	+1.55	All tiles (multi-bitrate)	Stable; sufficient bandwidth
3	10.51	4.50	+6.01	All tiles (reduced bitrate)	Bitrate reduced due to buffer drop
4	10.66	4.50	+6.16	All tiles (reduced bitrate)	Stable conservative transmission
5	12.22	5.90	+6.32	All tiles (adaptive FoV focus)	Increased FoV quality (good BWE)
6	10.92	5.90	+5.02	All tiles (adaptive FoV focus)	Stable adaptive allocation
7	10.74	4.50	+6.24	All tiles (reduced bitrate)	Conservative bitrate control
8	10.34	5.90	+4.44	All tiles (adaptive FoV focus)	Balanced adaptive strategy
9	10.53	4.50	+6.03	All tiles (reduced bitrate)	Stable conservative mode
10	10.89	4.50	+6.39	All tiles (reduced bitrate)	Final stable transmission

Chapter 5

Conclusions

This thesis is structured around four main components. First, it presents the fundamental concepts of 360° video streaming. Second, an overview of recent studies and their trade-offs in viewport-adaptive streaming systems. Third, it presents a complete viewport prediction pipeline which covers pre-processing of user HMD orientation data (pitch, yaw angles) to predict user FoV. This pipeline is also compared with a state-of-the-art approach. The last part contains a complete tile-based viewport adaptive 360° video streaming pipeline using QUIC transport protocol. The thesis addresses the key challenge of delivering immersive 360° video at high quality within the user's FoV and lower quality at the remaining regions under bandwidth constraints and variable network conditions. The growing deployment of HMD devices has increased the need to meet the QoE requirements of users watching immersive content. The research is organized into two experimental parts: viewport prediction using ML models and tile-based viewport-adaptive streaming using QUIC transport protocol.

Chapters 3 and 4 consist of experiments on viewport prediction and immersive multimedia delivery using tile-based streaming. Chapter 3 contains a comparison of various ML models used for viewport prediction and their performance using viewport prediction accuracy. ML models used in this study are LSTM, RNN, GRU, CNN, LR

and MLP. Identical hyperparameters were selected for all these models to analyze their performance on predicting user FoV from next-frame up to 5 seconds prediction horizon. Dataset used for this study was taken from E3PO website whose official link to dataset is mentioned in chapter 3. These findings show that for next-frame viewport prediction accuracy, CNN model outperforms to all other models on selected motion trace (trace 55, 48 users) while comparing at 5 second horizon, GRU and LSTM models performed well as compared to other models used.

The robustness of next-frame model comparison was investigated through cross-video validation experiment using 27 remaining traces. Across these traces, CNN achieved the highest mean exact tile-set accuracy of $81.64 \pm 3.62\%$ and a mean tile rank of 1.00 ± 0.00 , ranking first on all 27 external videos. The analysis across 1314 user-video trajectories showed the same overall performance trend. These results show that the superior short-term performance of CNN is not limited to a single evaluated motion trace.

In addition, recent state-of-the-art MFTR model is used as a benchmark for comparison with LSTM-based framework. Across all prediction horizons, the LSTM model achieves better AP accuracy that improves from 92.47% against MFTR's 87.09% at next-frame prediction, to 44.20% against MFTR's 27.11% at the 3-second horizon with only gaze data as input, while keeping competitive results of AO values. The results show that a properly tuned LSTM model can outperform more complex transformer architectures in terms of viewport prediction accuracy.

Chapter 4 focuses on a complete framework, from viewport prediction to adaptive streaming and delivery of media over QUIC. This system contains ERP-based video frames divided into 9 tiles, viewport prediction using LSTM model, ABR decision and QUIC-based transport in a combined streaming pipeline. Priority is given to FoV tiles at highest bitrate (1.0 Mbps) while medium bitrate allocation to neighboring tiles (0.5 Mbps) and rest of the tiles are transmitted at lowest bitrate (0.3 Mbps), the system gets efficient bandwidth utilization while keeping high perceptual quality within the

user's FoV. The results observed from this experimental setup confirmed that the requested bitrate (bitrate of sum of all 9 tiles) behaves consistently and remained within the available network bandwidth across all streaming segments, which indicates that adaptive behavior performed well with network constraints. Parallel streams of QUIC protocol eliminates head-of-line blocking and enable stable segment download time and smooth buffer values with variable network conditions. Buffer behavior is almost stable after initial buffering segments with slightly high peaks observed during playback, showing given systems validation and real time suitability in immersive multimedia applications.

Besides promising results mentioned in above chapters, this thesis acknowledges some limitations which are important to work for future enhancement of this research field. viewport prediction which is higher for many tested models for next-frame prediction; however, it degrades when prediction horizon is increased up to 3 or 5 seconds. LSTM model at 3 seconds prediction horizon (gaze input) achieves AP accuracy only 44.20% which reflects unpredictable user head movement for long-term prediction. Viewport prediction errors can lead to the transmission of irrelevant tiles, resulting in high missing ratio that reduces bandwidth efficiency and degrades QoE.

Initial buffer in immersive streaming systems causes early delay which lead to create problems for low-latency. The transport evaluation is limited to a single-user scenario and ten video segments. The experiments consider throughput and buffer-driven adaptation under controlled network conditions and do not explicitly emulate time-varying propagation delay or packet loss. In addition, the rule-based bitrate controller used in this work is not benchmarked against formal ABR approaches such as MPC or DRL. These limitations reduce the generalizability of the transport results.

Based on these findings and limitations presented in this work, various future directions can be identified. First, consideration of advanced neural network models such as liquid neural network, hybrid models e.g. attention based mechanism and

transformer model for better viewport prediction accuracy. Integration of semantic content with saliency maps along with gaze and velocity coordinates may lead to improve viewport prediction results for long-term prediction horizon. The present study evaluates viewport prediction horizons up to 5 seconds; extending the prediction analysis to horizons beyond 5 seconds is left for future work.

Future work will extend the transport evaluation to longer streaming sessions and multi-user scenarios. The framework should also be tested under realistic mobile-network conditions with varying network characteristics. In addition, the current rule-based adaptation strategy should be compared with formal ABR approaches such as MPC and DRL. These approaches could improve bitrate adaptation and viewport-aware streaming under changing network conditions.

Bibliography

- [1] H. Yuan, S. Zhao, J. Hou, X. Wei, and S. Kwong, “Spatial and temporal consistency-aware dynamic adaptive streaming for 360-degree videos,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 1, pp. 177–193, 2020.
- [2] M. Zink, R. Sitaraman, and K. Nahrstedt, “Scalable 360° Video Stream Delivery: Challenges, Solutions, and Opportunities: This article introduces a novel networking scenario, namely 360° video streaming, discussing its challenges, existing approaches, and the research opportunities which it enabled.,” *IEEE Proceedings*, vol. 107, pp. 639–650, Jan. 2019.
- [3] F. Islam, Z. Bright, M.-H. Chen, L. Zhan, and C. Shi, “Evaluating the effect of immersive virtual reality technologies on cognitive load, situation awareness, and engagement using eye movements,” *Applied Ergonomics*, vol. 134, p. 104720, 2026.
- [4] D.-H. Lee, K.-M. Jang, and H. K. Lim, “Cybersickness-induced eeg responses in curved monitor and head-mounted display,” *Scientific Reports*, vol. 16, no. 1, p. 2467, 2026.
- [5] V. H. Nguyen, D. T. Bui, T. L. Tran, C. T. Truong, and T. H. Truong, “Scalable and resilient 360-degree-video adaptive streaming over http/2 against sudden network drops,” *Computer Communications*, vol. 216, pp. 1–15, 2024.

- [6] I. Agtzidis, M. Startsev, and M. Dorr, “360-degree video gaze behaviour: A ground-truth data set and a classification algorithm for eye movements,” in *Proceedings of the 27th ACM International Conference on Multimedia*, MM ’19, (New York, NY, USA), p. 1007–1015, Association for Computing Machinery, 2019.
- [7] X. Feng, Z. Bao, and S. Wei, “Liveobj: Object semantics-based viewport prediction for live mobile virtual reality streaming,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 5, pp. 2736–2745, 2021.
- [8] R. Monica and J. Aleotti, “Point cloud annotation in virtual reality based on selection of control points by gaze or hand pointing,” *Virtual Reality*, vol. 30, no. 2, p. 62, 2026.
- [9] B. A. Zulkiewicz, V. Boudewyns, C. Gupta, A. Kirschenbaum, and M. A. Lewis, “Using 360-degree video as a research stimulus in digital health studies: Lessons learned,” *JMIR Serious Games*, vol. 8, no. 1, 2020.
- [10] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, “Viewport-adaptive navigable 360-degree video delivery,” in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–7, 2017.
- [11] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, “Optimizing 360 video delivery over cellular networks,” in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*, ATC ’16, (New York, NY, USA), p. 1–6, Association for Computing Machinery, 2016.
- [12] H. Nguyen, T. N. Dao, N. S. Pham, T. L. Dang, T. D. Nguyen, and T. H. Truong, “An accurate viewport estimation method for 360 video streaming using deep learning,” *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, vol. 9, p. e2, Sep. 2022.

- [13] C.-L. Fan, J. Lee, W.-C. Lo, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, “Fixation prediction for 360° video streaming in head-mounted virtual reality,” *Proceedings of the 27th Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2017.
- [14] A. Hamara, B. Kilpatrick, A. Baratta, B. Kofink, and A. C. Freeman, “Towards low-latency adaptive streaming for event-based vision,” in *2025 International Conference on Multimedia Computing, Networking and Applications (MCNA)*, pp. 17–24, 2025.
- [15] M. Graf, C. Timmerer, and C. Mueller, “Towards bandwidth efficient adaptive streaming of omnidirectional video over http: Design, implementation, and evaluation,” in *Proceedings of the 8th ACM on Multimedia Systems Conference, MMSys’17*, (New York, NY, USA), p. 261–271, Association for Computing Machinery, 2017.
- [16] X. Chen, B. Cao, I. Ahmad, and X. Xue, “Lightweight neural network-based viewport prediction for live vr streaming in wireless video sensor network,” *Mob. Inf. Syst.*, vol. 2021, Jan. 2021.
- [17] J. Vielhaben, H. Camalan, W. Samek, and M. Wenzel, “Viewport forecasting in 360° virtual reality videos with machine learning,” in *2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pp. 74–747, 2019.
- [18] J. Adhuran and M. G. Martini, “Efficient viewport prediction and tiling schemes for 360 degree video streaming,” in *Proceedings of the 15th ACM Multimedia Systems Conference, MMSys ’24*, (New York, NY, USA), p. 374–380, Association for Computing Machinery, 2024.

- [19] F.-Y. Chao, C. Ozcinar, and A. Smolic, “Transformer-based long-term viewport prediction in 360° video: Scanpath is all you need,” in *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6, 2021.
- [20] X. Li, S. Wang, C. Zhu, L. Song, R. Xie, and W. Zhang, “Viewport prediction for panoramic video with multi-cnn,” in *2019 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pp. 1–6, 2019.
- [21] J. Shao, Z. Li, and J. Song, “A hybrid approach for accurate viewport prediction in 360° video streaming,” in *2025 IEEE 6th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*, pp. 1303–1306, 2025.
- [22] X. Feng, Z. Bao, and S. Wei, “Exploring cnn-based viewport prediction for live virtual reality streaming,” in *2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pp. 183–1833, 2019.
- [23] C.-L. Fan, J. Lee, W.-C. Lo, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, “Fixation prediction for 360° video streaming in head-mounted virtual reality,” in *Proceedings of the 27th Workshop on Network and Operating Systems Support for Digital Audio and Video, NOSSDAV’17*, (New York, NY, USA), p. 67–72, Association for Computing Machinery, 2017.
- [24] Y. Xu, Y. Dong, J. Wu, Z. Sun, Z. Shi, J. Yu, and S. Gao, “Gaze prediction in dynamic 360° immersive videos,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5333–5342, 2018.
- [25] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar, J. Bailey, J. Dorfman, J. Roskind, J. Kulik, P. Westin, R. Tenneti, R. Shade, R. Hamilton, V. Vasiliev, W.-T. Chang, and Z. Shi, “The quic transport protocol: Design and internet-scale deployment,” in *Proceedings of the Conference of the ACM Special Interest Group on Data*

- Communication*, SIGCOMM '17, (New York, NY, USA), p. 183–196, Association for Computing Machinery, 2017.
- [26] J. Iyengar and M. Thomson, “Quic: A udp-based multiplexed and secure transport,” *RFC*, vol. 9000, pp. 1–151, 2020.
- [27] X. Corbillon, F. De Simone, and G. Simon, “360-degree video head movement dataset,” in *IEEE International Conference on Multimedia Systems (MMSys)*, 2017.
- [28] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, “Viewport-adaptive navigable 360-degree video delivery,” in *Proceedings of the ACM Multimedia Systems Conference (MMSys)*, 2017.
- [29] H.-Y. Chen and C.-S. Lin, “Tiled streaming for layered 3d virtual reality videos with viewport prediction,” *Multimedia Tools and Applications*, vol. 81, no. 4, 2022.
- [30] A. Yaqoob, G.-M. Muntean, A. Al-Hamoodi, M. Gajewski, B. T. Joukov, and A. Moldovan, “A survey on adaptive 360° video streaming: Solutions, challenges and opportunities,” *IEEE Communications Surveys & Tutorials*, 2020.
- [31] M. Mahmoud, I. Valiandi, A. S. Panayides, S. Rizou, P. I. Lazaridis, G. K. Karagiannidis, N. V. Kantartzis, and Z. D. Zaharis, “Versatile video coding performance evaluation for tiled 360 deg videos,” in *European Wireless 2023; 28th European Wireless Conference*, pp. 191–196, 2023.
- [32] Y. Ye, J. M. Boyce, and P. Hanhart, “Omnidirectional 360° video coding technology in responses to the joint call for proposals on video compression with capability beyond hev1,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 5, pp. 1241–1252, 2020.

- [33] F. Chiariotti, “A survey on 360-degree video: Coding, quality of experience and streaming,” *Comput. Commun.*, vol. 177, p. 133–155, Sept. 2021.
- [34] R. Shafi, W. Shuai, and M. U. Younus, “360-degree video streaming: A survey of the state of the art,” *Symmetry*, vol. 12, no. 9, 2020.
- [35] F. Zhang, J. Zhao, Y. Zhang, and S. Zollmann, “A survey on 360° images and videos in mixed reality: Algorithms and applications,” *J. Comput. Sci. Technol.*, vol. 38, p. 473–491, May 2023.
- [36] S.-H. Lee, S.-T. Kim, E. Yip, B.-D. Choi, J. Song, and S.-J. Ko, “Omnidirectional video coding using latitude adaptive down-sampling and pixel rearrangement,” *Electronics Letters*, vol. 53, no. 10, pp. 655–657, 2017.
- [37] L. Xie, Z. Xu, Y. Ban, X. Zhang, and Z. Guo, “360probdash: Improving qoe of 360 video streaming using tile-based http adaptive streaming,” in *Proceedings of the 25th ACM International Conference on Multimedia*, MM ’17, (New York, NY, USA), p. 315–323, Association for Computing Machinery, 2017.
- [38] E. S. Wong, N. H. A. Wahab, F. Saeed, and N. Alharbi, “360-degree video bandwidth reduction: Technique and approaches comprehensive review,” *Applied Sciences*, vol. 12, no. 15, 2022.
- [39] A. Yaqoob and G.-M. Muntean, “Advanced predictive tile selection using dynamic tiling for prioritized 360° video vr streaming,” *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 20, Aug. 2023.
- [40] Z. Pang, “Vatp360: Viewport adaptive 360-degree video streaming based on tile priority,” 2023.
- [41] P. K. Yadav and W. T. Ooi, “Tile rate allocation for 360-degree tiled adaptive video streaming,” in *Proceedings of the 28th ACM International Conference*

- on Multimedia*, MM '20, (New York, NY, USA), p. 3724–3733, Association for Computing Machinery, 2020.
- [42] C. Kattadige and K. Thilakarathna, “Vad360: Viewport aware dynamic 360-degree video frame tiling,” 2021.
- [43] S. K. Murthy Naik, M. Wang, and D. Krishnamurthy, “Tyle: Tile-based dynamic quality enhancement for 360-degree video streaming,” in *2024 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, pp. 1–8, 2024.
- [44] H. Nguyen, T. N. Dao, N. S. Pham, T. L. Dang, T. D. Nguyen, and T. H. Truong, “An accurate viewport estimation method for 360 video streaming using deep learning,” *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, vol. 9, 9 2022.
- [45] X. Xu, X. Tan, S. Wang, Z. Liu, and Q. Zheng, “Multi-features fusion based viewport prediction with gnn for 360-degree video streaming,” in *2023 IEEE International Conference on Metaverse Computing, Networking and Applications (MetaCom)*, pp. 57–64, 2023.
- [46] M. Mahmoud, S. R. R. Stamatia, A. S. Panayides, P. I. Lazaridis, N. V. Kantartzis, G. K. Karagiannidis, and Z. D. Zaharis, “A comparative analysis of viewing prediction techniques for 360° video streaming applications,” in *2024 Panhellenic Conference on Electronics Telecommunications (PACET)*, pp. 1–4, 2024.
- [47] S. Van Damme, M. T. Vega, and F. De Turck, “Machine learning based content-agnostic viewport prediction for 360-degree video,” *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 18, Feb. 2022.

- [48] G. Manfredi, V. A. Racanelli, L. De Cicco, and S. Mascolo, “Lstm-based viewport prediction for immersive video systems,” in *2023 21st Mediterranean Communication and Computer Networking Conference (MedComNet)*, pp. 49–52, 2023.
- [49] B. Gao, D. Sheng, L. Zhang, Q. Qi, B. He, Z. Zhuang, and J. Wang, “Star-vp: Improving long-term viewport prediction in 360° videos via space-aligned and time-varying fusion,” in *Proceedings of the 32nd ACM International Conference on Multimedia*, MM ’24, (New York, NY, USA), p. 5556–5565, Association for Computing Machinery, 2024.
- [50] D. Nguyen, “An evaluation of viewport estimation methods in 360-degree video streaming,” in *2022 7th International Conference on Business and Industrial Research (ICBIR)*, pp. 161–166, 2022.
- [51] J. Chen, X. Luo, M. Hu, D. Wu, and Y. Zhou, “Sparkle: User-aware viewport prediction in 360-degree video streaming,” *IEEE Transactions on Multimedia*, vol. 23, pp. 3853–3866, 2021.
- [52] H. Zhou, F. Zhao, and C. Li, “Multi-scale historical trajectory decomposition for viewport prediction in 360-degree videos,” *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 21, Nov. 2025.
- [53] A. Bentaleb, B. Taani, A. C. Begen, C. Timmerer, and R. Zimmermann, “A survey on bitrate adaptation schemes for streaming media over http,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 562–585, 2019.
- [54] C. Timmerer, H. Amirpour, F. Tashtarian, S. Afzal, A. Rizk, M. Zink, and H. Hellwagner, “Http adaptive streaming: A review on current advances and future challenges,” *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 21, July 2025.

- [55] H. K. Ravuri, M. T. Vega, J. D. Van Hooft, T. Wauters, and F. De Turck, “Adaptive partially reliable delivery of immersive media over quic-http/3,” *IEEE Access*, vol. 11, pp. 38094–38111, 2023.
- [56] S. Huang, Z. Wang, W. Zhang, H. Du, and Q. Zheng, “Adaptive video streaming using dynamic server push over http/2,” in *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 673–678, 2021.
- [57] S. Petrangeli, V. Swaminathan, M. Hosseini, and F. De Turck, “An http/2-based adaptive streaming framework for 360° virtual reality videos,” in *Proceedings of the 25th ACM International Conference on Multimedia*, MM ’17, (New York, NY, USA), p. 306–314, Association for Computing Machinery, 2017.
- [58] M. Ben Yahia, Y. Le Louedec, G. Simon, and L. Nuaymi, “Http/2-based streaming solutions for tiled omnidirectional videos,” in *2018 IEEE International Symposium on Multimedia (ISM)*, pp. 89–96, 2018.
- [59] S.-C. Yen, C.-L. Fan, and C.-H. Hsu, “Streaming 360° videos to head-mounted virtual reality using dash over quic transport protocol,” in *Proceedings of the 24th ACM Workshop on Packet Video*, PV ’19, (New York, NY, USA), p. 7–12, Association for Computing Machinery, 2019.
- [60] F. Rosa, S. Ferlin, A. Brunstrom, and B. Kimura, “End-to-end 360° video streaming over http/3: Architecture and implementation,” in *Proceedings of the 2025 Applied Networking Research Workshop*, ANRW ’25, (New York, NY, USA), p. 9–16, Association for Computing Machinery, 2025.
- [61] Y. Wei, Q. Ye, K. Qu, W. Zhuang, and X. Shen, “Customized transmission protocol for tile-based 360° vr video streaming over core network slices,” *IEEE Transactions on Networking*, vol. 33, no. 1, pp. 340–354, 2025.

- [62] W.-C. Lo, C.-L. Fan, J. Lee, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, “360° video viewing dataset in head-mounted virtual reality,” in *Proceedings of the 8th ACM on Multimedia Systems Conference*, MMSys’17, (New York, NY, USA), p. 211–216, Association for Computing Machinery, 2017.
- [63] C. Wu, Z. Tan, Z. Wang, and S. Yang, “A dataset for exploring user behaviors in vr spherical video streaming,” in *Proceedings of the 8th ACM on Multimedia Systems Conference*, MMSys’17, (New York, NY, USA), p. 193–198, Association for Computing Machinery, 2017.
- [64] Y. Guan, C. Zheng, X. Zhang, Z. Guo, and J. Jiang, “Pano: optimizing 360° video streaming with a better understanding of quality perception,” in *Proceedings of the ACM Special Interest Group on Data Communication*, SIGCOMM ’19, (New York, NY, USA), p. 394–407, Association for Computing Machinery, 2019.
- [65] A. T. Nasrabadi, A. Samiei, A. Mahzari, R. P. McMahan, R. Prakash, M. C. Q. Farias, and M. M. Carvalho, “A taxonomy and dataset for 360° videos,” in *Proceedings of the 10th ACM Multimedia Systems Conference*, MMSys ’19, (New York, NY, USA), p. 273–278, Association for Computing Machinery, 2019.
- [66] L. Zhang, P. Chen, C. Zhang, C. Pan, T. Long, W. Xu, L. Cui, and J. Liu, “Optimizing mobile-friendly viewport prediction for live 360-degree video streaming,” *IEEE Transactions on Mobile Computing*, vol. 24, no. 10, pp. 10441–10455, 2025.
- [67] M. U. Younus, R. Shafi, A. Rafiq, M. R. Anjum, S. Afridi, A. A. Jamali, and Z. A. Arain, “Encoder-decoder based lstm model to advance user qoe in 360-degree video,” *Computers, Materials and Continua*, vol. 71, no. 2, pp. 2617–2631, 2021.
- [68] H. He, F. Zhao, Z. Su, H. Zhou, Y. Zhang, and J. Xiong, “360-degree video long-term viewport prediction model based on lstm and self-attention mechanism,” in *Third Asia Conference on Computer Vision, Image Processing, and Pattern Recognition (CVIPPR 2025)*, vol. 13697, pp. 75–84, SPIE, 2025.

- [69] M. Jamali, S. Coulombe, A. Vakili, and C. Vazquez, “Lstm-based viewpoint prediction for multi-quality tiled video coding in virtual reality streaming,” in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2020.
- [70] Z. Zhang, Y. Chen, W. Zhang, C. Yan, Q. Zheng, Q. Wang, and W. Chen, “Tile classification based viewport prediction with multi-modal fusion transformer,” in *Proceedings of the 31st ACM International Conference on Multimedia*, MM ’23, p. 3560–3568, ACM, Oct. 2023.
- [71] M. Xu, Y. Song, J. Wang, M. Qiao, L. Huo, and Z. Wang, “Predicting head movement in panoramic video: A deep reinforcement learning approach,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, p. 2693–2708, Nov. 2019.
- [72] J. S. Sidhu and A. Bentaleb, “Video streaming over quic: A comprehensive study,” *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 22, Mar. 2026.