Artificial Intelligence–Driven Rate Adaptive and Perceptual-Aware Rich Media Delivery in 5G and Beyond Networks

Fazal E Subhan, MSc

A Dissertation submitted in fulfillment of the

requirements for the award of

Doctor of Philosophy (PhD)

to the



School of Electronic Engineering

Dublin City University, Ireland

Supervisor:

Professor Gabriel-Miro Muntean, Dublin City University, Ireland

Co-Supervisor:

Professor Cristina Muntean, National College of Ireland, Ireland

August 2026

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, and that I have exercised reasonable care to ensure that the work is original and have conformed to the regulations on the use and declaration of Generative AI, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: Fazal E Subhan

ID No.: 23266615

Date: 18 August 2026

Reporting of Generative AI Tool Use

Tools

OpenAI ChatGPT GPT-5.1 and Gemini 2.5 Flash have been used between 2025 and 2026 during the preparation and revision of this thesis.

Outputs

The outputs generated and selectively used in this thesis preparation process included:

- draft revisions of thesis paragraphs to improve clarity, structure, and academic style;
- suggested wording for technical explanations, captions, and section transitions;
- LaTeX-code refinement for formatting, table refinement, and figure placement;

Prompts

Representative prompts included:

- Revise this paragraph for clarity and academic tone in Overleaf format.
- Rewrite this explanation to get to the desired technical depth.
- Suggest a technical caption for the figures.
- Improve the logical flow of this section while preserving the technical meaning.

Where needed, outputs were further refined through follow-up prompts requesting concision, stronger technical emphasis, or better alignment with reviewer concerns.

Iterations

The use of the tool was iterative. Initial outputs were treated as draft support only and were repeatedly refined through multiple follow-up prompts. All generated text, captions, and code suggestions were manually reviewed, edited, and adapted by the author before any inclusion in the thesis.

Critical Reflection

Generative AI was used in a limited support role for editing, restructuring, drafting assistance, and code refinement. It was not used to generate the underlying research contributions, experimental results, or scientific conclusions of the thesis. The author remained fully responsible for the conception of the research, implementation decisions, interpretation of the findings, verification of technical accuracy, and final form of all thesis content.

"Man is primarily governed by passion and instinct"

— Allama Muhammad Iqbal

"With faith, discipline, and selfless devotion to duty,
there is nothing worthwhile that you cannot achieve"

— Muhammad Ali Jinnah (Quaid-e-Azam)

Acknowledgements

First of all, I wish to thank my supervisor, Professor Gabriel-Miro Muntean, and my co-supervisor, Professor Cristina Muntean, for their constant support and guidance throughout my PhD. Gabriel's constant encouragement, special support, and experience have helped me keep my focus on what is important. Moreover, this PhD journey would not have been possible without Gabi's special support in every sense, his technical guidance, consistent encouragement both professionally and personally, and genuine care for my progress and well-being. His belief in my work helped me stay focused and resilient throughout the process.

I would like to extend special thanks to Dr. Abid Yaqoob, whose support and mentorship have been crucial to this thesis and to my research as a whole.

I dedicate this work to my late father, whose unwavering support and belief in my abilities laid the foundation for every step of my academic and professional journey, and to my whole family, whose constant encouragement and love have been my greatest strength throughout this endeavor.

I am also grateful to my fellow labmates for their encouragement, insightful discussions, and camaraderie throughout this research.

Dublin, August 2026

Fazal E Subhan

List of Publications

Journal Papers

1. F. E. Subhan, A. Yaqoob, C. H. Muntean and G.-M. Muntean, "A Survey on Artificial Intelligence Techniques for Improved Rich Media Content Delivery in a 5G and Beyond Network Slicing Context," in IEEE Communications Surveys & Tutorials, vol. 27, no. 2, pp. 1427-1487, April 2025, doi: 10.1109/COMST.2024.3442149.

2. F. E. Subhan, A. Yaqoob, C. H. Muntean and G.-M. Muntean, "EDGE360: Edge-Enabled Multi-Agent DRL for Region-Aware Rate Adaptation Solution to Enhance Quality of 360° Video Streaming," in IEEE Transactions on Mobile Computing, vol. 25, no. 2, pp. 1918-1935, February 2026, doi: 10.1109/TMC.2025.3605849.

3. F. E. Subhan, A. Yaqoob, C. H. Muntean and G.-M. Muntean, "A Viewport-Adaptive Spatial Saliency Based Meta-Reinforcement Learning Framework for Perceptual-Aware Volumetric Video Streaming," in IEEE Transactions on Network Science and Engineering, July 2026, doi: 10.1109/TNSE.2026.3715507

Conference Papers

4. F. E. Subhan, A. Yaqoob, C. H. Muntean and G.-M. Muntean, "EDQD: An Edge-Driven Multi-Agent DRL Solution for Improving Joint QoE in DASH-based Rich Media Content Delivery," 2024 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Toronto, ON, Canada, 2024, pp. 1-7, doi: 10.1109/BMSB62888.2024.10608238.

Funding Acknowledgment

This work was funded by the European Regional Development Fund through the Research Ireland Centres Programme via grant 12/RC/2289_P2 for the Insight Centre for Data Analytics and the Research Ireland Research Frontiers Projects Programme via grant 21/FFP-P/10244 for the FRADIS project.

Contents

Declaration i

Reporting of Generative AI Tool Use ii

Acknowledgements v

List of Publications vi

Table of Contents viii

List of Figures xiii

List of Tables xviii

List of Acronyms xix

Abstract xxi

Chapter 1: Introduction 1

1.1 Background 1

1.2 Challenges for Rich Media Delivery in 5G and Beyond Networks 4

1.3 Problem Statement 7

1.4 Research Motivation 9

1.5 Architecture for AI-Driven Immersive Streaming 11

1.6 Research Contributions 13

1.6.1 Overview of Research Contributions in Immersive Media Delivery 16

1.7 Organization of the Thesis 17

1.8 Chapter Summary 18

Chapter 2: Technical Background **19**

2.1 Adaptive Bitrate Streaming and DASH 19

2.2 Network Slicing and Edge Computing Landscape for Rich Media Delivery 21

2.2.1 MEC-Enabled 5G and Beyond Network for Rich Media Applications 22

2.3 Perceptual Quality Metrics for Immersive Media 24

2.3.1 VMAF for 360° Video 25

2.3.2 PCQM for Volumetric Point Clouds 25

2.4 Viewport Prediction and Region-Based Bitrate Prioritization . 26

2.4.1 Viewport Prediction with FoV Probability Maps 27

2.4.2 Region Prioritization and Bitrate Allocation via MEC . . 28

2.4.3 Mathematical Formulation 29

2.5 Fairness in Adaptive Streaming Systems 30

2.6 AI-Assisted Optimization Techniques for 5G and Beyond Networks 32

2.6.1 Reinforcement Learning 32

2.6.2 Deep Reinforcement Learning 33

2.6.3 Counterfactual Multi-Agent Policy Gradients (COMA) . . 35

2.6.4 Meta-Reinforcement Learning for Adaptive Streaming . 36

2.6.5 Categories of Meta-Reinforcement Learning 37

2.6.6 Probabilistic Embeddings for Actor-Critic Reinforcement Learning (PEARL) 38

2.6.7 Applications of Meta-Reinforcement Learning 39

2.7 Volumetric Video Streaming for Immersive Applications 39

2.8 Insights from Surveyed Literature 42

2.9 Chapter Summary 42

Chapter 3: Related Works **45**

3.1 Existing Surveys and Tutorials on AI-Enabled Network Slicing 46

3.2 MEC-Enabled 5G and Beyond Network Slicing for Rich Media Applications 50
3.2.1 Edge Intelligence for Media-Aware Adaptation 52
3.2.2 MEC-based QoE Fairness and Multi-Client Optimization 53
3.3 MEC-enabled CDN for DASH-based Rich Media Content Streaming 56
3.3.1 CDN-based Resource allocation for efficient QoE requirements 60
3.4 Adaptive Streaming for Volumetric Video 61
3.4.1 DASH-compliant segmentation and adaptive VV delivery 62
3.4.2 Learning-based bitrate and geometry optimization . . . 63
3.4.3 Generalization-centric quality adaptation frameworks . 64
3.5 Viewport Prediction and User Behavior Modeling for 360° VR and Volumetric Video 65
3.6 Chapter Summary 67

Chapter 4: EDGE360: Edge-Enabled Multi-Agent DRL for A Region-Aware Rate Adaptation Solution to Enhance Quality of 360° Video Streaming 69

4.1 Architecture 70
4.1.1 Clients 71
4.1.2 Intelligent Edge Node and Adaptation Modules 71
4.1.3 360° Video Server 74
4.1.4 Edge-Assisted Adaptive Streaming in the *EDGE360* Framework 75
4.2 *EDGE360* Adaptive Video Streaming Process 76
4.3 *EDGE360*-Driven Bitrate Adaptation and QoE Computation . 79
4.3.1 QoE Model 79
4.3.2 Mapping, Tile Chunk Data Size, and Bandwidth Constraints 84
4.3.3 QoE Calculation 89

4.3.4 Fairness Assessment 90
4.3.5 *EDGE360* Reward Function 91
4.3.6 Adaptive Bitrate Allocation for Predicted Regions in *EDGE360* 93
4.4 *EDGE360* Rate Adaptation and Problem Formulation 95
4.4.1 Foundation of MDP 96
4.4.2 State Formulation for *EDGE360* Adaptive 360° Streaming 97
4.4.3 Actor-Central Critic Policy Training for *EDGE360* 99
4.5 Implementation and Evaluation 103
4.5.1 Simulation Settings 103
4.5.2 Viewport Dataset 104
4.5.3 Rich Video Dataset and Network Traces 107
4.5.4 Reproducibility Summary for the *EDGE360* Evaluation 109
4.5.5 Results and Analysis 111
4.5.6 Statistical Significance Analysis 126
4.6 Limitations of the *EDGE360* Evaluation 133
4.7 Chapter Summary 134

Chapter 5: METAVV: A Perceptual-Aware Meta-Learning Framework for Viewport-Adaptive Volumetric Video Streaming 137
5.1 Introduction 137
5.1.1 Overview of the *METAVV* solution 138
5.1.2 View-Centric Content Block (VAQM) 139
5.1.3 Client Adaptation with Meta-Learner 144
5.1.4 Viewport and Distance-Aware Region-Based Adaptation Mechanism 148
5.2 *METAVV* Rate Adaptation and Problem Formulation 149
5.2.1 Foundation of Meta-POMDP 151
5.2.2 State Formulation for *METAVV* Adaptive Volumetric Streaming 152
5.2.3 Meta-RL Policy Learning for *METAVV* 154

5.3 Implementation and Evaluation 158
5.3.1 Simulation Settings 158
5.3.2 Video Database and Network Traces 159
5.3.3 Viewport Dataset 161
5.3.4 Reproducibility Summary 163
5.3.5 Results and Analysis 166
5.3.6 Variance and Statistical Significance Analysis 181
5.4 Limitations of the *METAVV* Evaluation 186
5.5 Conclusions 187

Chapter 6: Conclusions and Future Works **189**
6.1 Conclusions 189
6.2 Future Works 192

Bibliography **193**

List of Figures

1.1 Network slicing: a physical infrastructure is used via multiple virtual network slices to support various rich media services [1]. 4

1.2 Estimated global mobile data traffic by application category from 2017 to 2027, highlighting the dominant contribution of video content [2]. 9

1.3 AI-driven architecture for immersive media streaming across 360° and volumetric video. 11

1.4 Overview of Research Contributions in Immersive Media Delivery. 16

2.1 AI-enabled DASH-based streaming over SDN/NFV 5G network slices with client-side and server-side coordination [1]. 21

2.2 AI-Enabled MEC-Based Network Slicing Architecture for 5G and Beyond Networks. The MEC servers serve as local adaptation and decision-making nodes to support low-latency and high-reliability delivery for immersive media such as AR/VR/XR and VV across verticals including healthcare, automotive, smart cities, and tourism. 24

2.3 Viewport prediction and region-prioritized bitrate allocation with MEC-based streaming. 27

2.4 Reinforcement Learning (RL)/Deep Reinforcement Learning (DRL) Techniques: Model-based vs. Model-free [3] 33

2.5 DRL techniques for handling continuous, discrete, and hybrid state-action spaces [3] 35

2.6 Client-based adaptive volumetric video streaming architecture. A meta-learning model dynamically selects mesh levels for bitrate-efficient playback based on viewport prediction and 3D content complexity. 41

4.1 The *EDGE360* video streaming architecture comprises a CDN server, smart edges, and multiple rich media clients. 70

4.2 Core intelligent-edge components in *EDGE360*: the client reports CSI (viewport/head-rotation trajectory, buffer, and throughput) per chunk C_j, while the edge performs 360° tiling, VMAF-based tile quality monitoring, caching, and viewport-prioritized tile/bitrate recommendation 72

4.3 COMA-based multi-agent adaptation mechanism, with multiple agents and central critic 73

4.4 360° video server component: stores multi-quality video chunks and serves requested quality levels to the edge/client. . . . 75

4.5 Sequence Diagram of the Client Tiled Video Chunk Download Session 77

4.6 (a) shows the segmentation of a 360° video frame into three spatial regions: *Viewport* ($\mathcal{F}_v$) with fine-grained tiles, *Marginal* ($\mathcal{F}_m$) with average-grained tiles, and *Background* ($\mathcal{F}_b$) with coarse-grained tiles. (b)–(d) illustrate region-specific tiling, where finer tiles are allocated to high-priority regions for perceptual quality, and coarser tiles to low-priority areas for bitrate efficiency. This demonstrates *EDGE360*'s hierarchical tiling strategy for optimizing 360° VR streaming. 80

4.7 Neural Network Architecture for *EDGE360* Adaptive 360° VR Streaming Using COMA 93

4.8 *EDGE360*'s adaptive bitrate distribution among predicted regions (viewport, marginal, and background) for three test videos. 94

4.9 Sensitivity analysis of *EDGE360* for viewport prediction error, showing the impact on normalized QoE, Jain's fairness index, and average rebuffering time per chunk. 106
4.10 Bitrate Vs VMAF of 2D Video and 360° videos (*Sunset*, *Alaska*, and *Beach*) 112
4.11 Average VMAF score Vs Tile size for *Sunset*, *Alaska*, and *Beach* 360° Videos 113
4.12 Comparison of average chunk QoE values for different adaptive streaming strategies across three test videos, with error bars representing ± one standard deviation. 114
4.13 Log-scaled rebuffering penalty for three test videos (*Sunset*, *Alaska*, and *Beach*) through different adaptive streaming techniques (*EDGE360*, *360SRL*, *RAPT360*, and *BOLA360*), with error bars representing ± one standard deviation. . . . 115
4.14 CDF vs. unfairness for different streaming strategies ensuring balanced resource allocation across clients. 116
4.15 Comparison of the CDF vs. normalized average chunk QoE for sessions simulated on three test videos employing different adaptive streaming strategies. 117
4.16 Variations in chunk-wise QoE values for adaptive streaming strategies, randomly selected from three test video session simulations. 118
4.17 Average training reward (QoE) achieved by EDGE360, 360SRL, RAPT360, and BOLA360 for approximately 50 epochs. . . . 119
4.18 Ablation study showing the impact of disabling each *EDGE360* component individually on normalized QoE, Jain's fairness index, and rebuffering time. 120
4.19 Comparison of QoE, fairness, and rebuffering time across state-of-the-art ABR streaming schemes. 122
4.20 Sensitivity of *EDGE360* to reward weight λ, showing optimal QoE, fairness, and rebuffering near $\lambda = 1.2$. 122

4.21 Scalability of *EDGE360* under increasing client load: slight QoE and fairness decline due to bandwidth contention, with low rebuffering maintained by decentralized actors and region-aware tiling. 124

4.22 Statistical comparison of *EDGE360* and baseline methods using seed-level mean values and standard deviations for normalized QoE, Jain's fairness index, and average rebuffering time per chunk. 129

5.1 *METAVV* solution's architecture: a client-based meta-learning framework for adaptive VV streaming. 138

5.2 6-DoF coordinates, viewport, and view frustum in the visual space illustrating adaptive tiling of volumetric content. . . . 141

5.3 Client-side adaptation logic in *METAVV* using latent inference and meta-policy networks. 142

5.4 Distance-aware point cloud adaptation in *METAVV*. Illustration of region-based tiling and density adaptation for a volumetric client at (a) 0.5 m, (b) 1.2 m, and (c) 2.2 m distances. The predicted FoV (orange) expands with distance, while the point cloud density decreases to reduce rendering and bandwidth costs. The marginal (green) and background (blue) regions are assigned moderate and low densities, respectively. Insets (a)–(c) highlight varying point cloud densities mapped to the viewport region for each distance scenario. 145

5.5 Visual quality representation of transmitted volumetric test video sequences used in the evaluation: (a) 8iVSLF and (b) Owlii datasets. 159

5.6 Bitrate vs. PCQM for 8iVSLF and Owlii volumetric video sequences under geometry and luma distortion 166

5.7 Per-frame PCQM analysis of the volumetric video sequence under different LoDs: 100%, 50%, and 20%. 167

5.8 PCQM values across varying viewing distances for different Levels of Details (LoDs) in tile volumetric video. 168
5.9 Sensitivity analysis of *METAVV* for 6-DoF viewport prediction error, showing the impact on normalized QoE, and average rebuffering time per chunk. 169
5.10 Chunk quality variation and rebuffering time across viewing distances for different adaptive streaming schemes. 170
5.11 Normalized average QoE comparison across three volumetric test video sequences: *Longdress+Dancer*, *Soldier+Basketball*, *Red&Black*, under (a) near and (b) far distance viewing viewing conditions. 171
5.12 (a) Average training reward (normalized QoE) achieved by *METAVV*, *Sparkle*, *FewVV*, *Rolling POT*, and *ViVo* over approximately 100 epochs. (b) CDF distribution of the normalized QoE for all adaptive streaming schemes. 175
5.13 Comparison of normalized average QoE for *METAVV*, Sparkle, FewVV, Rolling POT, and ViVo across different network types: 4G/LTE, 5G, and WiFi. 177
5.14 Evaluation of five adaptive volumetric streaming schemes across (a, d) Scene Generalization, (b, e) Temporal Consistency, and (c, f) Semantic Diversity, plotted against average QoE for Near and Far viewing distances. Elliptical patches indicate performance dispersion per scheme. 178
5.15 Ablation study demonstrating the impact of disabling each *METAVV* component individually on normalized QoE, and rebuffering time. 179
5.16 *METAVV* architecture efficiency analysis, (a) Breakdown of module-wise optimized run-time per decision, and (b) resource overhead of *METAVV*, including GPU utilization and memory usage during inference execution 180

List of Tables

3.1 Summary of related surveys and tutorials [3]. 48
3.2 AI-based optimised content-delivery approaches over MEC [3]. 54
3.3 AI-based optimised content-delivery approaches over CDN [3]. 59

4.1 Notation Table 82
4.2 Reproducibility summary for the *EDGE360* evaluation. . . . 109
4.3 Optimized per-decision runtime breakdown of *EDGE360* at the edge server under varying client loads 125
4.4 Seed-level statistical summary for *EDGE360* and baseline methods. 127
4.5 Repeated-measures ANOVA results for *EDGE360* and the baseline methods across five matched seed-level evaluations. 130
4.6 Holm-corrected post-hoc paired comparisons between EDGE360 and the baseline methods across five matched seed-level evaluations. 131

5.1 Reproducibility summary for the *METAVV* evaluation. . . . 163
5.2 Seed-level variance summary for *METAVV* and baseline methods under near- and far-distance viewing. Values are reported as mean ± standard deviation across five seed-level evaluations. 182
5.3 Repeated-measures ANOVA results for *METAVV* and baseline methods. 183
5.4 Holm-corrected post-hoc paired comparisons between *METAVV* and baseline methods. 185

List of Acronyms

360SRL 360° Sequential Reinforcement Learning.

3GPP 3rd Generation Partnership Project.

5G Fifth Generation.

6G Sixth Generation.

ABR Adaptive Bitrate.

AI Artificial Intelligence.

AR Augmented Reality.

BOLA360 Buffer Occupancy-based Lyapunov Algorithm for 360° video.

CDN Content Delivery Network.

COMA Counterfactual Multi-Agent.

DASH Dynamic Adaptive Streaming over HTTP.

DDPG Deep Deterministic Policy Gradient.

DRL Deep Reinforcement Learning.

gNB next-generation Node B.

HD High Definition.

HMD Head-Mounted Display.

HTTP Hypertext Transfer Protocol.

ICN Information-Centric Networking.

ITU International Telecommunication Union.

LoD Levels of Detail.

MADRL Multi-Agent Deep Reinforcement Learning.

MDP Markov Decision Process.

MEC Multi-Access Edge Computing.

Meta-RL Meta-Reinforcement Learning.

ML Machine Learning.

MPD Media Presentation Description.

NFV Network Functions Virtualization.

Open-RAN Open Radio Access Network.

PA-DDPG Prioritized Experience Replay Actor-Critic Deep Deterministic Policy Gradient.

PCQM Point Cloud Quality Metric.

POMDP Partially Observable Markov Decision Process.

QoE Quality of Experience.

QoS Quality of Service.

RAPT360 Reinforcement Learning-based Adaptive Prediction and Tiling for 360° video.

RL Reinforcement Learning.

SAND Server and Network DASH.

SDN Software-Defined Networking.

UHD Ultra High Definition.

VMAF Video Multimethod Assessment Fusion.

VNF Virtualized Network Functions.

VR Virtual Reality.

VV Volumetric Video.

XR Extended Reality.

Abstract

Artificial Intelligence–Driven Rate Adaptive and Perceptual-Aware Rich Media Delivery in 5G and Beyond Networks

Fazal E Subhan

The rapid evolution of immersive and bandwidth-intensive applications such as 360-degree video, Virtual Reality (VR), Augmented Reality (AR), Extended Reality (XR), and Volumetric Video (VV) poses substantial challenges to contemporary network infrastructures, especially in delivering high perceptual Quality of Experience (QoE) and real-time content adaptability. These challenges are intensified in Fifth Generation (5G) and Sixth Generation (6G) environments, where heterogeneous user demands, device capabilities, and network conditions necessitate intelligent, context-aware streaming solutions. To address these complexities, this thesis investigates the integration of Artificial Intelligence (AI) techniques with enabling technologies in the context of network slicing, including Software-Defined Networking (SDN), Network Functions Virtualization (NFV), and a specifically proposed Multi-Access Edge Computing (MEC)-based Adaptive Bitrate (ABR) streaming strategy to support rich media delivery.

The thesis proposes *EDGE360*, a novel edge-assisted Multi-Agent Deep Reinforcement Learning (MADRL) framework for region-aware ABR control in tile-based 360-degree video streaming. *EDGE360* employs the Counterfactual Multi-Agent (COMA) policy gradient method to optimize bitrate allocation across viewport, marginal, and background regions, using the Video Multimethod Assessment Fusion (VMAF) metric and viewport trajectory predictions. A centralized critic collaborates with distributed agent networks deployed at the edge, leveraging real-time feedback from client buffers, throughput estimations, and tile-level monitors. The proposed solution models the adaptation task as a Partially Observable Markov Decision Process (POMDP) and ensures joint QoE optimization and fairness across multiple clients. Experimental results confirm that *EDGE360* significantly outperforms existing state-of-the-art ABR schemes by achieving the highest average reward, outperforming RAPT360, 360SRL, and BOLA360 by 8.12%, 11.86%, and 18.00%, respectively, demonstrating superior convergence and refinement under dynamic network conditions.

Extending adaptive streaming to next-generation VV scenarios, the thesis also introduces *METAVV*, a novel saliency-aware viewport-adaptive Meta-Reinforcement Learning (Meta-RL) framework built on the PEARL (Probabilistic Embeddings for Actor–Critic Reinforcement Learning) paradigm. Formulated as a POMDP, *METAVV* incorporates 6-DoF viewport prediction, Levels of Detail (LoD) mesh segmentation, and perceptual quality assessment based on the Point Cloud Quality Metric (PCQM). By inferring latent belief vectors from session-level observations, *METAVV* dynamically conditions the ABR policy for fast adaptation across out-of-distribution network and user mobility scenarios. This context-aware design enables rapid convergence, personalized streaming decisions, and consistent perceptual quality even under constrained or unseen conditions. Evaluations using public VV datasets show that, under near-distance viewing, METAVV improves average QoE by 3.25%, 3.78%, 11.98%, and 18.91% over FewVV, Sparkle, Rolling POT, and ViVo, respectively. Under far-distance viewing, the corresponding QoE gains are 8.81%, 7.17%, 16.94%, and 24.93%, respectively. These results demonstrate that the proposed solution outperforms state-of-the-art solutions in rate adaptation under dynamic and out-of-distribution conditions.

The thesis also includes a comprehensive survey that discusses AI-driven advancements for intelligent content-centric networking, predictive streaming, cache optimization, and secure slice-aware traffic management. It highlights how AI models ranging from Machine Learning (ML) to DRL are reshaping slice orchestration and media delivery in future networking environments.

Chapter 1

Introduction

1.1 Background

Rich media content, sometimes referred to also as immersive media, offers users the opportunity to interact with other worlds, real or virtual, by using diverse stimuli [3, 4]. Applications supporting rich media have gained significant traction and are no longer limited to online gaming and entertainment; they are increasingly used in teaching and learning, training, museum tours, healthcare, advertising, social networking, and user-generated scenarios [5]. AR, VR, XR, multi-sensorial, omnidirectional (360-degree), and volumetric media are among the key immersive technologies driving this growth [6].

360° video is an immersive rich-media format that captures an omnidirectional scene, enabling users to freely look around from a fixed viewpoint by interactively changing their viewing direction (i.e., three-degrees-of-freedom, 3-DoF) in VR/XR applications. In addition, volumetric video (VV) is an emerging rich-media format that captures real-world scenes as fully three-dimensional (3D), six-degrees-of-freedom (6-DoF) content typically represented as dynamic point clouds and meshes, allowing users to move around and view the scene from any viewpoint in immersive VR and XR applications. Major industry players such as Meta, Google, Samsung,

Apple, and VIVE have invested in diverse hardware and software ecosystems to provide users with high-quality immersive experiences [7].

From a networking perspective, such services fall squarely within the *enhanced Mobile Broadband (eMBB)* use case of 5G and beyond networks, which targets ultra-high data rates, low latency, and high reliability [8]. Immersive media streaming, however, pushes eMBB requirements to their limits due to its stringent demands on sustained throughput, tight delay constraints, and real-time adaptation to user interactions and mobility. Typically, users consume such content using Head-Mounted Displays (HMDs) , gaming consoles, or high-end mobile devices demanding High Definitions (HDs), Ultra High Definitions (UHDs), 4K/8K resolutions for their video components, and strong network Quality of Service (QoS) requirements such as ultra-low latency and very high throughput.

Unfortunately, the delivery of rich media content over networks is a major bottleneck. For instance, according to the most recent Cisco Annual Internet Report (2018-2023), the minimum bandwidth required for HD VR and UHD VR streaming is 167 Mbps and 500 Mbps, respectively, whereas the current average mobile network speed is 43.9 Mbps [9]. Along with mobile networks, the broadband networks have a much larger bandwidth, which should help. However, they also carry increasing amounts of traffic. In 2023, an estimated 90% of all IP traffic is video-like, and 66% of the installed flat-panel TVs, which also use this infrastructure, are using high bitrate UHD content [10]. The rapid proliferation of immersive multimedia such as AR, VR, XR, and VV Services (VVSs), coupled with the demand for UHD video content, is significantly impacting modern mobile networks. According to the Ericsson Mobility Report [2], global mobile data traffic has seen a dramatic increase, rising from 10.9 Exabytes (EB) per month in 2017 to 90.4 EB per month in 2022, and it is projected to reach 282.8 EB per month by 2027. As illustrated in Fig. 1.2, video content remains the dominant contributor, accounting for the vast majority of mobile data usage. This trend underscores the central role of multimedia, particularly

video in driving future mobile traffic growth. In parallel, monthly traffic per smartphone has also surged, increasing from 2.6 GB in 2017 to 15.2 GB in 2022, and is expected to reach 39.7 GB by 2027, further emphasizing the growing demand for high-bandwidth, content-rich applications in mobile environments.

Although broadband networks offer higher capacity, they also carry rapidly growing traffic loads. Recent operator forecasts continue to show that video dominates mobile usage; Ericsson reports that video accounted for about 73% of mobile data traffic at the end of 2023 and is expected to reach 76% by the end of 2025. In parallel, global mobile traffic volumes have reached unprecedented levels, with total monthly mobile network data traffic reported at approximately 200 EB in Q4 2025 [11]. Despite these improvements, typical user-access speeds still lag behind immersive-media requirements at scale; the ITU's Global Connectivity Report, i.e., drawing on Speedtest by Ookla, reports average global mobile download speeds on the order of $\sim$92 Mbit/s in 2025 [12]. Taken together, these trends reinforce that immersive services can quickly saturate available capacity and remain highly sensitive to throughput variability and delay under real-world network dynamics.

In this regard, improving high-bitrate content transmissions in the current network environment is fundamental for the success of rich media services. Among several innovative solutions for network resource management [13], there are network slicing [14] and edge computing, which can be used to orchestrate and manage network resources to meet the requirements of various rich media services, as illustrated in Fig. 1.1. Network slicing enables the division of physical infrastructure into multiple virtualized networks, each tailored to specific service-level requirements and user contexts. Multi-Access Edge Computing (MEC), in this context, brings computation and storage closer to users by deploying resources at the edge of the network, thereby reducing network delay and enabling real-time, context-aware service delivery.

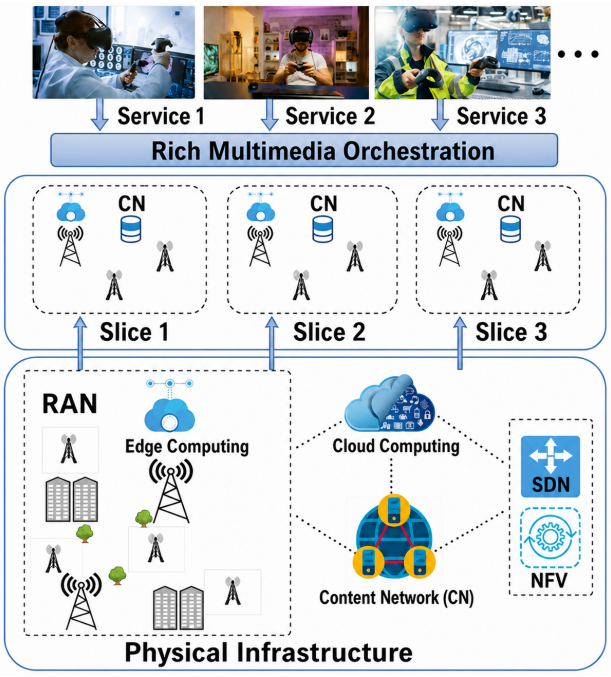


Figure 1.1: Network slicing: a physical infrastructure is used via multiple virtual network slices to support various rich media services [1].

Furthermore, the integration of AI techniques with network slicing and MEC enhances the adaptability and efficiency of multimedia delivery systems. By deploying intelligent decision-making to the network edge, such systems can dynamically respond to fluctuating bandwidth, user behavior, and content-specific requirements. This paradigm is central to the research presented in this thesis, which proposes an edge-assisted and client-driven, AI-driven solution for adaptive bitrate selection in immersive 360° VR and VVS. The proposed framework leverages edge intelligence, processing, video rate adaptation, and perceptual video quality metrics to optimize transmission QoS and maximize user-perceived Quality of Experience (QoE) in multi-user, bandwidth-constrained environments.

1.2 Challenges for Rich Media Delivery in 5G and Beyond Networks

Despite advances in multimedia hardware and software, delivering high-bitrate rich media content over existing network infrastructures remains a

substantial challenge. Although broadband networks offer higher capacity, they are also burdened with a growing volume of traffic (e.g., 90% of all IP traffic in 2023 was video-like, with 66% of flat-panel TVs streaming UHD content) [10]. In 5G and beyond networks, an innovative concept named *network slicing* [14] can allocate dedicated virtual resources for different service classes, but slice-level provisioning alone does not guarantee stable user experience under fast wireless fluctuations, mobility, and heterogeneous device capabilities. Hence, adaptive mechanisms are required at the application layer, where ABR algorithms [15, 16] adjust bitrate based on real-time bandwidth and buffer observations and Dynamic Adaptive Streaming over HTTP (DASH) [17] enables client-driven rate selection. However, immersive rich media introduces additional challenges that make ABR/DASH-only solutions insufficient in multi-client and constrained environments. To address these challenges in a principled manner, this thesis develops an algorithmic adaptation framework that tightly couples (i) perceptual region/viewport-aware quality optimization with (ii) network-aware sequential decision-making, enabling robust, low-latency bitrate/quality control under uncertainty and multi-client contention.

A fundamental difficulty is that immersive content has *non-uniform perceptual importance* across space and time, which requires *user viewport* awareness. The *user viewport* denotes the subset of content that is currently within the user's visible Field-of-View (FoV). For 360° video, the viewport is the user's FoV on the spherical frame (primarily 3-DoF head rotation), whereas for volumetric video, the visible content depends on both head orientation and user position (6-DoF), forming a view frustum in 3D space. Since the viewport changes over time and must typically be *predicted*, any viewport prediction error can cause bandwidth to be spent on less relevant regions or degrade quality in the actually viewed region, directly harming QoE.

Building on viewport awareness, *region saliency* is introduced as a notion of *how important a spatial region is to perceived quality*. In practice,

saliency can be represented by region weights that prioritize viewport and near-viewport areas (and, for volumetric content, often incorporate viewing distance), enabling *region-aware* resource allocation. This naturally leads to region segmentation of the scene into multiple areas (e.g., viewport/marginal/background). A key design dimension is *region segmentation granularity (or graduality)*, which describes (i) *how finely* the content is partitioned (e.g., tile-level vs. coarse regions, or multiple geometry LoD levels) and (ii) *how smoothly* quality changes are applied across regions to avoid noticeable boundaries and flicker. Finer granularity can improve perceptual targeting but increases encoding/packaging overhead, signaling cost, and decision complexity; coarser granularity reduces overhead but can waste bandwidth and reduce QoE when the viewport shifts rapidly.

Another challenge is that immersive streaming must optimize *perceptual quality metrics* that are media-specific. For 360° video, perceptual video quality is often quantified using the *Video Multi-method Assessment Fusion* (*VMAF*), which provides a quality score that better correlates with perceived quality than simple fidelity metrics; in immersive settings, it is common to consider viewport- or region-weighted quality to reflect what the user actually sees. For volumetric video, perceptual quality must account for 3D reconstruction artifacts (geometry and/or texture distortions). Therefore, the *Point Cloud Quality Metric* (*PCQM*) is used to quantify point-cloud content quality, and it can likewise be extended with region/viewport and distance awareness. Importantly, these perceptual objectives must be balanced against network-driven penalties such as rebuffering, latency, and excessive quality switches, which can be particularly disruptive in XR and data-intensive VV content.

Finally, rich media adaptation is inherently a *sequential decision-making* problem under uncertainty, since future bandwidth, latency, and user motion are not perfectly known. This motivates modeling the streaming control problem as a *Markov Decision Process* (*MDP*) or, more realistically, a *Partially Observable MDP* (*POMDP*), where the agent makes decisions based on par-

tial observations (e.g., noisy throughput samples and imperfect viewport estimates) rather than full system state. Moreover, realistic deployments involve *multiple clients* competing for shared resources, where independent adaptation can lead to instability and unfairness. This motivates *multi-agent DRL*, in which multiple agents (e.g., per-client decision makers) learn coordinated policies under shared constraints. Within this space, the *Counterfactual Multi-Agent Policy Gradients* (*COMA*) is a representative actor-critic approach that addresses multi-agent credit assignment by using a counterfactual baseline to estimate each agent's marginal contribution to a global objective [18]. In addition, because network conditions and user behaviors can change abruptly across sessions, *Meta-RL* is introduced as a learning paradigm that aims to train policies capable of *fast adaptation* to new tasks/environments with limited additional experience, a property that becomes important when streaming systems must generalize across diverse and previously unseen conditions.

In summary, the core challenges for rich media delivery over 5G and beyond networks include: (i) extreme and variable bandwidth demand, (ii) strict latency and rebuffering sensitivity, (iii) viewport-driven and saliency-driven non-uniform perceptual importance, (iv) segmentation granularity-overhead trade-offs, (v) media-specific perceptual quality metrics (VMAF for 360° video and PCQM for volumetric video), and (vi) partial observability and multi-client coupling that motivate POMDP-based reasoning and learning-based control (including multi-agent DRL, and Meta-RL). These concepts provide the foundation for the methodology and system designs presented later in this thesis.

1.3 Problem Statement

The class of eMBB services for 5G and beyond networks is expected to support immersive rich media applications such as 360° video and VV streaming, which impose extreme requirements on throughput, latency,

reliability, and adaptability [3]. Unlike conventional 2D video, immersive media exhibit strong spatial, perceptual, and user-centric characteristics, where only a subset of the content (e.g., viewport and near-field 3D regions) directly impacts the user QoE. Efficient delivery, therefore, benefits from region-prioritized, perceptually aware, and context-adaptive streaming mechanisms.

The problem addressed in this thesis is *to design AI-driven, user-perception-aware, and region-prioritized adaptive streaming architectures* for 360° video and VV content in eMBB environments, such that scalable, high-quality immersive experiences can be delivered over 5G and beyond latency-sensitive and broadband networks under dynamic and multi-user conditions.

However, existing ABR solutions primarily designed for monolithic 2D video remain inadequate for immersive streaming in eMBB environments due to the following fundamental limitations:

- **Absence of perceptual and spatial awareness:** Current ABR schemes optimize bitrate based on throughput and buffer occupancy, but fail to incorporate perceptual quality metrics (e.g., VMAF, PCQM) and spatial importance, which are critical for viewport-centric 360° video and distance-sensitive volumetric content.

- **Poor coordination and fairness in multi-user:** Client-centric and heuristic ABR approaches lack cooperative decision-making, resulting in inefficient bandwidth utilization, QoE imbalance, and unfair resource allocation when multiple immersive users compete for shared radio and edge resources.

- **Limited adaptability and generalization under dynamic conditions:** Rapid fluctuations in wireless links, user motion (3-DoF/6-DoF), and content heterogeneity require learning-based solutions capable of real-time inference, fast adaptation, and generalization across unseen network and content conditions capabilities that conventional DRL-based ABR models struggle to provide.

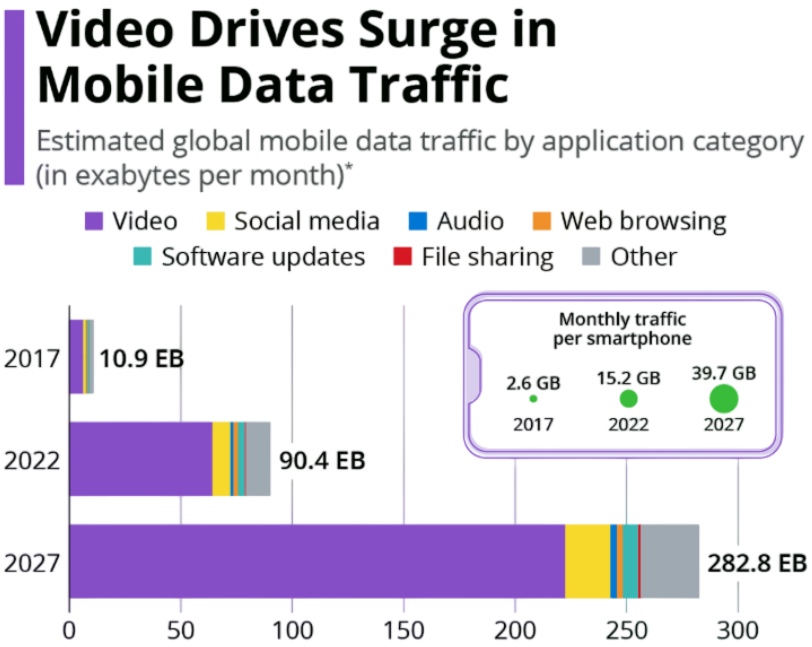


Figure 1.2: Estimated global mobile data traffic by application category from 2017 to 2027, highlighting the dominant contribution of video content [2].

While architectural enablers such as network slicing, SDN, and MEC provide the necessary infrastructure for scalable service provisioning, they do not inherently resolve the decision-making complexity of immersive media adaptation. Consequently, there is a clear need for AI-driven control frameworks that can jointly exploit edge intelligence, perceptual modeling, cooperative and meta-learning strategies to enable scalable, fair, and QoE-aware immersive media delivery.

1.4 Research Motivation

The integration of immersive applications, including VR streaming, volumetric video, and high-definition video conferencing, is placing unprecedented strain on existing wireless infrastructures. As immersive experiences become increasingly integral to education, communication, and professional workflows, user expectations for seamless, low-latency, and high-quality interactions are rising accordingly. These expectations translate

into stringent requirements for end-to-end Quality of Service (QoS) and user-perceived QoE. Conventional video delivery systems rely heavily on centralized, heuristic-based ABR algorithms that adjust bitrate based on estimated throughput and buffer levels. However, these solutions are not sufficient for supporting multi-user environments in 5G and beyond networks, where variable link conditions, heterogeneous device capabilities, and limited edge resources must be considered. They also lack the intelligence to account for perceptual quality differences, especially in dynamic and user-centric immersive contexts such as viewport-based VR streaming.

Furthermore, recent advancements in MEC bring computation and storage capabilities closer to end-users, making it feasible to deploy intelligent video adaptation strategies at the network edge. This opens new opportunities for context-aware, delay-sensitive streaming applications. However, the efficient use of MEC in immersive media delivery still demands intelligent mechanisms capable of making rapid, decentralized decisions in real-time.

In this context, there is a strong research impetus to develop intelligent edge-based ABR mechanisms that:

- Leverage network conditions, buffer occupancy, and perceptual quality indicators such as VMAF for 360° video and PCQM for volumetric content to guide bitrate decisions.
- Adapt effectively across diverse user behaviors and fluctuating network environments through advanced ML models, including RL, DRL, and meta-learning–based strategies that support cross-content generalization.
- Scale to support multiple users in real time while ensuring QoE fairness and resource efficiency at the MEC-enabled edge.

Our previously published survey [3] systematically explored how AI-enabled network slicing, when combined with enabling technologies such as SDN, NFV, Information-Centric Networking (ICN), and Content Delivery Network (CDN), can improve multimedia content delivery. Among these, MEC emerges

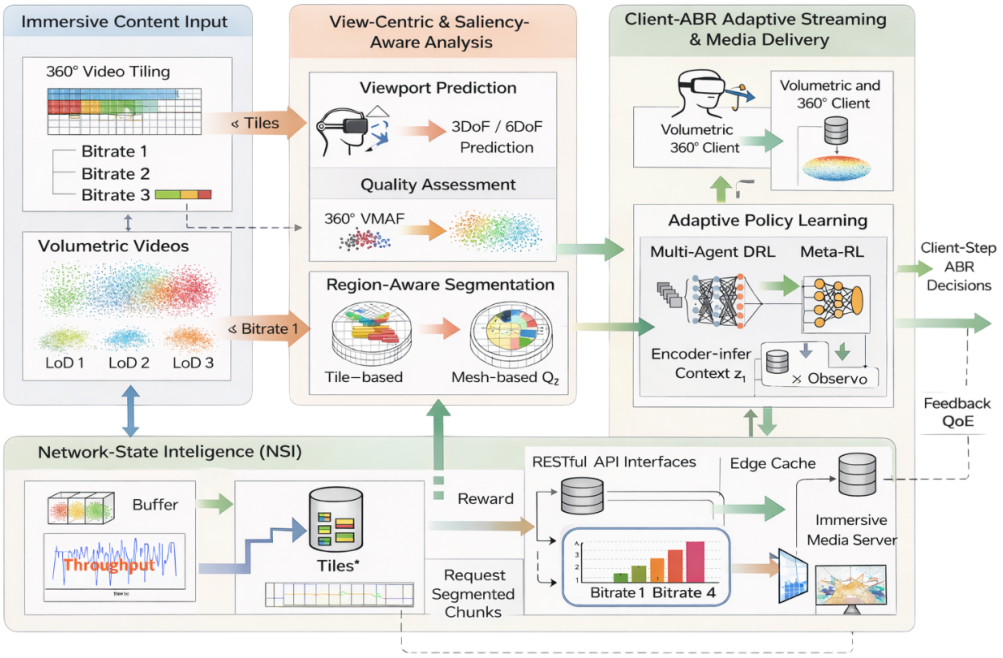


Figure 1.3: AI-driven architecture for immersive media streaming across 360° and volumetric video.

as a pivotal technology for supporting latency-sensitive and bandwidth-intensive applications such as immersive media streaming. By bringing computation and storage closer to end-users, MEC holds the promise of reducing end-to-end latency and enabling more responsive service delivery. Despite its theoretical advantages, the simulation and evaluation of MEC-assisted adaptive streaming systems under realistic network conditions remain limited in existing literature. This thesis addresses this gap by proposing an MEC-centric, AI-driven framework designed to support scalable, QoE-aware immersive content delivery in 5G and beyond networks.

1.5 Architecture for AI-Driven Immersive Streaming

Fig. 1.3 illustrates the unified AI-driven adaptive streaming architecture proposed in this thesis, which serves as the foundation for the major solutions presented in various chapters. This architecture encapsulates the full pipeline for immersive media streaming and adaptive bitrate decision-making across both 360° video and VV content. It is divided into five tightly

coupled functional blocks, each addressing a specific challenge in the intelligent media delivery lifecycle.

- **Immersive Content Input:** This block handles the preparation of streaming media. It supports tiled encoding of 360° videos at multiple bitrates and mesh-based LoDs for volumetric videos. The content is pre-segmented into spatio-temporal chunks to facilitate adaptive transmission based on user viewport and region saliency.

- **View-Centric & Saliency-Aware Analysis:** This module performs key perceptual tasks. First, a *viewport prediction* model forecasts future user views based on 3-DoF (head orientation) and 6-DoF (head + position) trajectories. Next, *quality assessment* metrics such as 360° VMAF and point-cloud-based PCQM are employed to quantify the perceptual relevance of content. Finally, *region-aware segmentation* determines spatial granularity using tile-based (for 360°) and mesh-based (for VV) segmentation strategies to prioritize bitrate allocation according to predicted importance.

- **Adaptive Policy Learning:** This is the intelligence core of the architecture. It incorporates both MADRL (multi-agent DRL) and Meta-RL strategies to enable robust adaptive bitrate selection. The *Multi-Agent DRL* module allows individual actors to collaboratively learn content and network-aware ABR decisions. The *Meta-RL* module uses an encoder-inferer mechanism to derive task-specific context vectors z_t, enabling few-shot generalization across unseen content and environments. These policies are trained to maximize a reward signal based on QoE, rebuffering, and visual quality smoothness.

- **Network-State Intelligence (NSI):** This module continuously monitors real-time network conditions, including *throughput*, *buffer occupancy*, and historical *latency* trends. This dynamic information is used

to influence both the ABR decision process and the segmentation granularity, forming a feedback loop with the learning models.

- **Client-ABR and Edge-Assisted Delivery:** This final module implements adaptive delivery. ABR decisions made by DRL/Meta-RL agents are transmitted to request specific bitrate segments. An *edge cache* stores frequently accessed content tiles or segments to minimize latency. The immersive media server delivers the requested chunks to the client, completing the closed-loop system with real-time *QoE feedback* for further optimization.

In general, this architecture supports 360° and volumetric media streaming scenarios by combining perceptual saliency modeling, adaptive region segmentation, network-aware learning, and client-step delivery coordination. It is versatile enough to adapt to fluctuating network conditions, dynamic user interactions, and content-specific perceptual constraints that lay the groundwork for the thesis's contributions to robust, generalizable ABR strategies in immersive media streaming.

1.6 Research Contributions

This thesis makes three original contributions to the field of adaptive bitrate immersive multimedia streaming across diverse network scenarios. The main novel contributions of this thesis are as follows:

Contribution 1: *EDGE360*: Edge-Enabled Multi-Agent DRL for Region-Aware Rate Adaptation Solution to Enhance Quality of 360° Video Streaming [19].

Contribution 2: *METAVV*: A Viewport-Adaptive Spatial Saliency Based Meta-Reinforcement Learning Framework for Perceptual-Aware Volumetric Video Streaming [20].

Contribution 3: A Survey on Artificial Intelligence Techniques for Improved Rich Media Content Delivery in a 5G and Beyond Network Slicing Context [1].

The novel contribution *EDGE360* [19], is an edge-assisted MADRL framework that applies COMA-based bitrate adaptation across spatial video regions using viewport prediction and VMAF-guided quality estimation for 360-degree video streaming. The strategy employs the COMA policy gradient algorithm to optimize bitrate distribution across three spatial regions: viewport, marginal, and background, using real-time viewport prediction and VMAF-based tile quality estimation. The streaming problem is modeled as a POMDP [21], where agents operate cooperatively under the guidance of a centralized critic. Each agent observes local environment metrics such as throughput, buffer occupancy, and predicted head orientation to make informed bitrate decisions.

EDGE360 is evaluated against state-of-the-art approaches, including Reinforcement Learning-based Adaptive Prediction and Tiling for 360° video (RAPT360) [22], 360° Sequential Reinforcement Learning (360SRL) [23], and Buffer Occupancy-based Lyapunov Algorithm for 360° video (BOLA360) [24], demonstrating notable gains in average reward, QoE fairness, and bitrate utility. By combining deep ML with region-based tiling and edge processing, *EDGE360* addresses both the delivery efficiency and quality adaptation challenges faced by immersive content systems in 5G and beyond networks. This makes it a strong candidate for future deployment in real-world multimedia platforms seeking scalable, intelligent content distribution mechanisms.

Building on the insights and limitations identified in earlier adaptive streaming frameworks such as *EDGE360*, this research extends region-aware bitrate adaptation into the more complex domain of volumetric video streaming (VVS) through a novel client-centric Meta-RL based solution termed *METAVV*. Unlike conventional DRL-based ABR solutions that rely on fixed policies and extensive retraining, *METAVV* equips each client with a lightweight

Meta-RL agent that rapidly adapts bitrate and LoD decisions using latent context inference from recent network conditions, viewport dynamics, and perceptual feedback.

The proposed adaptive VVS solution targets segmented point-cloud content and integrates a distance-aware PCQM-based perceptual quality model, enabling bitrate decisions that explicitly align resource allocation with the perceptual relevance of spatial regions in a 6-DoF environment. In contrast to throughput-driven ABR strategies, *METAVV* employs region-prioritized adaptation, assigning differentiated importance to viewport, marginal, and background regions based on predicted user attention and viewing distance. This design allows the system to preserve high-fidelity geometry in perceptually critical regions while aggressively reducing bitrate for less salient content.

By leveraging meta-learning–based policy generalization, *METAVV* demonstrates the effectiveness of fast-adapting, client-driven intelligence in addressing the unique challenges of volumetric streaming, including massive data volumes, spatial–temporal complexity, and highly dynamic network conditions. Extensive evaluation confirms that *METAVV* delivers improved QoE stability, faster adaptation, and superior generalization across unseen content and network scenarios.

The thesis also includes a comprehensive survey that discusses AI-driven advancements for intelligent content-centric networking, predictive streaming, cache optimization, and secure slice-aware traffic management. Central to these advancements is the integration of MEC and CDN, which shifts computational resources and storage to the network edge to satisfy the ultra-low latency and high-throughput demands of immersive media [3]. By leveraging edge computing functionalities, the proposed survey highlights how AI models ranging from standard ML to complex DRL are reshaping edge-assisted slice orchestration and media delivery through localized, real-time decision-making. These edge-centric intelligence models enable proactive content prefetching and dynamic resource allocation, successfully miti-

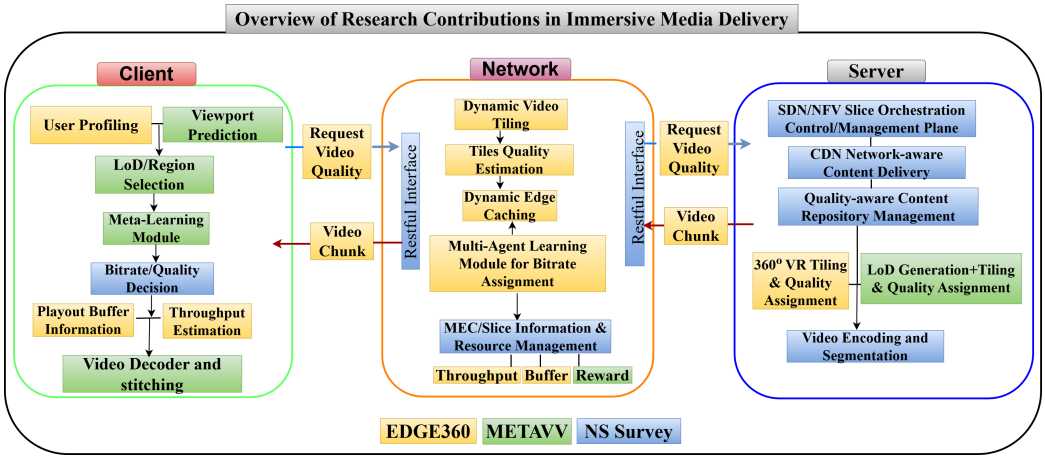


Figure 1.4: Overview of Research Contributions in Immersive Media Delivery.

gating the bottlenecks of traditional core-dependent architectures in future 5G and beyond environments.

1.6.1 Overview of Research Contributions in Immersive Media Delivery

Fig. 1.4 presents a high-level overview of the immersive media delivery solution proposed in this thesis. This illustrates the interaction among the client, network, and server entities through RESTful interfaces. The yellow-colored modules represent functionalities primarily aligned with the *EDGE360* solution, such as dynamic video tiling, tile quality estimation, and multi-agent learning for bitrate adaptation. The green-colored modules reflect contributions of the *METAVV* framework, including region-based LoD selection, viewport prediction, and meta-learning-based bitrate decision. The modules highlighted in blue pertain to the contribution of the network slicing survey, including SDN/NFV slice orchestration, MEC-assisted resource management, and content-aware delivery through CDNs. Arrows represent control and data flow across the pipeline, from viewport-driven bitrate requests to chunk delivery. This architectural breakdown captures the integration of AI-based decision-making, edge intelligence, and network

slicing technologies across the media pipeline from content generation on the server to viewport-aware decoding on the client.

1.7 Organization of the Thesis

This thesis is organized as follows:

- **Chapter 1** - Presents the research motivation, problem statement, research objectives, and main contributions of this thesis.

- **Chapter 2** - Provides a technical background on enabling technologies, including network slicing, MEC, video quality metrics, AI, Meta-RL, and DRL-based rate optimization frameworks.

- **Chapter 3** - Surveys recent research efforts and identifies the research gaps related to intelligent rich media delivery over 5G and beyond networks.

- **Chapter 4** - Introduces the *EDGE360* framework, a COMA-based MADRL solution for viewport-aware adaptive 360° video streaming, leveraging tile granularity, VMAF scoring, and edge-assisted bitrate allocation to enhance QoE and fairness.

- **Chapter 5** – This chapter introduces *METAVV*, a client-centric framework Meta-RL for adaptive streaming of segmented VV content. The proposed system employs a meta-learner that enables perceptually guided, distance-aware bitrate adaptation at the client side, improving generalization across diverse network scenarios and unseen content distributions.

- **Chapter 6** - Summarizes the main findings of this research and outlines potential directions for future work.

1.8 Chapter Summary

The introduction chapter presents the motivation, challenges, problem statement, and research context for the adaptive and efficient delivery of rich media content in 5G and beyond networks. It outlines the limitations of traditional streaming architectures and emphasizes the need for intelligent and perceptually aware adaptation strategies. The chapter highlights the role of AI-enabled network slicing, edge intelligence, and client-side learning in improving the QoE of immersive media services. This chapter highlights the main novel contributions of the research work performed, such as the *EDGE360* framework for tile-based 360° video streaming using viewport prediction and COMA-based bitrate adaptation, and the *METAVV* framework for VV streaming, which employs client-side Meta-RL and PCQM-guided bitrate selection to enable scalable and generalizable streaming under dynamic network conditions. In the following chapters, these contributions will be described in detail.

Chapter 2

Technical Background

The rapid growth of immersive media applications such as VR, AR, XR, and VV has introduced stringent requirements on latency, throughput, and user-perceived QoE in modern networks. These demands have accelerated the evolution of 5G and 6G network systems and highlighted the importance of AI-driven adaptation mechanisms deployed at the network edge. This chapter presents the technical background relevant to this thesis, focusing on ABR streaming, immersive media representations, and learning-based optimization techniques. Particular emphasis is placed on MEC-enabled architectures and reinforcement learning paradigms, including DRL, MADRL, and Meta-RL, which enable real-time, perceptually aware, and client-centric streaming decisions. These foundations underpin the proposed EDGE360 and METAVV frameworks for scalable and intelligent immersive media delivery.

2.1 Adaptive Bitrate Streaming and DASH

The Dynamic Adaptive Streaming over HTTP (DASH)[1] has become the de facto standard for Hypertext Transfer Protocol (HTTP)-based video delivery, enabling streaming clients to dynamically adapt media quality based on real-time measurements such as estimated throughput and buffer occupancy [25]. As depicted in Fig. 2.1, a rich media source—such as 360° or

[1] https://github.com/dash-industry-forum/dash.js/

VV is encoded into multiple quality representations (e.g., 8–20 Mbps) and described through a Media Presentation Description (MPD). These representations are hosted on a Server and Network DASH (SAND)/DASH server that responds to client-side HTTP segment requests.

Traditionally, ABR algorithms reside at the client side and operate heuristically to select bitrate levels based on local observations. However, in heterogeneous wireless environments characterized by dynamic bandwidth, congestion, and competing users, particularly for immersive media like VR, AR, XR, and VV, these heuristic schemes often fail to deliver consistent QoE [26]. This limitation becomes even more pronounced when streaming high-resolution tiled and volumetric content, which demands fine-grained bitrate allocation, low latency, and responsiveness to spatial importance (e.g., viewport focus areas) [27, 28].

To address these limitations, recent research has focused on AI-driven and network-assisted solutions. Notably, SAND introduces server and network-assisted streaming enhancements by exposing performance metrics and network information to guide adaptation decisions. Moreover, DRL-based ABR approaches have demonstrated promise in learning policies that optimize long-term QoE, outperforming hand-crafted heuristics.

The illustrated architecture (Fig. 2.1) integrates SAND/DASH streaming with a 5G network slice managed by SDN/NFV controllers. The AI-enabled slice dynamically selects the optimal path and bandwidth allocation based on application-level metadata (e.g., MPD) and predicted user demand. Through integration with CDN and ICN mechanisms, this design ensures that content segments are delivered efficiently, while AI modules at the edge support intelligent ABR decision-making for the DASH client.

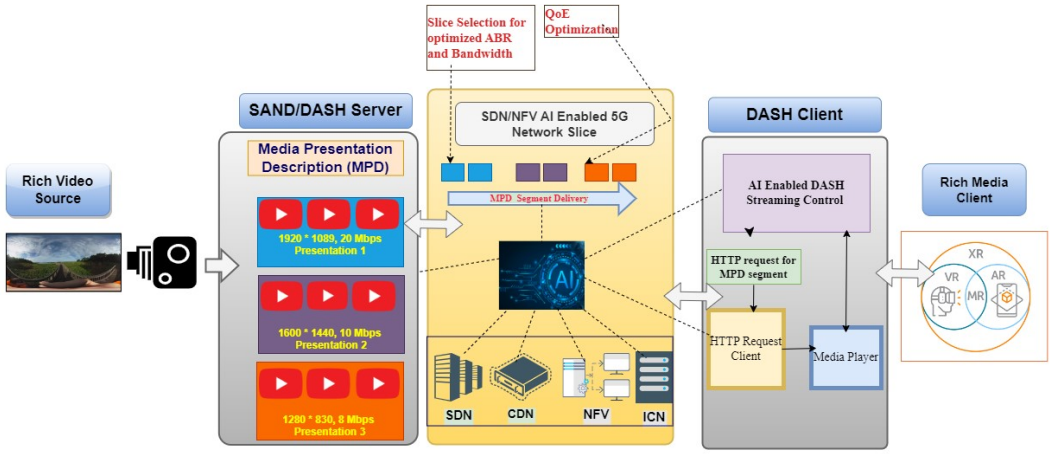


Figure 2.1: AI-enabled DASH-based streaming over SDN/NFV 5G network slices with client-side and server-side coordination [1].

2.2 Network Slicing and Edge Computing Landscape for Rich Media Delivery

5G and future 6G networks are poised to revolutionize the delivery of AR, VR, XR, and 360-degree content by supporting ultra-reliable, low-latency, and high-throughput transmission via edge computing enabling technologies. A foundational enabler of this transformation is *network slicing*, which allows multiple virtualized and service-specific logical networks to run on top of shared physical infrastructure. According to recent industry forecasts, network slicing is expected to support around 30% of future 5G and 6G use cases, especially for immersive content delivery, with market valuations ranging from $45 billion to $200 billion by 2030 [29, 30]. Organizations such as 3rd Generation Partnership Project (3GPP) and International Telecommunication Union (ITU) have specified minimum service and infrastructure requirements for 5G and 6G systems to enable differentiated performance across slices. These include significant improvements in data rates (up to 1 Tbps), latency (down to 0.1 ms), bandwidth utilization (up to 100 GHz), and device connectivity (up to 10^8 devices/km^2) [8, 31]. These capabilities directly support resource-intensive applications such as UHD/8K video and immersive VR/XR experiences.

However, to meet such demands effectively, especially for rich media streaming scenarios, intelligent resource allocation is required at both core and edge levels. Our research work targeted this problem by proposing a MADRL-based rate adaptation framework for DASH-enabled video delivery. It emphasized fairness-aware bitrate optimization across multiple clients sharing edge bandwidth and laid the groundwork for the more advanced edge-enabled framework.

The later chapters in this thesis extend the network slicing vision by leveraging edge intelligence and viewport-based tiling to dynamically allocate bitrates across video regions employing multi-agent DRL-based policy gradients frameworks. These systems align directly with the performance objectives of 5G and 6G slices and demonstrate how AI-driven content delivery architectures can achieve scalability, responsiveness, and perceptual quality in future media ecosystems.

2.2.1 MEC-Enabled 5G and Beyond Network for Rich Media Applications

The integration of AI with MEC-enabled 5G and beyond networks offers considerable advantages for enhancing the delivery of rich media content. MEC extends traditional network capabilities by bringing computing and storage closer to the end-user at the network edge, enabling ultra-low latency operations, context-aware processing, and real-time service responsiveness [32]. This edge shift significantly enhances QoS and QoE for immersive media services such as AR, VR, XR, and VV streaming.

The MEC servers deployed at the edge can locally cache, transcode, and deliver media segments tailored to individual users. Real-time inference using ML models allows the system to predict user behavior, content popularity, and network load patterns. Based on these predictions, content is adaptively compressed and delivered through optimal paths, reducing bandwidth overhead and minimizing playback interruptions.

In advanced frameworks MEC nodes can also host reinforcement learning-based models (e.g., MADRL), allowing distributed actors (at clients) and centralized critics (at the edge) to collaboratively optimize video delivery [33]. The actor-critic framework enables adaptive bitrate and tile quality allocation based on viewport prediction, buffer status, and fluctuating network conditions [34].

Moreover, AI-driven analytics at the MEC layer enhance proactive scheduling and resource provisioning. For instance, if multiple users in a smart city environment are accessing volumetric content, the MEC node can predict peak usage intervals and prefetch high-demand content to avoid congestion [35].

Fig. 2.2 illustrates an AI-enabled MEC-based network slicing architecture supporting diverse verticals, including smart cities, automotive systems, remote healthcare, volumetric media, and industrial XR use cases. Each vertical is served by dedicated and distributed MEC servers positioned close to the next-generation Node B (gNB) (next-generation Node B) units to enable ultra-low-latency and context-aware content delivery. In this architecture, MEC servers are not merely caching or forwarding units—they also host AI models and MADRL agents responsible for making real-time decisions related to resource allocation, media adaptation, and network orchestration. For instance, in healthcare and aviation, where latency and security are paramount, the MEC nodes dynamically adapt the content resolution and delivery paths. Similarly, in automotive and smart home environments, the MEC layer ensures seamless mobility and personalized media rendering. VV streaming, which requires high bandwidth and dynamic tiling, benefits from MEC-based predictive analytics, where AI models forecast user interaction patterns and prefetch relevant media segments accordingly. This distributed intelligence results in highly efficient and immersive service delivery across a wide range of network slicing scenarios in 5G and beyond.

To conclude, MEC-enabled architectures form a foundational pillar of 5G and future 6G networks by decentralizing computation and enabling intelligent, localized service delivery. Through the synergy of AI, MADRL, and network slicing, MEC not only meets the stringent QoS requirements of rich media content but also proactively enhances user QoE across diverse verticals. By embedding decision-making capabilities at the network edge, the architecture supports scalable, adaptive, and personalized delivery of AR/VR/XR and VV services. This design ultimately paves the way for context-aware, resilient, and high-performance immersive media ecosystems, making it indispensable for future network infrastructures.

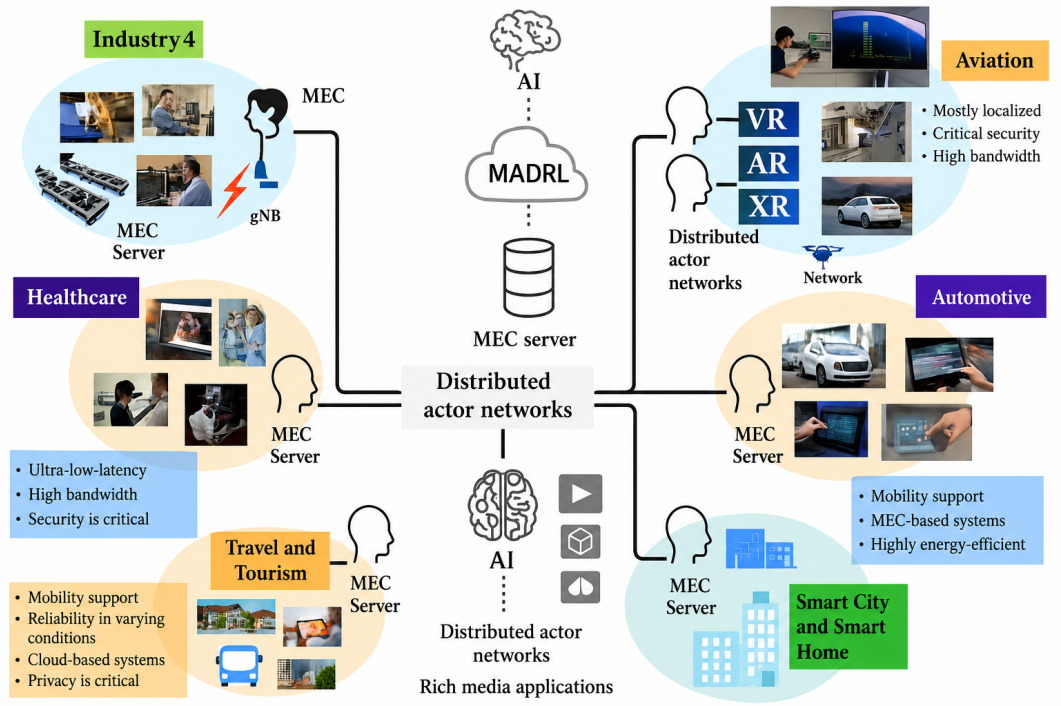


Figure 2.2: AI-Enabled MEC-Based Network Slicing Architecture for 5G and Beyond Networks. The MEC servers serve as local adaptation and decision-making nodes to support low-latency and high-reliability delivery for immersive media such as AR/VR/XR and VV across verticals including healthcare, automotive, smart cities, and tourism.

2.3 Perceptual Quality Metrics for Immersive Media

Delivering immersive media such as 360° video and volumetric content requires quality assessment mechanisms that go beyond traditional signal-

based metrics (e.g., PSNR or SSIM), which often fail to correlate with human perception in highly interactive and spatially complex environments. In this context, *perceptual quality metrics* play a critical role in guiding adaptive streaming decisions by explicitly modeling how users perceive visual quality under varying content, viewpoint, and network conditions.

2.3.1 VMAF for 360° Video

VMAF is a perceptual video quality metric developed by Netflix [2]. It uses ML to fuse multiple elementary metrics. It is currently the gold standard for high-resolution video streaming. VMAF is a perceptual video quality metric developed to reflect subjective human viewing experience better. VMAF combines multiple quality indicators, such as detail loss, structural distortion, and temporal artefacts, using machine learning models trained on large-scale subjective datasets. For 360° video, VMAF can be extended with *viewport-aware weighting*, where tiles within the user's FoV are assigned higher importance than marginal and background regions. This is particularly important for HMD-based viewing, where users only perceive a subset of the full spherical scene at any given time.

Within ABR streaming architectures, VMAF enables quality-driven adaptation by allowing bitrate decisions to be optimized directly against perceptual gain rather than raw bitrate or resolution. As a result, bandwidth can be preferentially allocated to visually important regions, improving perceived quality while maintaining network efficiency.

2.3.2 PCQM for Volumetric Point Clouds

PCQM is a *perceptual quality metric for 3D point clouds* that quantifies the visual quality of volumetric content by accounting for both geometry and color-related impairments. For VV, which is typically represented using point clouds or mesh-based 3D structures, conventional 2D video quality metrics are not applicable. PCQM has emerged as an effective perceptual

[2] https://github.com/Netflix/vmaf

metric for evaluating volumetric content by jointly considering geometric distortion, point density variation, and color fidelity [36]. PCQM models how spatial inconsistencies and surface degradation impact human perception during 6-DoF navigation.

PCQM is particularly well-suited for adaptive volumetric streaming, in which different LoDs or mesh resolutions are transmitted based on the user's viewpoint, distance, and interaction patterns. By quantifying perceptual degradation across spatial regions, PCQM enables intelligent bitrate and LoD selection that aligns transmission decisions with perceived visual importance.

Both VMAF and PCQM serve as critical feedback signals for perceptually aware ABR control. Instead of relying solely on network-centric indicators such as throughput and buffer occupancy, these metrics allow adaptive streaming systems to optimize decisions toward maximizing user-perceived QoE. In immersive media architectures, perceptual metrics are integrated into the reward formulation or utility models used by learning-based adaptation algorithms, ensuring that bitrate selection, region prioritization, and quality switching directly reflect visual impact. This perceptual grounding is essential for achieving efficient, scalable, and user-centric delivery of immersive media in 5G and beyond networks.

2.4 Viewport Prediction and Region-Based Bitrate Prioritization

In immersive media applications such as 360° video, VR, AR, XR, and VV, users are presented with a panoramic or fully navigable scene that far exceeds the instantaneous human field of view. At any given moment, only a limited spatial region referred to as the *viewport* is perceptually relevant to the user. Delivering uniformly high quality across the entire scene is therefore inefficient and often infeasible under bandwidth constraints. This observation has motivated the development of viewport-aware streaming

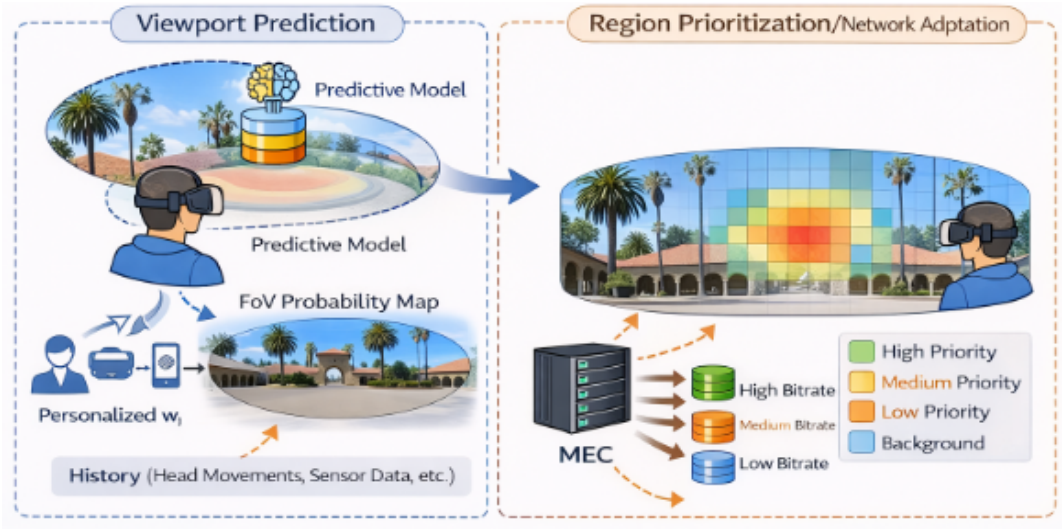


Figure 2.3: Viewport prediction and region-prioritized bitrate allocation with MEC-based streaming.

mechanisms that selectively allocate resources according to user attention [37, 38].

Conventional ABR algorithms operate on a segment-level abstraction and are largely agnostic to spatial perception, treating all regions of a video frame equally [39]. Such approaches are suboptimal for immersive content, where perceptual importance varies significantly across spatial regions. Viewport prediction combined with region-based bitrate prioritization addresses this limitation by aligning transmission decisions with predicted user focus, thereby improving perceptual efficiency while reducing unnecessary bandwidth consumption.

2.4.1 Viewport Prediction with FoV Probability Maps

Viewport prediction aims to estimate a user's future viewing direction over a short prediction horizon, enabling proactive adaptation of content quality. These models typically leverage temporal correlations in head orientation, eye gaze trajectories, and inertial sensor readings collected from head-mounted displays or client devices. Learning-based predictors may further exploit personalized viewing patterns to improve accuracy over time [40, 41].

As illustrated in Fig. 2.3, a user-specific prediction model $\mathcal{M}_i$ generates a Field-of-View (FoV) probability map $\mathcal{P}_i(\theta, \phi)$ over the spherical projection of the content. Here, (θ, ϕ) denote angular coordinates corresponding to longitude and latitude, and $\mathcal{P}_i(\theta, \phi)$ quantifies the likelihood that a given spatial location will be viewed by user i in the near future. Regions with higher probability mass correspond to anticipated gaze directions, while lower values indicate background areas with limited perceptual relevance.

FoV probability maps provide a continuous and interpretable representation of user attention, allowing adaptive streaming systems to move beyond binary viewport definitions. They also offer robustness against minor prediction errors by distributing importance across neighboring regions. In this thesis, the detailed modeling assumptions and learning mechanisms used for viewport prediction in both 360° and VV formats are presented in Chapters 4 and 5.

2.4.2 Region Prioritization and Bitrate Allocation via MEC

Once the FoV probability map is available, the immersive scene is spatially partitioned into tiles or regions, each assigned a priority level according to its predicted perceptual importance. As depicted in the right-hand side of Fig. 2.3, tiles are commonly grouped into three categories: *high priority* tiles corresponding to the predicted viewport, *medium priority* tiles covering the marginal region surrounding the viewport, and *low priority* tiles representing the background.

Each priority class is associated with a target bitrate level, denoted as $R^{(H)}$, $R^{(M)}$, and $R^{(L)}$, respectively. These bitrates determine the encoding resolution or fidelity allocated to each tile group. In edge-assisted architectures, this mapping is performed by a MEC-enabled server that has access to both user-level prediction information and real-time network state. By situating this decision-making process at the edge, latency is reduced, and adaptation becomes more responsive to rapid changes in user behavior or channel conditions.

This region-aware bitrate allocation strategy ensures that visual fidelity is maximized where it matters most, while conserving bandwidth in perceptually less significant areas. At the same time, proximity-based delivery from the MEC layer helps mitigate stalls and rebuffering events, which are particularly detrimental in immersive applications.

2.4.3 Mathematical Formulation

Let $B_i(t)$ denote the available bitrate budget for user i at time t, as estimated from throughput and buffer dynamics. The objective of region-based bitrate allocation is to maximize the expected perceptual utility subject to this budget constraint. This can be expressed as:

$$\max_{R_k} \sum_{k=1}^{K} \mathcal{P}_i(k)\, \mathcal{Q}(R_k) \quad \text{s.t.} \quad \sum_{k=1}^{K} R_k \leq B_i(t), \tag{2.1}$$

where k indexes the set of spatial tiles, $\mathcal{P}_i(k)$ denotes the viewport probability associated with tile k, and $\mathcal{Q}(R_k)$ represents a perceptual quality function evaluated at bitrate R_k. Depending on the media format, $\mathcal{Q}(\cdot)$ may correspond to metrics such as VMAF for 360° video or PCQM for volumetric point cloud content.

This formulation explicitly couples perceptual relevance with bitrate selection, enabling adaptive streaming policies that are aligned with human visual attention. Rather than maximizing raw bitrate or minimizing rebuffering alone, the system optimizes a weighted perceptual objective that reflects the user's expected viewing experience [24, 27].

The joint use of viewport prediction and region-prioritized bitrate adaptation has been shown to significantly reduce transmission overhead while preserving or even enhancing user-perceived quality. Nevertheless, several challenges persist, including the accurate prediction of rapid head movements, the graceful handling of prediction errors, and scalable coordination across multiple users sharing edge resources. The algorithmic strate-

gies employed to address these challenges in the proposed frameworks are detailed in the subsequent chapters of this thesis.

2.5 Fairness in Adaptive Streaming Systems

In multi-user adaptive streaming environments, fairness is a fundamental design objective that complements throughput efficiency and perceptual quality optimization. As immersive media services such as 360° video and VV increasingly rely on shared wireless and edge resources, ABR decisions made for one client can directly impact the QoE of other users. Consequently, adaptive streaming systems must ensure that resource allocation mechanisms do not systematically favor a subset of users while degrading the experience of others, particularly under bandwidth-constrained or highly dynamic network conditions.

Fairness in adaptive streaming generally refers to the equitable distribution of network and computation resources such that all active clients achieve a comparable level of perceived quality over time. In practice, unfairness may manifest as persistent bitrate starvation, excessive rebuffering, or frequent quality oscillations for certain users, especially those with weaker channel conditions or disadvantaged spatial locations. These effects are amplified in immersive media delivery, where viewport-dependent quality, tile-level adaptation, and heterogeneous device capabilities introduce additional complexity beyond conventional 2D video streaming.

Several notions of fairness have been explored in the context of networked multimedia systems. One widely adopted quantitative measure is Jain's Fairness Index, which evaluates how evenly a given performance metric, such as bitrate allocation or QoE, is distributed among users. For a system serving N clients with QoE values $\{x_1, x_2, \ldots, x_N\}$, the index is defined as:

$$J = \frac{\left(\sum_{i=1}^{N} x_i\right)^2}{N \sum_{i=1}^{N} x_i^2}. \tag{2.2}$$

A value of J close to unity indicates a high degree of fairness, whereas lower values reveal increasing disparity among users. While this metric provides a concise numerical assessment, it does not capture temporal variations or perceptual relevance, which are critical in immersive streaming scenarios [42].

Beyond static fairness, adaptive streaming systems must also address temporal fairness, ensuring that each user experiences stable quality over time without abrupt degradations. This requirement is particularly important for virtual and augmented reality applications, where sudden drops in visual quality or prolonged buffering events can significantly disrupt user immersion. In region-aware streaming systems, fairness must further account for spatial heterogeneity, as different regions of the media content (e.g., viewport versus background) carry different perceptual importance. Without appropriate safeguards, aggressive prioritization of high-importance regions for some users may inadvertently reduce the overall fairness of the system.

The emergence of edge-assisted and learning-based streaming architectures has enabled more sophisticated fairness-aware control mechanisms. In such systems, fairness can be incorporated directly into the decision-making process by coupling perceptual quality metrics with global system observations. For instance, centralized and semi-centralized controllers deployed at the MEC layer can observe aggregate network conditions and user-level performance indicators, allowing them to moderate individual adaptation decisions to maintain balanced QoE across clients. This approach is particularly relevant in multi-user environments where independent client-driven ABR algorithms may otherwise behave selfishly [43].

RL-based adaptive streaming frameworks further extend fairness modeling by embedding it within the optimization objective. By augmenting the reward function with fairness-sensitive terms, learning agents can be encouraged to maximize long-term collective QoE rather than instantaneous individual gains. This formulation enables adaptive policies that dynam-

ically trade off efficiency and equity, leading to more stable and socially optimal outcomes under fluctuating network conditions [44].

Overall, fairness-aware adaptive streaming is a critical component of scalable immersive media delivery in 5G and beyond networks. By jointly considering perceptual quality, temporal stability, and multi-user resource sharing, fairness mechanisms ensure that advanced adaptive streaming architectures deliver not only high average performance but also consistent and equitable user experiences. This foundational concept underpins the design of modern AI-driven streaming systems and motivates the integration of fairness considerations in subsequent algorithmic and architectural developments presented in this thesis.

2.6 AI-Assisted Optimization Techniques for 5G and Beyond Networks

To support scalable and intelligent adaptation for rich media content, this thesis leverages a range of AI paradigms:

2.6.1 Reinforcement Learning

RL enables an agent to learn optimal decision-making by interacting with an environment to maximize cumulative rewards over time. RL methods are broadly classified as *model-based* and *model-free* approaches [45], as shown in Fig. 2.4. Model-based methods rely on accurate environment modeling and utilize Markov Decision Process (MDP) for planning through dynamic programming:

$$V(s) = \sum_{s',r'} p(s', r|s, \pi(s))[r + \gamma V(s')] \tag{2.3}$$

where $V(s)$ is the expected return from state s under policy π, and γ is the discount factor. The corresponding policy improvement step is:

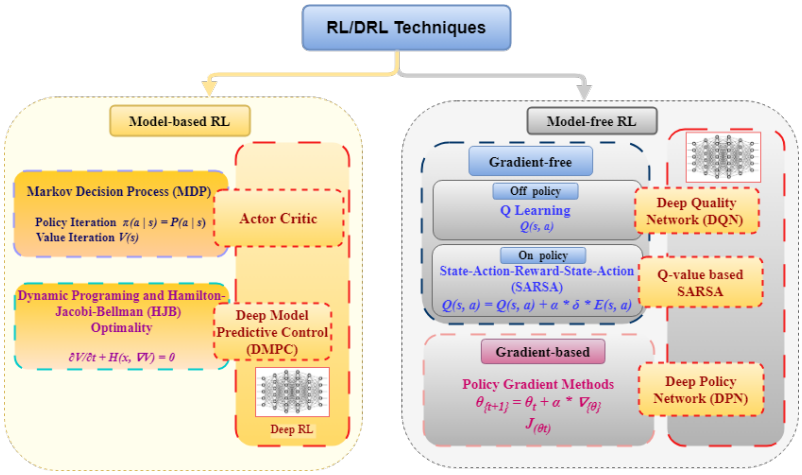


Figure 2.4: RL/DRL Techniques: Model-based vs. Model-free [3]

$$\pi(s) = \arg\max_{a} \sum_{s',r'} p(s', r|s, a)[r + \gamma V(s')] \tag{2.4}$$

Model-free RL, such as Q-learning, bypasses environment modeling and directly learns value functions. It is particularly suited for complex, high-dimensional problems but suffers from sample inefficiency and instability. The Q-learning update is given by:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right] \tag{2.5}$$

Here, $Q(s, a)$ denotes the expected reward of taking action a in state s, with updates guided by observed reward r, learning rate α, and future Q-values [46, 47].

2.6.2 Deep Reinforcement Learning

The heterogeneous services and immersive applications demanded by rich media consumers within 5G and 6G MEC-enabled SDN/NFV-based network slicing architectures require intelligent and adaptive control strategies. Among various AI-based models, DRL techniques have emerged as the most promising approaches due to their robustness, scalability, generalization ability, transparency, and security. These models support *multi-objective optimization* and MADRL-based learning, making them ideal for resource

allocation, caching, and scheduling tasks in dynamic and heterogeneous network environments, as shown in Fig. 2.5.

Unlike traditional RL, DRL algorithms can autonomously learn abstract representations of complex state spaces, manage high-dimensional and continuous action domains, and dynamically adapt policies based on QoE-driven metrics for personalized media delivery. For example, COMA-based actor-critic architectures and hybrid MADRL strategies have been applied to optimize ABR decisions in tile-based 360° VR streaming and VV rendering.

Hybrid DRL combines discrete and continuous state-action representations. Algorithms like H-PPO[48], H-MPO[49], MP-DQN[50], and Prioritized Experience Replay Actor-Critic Deep Deterministic Policy Gradient (PA-DDPG) leverage the strengths of both paradigms, optimizing user experience in adaptive streaming and fine-grained QoS resource allocation. They reduce dimensionality, enable better decision interpretation, and offer dynamic policy adaptation to real-time network and user context variations.

In MEC-enhanced network slicing, MADRL agents can be deployed at multiple network tiers:

- **MEC Tier:** Agents can adapt media chunk bitrates based on real-time network feedback and QoE predictions.

- **Core Tier:** Orchestrators and controllers utilize DRL-based policies to optimize traffic routing, Virtualized Network Functions (VNF) placement, inter-slice coordination, and MEC resource and cache management.

Efficient communication among these agents is enabled through RESTful APIs and interfaces such as E2AP in Open Radio Access Network (Open-RAN). Hybrid models like PA-DDPG and multi-agent Deep Deterministic Policy Gradient (DDPG) rely on such infrastructures to share environment feedback and optimize policies collaboratively.

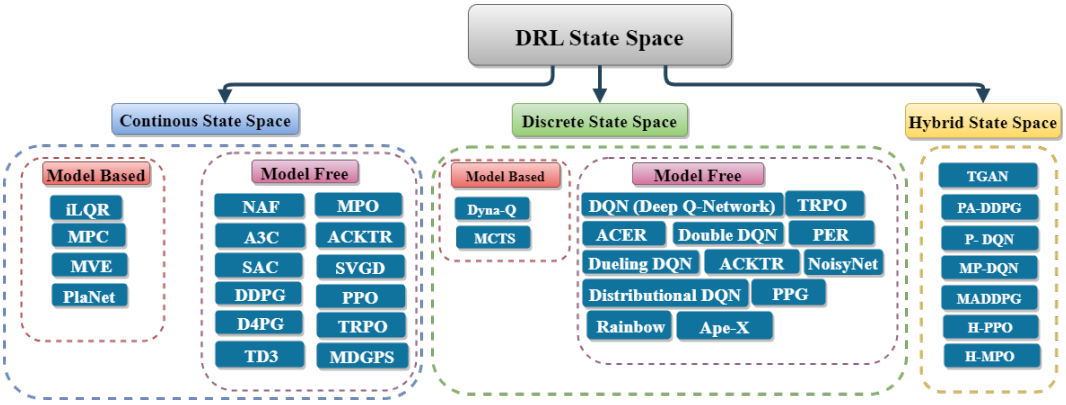


Figure 2.5: DRL techniques for handling continuous, discrete, and hybrid state-action spaces [3]

2.6.3 Counterfactual Multi-Agent Policy Gradients (COMA)

COMA [18] is a multi-agent actor-critic method designed for settings in which multiple agents must coordinate under a shared objective while learning *decentralized* policies. A common training paradigm used by COMA is *centralized training with decentralized execution (CTDE)*: each agent executes its own policy based on local observations, but training leverages a centralized critic that can condition on global state information and the *joint action* of all agents.

Let s_t denote the global state at time t, o_t^i the local observation of agent i, and $\mathbf{u}_t = (u_t^1, \ldots, u_t^n)$ the joint action. Each agent has a stochastic actor policy $\pi_{\theta_i}(u_t^i \mid o_t^i)$, while the centralized critic estimates the joint action-value function $Q_\phi(s_t, \mathbf{u}_t)$. A key challenge in cooperative multi-agent learning is *credit assignment*: determining how each agent's action contributed to a global outcome. COMA addresses this using a *counterfactual baseline* that marginalizes over agent i's actions while keeping other agents' actions fixed.

The counterfactual baseline for agent i is:

$$b_i(s_t, \mathbf{u}_t^{-i}) = \sum_{u'^i} \pi_{\theta_i}(u'^i \mid o_t^i)\, Q_\phi\big(s_t, (u'^i, \mathbf{u}_t^{-i})\big)\,, \tag{2.6}$$

where $\mathbf{u}_t^{-i}$ denotes the actions of all agents except i. The corresponding counterfactual advantage is:

$$A_i(s_t, \mathbf{u}_t) = Q_\phi(s_t, \mathbf{u}_t) - b_i(s_t, \mathbf{u}_t^{-i}). \tag{2.7}$$

Using this advantage, each actor is updated via a policy-gradient objective:

$$\nabla_{\theta_i} J(\theta_i) = \mathbb{E}\Big[\nabla_{\theta_i} \log \pi_{\theta_i}(u_t^i \mid o_t^i)\, A_i(s_t, \mathbf{u}_t)\Big]. \tag{2.8}$$

Intuitively, COMA increases the probability of agent actions that improve the joint action-value beyond what would be expected if the agent sampled alternative actions from its current policy, thus encouraging coordinated behavior and reducing misattribution of global rewards.

2.6.4 Meta-Reinforcement Learning for Adaptive Streaming

Meta-RL refers to a class of learning algorithms that aim to optimize the capacity of an agent to adapt rapidly to new tasks and environments employing limited interaction data [51]. Rather than learning a fixed policy for a specific environment, Meta-RL methods seek to learn a learning strategy effectively, how to learn which enables generalization across a distribution of tasks. This capability is particularly valuable in dynamic and heterogeneous networked environments, such as those supporting immersive media streaming, where real-time adaptation to unseen content and network conditions is critical.

Meta-RL extends conventional reinforcement learning by enabling agents to rapidly adapt their decision-making policies to new tasks or environments using limited interaction data. Instead of optimizing a policy for a single stationary MDP, Meta-RL considers a *distribution of tasks*, where each task corresponds to a distinct MDP sampled from a task distribution $\mathcal{P}(\mathcal{T})$. The primary objective is to learn a meta-policy that can efficiently generalize across tasks and adapt to unseen conditions with minimal additional experience.

Formally, each task $\mathcal{T}_i$ is defined by an MDP:

$$\mathcal{T}_i = \langle \mathcal{S}, \mathcal{A}, p_i(s'|s,a), r_i(s,a), \gamma \rangle, \tag{2.9}$$

where $\mathcal{S}$ and $\mathcal{A}$ denote the shared state and action spaces, p_i and r_i represent task-specific transition and reward functions, and $\gamma \in (0,1)$ is the discount factor. The goal of Meta-RL is to learn a policy π_θ that maximizes the expected return across the task distribution:

$$\max_\theta \ \mathbb{E}_{\mathcal{T}_i \sim \mathcal{P}(\mathcal{T})} \left[\mathbb{E}_{\tau \sim \pi_\theta^{\mathcal{T}_i}} \left[\sum_{t=0}^{T} \gamma^t r_i(s_t, a_t) \right] \right], \tag{2.10}$$

where τ denotes a trajectory sampled under task $\mathcal{T}_i$.

2.6.5 Categories of Meta-Reinforcement Learning

Existing Meta-RL approaches can be broadly categorized into two major paradigms: *gradient-based meta-learning* and *context-based (latent-variable) meta-learning* [52].

2.6.5.1 Gradient-Based Meta-Reinforcement Learning

Gradient-based Meta-RL methods aim to learn policy parameters that can be rapidly fine-tuned to a new task using a small number of gradient updates. A representative example is Model-Agnostic Meta-Learning (MAML), where the meta-objective is formulated as:

$$\min_\theta \sum_{\mathcal{T}_i \sim \mathcal{P}(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i} \left(\theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(\theta) \right), \tag{2.11}$$

with α denoting the inner-loop learning rate. While effective, such approaches require differentiable environments and can suffer from high computational overhead due to second-order gradient computations, limiting their applicability in real-time systems.

2.6.5.2 Context-Based Meta-Reinforcement Learning

Context-based Meta-RL methods address these limitations by inferring a latent task representation directly from observed experience, avoiding explicit gradient-based adaptation. In this paradigm, a latent variable z is introduced to encode task-specific information:

$$z \sim q_\phi(z \mid \mathcal{D}), \tag{2.12}$$

where $\mathcal{D} = \{(s_t, a_t, r_t, s_{t+1})\}$ represents a set of recent transition tuples, and q_ϕ is a learned inference model. The policy is then conditioned on this latent context:

$$\pi_\theta(a_t \mid s_t, z). \tag{2.13}$$

This formulation enables the agent to adapt its behavior by updating the inferred context z rather than directly modifying policy parameters, resulting in faster and more stable adaptation [53].

2.6.6 Probabilistic Embeddings for Actor-Critic Reinforcement Learning (PEARL)

PEARL is a prominent context-based Meta-RL framework that models task uncertainty through probabilistic latent variables [54]. PEARL employs an encoder network that infers a posterior distribution over latent contexts:

$$q_\phi(z \mid \mathcal{D}) = \mathcal{N}(\mu_\phi(\mathcal{D}), \sigma^2_\phi(\mathcal{D})), \tag{2.14}$$

where μ_ϕ and σ_ϕ are learned functions of the experience buffer $\mathcal{D}$. The latent variable z conditions both the actor $\pi_\theta(a|s, z)$ and critic $Q_\psi(s, a, z)$ networks within an off-policy actor–critic framework.

The learning objective combines the expected return and a regularization term that encourages informative yet compact latent representations:

$$\mathcal{L} = \mathbb{E}_{z\sim q_\phi}\left[\mathcal{L}_{\text{RL}}(\theta, \psi; z)\right] + \beta\,\mathrm{KL}\left(q_\phi(z \mid \mathcal{D}) \parallel p(z)\right), \tag{2.15}$$

where $p(z)$ is a prior distribution and β controls the strength of the regularization. This probabilistic treatment enables structured exploration and robust generalization across diverse tasks.

2.6.7 Applications of Meta-Reinforcement Learning

Meta-RL has been successfully applied in robotics, autonomous navigation, recommendation systems, and networked systems, where rapid adaptation to non-stationary environments is essential. In communication and multimedia systems, Meta-RL is particularly attractive due to its ability to generalize across heterogeneous network conditions, user behaviors, and content characteristics. By learning task-invariant priors and inferring task-specific context online, Meta-RL frameworks offer a principled approach to real-time decision-making in dynamic environments, making them well-suited for next-generation adaptive systems [55, 56].

2.7 Volumetric Video Streaming for Immersive Applications

VV is an emerging media format that captures three-dimensional (3D) content, providing users with immersive six degrees of freedom (6-DoF) experiences in AR, VR, and XR environments [57]. Unlike traditional 2D or 360° video, volumetric content is rendered as dense point clouds or triangle-based mesh representations, enabling users to navigate scenes freely from any viewpoint. These rich spatial experiences, however, pose significant challenges for content transmission due to the high spatial complexity, large file sizes, and real-time rendering requirements.

To construct such immersive representations, VVS rely on the accurate capture of 3D geometry, typically represented as navigable mesh grids [58]. One of the most effective techniques for acquiring these mesh grid points is Laser Detection and Ranging (LADAR), commonly referred to as LiDAR. LiDAR sensors work by emitting laser pulses and measuring the time it takes

for each pulse to reflect off surfaces and return to the sensor, thereby generating dense 3D point clouds. These point clouds are then processed into mesh structures suitable for compression and streaming using standards like V-PCC (Video-based Point Cloud Compression) and MPEG-I [59].

In addition to LiDAR, other depth-sensing technologies, including structured light sensors (e.g., Microsoft Kinect), time-of-flight cameras, and multi-view stereo camera arrays, are also utilized to enhance geometric accuracy and surface detail. These high-resolution sensing systems are crucial for supporting the 6-DoF experience, enabling users to interact naturally within immersive environments. The resulting detailed 3D data enables advanced streaming mechanisms, including adaptive mesh level selection and perceptual quality optimization, which are critical for delivering efficient and high-quality volumetric media experiences across bandwidth-constrained networks.

One of the primary challenges in VVS is the massive data size and high bandwidth requirement associated with transmitting dense 3D mesh content in real time. To address these challenges, client-based rate adaptation strategies have been developed, wherein the user device makes bitrate selection decisions based on viewport prediction, mesh complexity, and playback conditions. Fig. 2.6 illustrates a client-driven adaptive volumetric streaming architecture, where mesh data is transmitted from a server and processed locally for rendering. The client predicts the user's viewport and selects a mesh level (e.g., Level 1, Level 2, Level 3) according to the user's field of view and motion patterns. Each mesh level corresponds to different resolutions and triangle densities, allowing for efficient adaptation based on available bandwidth.

A key innovation in this pipeline is the integration of meta-learning for rate adaptation. Instead of relying solely on static rules or traditional ABR heuristics, a meta-learner is employed to learn and generalize across diverse streaming environments. This allows the client to rapidly adapt mesh-level decisions using real-time features such as network throughput, buffer sta-

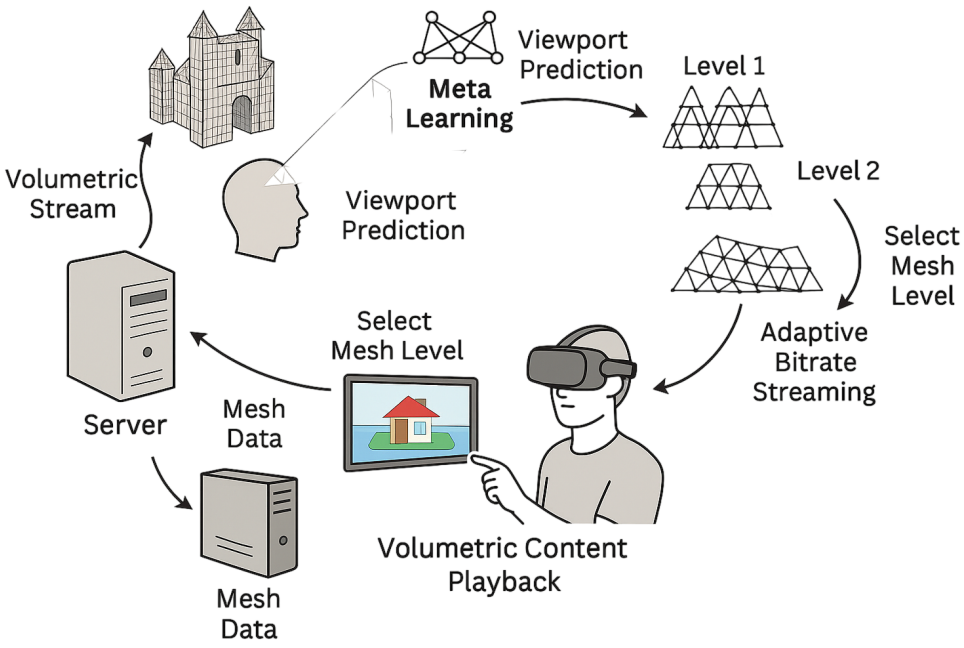


Figure 2.6: Client-based adaptive volumetric video streaming architecture. A meta-learning model dynamically selects mesh levels for bitrate-efficient playback based on viewport prediction and 3D content complexity.

tus, and spatial complexity of the 3D scene. The meta-learning component effectively tunes the bitrate selection policy in a personalized and context-aware manner, making it more robust to dynamic changes in user attention and device capabilities. Meta-reinforcement learning approaches are being explored to dynamically tune policies for such high-dimensional content [60], making this an exciting future direction.

In summary, VVS benefits significantly from combining client-side intelligence with meta-learning-based adaptation. This approach ensures optimal visual quality within constrained bandwidth environments by focusing bitrate allocation on user-visible regions while maintaining low latency and high responsiveness. As immersive media continues to evolve, such intelligent frameworks are critical for scalable and personalized 3D content delivery in 5G and beyond networks. Hybrid DRL, paired with meta-learning strategies, offers potential for real-time policy tuning. A MADRL and Meta-Learning ABR framework can predict viewport-aware bandwidth needs for

360° Video and volumetric tiles and update streaming strategies based on evolving user attention and network load.

2.8 Insights from Surveyed Literature

The key concepts above are deeply aligned with findings from the author's published survey [3], which reviewed AI-enabled rich media delivery frameworks within 5G and beyond environments. The survey revealed:

- Traditional client-centric strategies cannot cope with emerging content types or collaborative streaming scenarios.
- Intelligent edge-assisted adaptations are crucial for meeting the low-latency, high-fidelity demands of XR and volumetric media.
- There is a gap in unified frameworks that jointly optimize QoE, fairness, and edge resource utilization.

2.9 Chapter Summary

This chapter has presented a comprehensive overview of the technical foundations underpinning immersive media streaming in 5G and beyond network environments. It introduced the pivotal role of MEC in enabling ultra-low latency, edge-assisted processing, and intelligent rate adaptation critical for emerging applications such as VR, AR, XR, and VV delivery. The chapter detailed the architectural and streaming challenges associated with next-generation content formats, particularly *360° video* and *volumetric point clouds*, focusing on their encoding, tiling, and delivery characteristics. It also discussed the integration of perceptual quality metrics such as VMAF and PCQM, which offer objective yet human-aligned assessments of visual fidelity, thus guiding adaptive bitrate decisions.

Furthermore, the chapter introduced MADRL and *meta-reinforcement learning* as key enablers for personalized, context-aware adaptation strategies. Techniques like viewport prediction, region prioritization, and proba-

bilistic embeddings were explored to support user-centric, QoE-driven streaming. Collectively, these enabling technologies form the conceptual basis for the advanced immersive streaming architectures proposed in the following chapters, offering a robust and scalable foundation for real-time, perceptual-aware multimedia delivery in heterogeneous network settings.

Chapter 3

Related Works

In recent years, the delivery of rich media content such as 360° VR/AR/XR and VV experiences over 5G and beyond networks has become increasingly complex due to diverse application demands, user mobility, and stringent QoS/QoE requirements. This chapter reviews the most relevant research directions that underpin this domain and, crucially, highlights the shortcomings and limitations of existing approaches at the intersection of intelligent network slicing, MEC, CDN, and adaptive streaming. It begins by examining AI-assisted network slice orchestration and service differentiation in heterogeneous environments, while emphasizing persistent limitations such as slow responsiveness to rapid traffic dynamics, limited cross-layer coordination, and inadequate support for immersive traffic bursts. It then surveys MEC-enabled CDN architectures for DASH-based immersive streaming (e.g., tiling, caching, and quality estimation), noting common gaps including imperfect viewport awareness, overheads due to fine-grained tiling/packaging, and suboptimal fairness under multi-user contention. The chapter also reviews recent work on VV delivery, including DASH-compliant segmentation, learning-based optimization of bitrate and geometry levels, and generalization-oriented adaptation; however, many existing solutions still struggle with robust perceptual modeling (e.g., geometry-aware metrics), partial observability of user behavior, and fast adaptation across unseen network and content conditions. Overall, the identified limitations motivate the need for innovative AI-driven,

user-perception-aware, and scalable adaptation solutions, which are introduced in this thesis.

Relationship to published work: Parts of the literature review and taxonomy presented in this chapter build upon the author's published survey article: F. E. Subhan, A. Yaqoob, C. H. Muntean, and G.-M. Muntean, "A Survey on Artificial Intelligence Techniques for Improved Rich Media Content Delivery in a 5G and Beyond Network Slicing Context," *IEEE Communications Surveys & Tutorials*, vol. 27, no. 2, pp. 1427–1487, April 2025. The published material has been reorganised, updated, expanded, and integrated into the broader technical context of this thesis.

3.1 Existing Surveys and Tutorials on AI-Enabled Network Slicing

The growing demand for intelligent network management in 5G and beyond networks has led to a surge in the literature addressing the role of AI in network slicing and orchestration. Barakabitze et al. [61] presented an extensive survey on 5G network slicing based on SDN and NFV, including orchestration strategies across multiple domains. Although the authors outlined major industry efforts and architectural components; the survey lacked focus on AI-based adaptive mechanisms for immersive content delivery. In a related work, Barakabitze et al. [62] categorized QoE management into SDN/NFV, MEC/ICN/cloud, and service-level-based strategies for rich media applications. However, neither survey addressed DASH-based streaming or advanced ML solutions for perceptual quality-aware adaptation.

Dangi et al. [63] reviewed ML-based slice lifecycle security but did not assess the impact of these algorithms on performance metrics such as QoS, latency, or throughput. Similarly, Ssengonzi et al. [64] investigated Deep RL in 5G slicing but overlooked commercial media delivery use cases. Yaqoob et al. [15] addressed DASH-based 360° streaming, highlighting spatial tiling

and network-assisted edge delivery, but did not explore ML-based optimization within network slices.

Phyu et al. [65] provided a comprehensive view of ML for RAN and core slice resource allocation, while Nauman et al. [66] examined QoS and QoE aspects for multimedia IoT. Neither explored DASH integration nor immersive content in AI-sliced environments. Spinelli et al. [67] focused on MEC standardization but did not compare intelligent ICN/CDN-assisted delivery techniques for multimedia.

Shen et al. [68] discussed AI-enabled RAN slicing, user association, and edge caching, yet lacked insight into SDN/NFV-based content-aware services. Agarwal et al. [13] emphasized resource management for 5G HetNets but missed out on integrating AI-driven slicing for high-throughput rich content. Cao et al. [69] tackled slice-level security in SDN/NFV architectures but did not account for multimedia traffic patterns or ML-enhanced content protection.

Wyszkowski et al. [70] contributed a vendor-agnostic design-time slicing model but ignored streaming-specific use cases or AI-supported provisioning. Coronado et al. [71] outlined Zero-Touch Management (ZTM) using AI for slice automation, although without consideration of multimedia demands. Similarly, Santos et al. [72] focused on MEC for ultra-low-latency services (e.g., VR, AV) but lacked discussion on high-throughput streaming or DASH frameworks.

Shen et al. [73] introduced a 6G architecture integrating AI, digital twins, and virtualization, yet concentrated only on vehicular networks. Lee et al. [74] proposed a federated learning-based network orchestration framework but did not consider DRL or rich media-specific scenarios. Finally, Ebrahimi et al. [75] evaluated E2E slice resource management but overlooked MEC/CDN-based delivery for immersive multimedia.

As summarized in Table 3.1, while many surveys address aspects of AI-enabled network slicing, the authors either overlook the role of DASH streaming or limit their scope to specific optimization problems or vertical

use cases. For example, [61, 62, 69, 70] do not integrate AI with adaptive rich media delivery, and [15] lacks slicing integration. Others like [13, 63–65] are narrow in focus, while [71–75] explore broader architectures without grounding in content-centric streaming challenges. This thesis bridges these gaps by proposing AI-enabled, DASH-based network slicing for rich media delivery. It focuses on joint QoE optimization, intelligent resource management, and perceptual-quality-driven adaptation. The following sections review research leveraging SDN, NFV, MEC, and CDN to support immersive, quality-oriented streaming in 5G and beyond networks.

Table 3.1: Summary of related surveys and tutorials [3].

Ref.	**Main contribution**	**AI techniques**	**Rich media / DASH**	**Scope and limitations**
[61]	Management and orchestration of multi-domain network slices.	AI-enabled network slicing is not considered.	×	Covers 5G slicing based on SDN, NFV, MEC, cloud computing, and virtual containers.
[62]	QoE management for immersive AR, VR, mulsemedia, and gaming applications.	AI-enabled network slicing is not considered.	✓	Reviews SDN/NFV-based slicing for QoE-aware DASH delivery using MEC, ICN, and cloud computing.
[63]	Network-slice lifecycle management and security.	ML-based slice management and security.	×	Focuses on SDN/NFV-enabled slicing for security and lifecycle management.
[64]	ML applications in network slicing for 5G and beyond.	ML, RL, and DRL for slice lifecycle management and operation.	×	Reviews SDN-, NFV-, and MEC-assisted slice management and operation.
[15]	End-to-end adaptive delivery of 360° video.	Partial coverage of RL- and DRL-based adaptation.	✓	Reviews adaptive 360° video delivery but does not address network slicing.
[65]	Traffic forecasting, admission control, and resource allocation in network slicing.	ML-based slice control and resource management.	×	Covers SDN-, NFV-, and MEC-assisted traffic and resource management.
[66]	Network-slicing technologies for Multimedia Internet of Things applications.	Partial ML-based M-IoT resource management.	×	Focuses on SDN-, NFV-, and ICN-enabled M-IoT applications.

Continued on the next page

Table 3.1 – *continued from the previous page*

Ref.	Main contribution	AI techniques	Rich media / DASH	Scope and limitations
[67]	MEC standardisation and resource provisioning for VR, AR, and e-health.	ML-based resource management for selected use cases.	×	Reviews SDN- and MEC-assisted application-level resource management.
[68]	AI-enabled RAN slicing, RAT selection, edge caching, and content delivery.	AI-based RAN slicing, RAT selection, and user association.	×	Addresses SDN/MEC-assisted caching, content delivery, and RAN resource management.
[13]	Interference, power control, and user association in 5G heterogeneous networks.	RL- and DRL-based RAT selection, power control, and user association.	×	Does not comprehensively examine network slicing for RAT, power, and user-association management.
[69]	Security and privacy in 3GPP-compliant 5G network slicing.	AI-enabled network slicing is not considered.	×	Reviews SDN/NFV-based slice security for applications including D2D communication and IoT.
[70]	Vendor-agnostic and standards-aligned design of network slices.	AI-enabled network slicing is not considered.	×	Focuses on design-time aspects of network slicing for 5G and beyond networks.
[71]	AI-assisted network slicing and zero-touch network management.	DL, RL, DRL, and FL.	×	Reviews zero-touch management for 5G/6G networks, with emphasis on slicing and AI integration.
[72]	MEC-based slicing for low-latency VR, XR, and autonomous-driving applications.	DL, RL, DRL, and FL.	×	Examines AI integration in MEC-assisted slicing for emerging low-latency services.
[73]	6G architecture based on virtualisation, network slicing, and AI.	Digital twins, AI-enabled slices, and data-driven methods.	×	Proposes AI-assisted 6G network management and slice orchestration.
[74]	Federated-learning-enabled mobile-network and slice management.	FL and partial DRL coverage.	×	Focuses on FL-assisted mobile-network management and network slicing.
[75]	Cross-domain and end-to-end network-slice resource management.	Partial coverage of DL, RL, and DRL.	×	Reviews resource-management challenges in cross-domain and end-to-end 5G/6G slicing.

Continued on the next page

Table 3.1 – *continued from the previous page*

Ref.	Main contribution	AI techniques	Rich media / DASH	Scope and limitations
This thesis	AI-enabled network slicing for improved rich-media delivery.	DL, RL, DRL, and FL for intelligent content delivery.	✓	Examines how AI and slicing-enabling technologies jointly improve adaptive rich-media delivery.

3.2 MEC-Enabled 5G and Beyond Network Slicing for Rich Media Applications

Integrating AI with MEC-enabled network slicing presents substantial advantages for rich media delivery by pushing compute and storage resources closer to end users. MEC enables latency-sensitive processing tasks such as video prefetching, caching, encoding, and adaptation to be executed at the network edge. When empowered with ML, MEC can predict content demand, classify traffic, and allocate resources in a manner that ensures seamless adaptive streaming and enhanced user QoE [76].

Behravesh et al. [77] applied Random Forest and Inductive Logic Programming (ILP) techniques at the MEC node for predictive caching of DASH segments, reducing access delay and backhaul utilization. Similarly, Uriol et al. [78] compared ML-based classifiers for segment bitrate prefetching, with Random Forest yielding high CHR and bandwidth savings. However, these approaches are primarily optimized for 2D content delivery and do not account for the complex spatiotemporal requirements of volumetric or 360° video content.

Siriwardhana et al. [79] provided a comprehensive review of MEC-enabled mobile AR systems and technical network slice challenges. Yang et al. [80] utilized Graph Neural Networks (GNNs) for collaborative cloud-edge VR caching, improving Cache Hit Rate (CHR) and QoE. Domeke et al. [81] proposed RL-based orchestration among SDN controllers for edge slicing, while Kao et al. [82] used regression-based QoE prediction to enhance MEC slice

planning. These solutions, although innovative, lack mechanisms to adapt to rapidly changing user viewport patterns in immersive streaming sessions.

Several studies leverage DASH-enabled segment prefetching and ML for low-latency multimedia delivery. Armijo et al. [83] adopted segment transcoding and popularity-based caching at the MEC using Least Popular Used (LPU) strategy, outperforming Least Frequently Used (LFU) and Least Recently Used (LRU) in CHR and QoE metrics. Other DASH-based ML caching systems [84–89] optimized ABR and resource allocation. However, these works often ignore viewport quality variations across 360° videos, and do not incorporate cross-user adaptation strategies to maximize fairness and consistency.

To address this, Xiao et al. [90] proposed a MADRL-enabled framework that jointly considers 360° VR transcoding, caching, and streaming, improving CHR and delay. Zeng et al. [91] further modeled FoV-aware bitrate-caching optimization for multi-category videos using Multi-agent Soft Actor-Critic, showing substantial QoE gains. Nonetheless, these works still focus primarily on individual user optimization and do not fully consider client diversity or fairness in resource allocation.

EdgeDRL-based tile-level solutions also show promise. Chen et al. [92] proposed edge-assisted multi-user bitrate adaptation, while Shi et al. [93] applied actor-critic MADRL for cooperative multicasting across multi-cloud MECs. The latter outperformed Advantage Actor-Critic (A2C), LRU, LFU, and First in First out (FIFO) caching on the large dataset for multi-modal person identification (iQIYI) [94]. However, many of these frameworks were evaluated in simplified settings and do not consider long-term fairness or video perceptual quality as optimization targets.

AI-aided offloading mechanisms are essential for reducing complexity and energy consumption. Pan et al. [95] presented a MEC-based offloading framework using mixed-integer nonlinear programming and Alternating Direction Method of Multipliers (ADMM) [96] to balance inference load, latency, and recognition accuracy for AR. Similarly, Xu et al. [97] applied Deep Deter-

ministic Policy Gradient (DDPG) [98] to optimize VR computation offloading, while Jia et al. [99] combined Reconfigurable Intelligent Surface (RIS) and MADRL for VR streaming optimization, improving association, power, and phase-shift controls. However, these systems still lack fully decentralized, self-adaptive coordination among edge clients in real time.

Resource allocation using Proximal Policy Optimization (PPO) [100] and DRL for hierarchical slicing was explored by Kim et al. [101], while Wang et al. [102, 103] modeled Mobile Virtual Network Operator (MANO) utility maximization in MEC slicing using DDPG and twin-actor DRL. These works showed significant utility and performance improvements for multimedia slice orchestration. Yet, the authors did not dynamically adjust adaptation strategies based on real-time perceptual feedback (e.g., VMAF) or inter-client QoE comparisons.

For VR slicing, Liu et al. [104] and Du et al. [105] used Asynchronous Advantage Actor-Critic (A3C) [106] and decoupled DRL strategies to optimize energy, latency, and viewpoint rendering. Their results validated long-term QoE and energy savings in MEC-driven VR networks. Despite this, their architectures were limited to single-user environments and were not evaluated for highly concurrent multi-client streaming sessions.

3.2.1 Edge Intelligence for Media-Aware Adaptation

MEC-assisted techniques are particularly significant for 360° video streaming, as scalability is needed to handle large bandwidth requirements and multiple users engaging with the dynamic network. By utilizing network-based strategies, these approaches improve user monitoring and involvement while guaranteeing scalable and effective 360° video content delivery. Suzhi Bi et al. [107] introduced a two-stage ABR strategy that improves client QoE by employing FoV tiles prediction and multimedia pre-caching for 360° video streaming on MEC networks. The technique utilizes a double-agent DRL architecture to dynamically modify bitrates in real time to reduce wireless fading of channels and FoV tiles' incorrect predictions, resulting

in reliable video playback. The authors of [108] presented a framework that employs a reinforcement learning-based bitrate adaptation strategy and MEC for better FoV tiles prediction to optimize 360° video streaming over cellular networks. The adaptive bitrate strategy minimizes bandwidth costs and optimizes streaming quality, while the FoV prediction integrates time-series models and video attributes to increase reliability. Nevertheless, in multi-user settings, the algorithm may not take into account multiple viewers simultaneously viewing 360° videos, which could result in scalability challenges and reduced QoE. Recently, Shi et al. [109] presented a meta-RL-based framework for real-time edge video analytics to react swiftly to the dynamic state of the system and different QoE patterns. However, their single-user-based adaptation model does not allow coordinated multi-user optimisation and real-time 360° video streaming. These solutions do not provide cross-client perceptual fairness or dynamic coordination under bandwidth contention.

3.2.2 MEC-based QoE Fairness and Multi-Client Optimization

In contrast to the discussed work, to overcome the constraints of single-user-focused algorithms and optimize QoE for several users simultaneously streaming 360° videos. Li et al. [110] introduced high-quality interactive VR collaboration for multiple users by exploiting edge computing. The authors effectively reduced VR service latency, improved rendering performance, reduced computing load on the VR device, and ultimately improved users' QoE. Recently, an edge-based ABR solution was considered to optimize the client's QoE and QoE fairness at the edge. In [91, 111–113], authors utilized the edge-based intelligent ABR algorithms for joint QoE optimization. These researchers used MADRL techniques to improve QoE fairness among clients and obtained high average client QoE levels. However, these solutions struggled to fulfill the demands of multiple clients and were hesitant to make judgments that would ultimately be in their best interests, since rich media content optimization was not taken into consideration while design-

ing them. These solutions do not address client heterogeneity in spatial and perceptual preferences or adaptively allocate tiles based on VMAF-aware priorities across concurrent streams.

Table 3.2 summarizes AI-empowered MEC-based slicing strategies. DASH-assisted caching/prefetching methods leveraging EL and DL [77, 78, 84, 85] yielded reduced delay and enhanced CHR. In multi-user immersive setups, RL and MADRL solutions [90–92] showed scalable QoE enhancements. MEC orchestration schemes based on GNN, regression, and DRL [80–82] demonstrated improved adaptability and precision. In [91, 111–113], authors utilized the edge-based intelligent ABR algorithms for joint QoE optimization. These solutions do not yet provide robust cross-user coordination or perceptual-aware adaptation at the tile level. Future work should deepen integration of RL, MADRL, and GNN models for scalable 360° media delivery, computation offloading, and context-aware adaptive streaming over MEC-enabled network slicing.

Table 3.2: AI-based optimised content-delivery approaches over MEC [3].

Main objective	**Ref.**	**Model**	**Key contribution**	**Data and setup**	**Evaluation**
DASH-based prefetching and caching	[77]	EL	Improves cache-hit ratio (CHR) for DASH streaming.	Synthetic; simulation	Objective QoS metrics
	[78]	EL	Improves CHR and network-data utilisation for DASH segments.	Synthetic; testbed	Objective QoS metrics
	[83]	EL + RL	Jointly adapts segment prefetching, caching, and edge transcoding.	Synthetic and real-world; simulation	Objective QoS and QoE metrics
	[84]	EL + RL	Evaluates dynamic segment prefetching for ABR streaming.	Synthetic and real-world; simulation	Objective QoS and QoE metrics
	[85]	DL	Performs intelligent segment prefetching and client-side rate adaptation.	Synthetic; simulation	Objective QoS and QoE metrics
	[86]	DL	Increases average bitrate and reduces rebuffering.	Synthetic and real-world; simulation	Objective QoE metrics
	[87]	DRL	Uses edge prefetching and caching to improve adaptive-streaming QoE.	Synthetic; simulation	Objective QoE metrics
	[88]	DRL	Improves QoE using buffer-state and bandwidth-dynamics awareness.	Synthetic; simulation	Objective QoE metrics

Continued on the next page

Table 3.2 – *continued from the previous page*

Main objective	Ref.	Model	Key contribution	Data and setup	Evaluation
	[89]	EL	Reduces delivery latency, cost, and backhaul load for DASH.	Real-time data; simulation	Objective QoS metrics
MEC-enabled network management	[80]	GNN	Improves client QoE and CHR for VR-service requirements.	Real-time data; simulation	Objective QoS and QoE metrics
	[81]	RL	Enables intelligent edge-sliced networks for low-latency video.	Synthetic and real-world; simulation	Objective QoS metrics
	[82]	Regression	Supports QoE management for AR and VR applications.	Real-time data; testbed	Objective QoS and QoE metrics
	[93]	DRL	Reduces service delay and backhaul congestion through multicasting.	Real-time data; simulation	Objective QoS metrics
MEC-based caching and transcoding	[90]	MADRL	Enables multi-agent transcoding, caching, and streaming for 360° VR video.	Real-time data; simulation	Objective QoS metrics
	[91]	MADRL	Provides QoE-aware multi-agent transcoding and edge caching for 360° video.	Real-time data; simulation	Objective QoS metrics
	[92]	MADRL	Supports multi-user 360° VR caching, transcoding, and streaming.	Real-time data; simulation	Objective QoE metrics
Computation offloading	[95]	AI inference	Reduces inference delay and computation cost while improving recognition.	Real-time data; simulation	Objective QoS metrics
	[97]	DRL	Optimises computation offloading for VR-video processing.	Real-time data; simulation	Objective QoS metrics
	[99]	DRL	Uses RIS assistance to reduce latency and optimise VR-rendering offloading.	Real-time data; simulation	Objective QoS metrics
Resource management	[101]	RL	Performs adaptive resource allocation for rich media in multi-layer MEC.	Real-time data; testbed	Objective QoS metrics
	[102]	DRL	Jointly optimises radio and estimation resources under QoS constraints.	Synthetic; simulation	Objective QoS metrics
	[103]	DRL	Optimises utility-based resource distribution in MEC-enabled slicing.	Synthetic; simulation	Objective QoS metrics
	[104]	DRL	Improves viewport prediction and rendering while reducing VR latency.	Real-time data; simulation	Objective QoS metrics
	[105]	DRL	Supports immersive VR streaming over terahertz wireless networks.	Real-time data; simulation	Objective QoS metrics

3.3 MEC-enabled CDN for DASH-based Rich Media Content Streaming

CDNs are often employed for DASH-based delivery of rich media content like AR, VR, XR, and 360-degree videos, and many are enabled by MEC. Generally, CDNs are designed to optimize content delivery by distributing content among numerous servers that have been carefully placed in diverse geographical areas to offer the best and most efficient services. Particularly, MEC-enabled CDNs ensure that media content is cached and stored on the edge servers that are closer to end clients [114]. Such locations improve the overall streaming quality and reduce latency, especially for rich immersive content, which requires both high throughput and low latency.

Lately, CDNs have gained prominence for delivering rich video content [115]. The need for engaging and aesthetically pleasing experiences is a key driver of the rising consumer demand for viewing appealing video content. Video content meets customers' demands for multimedia participation in the digital age by offering a dynamic and engaging approach to consuming news, entertainment, and narratives. At the same time, edge media processing has been introduced as a critical component of 3GPP Release 17[1], taking advantage of the 5G system's edge characteristics to benefit various multimedia applications and content suppliers [116]. Apart from introducing new media transfer protocols, the 3GPP TR 26.804[2] and TR 21.905[3] also address network slicing, content preparation, edge-based rich multimedia processing (i.e., MEC), background data transmission, and improvements to backhaul and low-latency rich media streaming via DASH [117]. Therefore, MEC-enabled CDN in a network slicing context can be a promising solution to optimally assign resources to meet the specific service performance requirements of rich media content users. However, these efforts primarily target static content delivery and do not fully capture the real-time dynam-

[1] https://www.3gpp.org/specifications-technologies/releases/release-17
[2] TR 26.804: "Study on 5G media streaming extensions"
[3] TR 21.905: "Vocabulary for 3GPP Specifications"

ics and viewport variability of 360° streaming scenarios involving multiple concurrent users.

To meet the growing demand for high-bandwidth applications, major content providers such as Apple TV+, Amazon Prime Video, YouTube, Facebook, and others employ Over-The-Top (OTT) systems built on CDN architectures [118]. Abbasi et al. [119] introduced a novel Stall Duration Tail Probability (SDTP) model to minimize video delays in CDN-based OTT systems by considering constrained cache sizes, server selection, and throughput-aware allocation policies. Farahani et al. [120] proposed a hybrid peer-to-peer CDN model utilizing unsupervised learning through Self-Organizing Maps (SOM) to adaptively serve peer requests with low latency and improved QoE. These models are effective in reducing stall durations and improving average QoE, but the authors largely rely on content popularity estimation without considering perceptual quality scores such as VMAF or adapting to user-specific field-of-view data.

In support of dynamic live content delivery, Moreno et al. [121] used a DRL-based framework deploying a deep Q-Network for context-aware resource provisioning in streaming systems. Similarly, Ozcelik et al. [122] combined ABR and playback speed optimization using an A2C model to maintain high user QoE in live streaming scenarios. However, these works are primarily optimized for 2D video and do not consider the tile-based, viewport-adaptive demands of 360° video streaming.

Due to the limited visibility of traditional CDN topologies, server selection remains suboptimal. To address this, Tran et al. [123] proposed a multi-armed bandit (MAB)-based selection model integrated with a centralized SDN controller in a distributed CDN network. Their experimental setup, built using ONOS and MININET, showed superior response time and reward metrics. However, this framework lacks integration with perceptual quality metrics and user feedback loops, which are essential in immersive media applications.

Hayes et al. [124] developed a lightweight neural network model distributed over a peer-to-peer multipath transport network for optimizing ABR in 360° VR streaming. Their solution integrated RL-based ABR adaptation, achieving lower disruptive FoV switches and improved estimated Mean Opinion Score (eMOS) [125] in comparison with baseline models such as Deep Q Network (DQN) and A3C. While promising, this method does not generalize well to multi-user MEC environments with shared resources and fairness constraints.

Prefetching and caching have further enhanced CDN capabilities. DeepCache [126] introduced a deep Recurrent Neural Network(RNN)-based personalized caching mechanism using Long Short-term Memory (LSTM) encoders for predicting content popularity. Shi et al. [127] proposed a QoE-oriented DNN caching and prefetching framework using edge servers, achieving up to 57% chr improvement and 29% higher QoE over five objective metrics. However, these designs typically optimize at a session level and do not consider viewport-aware prediction or the granularity of tile-level adaptation necessary for volumetric or 360° video formats.

For 360° video content, Uddin et al. [128] employed a random forest model to predict video segment popularity for CDN-assisted caching. Sadeghi et al. [129] designed a hyper DQN-based two-level caching mechanism that coordinates caching actions across central and root nodes. Berger et al. [130] applied model-based RL for optimal CDN caching decisions, enhancing CHR and throughput with reduced tuning overhead. While these approaches introduce predictive mechanisms, the authors still fall short in synchronizing perceptual quality with caching and streaming decisions across heterogeneous user contexts.

CDN systems consist of multiple geographically distributed Points of Presence (PoPs) managed by Global Server Load Balancing (GSLB). DNS resolves client requests to the optimal PoP, and load balancing ensures effective caching and content delivery. At the mobile edge, Tahmasebi-Pouya et al. [131] proposed a Q-learning-based load balancing framework for time-

critical healthcare applications, achieving better response time and execution delay. Yet, similar intelligent load balancing strategies have not been widely adopted in 360° video streaming scenarios, where client demands and bitrate requirements vary significantly by viewport and device constraints.

Table 3.3: AI-based optimised content-delivery approaches over CDN [3].

Main objective	Ref.	Model	Key contribution	Data and setup	Evaluation
Mobile-edge caching	[132]	DL	Develops a content-replacement policy to improve cache efficiency.	Synthetic and real-world data; testbed	Objective QoE metrics
	[133]	DRL	Prioritises mobile VR content delivery at the edge.	Real-time data; simulation	Objective QoE metrics
	[134]	RL	Optimises CDN cache-admission decisions to improve CHR.	Real-time data; simulation	Objective QoE metrics
Edge-based ABR optimisation	[135]	DL + DRL	Personalises live crowd-casting and improves network assignment.	Real-time data; testbed	Subjective and objective QoE metrics
	[136]	DRL	Applies neural ABR control to improve user QoE.	Real-time data; simulation	Objective QoE metrics
	[137]	DRL	Scales neural ABR optimisation to Facebook's production video platform.	Real-time data; simulation	Objective QoE metrics
	[138]	DRL	Improves aggregate QoE and fairness among multiple clients.	Real-time data; simulation	Objective QoE metrics
	[124]	RL	Optimises ABR decisions over multipath networks.	Real-time data; simulation	Objective QoE metrics
OTT video streaming	[120]	Unsupervised SOM	Uses online learning to optimise OTT-video QoE.	Real-time data; testbed	Objective QoS and QoE metrics
	[121]	DRL	Improves QoE and QoS through higher session-acceptance rates.	Real-time data; simulation	Objective QoS and QoE metrics
	[122]	DRL	Jointly optimises user QoE and playback speed.	Real-time data; simulation	Objective QoE metrics
Content prefetching and caching	[126]	DL	Improves CHR compared with LRU and K-LRU policies.	Synthetic and real-world data; simulation	Objective QoS and QoE metrics
	[127]	DL	Uses cooperative learning to improve CHR and QoE.	Real-time data; simulation	Objective QoE metrics
	[128]	DL	Improves bandwidth efficiency and user QoE through edge caching.	Real-time data; simulation	Objective QoE metrics
	[129]	DRL	Applies two-level network caching to optimise user QoE.	Synthetic data; simulation	Objective QoS metrics

Continued on the next page

Table 3.3 – *continued from the previous page*

Main objective	Ref.	Model	Key contribution	Data and setup	Evaluation
	[130]	RL	Provides a lightweight and robust strategy for improving CHR.	Real-time data; simulation	Objective QoS metrics
Load balancing	[131]	RL	Reduces response time and network latency while improving load balance.	Synthetic data; simulation	Objective QoS metrics
	[139]	DL	Performs QoE-aware load balancing for OTT content over CDN.	Real-time data; testbed	Objective QoS and QoE metrics
	[140]	DL	Improves network utilisation while reducing delay and traffic load.	Synthetic data; simulation	Objective QoS metrics
Resource management	[141]	DL	Uses server logs for learning-based resource management and QoE improvement.	Real-time data; testbed	Objective QoS metrics
	[142]	DRL	Optimises resources in dynamic edge environments and reduces service time.	Real-time data; testbed	Objective QoS metrics
	[143]	LSTM + DNN	Improves CDN-performance prediction and resource scheduling.	Real-time data; simulation	Objective QoS metrics

3.3.1 CDN-based Resource allocation for efficient QoE requirements

Resource allocation based on CDN can meet the stringent latency and bandwidth needs of rich media streaming. Through QoE-driven analysis, Lobanov et al. [141] used CDN server logs to train a neural network that predicted QoE metrics (e.g., bitrate, caching events) without client-side measurement. However, such inference methods may not effectively capture perceptual satisfaction or temporal smoothness in dynamically changing user environments such as XR or volumetric applications.

Furthermore, Wang et al. [142] proposed a DRL-based resource orchestration scheme to minimize service delay in edge environments, surpassing traditional Open Shortest Path First (OSPF)-based strategies [144]. Li et al. [143] applied a deep learning model combining LSTM and Deep Neural Network (DNN) for predicting CDN performance, enabling better resource scheduling and improved QoE accuracy with minimal training overhead. Nevertheless, these studies lack attention to cross-client fairness and real-time feedback control in immersive environments, which are key for sup-

porting multi-user QoE stability. Table 3.3 summarizes key works on AI-driven CDN content delivery strategies. While traditional CDN platforms offer scalable distribution, integrating AI models such as DRL, DNNs, and classification algorithms enables predictive caching, dynamic routing, personalized QoE optimization, and latency-aware resource management. These methods have shown significant performance gains in immersive and delay-sensitive applications, including 360° VR, live events, and OTT streaming. Nevertheless, future research must explore federated learning, zero-touch orchestration, and fairness-aware models in CDN-enhanced network slicing for delivering high-throughput multimedia services. In particular, there is a need for viewport-aware caching and VMAF-driven adaptation that jointly optimize QoE and fairness across multiple concurrent 360° video users. Our proposed work addresses these gaps through an edge-intelligent COMA MADRL framework that aligns CDN resource allocation with perceptual quality metrics and multi-user adaptation priorities.

3.4 Adaptive Streaming for Volumetric Video

VVS introduced in chapter 1 has emerged as a pivotal technology in enabling immersive 6-DoF experiences where users can freely navigate within a 3D scene. Unlike conventional 2D or 360° video, volumetric content comprises dense spatial point clouds or mesh representations, offering both translational and rotational interaction. However, the massive data volumes and real-time rendering requirements present formidable challenges for streaming over bandwidth-constrained and variable network conditions. To address these challenges, several research efforts have explored diverse strategies for bitrate adaptation, geometry prioritization, and perceptual quality enhancement. These can broadly be categorized into three research directions: (i) DASH-compliant segmentation and adaptive delivery, (ii) learning-based bitrate and geometry optimization, and (iii) generalization-centric quality adaptation frameworks.

3.4.1 DASH-compliant segmentation and adaptive VV delivery

Early works in adaptive streaming for volumetric content focused on extending DASH principles to volumetric formats. For example, VV-DASH introduced a codec-agnostic segmentation framework aimed at facilitating rate adaptation for point cloud content using standard DASH mechanisms [145]. However, this segment-based approach operates at a coarse temporal level and lacks viewport awareness, which is crucial for immersive scenarios where user attention is spatially focused and dynamic. Moreover, it does not consider perceptual quality metrics specific to 3D content, such as PCQM or V-PCC quality indices, which limits its effectiveness in maximizing user QoE. Other streaming systems, such as ViVo [146] and Rolling Prediction-Optimization-Transmission (POT) [147], improve data transfer efficiency through viewport-based pruning techniques. By leveraging head-tracking or FoV predictions, the authors discard occluded or non-visible point cloud regions, thereby achieving significant reductions in bandwidth consumption. Rolling POT further integrates a prediction-optimization-transmission loop for real-time decision-making. Nevertheless, while these solutions enhance responsiveness and reduce latency, the authors often suffer from degraded performance when viewport prediction errors or sudden motion transitions occur, leading to rebuffering or visible degradation artifacts.

To address the limitations of viewport-agnostic methods, newer frameworks have incorporated real-time viewport estimation and region-aware geometry prioritization. For instance, the approach in [148] fuses pose prediction with adaptive mesh segmentation, allowing prioritized delivery of salient geometry segments under bandwidth constraints. Similarly, studies such as GROOT [149] and its variants [150] employ geometry-driven compression and integrate wireless adaptation layers to optimize delivery in mobile edge environments. These systems focus on content-aware point cloud delivery but lack robustness when faced with dynamic or unseen user

behavior and network conditions, often resulting in unstable QoE. This motivates the need for adaptive schemes that go beyond handcrafted heuristics and embrace learning-based approaches capable of generalizing across diverse scenarios.

3.4.2 Learning-based bitrate and geometry optimization

The integration of ML techniques has significantly improved the adaptability and intelligence of VVS solutions. Liu et al. proposed *Vues*, an edge-assisted multiview transcoding framework that adaptively selects multiple 2D views from volumetric sources using lightweight CNNs [151]. This system strategically balances perceptual gain and bandwidth usage, effectively improving QoE without overwhelming the network. Building on viewport-driven streaming, *YuzuVV* employs a neural network for real-time head movement prediction and integrates geometry-awareness to focus bandwidth on visible volumetric segments [152]. Such approaches demonstrate the potential of combining learning models with perceptual and spatial priors, but the authors remain tightly coupled to specific network conditions and content profiles, thereby struggling to generalize.

Further advancements are seen in *CaV3* [153], which utilizes Long Short-Term Sequential Prediction (LSTSP) models to forecast user viewing trajectories and integrates a contextual multi-armed bandit for caching decisions. This intelligent caching minimizes redundant tile transmission and improves responsiveness, though its policy remains brittle under severe viewport shifts or network jitter. In a different direction, VV-based implicit neural radiance field (V^2NeRF) [154] combines explicit point clouds with implicit neural radiance fields to enable scene-level streaming. By employing background removal, a two-stage decoupled architecture, and perceptual loss-based buffer-aware adaptation, it achieves perceptual efficiency, but again, the framework is not optimized for generalization across unseen content or network traces.

Several RL-based solutions have also been proposed for volumetric ABR streaming. A notable example is the Rolling POT framework [147], which introduces a rolling horizon approach that incorporates short-window bandwidth and FoV forecasts into a DRL-based decision system. This framework significantly improves low-latency streaming performance and real-time QoE. More recently, Spatial quality oriented rate control for volumetric video streaming (SQRC-VV) [155] applies DRL to perform spatially aware rate allocation. It considers mesh distortion, transmission delay, and viewport consistency in its reward formulation, resulting in better perceptual outcomes. Despite these innovations, these RL methods lack meta-adaptation mechanisms that allow rapid retraining or tuning when facing new types of volumetric content or unforeseen client behavior, leading to performance drops in out-of-distribution conditions.

3.4.3 Generalization-centric quality adaptation frameworks

Generalization has emerged as a central concern in VVS. The inherent variability in 3D scenes, ranging from static landscapes to highly dynamic human interactions, along with user-specific viewport trajectories, demands adaptive policies that are robust yet flexible. Meta-learning techniques have been proposed to address this gap. *Meta-INR* introduces a meta-learned initialization of implicit neural representations (INRs) that can quickly adapt to new volumetric datasets using only partial observations and few-shot gradient updates [156]. While effective in improving encoding generalization, Meta-INR does not address downstream tasks such as viewport prediction, adaptive bitrate selection, or QoE modeling, which are critical in end-to-end streaming systems.

In traditional video streaming, several studies have successfully applied meta-reinforcement learning to improve policy generalization under fluctuating network traces and heterogeneous user conditions. For example, works like [157–159] use meta-RL to quickly adapt ABR policies to diverse 2D video content, minimizing rebuffering and improving quality switches

across sessions. However, these frameworks are limited to 2D video delivery and are not directly applicable to volumetric contexts, where the challenges of 6-DoF navigation, point-cloud encoding, and region-specific quality are fundamentally different and more complex.

To this end, some recent volumetric ABR frameworks have started to incorporate generalization-aware design principles. Few-shot adaptive bitrate volumetric video streaming (FewVV) [160], for instance, formulates adaptive streaming as a sequence modeling problem and trains a causal transformer on large-scale interaction traces to enable few-shot and zero-shot generalization across content and network conditions. The model outperforms several heuristic and RL-based baselines in unseen environments. Likewise, the *saliency-guided meta-learning framework for Gaussian Splatting* [161] introduces a region-prioritized QoE model and meta-learned policy to achieve robust performance in VV delivery under varying spatial dynamics and latency conditions. These systems represent significant progress toward generalization in immersive streaming, though the authors often overlook the joint optimization of all contributing components, such as head prediction, point cloud quality, and viewport-driven rate control within a unified adaptive framework.

3.5 Viewport Prediction and User Behavior Modeling for 360° VR and Volumetric Video

Accurate modeling of user behavior and prediction of future viewports are critical components of immersive media streaming systems, particularly for 360° VR and VV formats. To optimize bandwidth and ensure high perceptual quality, accurate forecasting of users' future FoV is essential.

Wahba et al.[162] conducted a comprehensive survey categorizing viewport prediction techniques into ML and DL-based methods, outlining the evolution from classical statistical models to data-driven approaches. The authors emphasized how DL architectures, such as LSTMs, outperform lin-

ear predictors in capturing the temporal dependencies of head movement trajectories, especially in dynamic VR content. Building on this, Lee [163] introduced an LSTM-based model enhanced with attention mechanisms, which improved prediction accuracy by focusing on salient temporal patterns in user head orientation.

Chen et al.[40] proposed a personalized deep learning framework that integrates content saliency with user behavior to adaptively forecast user viewports in 360° VR videos. The developedmodel demonstrates the advantage of combining visual attention cues with historical viewing patterns. Similarly, Hou et al. [164] focused on predictive view generation using mobile trajectory data, introducing an online learning framework that minimizes latency and optimizes tile delivery based on anticipated user movement.

From a content delivery perspective, Khan et al. [165] investigated the coupling of viewport prediction with edge caching strategies. Their work demonstrated that deep learning-based popularity prediction, aligned with LSTM-driven user movement forecasts, can significantly reduce network load and improve cache hit rates in 360° VR environments.

Focusing on volumetric video, Hu et al.[166] addressed the complexities of 6DoF navigation by developing a novel data set and analysis framework for VV user behavior. Their transformer-based prediction model effectively captured the spatial-temporal variability in volumetric video exploration, setting a benchmark for 3D viewport forecasting. Similarly, the STVP model introduced in [167] leveraged both saliency features and user trajectory fusion to enhance viewport prediction accuracy in VV delivery scenarios.

In addition, Tian et al. [168] proposed a multi-modal multiscale transformer architecture that jointly processes gaze, saliency, and head motion cues to produce robust viewport forecasts. His work demonstrated the potential of transformer-based models to generalize across varying content types and user behaviors, thereby enabling more adaptive and intelligent streaming systems. Recently, Li et al. [169] proposed a transformer-based volumetric Video viewport prediction network (Vp-former) that efficiently ex-

tracted spatial-temporal features from point cloud videos. It constitutes a viewport prediction framework that combines the spatial-temporal features of point cloud videos of user trajectory information with reduced computational complexity, and successfully predicts future user viewpoints by combining the spatial-temporal features of point cloud videos with user head trajectory information.

In contrast to the aforementioned approaches, which are limited by imperfect viewport/saliency awareness, suboptimal perceptual modeling (e.g., geometry-level quality), and weak generalization under dynamic 6-DoF and network variability, the proposed *METAVV* framework presented in Chapter 5 represents a comprehensive and generalizable solution to the challenges of VVS. *METAVV* jointly integrates 6-DoF viewport prediction, distance-aware point cloud perceptual modeling (e.g., using PCQM metrics), and region-prioritized bitrate selection guided by a QoE-driven reward function. The core of *METAVV* is a multi-agent meta-reinforcement learning policy that learns generalizable adaptation strategies across diverse content, network traces, viewport trajectories, and user motion behaviors. By employing task distributions that reflect real-world variability in volumetric scenes and network conditions, *METAVV* enables rapid adaptation to unseen scenarios with minimal gradient updates, thereby addressing the scalability and generalization limitations of traditional DRL and heuristic-based approaches. Furthermore, *METAVV*'s ability to operate across spatial and temporal levels, from densely voxelized versions of the point cloud to adaptive bitrate switching positions it as a significant advancement in the design of intelligent, perceptually aware, and generalization-ready VVS for next-generation immersive media delivery.

3.6 Chapter Summary

This chapter presented a review of existing research on AI-enabled adaptive streaming for rich media delivery in 5G and beyond network environ-

ments. The discussion covered major advances in intelligent network support for immersive applications, including edge-assisted content delivery, learning-driven bitrate adaptation, and predictive resource management for AR, VR, 360° video, and volumetric media. Recent surveys and tutorial works were examined to contextualize the evolution of AI-driven solutions, while highlighting the growing role of MEC and edge intelligence in meeting stringent latency and throughput requirements. The chapter further reviewed adaptive streaming mechanisms for volumetric video, emphasizing DASH-compliant segmentation strategies, learning-based bitrate and geometry optimization, and viewport-aware delivery techniques. Reinforcement learning and multi-agent frameworks have demonstrated strong potential to address scalability and QoE optimization challenges in dynamic and multi-user environments. At the same time, emerging research on user behavior modeling and viewport prediction for 360° and VV formats revealed the importance of perceptual awareness and spatial prioritization in efficient media delivery.

Despite these advances, existing solutions often exhibit limited robustness when exposed to unseen network conditions, heterogeneous content characteristics, or diverse user interaction patterns. In particular, many approaches lack mechanisms for rapid adaptation and generalization across environments, and do not jointly consider 6DoF user behavior, perceptual quality metrics, and adaptive rate control within a unified framework. These limitations motivate the need for adaptive architectures that integrate edge intelligence, perceptual modeling, and generalization-centric learning. The insights drawn from this chapter provide the foundation for the proposed solutions presented in the subsequent chapters, which aim to address these gaps through region-aware multi-agent learning and meta-reinforcement learning–based adaptive streaming frameworks.

Chapter 4

EDGE360: Edge-Enabled Multi-Agent DRL for A Region-Aware Rate Adaptation Solution to Enhance Quality of 360° Video Streaming

*This chapter presents **EDGE360** [19], a novel edge-assisted adaptive streaming framework tailored for 360° VR video delivery in bandwidth-constrained and latency-sensitive environments. EDGE360 employs three innovations in terms of viewport prediction, tile-based VMAF-driven quality assessment, and MADRL to enable intelligent bitrate decisions that are personalized and perceptually optimized. By integrating a centralized critic with distributed actor agents, this proposed solution ensures collaborative yet client-specific adaptations, while accounting for viewport saliency, network state intelligence, and QoE fairness across users. EDGE360 serves as a foundational component in this thesis, laying the groundwork for subsequent AI-driven strategies for immersive content delivery in 5G and beyond networks.*

Relationship to published work: This chapter builds upon the author's published article: F. E. Subhan, A. Yaqoob, C. H. Muntean, and G.-M. Muntean, "EDGE360: Edge-Enabled Multi-Agent DRL for Region-Aware Rate Adaptation Solution to Enhance Quality of 360° Video Streaming," *IEEE Transactions on Mobile Computing*, vol. 25, no. 2, pp. 1918–1935, February 2026. The material has been extended for inclusion in this thesis, including additional methodological explanations, reproducibility information, sensitivity analysis conclusion, statistical significance testing, and discussion of evaluation limitations.

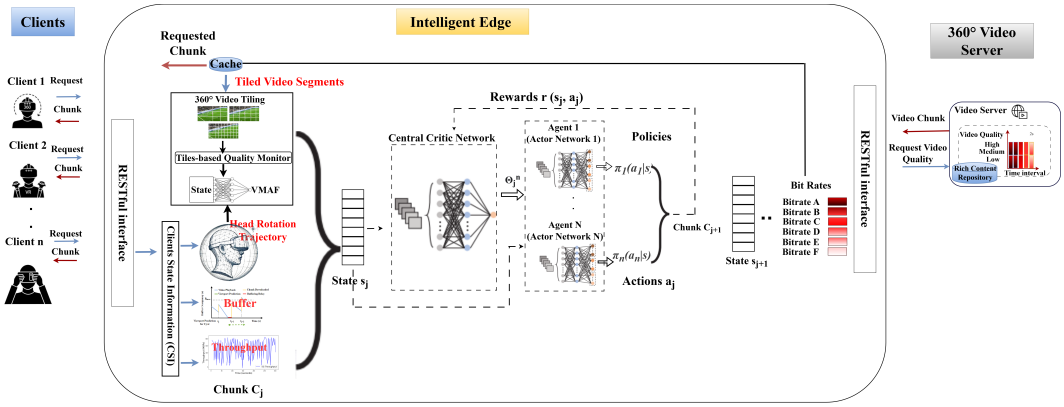


Figure 4.1: The *EDGE360* video streaming architecture comprises a CDN server, smart edges, and multiple rich media clients.

4.1 Architecture

Fig. 4.1 illustrates the architecture of the *EDGE360* framework. The architecture targets adaptive delivery of 360° VR video to multiple clients via an intelligent edge node. In order to achieve high video quality and a seamless streaming experience, a sophisticated multi-agent DRL-based ABR algorithm is executed at the edge, leveraging client-specific inputs such as viewport prediction, buffer status, and available network bandwidth/throughput. Through interaction with a central 360° video server, the edge node operates as an intermediary: it processes client requests, applies the learned adaptation policy, and delivers optimized tiled video content to users. By continuously updating decisions as network conditions and client demands vary, the edge-side adaptation mechanism sustains

high QoE for immersive VR streaming. The *EDGE360* solution comprises three primary components: clients, the intelligent edge, and the rich VR 360° video server, which are described in detail below.

4.1.1 Clients

A client in the *EDGE360* framework corresponds to an end-user device (e.g., a 360° video viewer or a VR headset) that requests and consumes immersive content. Using a RESTful interface, the client requests specific 360° video segments (and associated tiles) from the intelligent edge node. To enable informed adaptation decisions at the edge, each client periodically reports its playback and context status via the Client State Information (CSI) module. This CSI report includes the client's measured throughput, buffer occupancy, and viewport trajectory. The throughput measurements capture time-varying network capacity and are used by the edge-side adaptation algorithm to select feasible bitrate levels. The buffer status enables the adaptation mechanism to avoid playback disruptions by adjusting quality when the risk of re-buffering increases. The viewport trajectory informs region-aware prioritization, allowing the adaptation algorithm to allocate higher quality to tiles expected to fall inside the client's viewport. Overall, these real-time client reports provide the key inputs required for edge-side resource allocation and QoE optimization across multiple concurrent clients.

4.1.2 Intelligent Edge Node and Adaptation Modules

Within the *EDGE360 framework*, the *intelligent edge node* is the central architectural component that receives client requests and executes the edge-side adaptation *algorithms* responsible for bitrate/tile-quality selection and content delivery. The adaptation logic leverages client-specific inputs, including viewport estimations, buffer status, and bandwidth/throughput availability to compute streaming decisions that aim to reduce latency and improve resource utilization while maintaining high QoE.

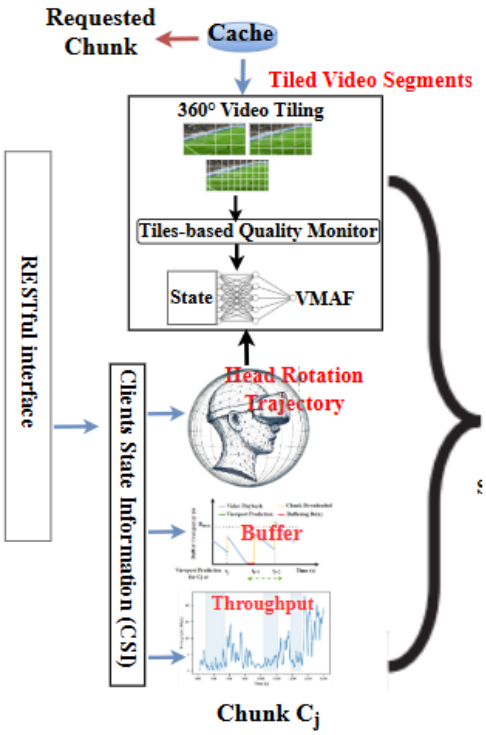


Figure 4.2: Core intelligent-edge components in *EDGE360*: the client reports CSI (viewport/head-rotation trajectory, buffer, and throughput) per chunk C_j, while the edge performs 360° tiling, VMAF-based tile quality monitoring, caching, and viewport-prioritized tile/bitrate recommendation

A core module hosted at the intelligent edge node is the *360° video tiling and quality monitoring* module as shown in Fig. 4.2, which enables region-aware adaptation. Specifically, the server-side 360° content is partitioned into tiles, where each tile corresponds to a distinct spatial area of the spherical scene. This tiling representation allows the edge-side adaptation algorithm to assign higher quality to tiles expected to fall inside the client's viewport and lower quality to tiles outside the FoV, thereby improving perceptual efficiency under constrained bandwidth. Using the client-reported CSI (head-rotation/viewport trajectory, buffer occupancy, and throughput) for each chunk C_j, the edge-side adaptation mechanism prioritizes higher quality for tiles likely to fall inside the client's viewport and allocates lower quality to tiles outside the FoV, improving perceptual efficiency under constrained bandwidth.

To quantify the perceptual quality of each tile, the intelligent edge node integrates a tile-based Quality Monitor that employs VMAF to score the per-

ceptual quality of candidate tile representations (quality levels) and guide viewport-prioritized bitrate allocation.VMAF is a perceptual video quality metric that uses ML-based fusion of elementary indicators (e.g., detail loss and structural distortion) to produce a score that correlates with human visual perception. In *EDGE360*, tile-level VMAF scores are used to map tile representations (e.g., small, medium, and large quality levels) into perceptually meaningful quality targets. This tile-specific monitoring supports the edge-side adaptation mechanism in meeting VR users' visual expectations while conserving bandwidth by reducing the quality of less perceptually relevant (non-visible) tiles.

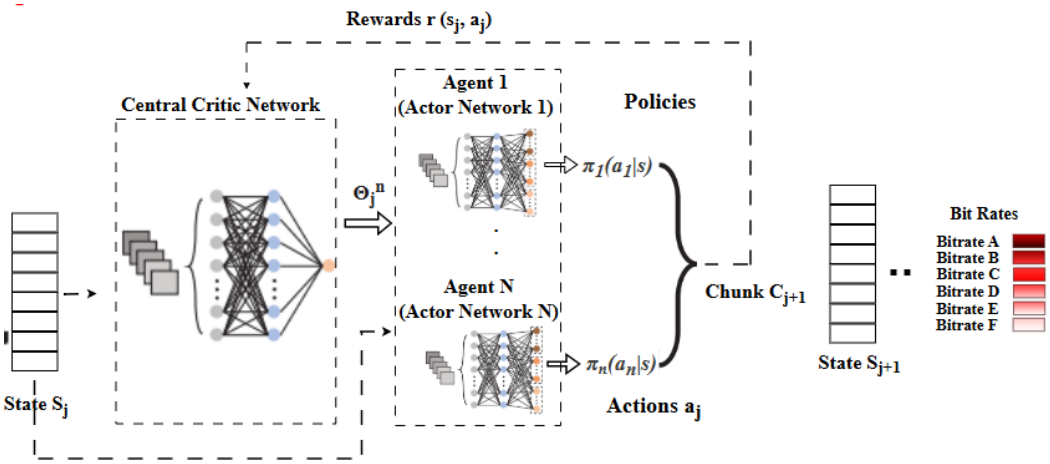


Figure 4.3: COMA-based multi-agent adaptation mechanism, with multiple agents and central critic

The COMA policy network is a specialized MADRL model as shown in Fig. 4.3, hosted within the intelligent edge node's *rate-adaptation module*, which is a component of *EDGE360* framework (shown in Fig. 4.1) as the decision-making component and manages cooperative decision-making for adaptive streaming. To carry out this task, the *EDGE360* system generates a state $\mathcal{S}_j$ for each client session based on real-time data. In the COMA-based multi-agent adaptation mechanism, each client serves as a separate agent with its actor-network that learns a policy $\pi_n\left(a_n \mid s_n\right)$, where a_n denotes actions like bitrate and tile quality selection for the current video chunk $\mathcal{C}_j$. After selecting the action a_j, the system updates the state to $\mathcal{S}_{j+1}$

based on an updated CSI module and a tiles-based quality monitor. The actor-network then repeats the process with this updated state for the next chunk $\mathcal{C}_{\text{j+1}}$. This strategy iteratively adjusts to maximize reward $r\left(s_j, a_j\right)$ given the current network circumstances. A central critic network assesses collective agent performance by computing the reward $r\left(s_j, a_j\right)$, incorporating measures such as video optimal streaming, tiled-chunk quality, and rebuffering minimization. The critic network computes an Advantage Function $\theta = A_n\left(s, a_n\right) = U_j\left(s, a_n\right) - \sum_{a'_n} \pi_n\left(a'_n \mid s\right) U_n\left(s, a'_n\right)$, where $U_j\left(s, a_n\right)$ is the value-action function of actor n, and $\sum_{a'_n} \pi_n\left(a'_n \mid s\right) U_n\left(s, a'_n\right)$ is a baseline to identify the expected value of alternative actions, resulting in isolating the influence of action a_n on overall performance. This function θ evaluates the efficiency of each agent's actions in terms of rewards, leading the system toward optimal resource allocation and adaptability across multiple clients.
Moreover, one of the edge's principal functions is to maintain a cache of video chunks. This cache momentarily caches frequently visited video items, reducing the need to send multiple queries to the central 360° video server. Whenever a client requests a video chunk, the edge node initially determines if it is accessible in the cache. If it is, the chunk is provided directly to the client, which results in quicker response times and less network demand for the central server. This caching approach is convenient when multiple viewers request overlapping content, since it allows for an effective reprise of cached video segments.

4.1.3 360° Video Server

The rich VR 360° video server is the primary source of 360° video material and includes a rich content repository with multiple quality levels (high, medium, and low) for each video tile, as shown in Fig. 4.4. Whenever the desired video chunk cannot be located in the edge's cache, the edge node retrieves it from the server, suggesting the quality required based on the COMA policy recommendations. The server dynamically reacts to client requests by supplying the appropriate video quality to ensure each client

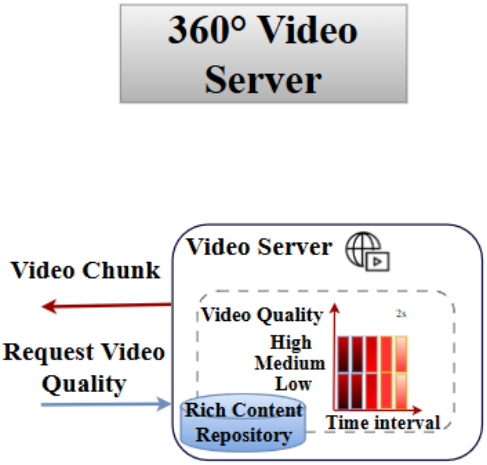


Figure 4.4: 360° video server component: stores multi-quality video chunks and serves requested quality levels to the edge/client.

achieves the highest degree of detail to enhance the viewing experience. The server allows the edge to scale effectively by transmitting only the required quality levels for each tile rather than an entire video with maximum quality. The server optimizes the bandwidth consumption and minimizes redundant data transmission by keeping a repository with split quality levels.

4.1.4 Edge-Assisted Adaptive Streaming in the *EDGE360* Framework

The *EDGE360* framework adopts an HTTP/DASH-style adaptive streaming workflow for tiled 360° video, where clients request MPD and subsequent video segments over time. In such workflows, the client remains responsible for issuing HTTP requests (e.g., MPD and segment/tile downloads) and maintaining playback continuity, while adaptation is driven by continuous feedback on network and playback dynamics.

As introduced earlier in *EDGE360*, an *intelligent edge node* is placed between clients and the origin 360° video server to enable edge-assisted decision making and delivery. Rather than altering the streaming protocol itself, the edge node hosts the adaptation and coordination *algorithms* that (i) interpret client-side context (e.g., buffer level, measured throughput, and viewport trajectory), (ii) compute region-aware tile/bitrate recom-

mendations, and (iii) accelerate delivery via caching and edge-side request handling. This edge-assisted adaptive streaming design enables responsiveness to fast wireless variations and multi-user contention while prioritizing perceptually important viewport regions.

Operationally, the adaptive streaming procedure proceeds in three recurring phases. First, a client establishes a streaming session and acquires the MPD that describes the available representations, segment duration, and tiling-related information. Second, during playback, the client periodically reports runtime measurements (e.g., throughput samples, buffer occupancy, and viewport trajectory/predictions) that serve as inputs to the edge-side adaptation algorithm. Third, based on these inputs and the available content representations, the edge node computes and returns a bitrate/tile-quality recommendation that guides the client's next download decision, while optionally serving requested objects directly from the edge cache to reduce latency.

The following section details this workflow step-by-step using the sequence diagram, clarifying how the client, the intelligent edge node (within the *EDGE360* framework), and the origin video server interact during a complete video download session.

4.2 *EDGE360* Adaptive Video Streaming Process

Fig. 4.5 illustrates the sequence of interactions among three entities during a video download session: (i) the *client device* (e.g., HMD/360° viewer), (ii) the *intelligent edge node* (a component within the *EDGE360* framework that hosts the adaptation and coordination algorithms), and (iii) the *origin 360° video server*. When a client initiates playback, it sends a *Session Up* message to the intelligent edge node to establish a streaming session. This request carries device and session descriptors (e.g., display resolution, operating system, network access information, and the 360° content URL). The edge node acknowledges the session and records the client's require-

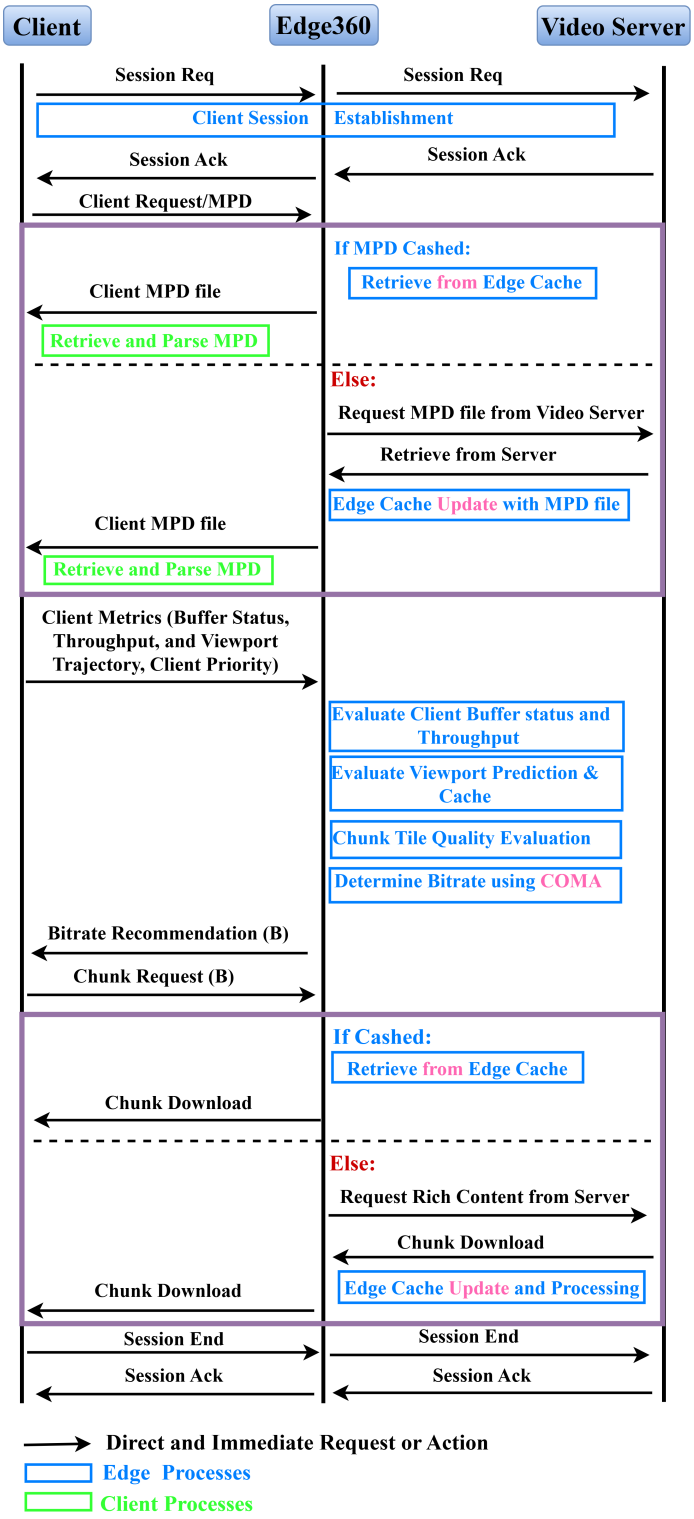


Figure 4.5: Sequence Diagram of the Client Tiled Video Chunk Download Session

ments so that subsequent adaptation decisions can be tailored to the client context.

Upon acknowledgment and client session establishment, the client requests the MPD file by sending a "Client Request/MPD" message after the session is initiated. The MPD file includes crucial details about the 360° video content, including bitrate selection, segment duration, and prospective resolutions. If the MPD is available in the edge cache, the edge node returns it immediately; otherwise, the edge node fetches it from the origin server and updates the cache. This step ensures that the edge node has the information required to interpret the available quality levels, tile layout (if applicable), and chunk duration used by the streaming session.

During playback, the client periodically reports runtime measurements, including buffer status, measured throughput, and viewport trajectory (and any derived region priority) to the intelligent edge node. These measurements serve as inputs to the *edge-side adaptation algorithm*, which computes the next bitrate/tile-quality recommendation while balancing perceptual quality, stability, and rebuffering risk.

In *EDGE360*, this decision-making is supported by a COMA-based multi-agent RL approach, enabling coordinated recommendations across multiple concurrent clients. The resulting recommendation is then communicated back to the client, guiding the next segment/tile request in a manner aligned with the client's buffer state, predicted FoV, and current network conditions, thereby reducing disruptive quality fluctuations and rebuffering events.

Furthermore, in the subsequent operations, the client sends a "Chunk Request" for a particular video chunk at the recommended bitrate after acquiring the bitrate recommendation. Next, the edge node searches its cache to check if the requested tiled video chunk is locally accessible. Alternatively, the edge node asks the video server for the necessary rich content. An important part of *EDGE360*'s architecture, this caching mechanism stores frequently requested video chunks at the edge to lower latency. To maximize the cache's effectiveness and improve the client's streaming performance, the content is first retrieved and then saved in the edge cache to speed up

subsequent requests for a similar video segment. In the following steps, the client starts the "Chunk Download" operation after getting the video chunk.

Finally, the client initiates the "Session End" step, which concludes the streaming session. The session termination message is acknowledged by the edge node and the video server, enabling the disengagement of allotted resources and the updating of client information, ensuring that the system is accessible to inbound session requests. *EDGE360* framework develops a scalable, adaptive streaming for immersive 360° content by handling caching, processing, and transmission effectively.

4.3 *EDGE360*-Driven Bitrate Adaptation and QoE Computation

4.3.1 QoE Model

In $EDGE360$, adaptive streaming of 360° media for many clients necessitates optimizing a number of QoE parameters. Each client in the network requests tiles from the video stream using expected viewports, and the network must distribute resources efficiently to provide seamless playback with minimal re-buffering and delay. In this instance, ABR choices are made using the COMA reinforcement learning model. In COMA, each client is represented by an actor-network, while a central critic network assesses the system's global state. This method tries to improve QoE for individual clients while retaining system fairness. The *EDGE360* framework evaluates QoE in a multi-client scenario using a step-wise QoE computation mechanism that starts from throughput distribution and then incorporates tile granularity, video quality switches, buffer dynamics, and delay, ultimately combining these factors into a comprehensive QoE metric.

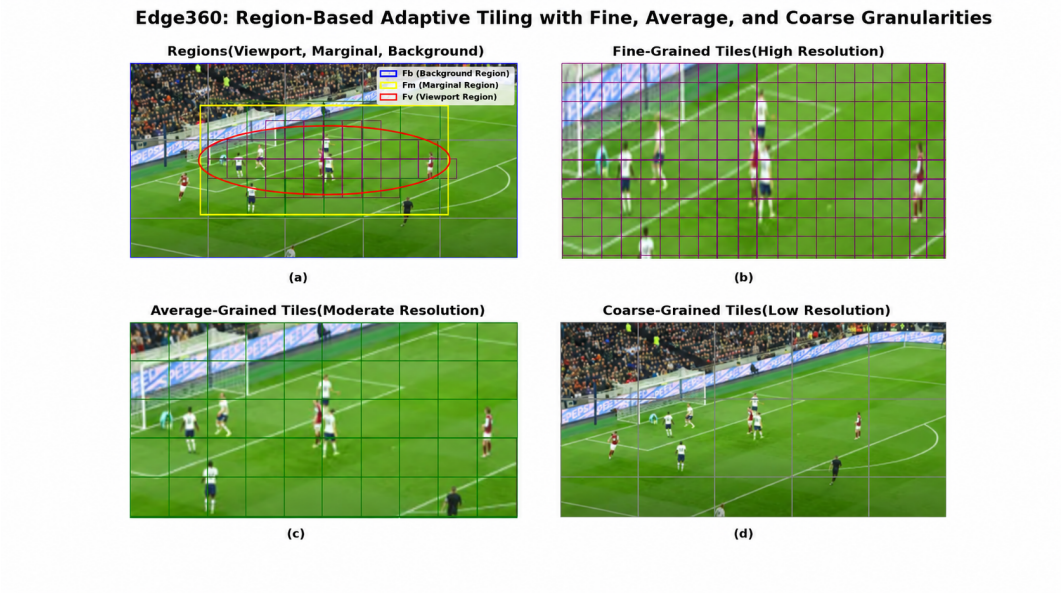


Figure 4.6: (a) shows the segmentation of a 360° video frame into three spatial regions: *Viewport* ($\mathcal{F}_v$) with fine-grained tiles, *Marginal* ($\mathcal{F}_m$) with average-grained tiles, and *Background* ($\mathcal{F}_b$) with coarse-grained tiles. (b)–(d) illustrate region-specific tiling, where finer tiles are allocated to high-priority regions for perceptual quality, and coarser tiles to low-priority areas for bitrate efficiency. This demonstrates *EDGE360*'s hierarchical tiling strategy for optimizing 360° VR streaming.

4.3.1.1 Network Throughput Distribution in a Multi-client Scenario

In a multi-client scenario, let $\mathcal{N}$ represent the number of clients, and $\Delta_{\text{BW}}(t)$ be the total bandwidth available at time t for distribution among clients. The allocated bandwidth for client n at time t, denoted $\mathcal{W}_n(t)$, is computed as:

$$\mathcal{W}_n(t) = \frac{\Delta_{\text{BW}}(t) \cdot \mathcal{C}_n(t)}{\sum_{n=1}^{\mathcal{N}} \mathcal{C}_n(t)} \tag{4.1}$$

where $n \in \{1, 2, \ldots, \mathcal{N}\}$ indexes the clients in the scenario and $\mathcal{C}_n(t)$ represents the bandwidth demand of each client, which is determined based on their buffer status and expected viewport probability. $\sum_{n=1}^{\mathcal{N}} \mathcal{C}_n(t)$ is therefore the total bandwidth demand across all clients.

$$\mathcal{C}_n(t) = c_1 \cdot \hat{\mathcal{B}}_n(t) \cdot \mathcal{R}_{\text{buf}} + c_2 \cdot \hat{\mathcal{P}}_n(t) \cdot \mathcal{R}_{view} \tag{4.2}$$

In Eq. 4.2, c_1 and c_2 are weighting coefficients that balance the contribution of buffer urgency and viewport relevance, respectively. $\hat{\mathcal{B}}_n(t)$ is the normalized buffer occupancy for client n, and $\mathcal{R}_{\text{buf}}$ is a bitrate factor reflecting the buffer-driven bandwidth demand. $\hat{\mathcal{P}}_n(t)$ is the normalized aggregated viewport probability, and $\mathcal{R}_{view}$ is the average bitrate expected per tile in the predicted viewport region.

The normalized buffer status $\hat{\mathcal{B}}_n(t)$ is defined as:

$$\hat{\mathcal{B}}_n(t) = \frac{\mathcal{B}_n(t)}{\max_{n\in\mathcal{N}} \mathcal{B}_n(t)} \tag{4.3}$$

with values within $[0, 1]$. $\mathcal{B}_n(t)$ (measured in seconds) is the buffer status and indicates the amount of buffered video available to the client at moment t.

$\hat{\mathcal{P}}_n(t)$, the normalized viewport probability, is computed as follows:

$$\hat{\mathcal{P}}_n(t) = \frac{\sum_{j=1}^{J}\sum_{k=1}^{K} \mathcal{P}_v(j,k,t)}{\max_{n\in\mathcal{N}} \sum_{j=1}^{J}\sum_{k=1}^{K} \mathcal{P}_v(j,k,t)} \tag{4.4}$$

where $\mathcal{P}_v(j,k,t)$ represents the probability tile $T_{j,k}$ is within the client's viewport at time t.

$\mathcal{W}_n(t)$ is used by the proposed *EDGE360* when performing adaptive bandwidth allocation. It dynamically prioritizes clients experiencing critical buffer depletion and those with higher viewport probability to achieve an optimal streaming experience.

4.3.1.2 Video Quality Assessment Based On Tile Granularity

In adaptive 360-degree video streaming, tile size plays a crucial role in balancing encoding complexity, perceptual quality, and network resource utilization. *EDGE360* employs a hierarchical tiling structure with three tile sizes *fine, average, and coarse* in 360° video, indicated by f, a, and c, respectively, as shown in Fig. 4.6. In Eq. 4.5, the impact of tile size on the perceived video quality is governed by scaling factors ψ_f, ψ_a, and ψ_c, which

Table 4.1: Notation Table

Notation	Description
$\mathcal{N}$	Represents the set of agents
$\mathcal{W}_n(t)$	The allocated bandwidth for client n at time t
$\Delta_{\text{BW}}(t)$	Total available bandwidth at time (t) for distribution among clients
$B_n(t)$	The bandwidth allocated to client n
$\mathcal{C}_n(t)$	The estimated bandwidth demand of client n at time t
$R_n(t)$	The reward of the action at time t by client n
$\alpha, \beta, \zeta, \lambda$	Non-negative parameters influencing the aggressiveness of client QoE
$\mathcal{Q}_{j,k}(t)$	The effective quality weight for each tile (j,k) at time t
$\mathcal{Q}^{\text{ref}}_{j,k}(t)$	The reference perceived quality level
$U_n(t)$	The utility factor of client n at time t
$B_n(t)$	The client buffer occupancy
$\hat{\mathcal{Q}oE}(t)$	Aggregated QoE across all clients at time t
$\mathcal{Q}oE_{\text{max},n}(t)$	The maximum QoE unfairness of client n at time t
$\mathcal{Q}oE_{\text{max},n}(t)$	The minimum QoE unfairness of client n at time t
$\Delta\mathcal{Q}_{j,k}(t)$	Quality switching penalty for tile (j,k) at time t
$\mathcal{X}_j$	Tiling region of tile j within the overall 360° video
$D_n(t)$	The latency of client n at time t
$G_{j,k}(t)$	Tile granularity of tile (j,k) at time (t)
$Q_{j,k}(t)$	Past video quality (VMAF Scores) of tile (j,k) at time (t)
$\hat{Q}_{j,k}(t+1)$	Future video quality (VMAF Scores) of tile (j,k) at time $(t+1)$
$L_{j,k}(t)$	Cache/Chunk loss rate of tile (j,k) at time (t)
$\rho_n(t)$	Rebuffering events for client (n) at time (t)
$\hat{\rho}_n(t)$	Aggregated rebuffering events across clients at time (t)
$P_v(j,k,t)$	Viewport prediction probability for tile (j,k) at time (t)
$\hat{P}_v(j,k,t+1)$	Future Predicted viewport probability for tile (j,k) at future time $(t+1)$
$\mathcal{K}_j$	Edge cache utilization for chunk (j)
$\mathcal{R}^{\chi}_{j,k}$	Bitrate assigned to tile (j,k) at time t for quality level $\chi \in \{v,m,b\}$
$\mathcal{F}_{\text{Jain}}(t)$	Jain's Fairness Index range from 0 to 1
F_J	Aggregate fairness index
M_n	Client identity
$U^v_{j,k}$	Viewport tile data size
$U^m_{j,k}$	Marginal tile data size
$U^b_{j,k}$	Background tile data size
$\mathcal{V}_{j,k}(t)$	The video quality score for tile (j,k) at time t

capture the differences in encoding efficiency and visual fidelity across various tile granularities. The term $\mathcal{G}_{j,k}(t)$ represents the granularity level of tile $T_{j,k}$ at time t and is defined as the normalized base quality score produced by the encoder, adjusted for spatio-temporal complexity. This formulation determines whether a tile is treated as fine, medium, or coarse within the segmentation hierarchy. *EDGE360* measures perceptual video quality using the VMAF metric. VMAF is expressed as a composite score (from 0 to 100) that reflects the human visual system's consideration of visual clarity, sharpness, and blockiness [170].

The video quality score $\mathcal{V}^{\chi}_{j,k}(t)_{\chi\in\{f,a,c\}}$ for a given tile $T_{j,k}$ at time t is computed as follows:

$$\mathcal{V}^{\chi}_{j,k}(t)_{\chi\in\{f,a,c\}} = \begin{cases} \mathcal{G}^{f}_{j,k}(t), & \text{if fine granularity,} \\ \mathcal{G}^{a}_{j,k}(t), & \text{if average granularity,} \\ \mathcal{G}^{c}_{j,k}(t), & \text{if coarse granularity.} \end{cases} \tag{4.5}$$

where $G^{f}_{j,k}(t) = \psi_f \cdot \mathcal{G}_{j,k}(t)$, $G^{a}_{j,k}(t) = \psi_a \cdot \mathcal{G}_{j,k}(t)$, and $G^{c}_{j,k}(t) = \psi_c \cdot \mathcal{G}_{j,k}(t)$ represent the weighted VMAF scores for tile (j,k) at time t under *fine*, *average*, and *coarse* granularities, respectively. This equation establishes a relationship between tile size, video quality, and required bandwidth consumption.

To characterize the region-based bitrate demand, the bitrates associated with all tiles $T_{j,k}$ in the region with a certain granularity level χ at time t are aggregated. The total bitrate required to deliver a target perceptual quality for each tile region is defined as:

$$\mathcal{R}^{\chi}(t) = \sum_{j=1}^{J}\sum_{k=1}^{K} \mathcal{R}^{\chi}_{j,k}(t) = \sum_{j=1}^{J}\sum_{k=1}^{K} f^{-1}_{\text{map}}\left(\mathcal{V}^{\chi}_{j,k}(t)\right) \tag{4.6}$$

where, $f^{-1}_{\text{map}}(\cdot)$ denotes the inverse of a quality-to-bitrate mapping function and $\chi \in \{f,a,c\}$ indicates the tile granularity level, respectively. The function $f^{-1}_{\text{map}}(\cdot)$ captures the encoding cost required to achieve a target VMAF

score for a tile at the granularity level χ. Finer tiles require higher bitrates to maintain the same visual quality due to their increased spatial details.

4.3.2 Mapping, Tile Chunk Data Size, and Bandwidth Constraints

As shown in Fig. 4.6 (a), a granularity level $\chi \in \{f, a, c\}$ is associated with a spatial region F_r with $r \in \{v, m, b\}$ representing the viewport, marginal, and background areas, respectively. This mapping allows us to jointly reason about spatial importance and encoding detail. Finer-grained tiles ($r = f$) are allocated to the viewport region F_v, while average and coarse tiles correspond to marginal F_m and background F_b regions, respectively. These aspects form the basis for region-aware bitrate and data size estimation.

Each tile $T_{j,k}$ is allocated a bitrate $\mathcal{R}^r_{j,k}$, where $r \in \{v, m, b\}$ denotes the viewport, marginal, and background regions, respectively. The corresponding tile data size is given by:

$$\mathcal{U}^r_{j,k} = f_{\text{size}}\left(\mathcal{R}^r_{j,k}, T_{j,k}\right) \tag{4.7}$$

Here, $f_{\text{size}}(\cdot)$ is a mapping function that converts bitrate $\mathcal{R}^r_{j,k}$ into data size for tile $T_{j,k}$ based on its encoding granularity and content complexity. This mapping is consistent with the quality-to-bitrate relationship described in Eq. 4.6. Across tile regions, the total data size $\mathcal{U}^r$ over all tiles belonging to region $\mathcal{F}_r$ is:

$$\mathcal{U}^r = \sum_{(j,k)\in\mathcal{F}_r} \mathcal{U}^r_{j,k} \tag{4.8}$$

where, $\mathcal{U}^v_{j,k}$ is the combined data size of tiles $T_{j,k}$ belonging to the viewport $T_{j,k} \in \mathcal{F}_v$, $\mathcal{U}^m_{j,k}$ is the overall data size of the marginal tiles $T_{j,k} \in \mathcal{F}_m$, and $\mathcal{U}^b_{j,k}$ is the total data size of background tiles $T_{j,k} \in \mathcal{F}_b$. This ensures that the most critical area (i.e., viewport region) receives the highest bitrate, while less significant regions (i.e., marginal and background) are allocated progressively lower bitrates to conserve bandwidth usage.

Since the client's bandwidth is constrained, the assigned bitrate across all tiles must satisfy:

$$R^v(t) + R^m(t) + R^b(t) = \sum_{\chi \in \{v,m,b\}} R^\chi(t) \leq \mathcal{W}_n(t) \tag{4.9}$$

$$\sum_{\chi \in \{v,m,b\}} \sum_{k=1}^{K} \sum_{j=1}^{J} \mathcal{R}_{j,k}^{\chi} \leq \mathcal{W}_n(t) \tag{4.10}$$

where $\mathcal{W}_n(t)$ is the available bandwidth for client n at time t (as defined in Eq. 4.1). This constraint ensures that rate allocation respects client-specific limits, enabling the edge node to optimize tile-based streaming while also adapting to network variability.

4.3.2.1 Viewport-Aware Quality Estimation

In *EDGE360*, viewport prediction is coupled with a probability-based viewport-aware quality estimation mechanism that weights each tile's perceived quality by its likelihood of being viewed by the client, thereby guiding region-prioritized adaptation decisions. The viewport quality $\mathcal{Q}_v(t)$ is computed by aggregating the VMAF-based quality scores of all tiles, weighted by their predicted probability of being within the viewport region at time t:

$$\mathcal{Q}_v(t) = \sum_{j=1}^{J} \sum_{k=1}^{K} \mathcal{P}_v(j,k,t) \cdot \mathcal{V}_{j,k}^{\chi}(t) \tag{4.11}$$

Here, $\mathcal{P}_v(j,k,t)$ denotes the probability that tile $T(j,k)$ falls within the client's viewport at time t, and $\mathcal{V}_{j,k}^{\chi}(t)$ represents the VMAF-based video quality score of that tile, already reflecting region-aware encoding based on its assigned granularity. The product $\mathcal{P}_v(j,k,t) \cdot \mathcal{V}_{j,k}^{\chi}(t)$ captures the contribution of each tile to the user's perceived quality, weighted by its likelihood of being viewed.

$\mathcal{P}_v(j,k,t)$ is computed using a Probability Density Function (PDF) based on previous and anticipated client head rotations:

$$\mathcal{P}_v(j,k,t) = \frac{1}{2b} \exp\left(\frac{-|\theta_{j,k} - \hat{\theta}_n(t)|}{b}\right) \tag{4.12}$$

Here, $\hat{\theta}_n(t)$ is the predicted center of the viewport for client n at time t, and $\theta_{j,k}$ represents the angular position of tile $T(j,k)$ in the 360° sphere. The parameter b is the scale factor that controls how quickly the probability decays as the tile moves away from the predicted viewport center, capturing the uncertainty in viewport prediction. This probability model follows similar methodologies found in [22, 171, 172], but instead of a Gaussian-based approach [27], it employs a Laplace distribution to better capture rapid and abrupt head movements in adaptive 360° video streaming.

The viewport-weighted quality in *EDGE360* is therefore computed in Eq. 4.11 by multiplying each tile's VMAF-based quality score $V^{\chi}_{j,k}(t)$ with its predicted viewport probability $P_v(j,k,t)$. The region weighting is introduced through the granularity index χ, where fine, average, and coarse granularities correspond to the viewport, marginal, and background regions, respectively. Thus, tiled-region that are more likely to be viewed and encoded with finer granularity contribute more strongly to $Q_v(t)$, while marginal and background tiled regions contribute less due to lower viewport probability and lower assigned granularity. This means that *EDGE360* does not compute QoE from a uniform average of all tiles; instead, it uses a probability-weighted and region-aware VMAF aggregation. This weighted quality term is then used later directly in the QoE formulation in section 4.3.3.

4.3.2.2 Video Quality Transitions

In *EDGE360*, quality switches often arise due to frequent variations in tile bitrate allocation caused by viewport movement and dynamic network conditions, particularly when viewing VR 360° video streaming. These transitions can degrade the user's QoE, particularly in regions of high visual priority. To ensure smooth visual playback and minimize perceptual discon-

tinuities, *EDGE360* incorporates a quality-switching penalty that accounts for both temporal and spatial variations in tile quality.

$$\Delta\mathcal{Q}_{j,k}(t) = \left|\mathcal{V}^{\chi}_{j,k}(t) - \mathcal{V}^{\chi}_{j,k}(t-1)\right| + \left|\psi(\theta_{j,k},\phi_{j,k}) \cdot \mathcal{V}^{\chi}_{j,k}(t) - \mathcal{Q}^{\text{ref}}_{j,k}(t)\right| \tag{4.13}$$

In Eq. 4.13, the first term quantifies the temporal quality variation in tile $T(j,k)$ by measuring the change in its VMAF-based perceptual quality score $\mathcal{V}^{\chi}_{j,k}(t)$ between two consecutive time steps t and $t-1$. The granularity-aware subscript $\chi \in \{f,a,c\}$ captures the dependency of perceptual quality on tile size (fine, average, and coarse). The second term models the spatial distortion penalty caused by the spherical projection in 360° video. It uses a distortion coefficient $\psi(\theta_{j,k},\phi_{j,k})$ based on the tile's angular coordinates, longitude $\theta_{j,k}$, and latitude $\phi_{j,k}$ relative to the viewer's viewport [173]. Tiles located farther from the viewport center typically suffer more from spherical stretching and compression, which $\psi(\cdot)$ penalizes accordingly. $\mathcal{Q}^{\text{ref}}_{j,k}(t)$ denotes the reference or target quality level assigned to tile $T(j,k)$ at time t, guiding bitrate assignment toward perceptual smoothness. The penalty $\Delta\mathcal{Q}_{j,k}(t)$ contributes to the *EDGE360* reward formulation by suppressing excessive switching and promoting seamless transitions, particularly in perceptually important regions.

4.3.2.3 Buffer Occupancy Dynamics

Maintaining adequate buffer levels is essential for ensuring uninterrupted playback during adaptive 360° video streaming. The buffer occupancy $\mathcal{B}_n(t)$ of client n evolves dynamically based on playback consumption, chunk downloads, and potential rebuffering events. The buffer level at time $t+1$ is estimated by:

$$\mathcal{B}_n(t+1) = \left(\mathcal{B}_n(t) - \frac{\sum_{j=1}^{J}\sum_{k=1}^{K}\mathcal{U}^{r}_{j,k}}{\mathcal{F}_n(t)}\right) + \frac{\kappa_n(t)}{\mathcal{F}_n(t)} - \delta_n(t) \tag{4.14}$$

Here, $\mathcal{B}_n(t)$ denotes the current buffer occupancy in seconds. The term $\sum_{j,k} \mathcal{U}^r_{j,k}$ represents the cumulative size of received tiles for region $r \in \{v, m, b\}$, while $\mathcal{F}_n(t)$ is the video playback rate (in bytes/s), used to normalize both consumption and chunk arrival durations. The term $\kappa_n(t)$ is the size of the newly received chunk, and $\delta_n(t)$ captures any rebuffering delay encountered due to an underflow in buffer content.

In order to ensure prompt bitrate and tile selection adaptations, *EDGE360*'s system must maintain a track of re-buffering and overflow occurrences. The following is the definition of the re-buffering condition for client n at time t:

$$\rho_n(t) = \begin{cases} 1, & \text{if} \quad \mathcal{B}_n(t) - \frac{\sum_{j=1}^{J} \sum_{k=1}^{K} \mathcal{U}^{\chi}_{j,k}}{\mathcal{F}_n(t)} < 0, \\ 0, & \text{otherwise.} \end{cases} \tag{4.15}$$

$\rho_n(t)$ is a binary indicator (0 or 1) denoting whether rebuffering occurred at time t; it is unitless.

A value of $\rho_n(t) = 1$ indicates that playback was interrupted due to buffer depletion, while $\rho_n(t) = 0$ means uninterrupted playback. Although $\rho_n(t)$ flags the occurrence of a stall, its impact on QoE is determined by the product $\rho_n(t)' = \rho_n(t) \cdot \delta_n(t)$, where $\delta_n(t)$ captures the stall duration. This modeling is especially useful in multi-client scenarios to reduce both the frequency and duration of rebuffering events.

4.3.2.4 Delay Calculation $\mathcal{D}_n(t)$

In adaptive bitrate streaming schemes such as $EDGE360$, latency plays a crucial role in determining the overall QoE for clients. The delay encountered by the client n at time t is expressed as $\mathcal{D}_n(t)$, determined by incorporating both the processing and transmission delays.

The processing delay $\mathcal{L}_{\text{proc}}(t)$ captures the time taken by the edge server to handle a chunk request, including tile checks and cache lookups. If tiles are missing, retrieval from the central server adds a delay. The transmission

delay $\mathcal{L}_{\text{trans}}(t)$ reflects the time to deliver tiles over the network based on their sizes and the client's available bandwidth. The total delay is given by:

$$\mathcal{D}_n(t) = \mathcal{L}_{\text{proc}}(t) + \mathcal{L}_{\text{trans}}(t) \tag{4.16}$$

The transmission delay $\mathcal{L}_{\text{trans}}(t)$ quantifies the time needed to deliver the requested video tiles to the client, which is determined by the cumulative data size of the tiles and the client's available bandwidth. Specifically, in *EDGE360*, each tile-aware data size $\mathcal{U}_{j,k}^{\chi}$ is derived from the selected bitrate and region class $\chi \in \{v, m, b\}$ corresponding to the viewport, marginal, and background regions, respectively.

Accordingly, the total end-to-end delay for client n at time t is given by:

$$\mathcal{D}_n(t) = \mathcal{L}_{\text{proc}}(t) + \frac{\sum\limits_{\chi \in \{v,m,b\}} \sum\limits_{j=1}^{J} \sum\limits_{k=1}^{K} \mathcal{U}_{j,k}^{\chi}}{\mathcal{W}_n(t)} \tag{4.17}$$

This approach ensures that delay computation is strongly connected with region-specific priority and bitrate selection. It represents the delay caused by sending differing quality tiles, allowing the *EDGE360* framework to make better streaming decisions in constrained network circumstances.

4.3.3 QoE Calculation

In the *EDGE360* framework, client-specific QoE estimation is essential for assessing streaming performance. The QoE model integrates factors such as effective tile quality, viewport enhancement, quality switches, rebuffering, and latency, key to ensuring a smooth and immersive experience under network variability. These metrics support fairness and effectiveness in multi-client scenarios. For client n at time t, the estimated QoE is defined as:

$$\mathcal{Q}oE_n(t) = \mathcal{Q}_v(t) - \alpha \cdot \Delta\mathcal{Q}_{j,k}(t) - \beta \cdot \rho_n(t)' - \zeta \cdot \mathcal{D}_n(t) \tag{4.18}$$

Here, $\mathcal{Q}_v(t)$ represents the aggregated perceptual quality for the predicted viewport tiles, as introduced in Eq. 4.11. Eq. 4.13, $\Delta\mathcal{Q}_{j,k}(t)$ tracks the variance in tile quality between segments and is penalized by α. Quality stability reduces visual interruptions, improving the viewing experience [174].

As defined in Eq. 4.15, $\rho_n(t)$ indicates a rebuffering event for client n at time t, with its duration given by $\delta_n(t)$. Their product models the stall penalty in QoE, weighted by β to discourage frequent and prolonged playback interruptions. Latency $\mathcal{D}_n(t)$ was introduced in Eq. 4.17 as the sum of transmission delay and processing delay and is weighted by ζ. This term highlights the significance of response time in adaptive streaming, particularly in a multi-client immersive environment. To evaluate the system-wide QoE level, the aggregated QoE was defined as $\hat{\mathcal{QoE}}_n(t)$, measuring the overall QoE across all clients, i.e., $\hat{\mathcal{QoE}}_n(t) = \frac{1}{N}\sum_{n=1}^{N}\mathcal{QoE}_n(t)$. This metric gives a system-wide (global) perspective on video quality and adaptation efficiency.

4.3.4 Fairness Assessment

To ensure fairness in resource allocation among multiple clients, Jain's Fairness Index is used as a well-established metric to quantify the equality of QoE across all clients. For $\mathcal{N}$ clients, the fairness index at time t is defined as:

$$\hat{\mathcal{F}}_{\text{Jain}}(t) = \frac{\left(\sum_{n=1}^{\mathcal{N}}\mathcal{QoE}_n(t)\right)^2}{\mathcal{N}\cdot\sum_{n=1}^{\mathcal{N}}\mathcal{QoE}_n^2(t)} \tag{4.19}$$

Here, $\mathcal{QoE}_n(t)$ denotes the perceived quality of experience of client n at time t, which incorporates viewport-aware video quality, rebuffering impact, delay penalties, and quality switches. A Jain's index value close to 1 indicates an equitable QoE distribution among clients, whereas lower values reflect increasing disparity.

This formulation aligns with *EDGE360*'s goal to not only optimize individual client performance but also ensure system-wide fairness in perceived video quality, avoiding resource bias and improving overall user satisfaction across diverse network and client conditions [175, 176].

4.3.5 *EDGE360* Reward Function

The *EDGE360* reward function is introduced to help optimise each client's QoE while preserving fairness across multiple clients in a resource-shared network. The reward function is preferable for adaptive streaming in a multi-client setting as it takes into account the QoE of each client, the consistency of QoE among clients, and fairness. The compact reward function for client n at time t is as follows:

$$\mathcal{R}_n(t) = \mathcal{Q}oE_n(t) - |\varepsilon| \cdot \frac{\left|\mathcal{Q}oE_n(t) - \mathcal{Q}oE_n(t)^k{}_{\text{avg}}\right|}{\mathcal{Q}oE_{\max,n}(t) - \mathcal{Q}oE_{\min,n}(t)} + \lambda \cdot \hat{\mathcal{F}}_{\text{Jain}}(t), \forall k = 1, 2, \ldots, N \tag{4.20}$$

In this context, ε functions as a local fairness penalty term, which quantifies the divergence of client n's instantaneous QoE from the moving average QoE of other clients, denoted as $\mathcal{Q}oE_n(t)^k{}_{\text{avg}}$. This term imposes penalties for significant discrepancies between individual clients within short temporal intervals, and is standardized using the observed maximum and minimum QoE values for the client, represented by $\mathcal{Q}oE_{\max,n}(t)$ and $\mathcal{Q}oE_{\min,n}(t)$, respectively. Furthermore, a global fairness measure across all clients is imposed via the term $\hat{\mathcal{F}}_{\text{Jain}}(t)$, specifically Jain's Fairness Index, which is computed based on system-wide QoE distributions at time t and weighted according to λ. While the local penalty promotes fairness among proximate clients within a rolling time window, the global term ensures prolonged equity in bandwidth distribution across all clients.

Together, these two fairness components allow *EDGE360* to achieve both short-term fairness responsiveness and long-term global balance, main-

Algorithm 1: Part 1: Input/Output Processing and Hierarchical Tile Selection with Bitrate Allocation

Input : Predicted Head Rotation $\mathcal{F}$, divided into: $\mathcal{F}_v$ (Viewport), $\mathcal{F}_m$ (Marginal), $\mathcal{F}_b$ (Background), Network Throughput $W_n(t)$, Buffer $B_n(t)$, Available Bitrates $\mathcal{R} = \{R_1, \ldots, R_M\}$, NN Inputs.

Output : Tile set $T_{j,k} \in T = \{T^v, T^m, T^b\}$ with optimized $\mathcal{R}^{\chi}_{j,k}, \forall \chi \in \{v, m, b\}$.

Initialization: $T^v = \emptyset$, $T^m = \emptyset$, $T^b = \emptyset$.;

foreach *tile* $T_{j,k}$ **do**
  if $T_{j,k} \cap \mathcal{F}_v \neq \emptyset$ **then**
    $T_{j,k} \mapsto T^v$, $\mathcal{R}^v_{j,k} \leftarrow R_v \in \mathcal{R}$;
  end
  if $T_{j,k} \cap \mathcal{F}_m \neq \emptyset$ **then**
    if $\mathcal{G}_{j,k}(t) = a$ **then**
      $T_{j,k} \mapsto T^m$, $\mathcal{R}^m_{j,k} \leftarrow R_m \in \mathcal{R}$;
    end
    if $\mathcal{G}_{j,k}(t) = c$ **then**
      Subdivide$(T^b_{j,k}) \mapsto T^m_{j,k}$;
    end
  end
  if $T_{j,k} \cap \mathcal{F}_b \neq \emptyset$ **then**
    if $\mathcal{G}_{j,k}(t) = c$ **then**
      Subdivide$(T^b_{j,k}) \mapsto T^m_{j,k}$;
    end
    if $\mathcal{G}_{j,k}(t) = a$ **then**
      Subdivide$(T^m_{j,k}) \mapsto T^v_{j,k}$;
    end
    $T_{j,k} \mapsto T^b$, $\mathcal{R}^b_{j,k} \leftarrow R_b \in \mathcal{R}$;
  end
end

Tile Merging for Compression:;
if *all* $T_{j,k} \in T^v$ *in a region belong to the same subset* **then**
  Merge them into $T^m_{j,k}$;
end
else if *all* $T_{j,k} \in T^m$ *in a region belong to the same subset* **then**
  Merge them into $T^b_{j,k}$;
end
else
  Keep the current tiling structure;
end

Algorithm 1: Part 2: State Preparation, *EDGE360* Evaluation, and Cache Management

State Preparation:
Local: $W_n(t)$ (Eq. 4.1), $V_{j,k}(t)$, $G_{j,k}(t)$ (Eq. 4.5), $\Delta\mathcal{Q}_{j,k}(t)$ (Eq. 4.13), $B_n(t)$ (Eq. 4.14).
Global: $P_v(j,k,t)$ (Eq. 4.4), $\mathcal{Q}o\bar{E}_n(t)$ (Eq. 4.18), $\rho_n(t)$ (Eq. 4.15), $D_n(t)$ (Eq. 4.17)

Evaluation: Central critic combines states to evaluate $Q(s,a)$;
Actors choose $\mathcal{R}^{\chi}_{j,k}$ for T^v, T^m, T^b to maximize QoE;

Dynamic Bitrate Adaptation:
if $W_n(t) < W_{th,min}$ **then**
  if $\mathcal{R}^b_{j,k} > R_{min}$ **then**
    $\mathcal{R}^b_{j,k} \leftarrow \mathcal{R}^b_{j,k} - \Delta R$
  end
  if $\mathcal{R}^m_{j,k} > R_{min}$ **then**
    $\mathcal{R}^m_{j,k} \leftarrow \mathcal{R}^m_{j,k} - \Delta R$
  end
  if $\mathcal{R}^v_{j,k} < R_{max}$ **then**
    $\mathcal{R}^v_{j,k} \leftarrow R_{\max}$
  end
end

Edge Cache Integration:
foreach $T_{j,k} \in T = \{T^v, T^m, T^b\}$ **do**
  if $T_{j,k}$ *is available in the edge cache* **then**
    Retrieve $T_{j,k}$ with its assigned quality and corresponding data size $U_{j,k}$ (Eq. 4.8);
  end
  else
    Request $T_{j,k}$ from the video server based on its region:
    - **Viewport** (T^v): High quality, data size $U^v_{j,k}$ (Eq. 4.8)
    - **Marginal** (T^m): Moderate quality, data size $U^m_{j,k}$ (Eq. 4.8).
    - **Background** (T^b): Low quality, data size $U^b_{j,k}$ (Eq. 4.8).

    Store the retrieved $T_{j,k}$ in the edge cache.;
  end
end
return $T = \{T^v, T^m, T^b\}$ *with bitrates* $\mathcal{R}^{\chi}_{j,k}$

taining efficient and fair system operation under dynamic network conditions.

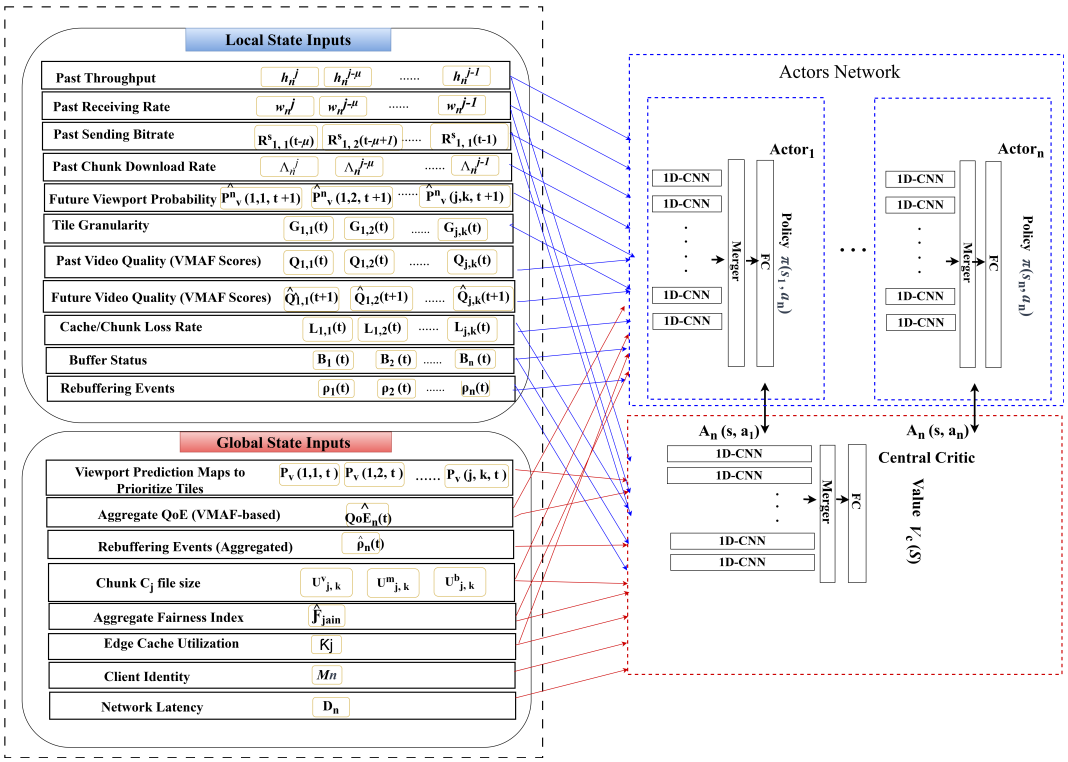


Figure 4.7: Neural Network Architecture for *EDGE360* Adaptive 360° VR Streaming Using COMA

4.3.6 Adaptive Bitrate Allocation for Predicted Regions in *EDGE360*

In *EDGE360*, efficient bitrate allocation is critical to ensuring high-quality user experiences while optimizing bandwidth usage in 360° video streaming. To achieve this, the video frame is divided into three predicted regions: the viewport region (F_v), the marginal region (F_m), and the background region (F_b). These regions are prioritized according to their relevance to the FoV of the user. The adaptive tiling mechanism dynamically assigns optimized bitrates ($\mathcal{R}^v_{j,k}$, $\mathcal{R}^m_{j,k}$, and $\mathcal{R}^b_{j,k}$) to tiles in each region to achieve a balance between visual quality and bandwidth efficiency. The set of available bitrates, denoted as $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_M\}$, represents the discrete bitrate levels available for allocation to tiles in the ($\mathcal{F}_v$), ($\mathcal{F}_m$), and ($\mathcal{F}_b$) regions of a 360° video frame.

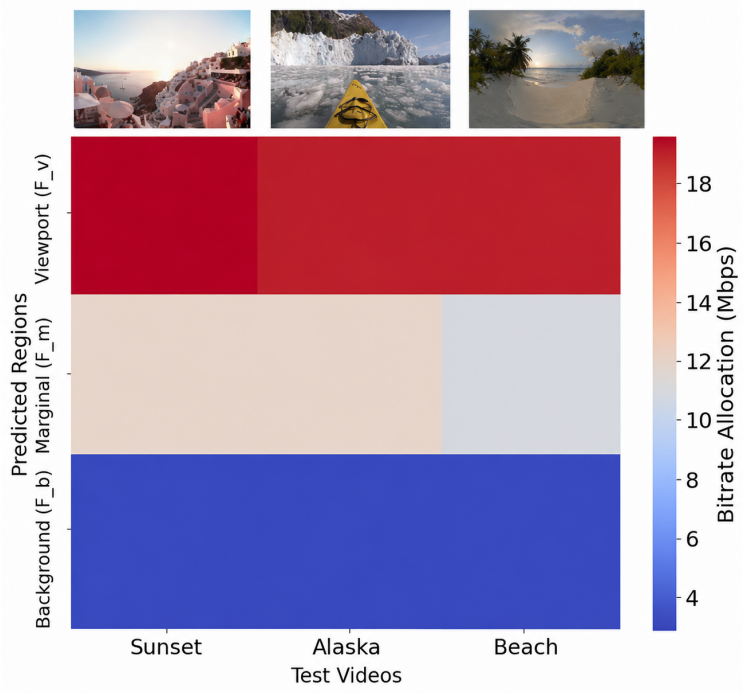


Figure 4.8: *EDGE360*'s adaptive bitrate distribution among predicted regions (viewport, marginal, and background) for three test videos.

The complete tile set in *EDGE360* is represented as $T = \{T^v, T^m, T^b\}$, where each subset corresponds to a specific region based on its priority and relevance to the user's FoV. The subset T^v consists of tiles belonging to the viewport region ($\mathcal{F}_v$), which have the highest priority. These tiles are directly within the user's focus and require the highest bitrates to ensure optimal visual quality. The subset T^m represents the tiles in the marginal region ($\mathcal{F}_m$), which are adjacent to the viewport but have a moderate priority. These tiles are allocated intermediate bitrates to balance quality and resource efficiency. Finally, the subset T^b includes tiles in the background region ($\mathcal{F}_b$), which have the lowest priority as they fall outside the user's immediate viewing area. These tiles are assigned the lowest bitrates, conserving bandwidth while maintaining a minimal level of quality. This hierarchical classification ensures efficient allocation of resources, prioritizing user experience in the most critical areas while optimizing bandwidth usage for less significant regions. As shown in Fig. 4.6, the viewport region $\mathcal{F}_v$ (red ellipse) receives fine-grained, high-resolution tiles to prioritize user focus. The marginal region $\mathcal{F}_m$ (yellow rectangle) is streamed with moderate gran-

ularity, while the background $\mathcal{F}_b$ uses coarse tiles to conserve bandwidth. Fig. 4.8 illustrates *EDGE360*'s bitrate allocation for three test videos (*Sunset*, *Alaska*, and *Beach*), assigning 18–20 Mbps to F_v, 8–12 Mbps to F_m, and 2.5–4 Mbps to F_b balancing visual quality and bandwidth efficiency across regions.

4.4 *EDGE360* Rate Adaptation and Problem Formulation

In *EDGE360* adaptive streaming, each client aims to choose the optimal tile bitrates during each adaptation span perpetually, maximizing aggregated QoE across all video tiles and delivering an immersive 360° viewing experience. *EDGE360* for rate adaptation, using the COMA framework [177] to optimize video streaming in 360° adaptive streaming in multiple client scenarios. The *EDGE360* design attempts to maximize aggregated QoE across multiple clients by considering individual client demands while also assuring system-wide fairness. This includes tackling constraints such as dynamic network circumstances, viewport dynamics, and quality trade-offs. Fig. 4.7 depicts the neural network architecture for *EDGE360*'s adaptive 360° VR streaming. COMA employs both local and global state inputs to optimise adaptive streaming action via a MARL architecture. The local state includes client-specific parameters, including previous throughput, receiving rate, transmitting bitrate, viewport probability, and video quality ratings. The global state includes aggregated system-level metrics such as viewport prediction maps, aggregate QoE, chunk file size, fairness index, delay, and edge cache utilization.

The *EDGE360* optimization problem may be represented quantitatively as follows:

Problem P1:

$$\arg\max \sum_{n\in[1,N]} \mathcal{Q}oE_n$$

EDGE360's aim for problem **P1** is to determine the optimum bitrate and tile quality selection approach with optimal policy $\pi^* : S \cdot A \rightarrow [0, 1]$, that maximizes the projected long-term discounted QoE while taking into consideration the limitations of viewport and tiled-based 360° video streaming in multi-client scenarios.

4.4.1 Foundation of MDP

A discrete-time finite Markov Decision Process (MDP) represented by $\mathcal{M}$, is defined by a 7-tuple $(\mathcal{S}, \{\mathcal{O}_n\}_{n\in\mathcal{N}}, \{\mathcal{A}_n\}_{n\in\mathcal{N}}, f, \{r_n\}_{n\in\mathcal{N}}, \gamma, \rho_0, H)$, which serves as the foundation for modeling decision-making in dynamic environments where multiple agents engage with both the environment and each other. Here, $\mathcal{N}$ represents the set of agents, and each agent makes individual decisions while interacting with a shared environment. This formulation indicates the action space $\mathcal{A}_n$ of agent n and the state space of the environment by $\mathcal{S}$ as they encompass both global and local information that influences decision-making. However, in multi-agent systems, each agent usually has access to partial observations of this state space, which is denoted by $\{\mathcal{O}_n\}_{n\in\mathcal{N}}$, where $\mathcal{O}_n$ represents the specific set of observations available to agent n. Such partial accessibility translates the MDP into a Partially Observable Markov Decision Process (POMDP), which reflects the actual constraints that agents encounter in real-world systems. $\mathcal{A}_n = \left\{a_n^v, a_n^m, a_n^b\right\}$ represents the set of actions the agent n can take and a_n^v, a_n^m, a_n^b are actions for viewport, marginal, and background tiles, respectively. The transition dynamics are defined by the deterministic function $f : \mathcal{S} \times A \rightarrow S$, which specifies the next state s' to which the environment will transition when an action a is performed in the current state s. The f accounts for joint actions of all agents (clients) and actions that affect the global state, especially regarding network throughput, buffer states, and tile quality switches. The reward for taking an action a in state s is given by the bounded reward function $r : S \times A \rightarrow \mathbb{R}$. It incorporates the discount factor $\gamma(0 \leq \gamma \leq 1)$ to balance the importance of current and future rewards. The initial state dis-

tribution can be determined using the formula $\rho_0 : S \to \mathbb{R}^+$, which reflects the initial conditions of network throughput, buffer states, and video quality across all clients, where H represents the horizon length or the length of the decision-making trajectory.

Furthermore, since each *EDGE360* agent perceives a partial view of the global state $\mathcal{S}$, the framework is a POMDP [178]. COMA's centralized critic calculates counterfactual baselines to influence individual agents' actor strategies, maximizing their value to the shared global reward. The objective of the POMDP framework for the agent n is to find the optimal policy $\pi_n^* : S \to A$ that maximizes the cumulative long-term discounted reward. This can mathematically be represented as $R = \sum_{t=0}^{H} \gamma^t \cdot r_t$, where r_t is the reward at time t and γ^t is a discount factor that balances the significance of immediate versus future rewards. Consequently, the edge-based optimal bitrate selection problem can be redesigned as:

Problem **P2:**

$$\max_{\{\pi_n\}_{n\in\mathcal{N}}} \mathbb{E}\left[\sum_{t=0}^{H} \gamma^t \cdot R_t \mid \{\pi_n\}_{n\in\mathcal{N}}\right] \tag{4.21}$$

The P2 goal is to collaboratively optimize the policies of all agents in the *EDGE360* framework to maximize the predicted cumulative discounted global reward across all clients in a multi-agent scenario.

4.4.2 State Formulation for *EDGE360* Adaptive 360° Streaming

In the multi-client *EDGE360*, the MDP is modified to a Markov game [179] to handle interactions among multiple agents and the environment. Each client behaves as an independent actor, requesting video tiles, and their collective decisions impact global state changes. The complicated dynamics of adaptive 360° video streaming, such as viewport prediction, tile-based adaptation, and limited network resources, require a precise state definition. The state of a client n requesting a video chunk C_i is defined as follows:

$$s_j^n = \left\{h_j^n, \omega_j^n, \Lambda_j^n, u_{j,k}^n, d_{j,k}^n, \mathcal{V}_{j-1}^n, B_j^n, \Delta\mathcal{Q}_{j,k}^n, P_v^n(j,k,t)\right\} \tag{4.22}$$

This state represents numerous facets of the client's interaction with the system. In a multi-client scenario, each client's local state s_j^n serves the joint global state $S_j = \{s_j^1, s_j^2, \ldots, s_j^N\}$. COMA's centralized critic utilizes S_j to optimize resource allocation, guarantee fairness, and maximize the client's QoE. The historical network throughput, presented as $h_n^j = \{h_n^{j-\mu}, \ldots, h_n^{j-1}\}$, incorporates the average throughput measurements over the past μ chunks, representing recent network performance to assist bitrate adaptation. Similarly, $\omega_n^j = \{\omega_n^{j-\mu}, \ldots, \omega_n^{j-1}\}$ contains the downloading instances of the previous μ chunks, providing insights regarding transmission efficiency and network variability. The historical data retrieval rate is expressed as $\Lambda_j^n = \{\Lambda_{j-\mu}^n, \ldots, \Lambda_{j-1}^n\}$, indicating the pace at which prior chunks were downloaded.

This metric helps to monitor network consistency and guide bitrate adaptation, resulting in smooth playback while preventing unnecessary rebuffering. The required chunk C_i tile sizes are represented as $u_{j,k}^n = [u_{j,k}^v, u_{j,k}^m, u_{j,k}^b]^T$, where $u_{j,k}^v$, $u_{j,k}^m$, and $u_{j,k}^b$ correspond to the viewport, marginal, and background regions, respectively. Each tile size is set for multiple bitrate levels, such as $u_{j,k}^v = \{u_{j,k}^v(R_1), \ldots, u_{j,k}^v(R_M)\}$, which enables the solution to change quality granularity swiftly. Similarly, $d_{j,k}^n = [d_{j,k}^v, d_{j,k}^m, d_{j,k}^b]^T$ records spherical distortions associated with these tiles, indicating spatial quality drop between regions.

The past chunk's video quality denoted as V_n^{j-1}, serves as a reference point to ensure quality consistency. The buffer occupancy metric B_n^j assesses the client's playback buffer during a chunk request C_i and is crucial for smooth playing and avoiding re-buffering occurrences. The quality switching penalty $\Delta\mathcal{Q}_{j,k}^n$ adjusts for disruptive quality variations, resulting in seamless transitions between video quality levels. The viewport probability mapping model $P_n^v(j,k,t)$ anticipates the likelihood of tile (j,k) being viewed based on the client's viewport trajectory, allowing for prioritized resource allocation.

4.4.3 Actor-Central Critic Policy Training for *EDGE360*

In *EDGE360*, the actor and central critic network leverage the COMA approach to handle the challenges of multi-client adaptive streaming in a partially observable setting. The technique merges client-specific decisions with global optimization via a POMDP framework, resulting in an effective resource allocation and improved QoE for all rich users.

4.4.3.1 Actor Policy Approximation

The actor policy for each client is parameterized by φ_n, representing the weights of the actor network. This network produces a probability distribution over possible actions, such as bitrate allocation and tile quality selection. The distribution is computed using a softmax function:

$$\pi_n(a_n \mid s) = \text{Softmax}(\phi(s; \varphi_n)) \tag{4.23}$$

where, $\phi(\cdot)$ is the feature extraction function that transforms the observable state s into a latent representation, while φ_n denotes the trainable parameters of the actor network for client n. This allows the policy to dynamically adapt to client-specific requirements, such as viewport movement, network fluctuations, and buffer conditions, while being influenced by the shared global environment. global environment.

4.4.3.2 Central Critic State-Value Function

The state-value function, which determines the predicted cumulative reward setting up from a particular global state s, is assessed by the centralized critic. It is represented as:

$$V_c(s) = \mathbb{E}\left[\sum_{t=0}^{H} \gamma^t R(s,a)\right], \tag{4.24}$$

where $R(s,a)$ is the global reward aggregating QoE tracks across all clients, γ is the discount factor harmonising current and future rewards, and H is

the decision horizon. This function provides a global view by taking into account all agents' combined states and actions, coordinating their actions with system objectives.

4.4.3.3 Advantage Function for Central Critic

In order to enhance actor training, the central critic evaluates the advantage function, which compares the effectiveness of an agent's action to a counterfactual baseline. This is stated as follows:

$$A_n(s, a_n) = Q(s, a) - \sum_{n=1}^{N} \pi_n(a_n \mid s) Q(s, a^{-n}) \tag{4.25}$$

Here, a denotes the joint action vector of all agents, i.e., $a = (a_1, a_2, \ldots, a_N)$, while a^{-n} represents the set of actions taken by all agents except agent n. The state-action value function is represented by $Q(s, a)$, and the counterfactual baseline by $Q(s, a^{-n})$ which minimizes the contribution of agent n by marginalizing over its actions. The counterfactual baseline $Q(s, a^{-n})$ estimates the global reward by marginalizing agent n's action while keeping other agents' actions fixed. This formulation is crucial for *EDGE360* with multiple agents, as the advantage function enables each agent to focus on optimizing its unique contribution to the global reward, guaranteeing fairness and efficient resource allocation for each client.

4.4.3.4 Actor Policy Gradient Update

In *EDGE360*, the actor policy gradient update is critical for enhancing each client's actor network. The policy gradient approach updates the actor networks, which promotes actions that result in higher expected rewards. The actor policy gradient update can be expressed as:

$$\nabla_{\varphi_n} J(\varphi_n) = \mathbb{E}\left[A_n(s, a_n) \nabla_{\varphi_n} \log \pi_n(a_n \mid s)\right] \tag{4.26}$$

Here, the function $A_n(s, a_n)$ measures the comparative advantage of an action compared to its baseline. This allows the actor network to maximize its distinct value to the shared global reward. This recurrent refinement guarantees that the policy responds dynamically to variations in network circumstances, such as bandwidth fluctuations and viewport trajectories, while also optimizing the QoE for each client. This strategy combines local decision-making with global system objectives by using COMA's multi-agent framework.

4.4.3.5 Entropy Regularization

To stimulate exploration while avoiding early convergence to suboptimal policies, the actor objective incorporates entropy regularisation as follows:

$$H(\pi_n) = -\sum_{a_n} \pi_n(a_n \mid s) \log \pi_n(a_n \mid s) \tag{4.27}$$

where $H(\pi_n)$ measures the randomness of the policy's action decision.

A higher entropy number implies a more exploratory perspective, whereas a lower entropy indicates a more predictable behavior. Entropy regularization encourages diversity in action choices, ensuring that agents explore different bitrate allocation and tile quality ways across the viewport, marginal, and non-viewport regions. The updated actor objective, which includes entropy regularization, becomes: The updated actor objective gradient, denoted by $\nabla_{\varphi_n} J'(\varphi_n)$, incorporates an entropy regularization term scaled by the tunable parameter η, which balances exploration and exploitation. This regularized formulation is especially important in *EDGE360*, where agents must adapt to dynamic network states and diverse viewport trajectories in a multi-client, tile-based streaming environment.

4.4.3.6 Critic Network Loss Function

The critic network is trained to reduce temporal difference (TD) errors, which improves the accuracy of value assessments. The loss function is stated as follows:

$$L(\Psi) = \mathbb{E}\left[\left(V_c(s) - \left(r + \gamma V_c(s')\right)\right)^2\right], \tag{4.28}$$

Here, $V_c(s)$ is the predicted value of the current state s, r is the immediate reward obtained following action, and $V_c(s')$ is the estimated value of the next state s'. The discount factor (γ) determines the relevance of future rewards compared to immediate ones. By minimizing this loss function, the critic network learns to deliver precise value estimations, allowing the centralized critic in *EDGE360*'s COMA system to develop effective counterfactual advantage functions.

EDGE360's actor central critic system integrates these components, i.e. $\pi_n(a_n \mid s)$, $V_c(s)$, $A_n(s, a_n)$, $\nabla_{\varphi_n} J(\varphi_n)$, $H(\pi_n)$, $\nabla_{\varphi_n} J'(\varphi_n)$, $L(\Psi)$, aligning client-specific edge-based adaptations with global QoE optimization. The COMA-based centralized critic ensures fairness, while actor networks dynamically adapt to local conditions, allowing for efficient and adaptable 360° video streaming in multi-agent settings.

Algorithm 1, parts 1 and 2, present the detailed procedure of the proposed viewport-aware adaptive tiling scheme in *EDGE360*. The algorithm returns the tile set $T = \{T^v, T^m, T^b\}$, where each subset corresponds to a different viewport-aware region: T^v (viewport tiles) forms the predicted user viewport, T^m (marginal tiles) covers adjacent areas with moderate granularity, and T^b (background tiles) consists of lower-priority tiles beyond the predicted FoV. These selected tiles collectively reconstruct a complete video chunk, ensuring optimized visual quality and efficient bandwidth utilization in *EDGE360*.

This approach ensures efficient bandwidth utilization while maintaining high-quality streaming for the user's viewport. By dynamically adjusting bitrates and calculating total file sizes based on tile regions and priorities,

EDGE360 achieves an optimal balance between visual quality and resource constraints. This method enables the framework to adapt to varying network conditions and user demands dynamically, ensuring improved QoE.

4.5 Implementation and Evaluation

4.5.1 Simulation Settings

To evaluate the performance of the proposed *EDGE360* algorithm, a Python-based simulator with the PyTorch framework was built [180]. The simulator was built on an Ubuntu 24.04 machine with a 64-bit Intel Core i7-8550 U CPU (1.8 GHz quad-core, turbo boost up to 4GHz) and 16 GB RAM.

As already mentioned, within the intelligent edge node, the *edge-side multi-agent rate adaptation module* uses a COMA-based DRL technique, which employs a single centralized critic network as well as several independent actor networks, with each client in a multi-client streaming situation acting as an autonomous agent. Unlike independent DRL approaches like A3C and DDQN, which rely on decentralized learning, COMA combines centralized training and decentralized execution to provide optimum collaborative policy acquisition for viewport-based bitrate selection while remaining fair between different clients. To simulate realistic multi-client streaming scenarios aligned with 5G network conditions, the proposed adaptation mechanism executed at the intelligent edge node is evaluated with an increasing number of clients: 16, 32, and 64 concurrent users, respectively. Each actor network is mapped to a unique client and independently performs tile-wise bitrate allocation decisions. The architecture supports decentralized execution, allowing each actor to operate with local observations that include viewport probability maps, buffer states, and past bitrate actions. Simultaneously, the client's observation space inherently captures the viewport, marginal, and background regions, enabling fine-grained control over spatial adaptation. This region-aware, modular structure not only supports accurate per-user decisions but also facilitates scalable operation,

as validated in scalability experiments. Even with 64 clients, the edge-side adaptation mechanism maintains acceptable levels of QoE, fairness, and rebuffering, demonstrating its robustness in high-density streaming environments.

Each network of client-specific actors selects bitrates based on local information such as viewport probability visualizations, network performance, buffer occupancy, and history bitrate decisions, where the actor policy gradient update $\nabla_{\varphi_n} J(\varphi_n)$ Eq. (4.26) optimizes policy learning by adjusting actions that result in higher expected rewards. These actor networks consist of an input layer and two fully connected (FC) layers, with the first layer comprising 128 neurons and the second containing 64 neurons, both with ReLU activation. The output layer implements a Softmax activation function $\phi(s; \varphi_n)$ Eq. (4.23) to generate action probabilities that decide the bitrate of each client's tiles in the viewport, marginal, and background regions. The critic examines the state action value function $Q(s, a)$ and compares the impact of each agent's action with a counterfactual baseline $Q(s, a^{-n})$ using an advantage function $A_n(s, a_n)$ Eq. (4.25) rather than directly calculating a Q value in a linear activation output layer. The counterfactual baseline marginalizes the contribution of the agent n's to global reward by marginalizing its activities. Taking into account future rewards over a broader horizon, the discount factor set to $\gamma = 0.99$ ensures long-term optimization. Using the Adam optimizer for effective convergence, the learning rate is set at 1×10^{-4} for actor networks and 5×10^{-3} for critic networks. To avoid early convergence to suboptimal policies, the entropy regularisation factor $H(\pi_n)$ Eq. (4.27) is initialized at 0.99 and decays exponentially at a rate of 0.995 every 10^3 iterations in the COMA DRL framework for exploration.

4.5.2 Viewport Dataset

To model the user viewport dynamics in *EDGE360*, a publicly available viewport dataset was used that includes head rotation traces from 153 volun-

teers while watching various 360° videos [181]. These traces, sampled at 10 Hz (i.e., 10 samples per second), offer high temporal resolution to support fine-grained head movement prediction.

The *viewport prediction module* used by the intelligent edge node adopts an LSTM-based deep learning model due to its ability to learn long-range temporal dependencies inherent in user head motion. In contrast to CNN-based methods that often struggle with temporal modeling [182], the LSTM network captures sequential user interactions effectively and enables forecasting of the future viewport center using historical rotation data. Additionally, *EDGE360* integrates a Laplace-based probabilistic viewport expansion scheme, which estimates tile-wise probability around the predicted center, exponentially decaying with distance. This allows for perceptually guided bitrate allocation, favoring high-probability tiles.

Although attention-based and transformer-based architectures have recently shown strong predictive capabilities in such domains [183], LSTM was chosen for its balance of accuracy and computational efficiency in real-time applications. Transformers, while potentially more accurate, often involve higher inference latency and parameter overhead, making them less practical for edge-based scenarios with resource constraints.

The prediction model was quantitatively evaluated using the Mean Absolute Error (MAE), and viewport hit rate metrics. The average MAE across all test users was 5.0°, and the average viewport hit rate exceeded 90%, indicating strong predictive accuracy. These metrics validate the use of the LSTM model in combination with the Laplace-based expansion strategy employed in *EDGE360*.

Furthermore, Fig. 4.9 quantifies the system's sensitivity to viewport prediction error through an ablation study. To assess the effect of degraded viewport prediction accuracy, controlled Gaussian noise was injected into the predicted viewport coordinates, and the resulting impact on *EDGE360* performance was analyzed. The LSTM-based viewport predictor used in the framework achieved a MAE of 5.1°, a hit rate within 5° of 89%, and an

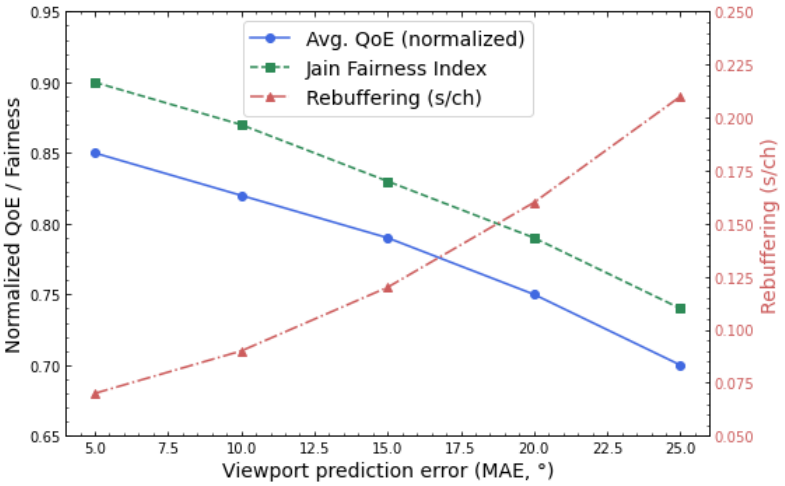


Figure 4.9: Sensitivity analysis of *EDGE360* for viewport prediction error, showing the impact on normalized QoE, Jain's fairness index, and average rebuffering time per chunk.

F1-score of 0.92 for FoV classification. As the injected MAE increases from 5° to 25°, the normalized average QoE steadily declines from 0.85 to 0.70, indicating an 18% loss in perceptual quality due to misallocated bitrate and increased stall events. Jain's fairness index similarly falls from 0.90 to 0.74, demonstrating imbalanced quality across users, as the COMA reward function dynamically reallocates bandwidth to clients with poor predictions via the deviation term $|\varepsilon|$. Rebuffering emerges as the dominant penalty beyond 15° MAE, where stall time per chunk increases from 0.07 s to 0.21 s, surpassing the penalty from quality switches.

This comprehensive degradation across all three dimensions, QoE, fairness, and rebuffering, highlights the pivotal role of accurate, low-latency viewport prediction. *EDGE360* maintains high QoE and fairness as long as the prediction MAE remains below 20°, validating the choice of an efficient LSTM + Laplace PDF model over heavier attention-based alternatives in real-time VR scenarios.

These results show that viewport prediction accuracy is an important factor in *EDGE360*'s performance. However, the conclusions remain valid under moderate prediction uncertainty because *EDGE360* maintains high QoE and fairness when the prediction MAE remains below approximately 20°. The marginal region also helps absorb small viewport shifts by pro-

viding medium-quality tiles around the predicted viewport. Therefore, the sensitivity analysis does not weaken the main conclusion; rather, it defines the operating range in which *EDGE360*'s viewport-aware adaptation remains robust and explains how performance degrades when prediction error becomes large.

4.5.3 Rich Video Dataset and Network Traces

A public video database [184] was employed to provide VR 360° videos ranging in length from 37 to 668 seconds, have a frame rate of 30 frames per second, and contain highly dynamic and static details. To provide adaptive 360° video streaming, the DASH360 tool was deployed, which divides each video into spatially adaptable tiled representations while optimizing bandwidth allocation based on viewport likelihood. DASH360 separates videos into three levels of tile granularity: coarse-grained (1024 × 1024 pixels), medium-grained (512 × 512 pixels), and fine-grained (256 × 256 pixels) to ensure effective bitrate distribution across regions. After tiling, the videos are encoded in HEVC/H.265 employing FFmpeg [185] and transformed into 2-second MPEG-DASH chunks with MP4Box [186]. Tile of each granularity is then pre-transcoded into VMAF-optimized bitrates based on its probability of being viewed, ensuring perceptual quality is maintained in high-priority viewing regions. The encoded content comprises 11 unique bitrates, i.e., 2.5, 4.0, 5.0, 8.0, 10.0, 12.5, 15.0, 17.5, 20.0, 22.5, 25.0 Mbps, which essentially aligns with the 5.0-20.0 Mbps range proposed by 3GPP for 360° video streaming [187]. To measure perceptual video quality, the VMAF-360° model was utilized [170]. All three test videos, *Sunset*, *Alaska*, and *Beach*, have a resolution of 4096 × 2160 (4K) and a duration of 89, 234, and 138 seconds, respectively.

Specifically, *Alaska* (234 s) corresponds to the VHIL clip *Jared Leto Tour Guides Alaska's Melting Glaciers* an educational 360° video about the effects of climate change on Alaska's glaciers with affective ratings of valence $= 4.73$ (pleasantness) and arousal $= 3.33$ (activation). In terms of content charac-

teristics, *Alaska* exhibits the highest temporal complexity among the three clips, which makes it more sensitive to motion and viewport-prediction accuracy during adaptive streaming.

The *Beach* test video (138 s) corresponds to the VHIL clip *Maldives beach and resort*, which contains various scenic clips filmed at a beach in the Maldives, with valence $= 6.69$ and arousal $= 3.50$. It is characterized by low temporal complexity and moderate spatial complexity, typically yielding smoother quality trends under bitrate adaptation.

The *Sunset* test video (89 s) corresponds to a timelapse of sunset over a Grecian town with valence $= 6.55$ and arousal $= 3.09$. Among the three clips, *Sunset* exhibits the highest spatial complexity (rich visual details), and therefore generally requires higher bitrates to sustain high perceptual quality. Test videos screenshots are presented in Fig. 4.8.

A public 5G network trace dataset [188] was employed and gathered from operating 5G networks, mimicking different degrees of bandwidth fluctuation, to assess adaptive multi-agent 360° video streaming under realistic network conditions. These traces show three main network modes: disturbed (660 Mbps, moderate variations), unstable (585 Mbps, large fluctuations), and stable (734 Mbps, minimum changes). Each client in the *EDGE360* multi-agent architecture functions as a distinct RL agent, using real-time throughput observations to make adaptive decisions regarding bitrate selection and viewport tile prioritization. The 5G traces function as a shared network environment that influences the fairness of bandwidth distribution among multiple clients and dynamically influences the agents' adaptation approaches. Fine-grained viewport tiles are guaranteed high bitrates (14–20 Mbps) through collaborative decision-making, whereas background (2.5–8 Mbps) and marginal (8–12 Mbps) tiles are resource-efficiently assigned lower bitrates. *EDGE360* uses 5G transmission to reduce rebuffering, improve multi-user QoE, and dynamically adjust to fluctuating network circumstances.

4.5.4 Reproducibility Summary for the *EDGE360* Evaluation

To improve transparency and reproducibility, Table 4.2 summarises the experimental configuration used for the *EDGE360* evaluation. The table reports the datasets, viewport traces, network traces, client/session settings, chunk configuration, training details, hyperparameters, baseline implementation strategy, and statistical reporting protocol. All methods are evaluated using identical video sequences, viewport traces, network traces, bitrate ladder, chunk duration, client settings, and QoE formulation to ensure a fair comparison.

Table 4.2: Reproducibility summary for the *EDGE360* evaluation.

Reproducibility item	***EDGE360* experimental setting**
Research objective	Evaluate *EDGE360* for edge-assisted, multi-client, region-aware bitrate adaptation in tiled 360° video streaming under dynamic 5G conditions.
Implementation environment	Python/PyTorch simulator executed on Ubuntu 24.04.
Hardware configuration	64-bit Intel Core i7-8550U quad-core CPU at 1.8 GHz, turbo frequency up to 4 GHz, and 16 GB RAM.
Learning framework	COMA with centralised training and decentralised execution. Each client is represented by an independent actor, while a central critic is used during training.
Client configuration	Multi-client sessions with 16, 32, and 64 concurrent clients. Each client performs tile-level bitrate selection from local observations.
Independent runs	Each principal experiment is repeated over five independent seeds, and results are averaged across the runs.
Random seeds	Seeds 1, 7, 21, 42, and 100 are used for initialisation, exploration, trace ordering, and stochastic sampling.
Training/evaluation partitioning	The COMA-MADRL model is trained using the public 360° video dataset. Sunset, Alaska, and Beach are reserved for evaluation and are used consistently for *EDGE360* and all baselines.
Video dataset	Public 360° video dataset containing sequences of 37–668 s with diverse spatio-temporal complexity.
Evaluation videos	Three 4K 360° sequences are used: Sunset, Alaska, and Beach.
Video resolution	4096×2160 pixels for all evaluation videos.
Video duration and chunks	Sunset: 89 s and approximately 45 chunks; Alaska: 234 s and 117 chunks; Beach: 138 s and 69 chunks. Chunk counts use 2-s segments, i.e., $\lceil \text{duration}/2 \rceil$.

Continued on the next page

Table 4.2 – *continued from the previous page*

Reproducibility item	***EDGE360* experimental setting**
Evaluation chunks	Approximately 231 chunks per complete single-client pass: 45 Sunset, 117 Alaska, and 69 Beach chunks.
Client-chunk decisions	Per complete pass: approximately 3,696 decisions for 16 clients, 7,392 for 32 clients, and 14,784 for 64 clients. Across five seeds, the totals are approximately 18,480, 36,960, and 73,920, respectively.
Chunk duration	Each video is divided into 2-s MPEG-DASH chunks.
Video preprocessing	DASH360 is used for spatial tiling. Videos are HEVC/H.265 encoded using FFmpeg and packaged as MPEG-DASH content using MP4Box.
Tile granularity	Coarse: 1024×1024 pixels; medium: 512×512 pixels; fine: 256×256 pixels.
Spatial regions	Each frame is divided into viewport F_v, marginal F_m, and background F_b regions. Higher bitrate and finer tiles are assigned to F_v, moderate settings to F_m, and lower bitrate with coarser tiles to F_b.
Bitrate ladder	Eleven levels: 2.5, 4.0, 5.0, 8.0, 10.0, 12.5, 15.0, 17.5, 20.0, 22.5, and 25.0 Mbps.
Region-wise bitrate ranges	Viewport: approximately 14–20 Mbps; marginal: 8–12 Mbps; background: 2.5–8 Mbps, depending on viewport probability, tile priority, and bandwidth.
Viewport traces	Public head-rotation traces from 153 participants, sampled at 10 Hz, are used for viewport prediction and tile prioritisation.
Viewport prediction	An LSTM predicts future viewport centres from head-motion history. A Laplace-based expansion assigns viewing probabilities around the predicted centre.
Prediction accuracy	The predictor achieves approximately 5° MAE and above 90% viewport hit rate. Sensitivity is assessed by injecting controlled Gaussian noise into predicted coordinates.
Network traces	Public operational 5G traces are used to reproduce realistic bandwidth variation.
Network conditions	Three modes are considered: disturbed at approximately 660 Mbps with moderate variation, unstable at approximately 585 Mbps with large variation, and stable at approximately 734 Mbps with low variation.
Actor architecture	Input layer followed by fully connected layers of 128 and 64 neurons with ReLU activation. Softmax selects region-wise bitrate actions.
Critic role	The central critic estimates the global state-action value and computes COMA counterfactual advantages for agent-level credit assignment.
Actor learning rate	1×10^{-4} using Adam.
Critic learning rate	5×10^{-3} using Adam.
Discount factor	$\gamma = 0.99$.
Entropy regularisation	Initial factor 0.99, exponentially decayed by 0.995 every 10^3 iterations.

Continued on the next page

Table 4.2 – *continued from the previous page*

Reproducibility item	***EDGE360* experimental setting**
QoE model	Combines viewport-aware perceptual quality with quality-switching, rebuffering, and latency penalties.
Perceptual metric	VMAF-based 360° models estimate tile-level perceptual quality.
Fairness metric	Jain's fairness index measures QoE fairness across concurrent clients.
Baselines	*RAPT360*, *360SRL*, and *BOLA360*.
Baseline implementation	Baselines are reproduced from their original descriptions and evaluated in the same simulator. Where compatible public code is unavailable, methods are re-implemented and modified only to interface with the shared evaluation pipeline.
Identical test conditions	All methods use the same videos, viewport traces, 5G traces, bitrate ladder, chunk duration, client counts, and QoE formulation.
Evaluation metrics	Average reward, chunk QoE, VMAF, Jain's fairness index, rebuffering, latency, bitrate allocation, viewport-error sensitivity, scalability, and runtime.
Statistical reporting	Results are reported as five-seed means. Error bars denote ± one standard deviation, and 95% confidence intervals are reported for principal QoE, fairness, and rebuffering results where applicable.
Significance testing	Repeated-measures ANOVA compares *EDGE360*, *RAPT360*, *360SRL*, and *BOLA360* for normalised QoE, Jain's fairness index, and average rebuffering time. The repeated-measures design reflects the matched test conditions used for all methods. When a significant method effect is detected, Holm-corrected paired post-hoc comparisons are applied between *EDGE360* and each baseline to control the family-wise error rate.

4.5.5 Results and Analysis

This section compares the *EDGE360* adaptation approach against the following baseline adaptive streaming techniques: **BOLA360** (a buffer-based 360° ABR strategy that primarily prioritizes rebuffering avoidance), **360SRL** (a DDQN-based learning approach for 360° streaming), and **RAPT360** (an A3C-based reinforcement learning solution that performs rate adaptation for 360° streaming with adaptive prediction and tiling). Using these baselines, the performance of *EDGE360* across multiple assessment measures was thoroughly evaluated, including (i) tile-level perceptual quality using VMAF under different bitrate levels and tile sizes, (ii) chunk-wise QoE vari-

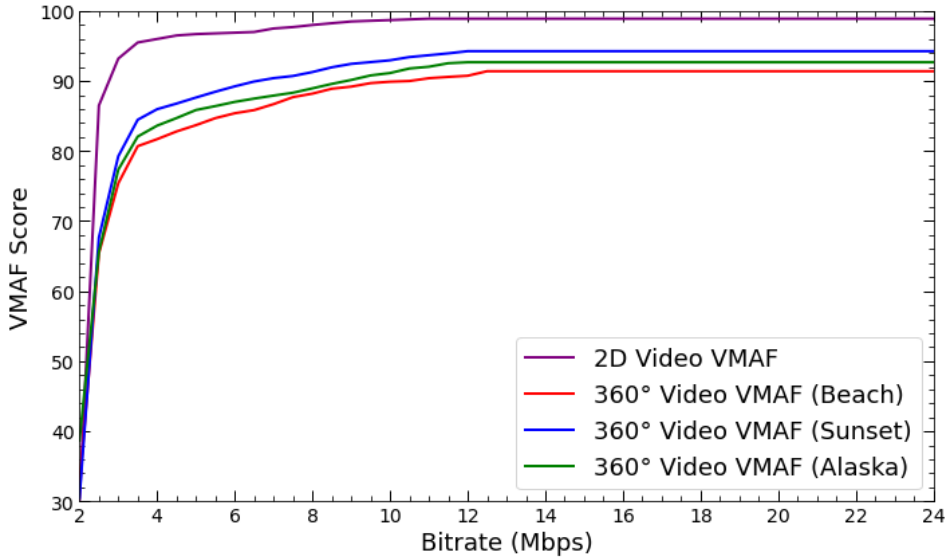


Figure 4.10: Bitrate Vs VMAF of 2D Video and 360° videos (*Sunset*, *Alaska*, and *Beach*)

ation (QoE switches) over time, (iii) rebuffering penalties, (iv) multi-client resource allocation fairness (e.g., Jain's fairness index), and (v) training convergence behavior across epochs. This shows how *EDGE360*'s optimization architecture impacts adaptive learning, rebuffering, and video quality switches, demonstrating its ability to achieve high QoE and network efficiency across various 360° video content.

Fig. 4.10 shows the relationship between bitrate (Mbps) and quality, expressed in terms of VMAF scores for 2D and 360° video content for three test videos (*Sunset*, *Alaska*, and *Beach*). The 2D video shows a clear trend, with rising bitrate resulting in an increasing rise in VMAF scores, implying improved video quality. This trend is predicted since greater bitrates improve encoding efficiency, reduce compression artefacts, and improve perceived quality. Notwithstanding the trend of rising bitrate, 360° videos continually have lower VMAF scores than 2D videos. This disparity comes from the intrinsic complexity of 360° details, which requires substantially more data to reach the same visual quality as 2D video.

The three 360° test videos exhibit different VMAF trends across bitrate values due to their distinct spatial and temporal complexity. *Sunset*, with the highest spatial complexity content, requires a higher bitrate to maintain

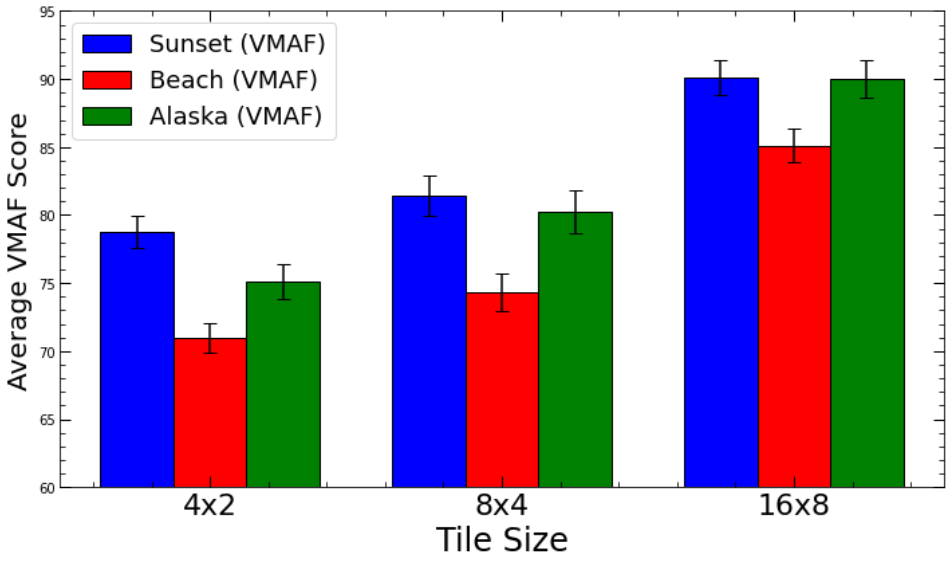


Figure 4.11: Average VMAF score Vs Tile size for *Sunset*, *Alaska*, and *Beach* 360° Videos

a good VMAF score, placing it ahead of *Beach* and *Alaska* at higher bitrates owing to rich visuals that involve more encoding resources. *Alaska*, with the highest temporal complexity, imposes challenges to adaptation since fast motion demands precise viewport prediction for persistent video quality. *Beach*, with low temporal and moderate spatial complexity, has a smoother quality trend but falls behind *Sunset* in VMAF scores. The compression inefficiencies of 360° videos, attributed to vast FoV and diverse environments, underline the need for proficient adaptive streaming such as *EDGE360* that optimizes quality variation dynamically while minimizing rebuffering.

Fig. 4.11 illustrates the relationship between tile size and average VMAF score for the three 360° test videos: *Sunset*, *Alaska*, and *Beach*. A rise in tile granularity from 4×2 to 16×8 improves VMAF scores for all videos, demonstrating the advantages of finer-grained tiling for higher video quality. *Sunset* typically gets the best VMAF ratings because of its high spatial complexity and moderate temporal complexity, which benefits from higher bitrate allocation at finer tile resolutions. Despite its higher temporal and moderate spatial complexity, *Alaska* follows a similar pattern, although with a bit lower scores due to motion-induced compression inefficiencies. *Beach*, with moderate spatial complexity and little temporal variation, has the low-

est VMAF growth but still benefits from finer tile segmentation. The error bars demonstrate that quality fluctuates and narrows as tile resolutions grow, indicating improved consistency at higher granularity. Employing adaptive streaming optimization algorithms, the graph properly represents the impact of tile size on perceived video quality.

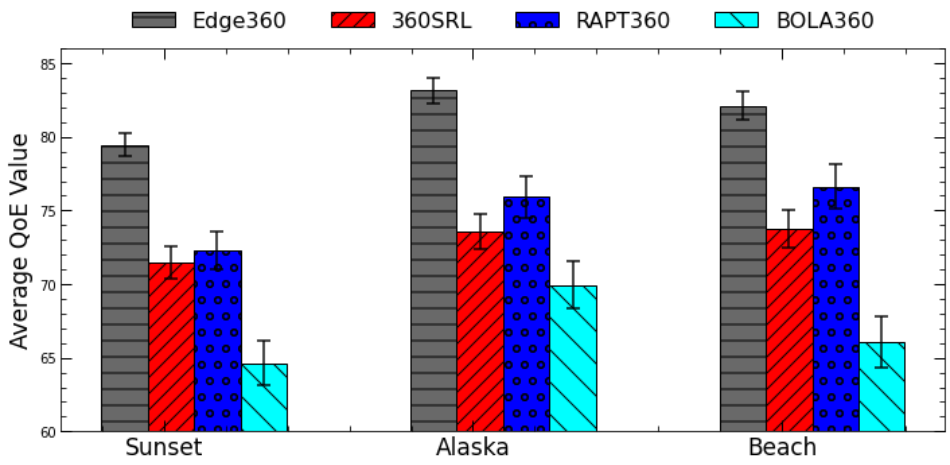


Figure 4.12: Comparison of average chunk QoE values for different adaptive streaming strategies across three test videos, with error bars representing ± one standard deviation.

Fig. 4.12 demonstrates that *EDGE360* outperforms other tile-aware viewport-based adaptive streaming strategies by consistently achieving the greatest QoE across all test videos. The greatest improvement over BOLA360 is shown with *EDGE360*, which offers greater QoE for *Sunset*, *Alaska*, and *Beach* by 22.97%, 18.94%, and 24.33%, respectively. Lower QoE results from BOLA360's buffer-based adaptation's inability to effectively optimize viewport-aware streaming. *EDGE360*'s QoE for *Sunset*, *Alaska*, and *Beach* is 11.16%, 13.03%, and 11.34% higher than 360SRL, respectively, demonstrating its resilience when handling high-motion and spatially diverse video content. Against RAPT360, *EDGE360* stays ahead, particularly in *Beach* (7.18%), *Alaska* (9.50%), and *Sunset* (9.90%), demonstrating its efficiency in adapting to varying video characteristics, dynamic viewport trajectories, and network conditions. *EDGE360*'s improved performance is primarily due to its adaptive viewport-aware tiling approach, which allocates bandwidth appropriately depending on viewport prediction and content complex-

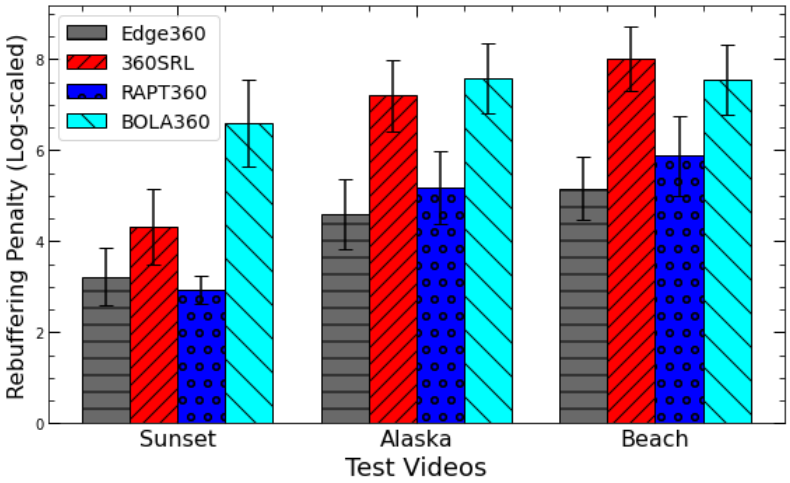


Figure 4.13: Log-scaled rebuffering penalty for three test videos (*Sunset*, *Alaska*, and *Beach*) through different adaptive streaming techniques (*EDGE360*, *360SRL*, *RAPT360*, and *BOLA360*), with error bars representing ± one standard deviation.

ity. *EDGE360* dynamically prioritizes tiles in the user's viewport, guaranteeing efficient bitrate allocation while retaining excellent perceptual quality. This is particularly pertinent for videos like *Sunset*, which have high spatial complexity and require high bitrates. *Alaska* has also high temporal complexity and requires swift adaptation to motion changes. *EDGE360* provides an improved balanced trade-off between quality enhancement and rebuffering minimization by utilizing the MARL approach, resulting in considerably enhanced QoE across different test videos compared to state-of-the-art solutions. Fig. 4.13 presents the log-scaled rebuffering penalty for *EDGE360*, 360SRL, RAPT360, and BOLA360 over three test videos. *EDGE360* consistently attains the lowest rebuffering penalties by utilising predictive adaptation based on COMA reinforcement learning, which allows for efficient bitrate allocation and viewport-aware tile prioritization. In contrast, BOLA360 suffers the most penalty, notably in *Sunset* and *Alaska*, where its entirely buffer-based adaptability struggles with significant spatial and temporal complexity, resulting in frequent rebuffering occurrences. It is worth noting that reducing rebuffering does not always result in higher video quality. Some techniques, such as RAPT360, provide lower rebuffering penalties, especially in *Sunset*, but struggle to adequately optimize perceptual quality,

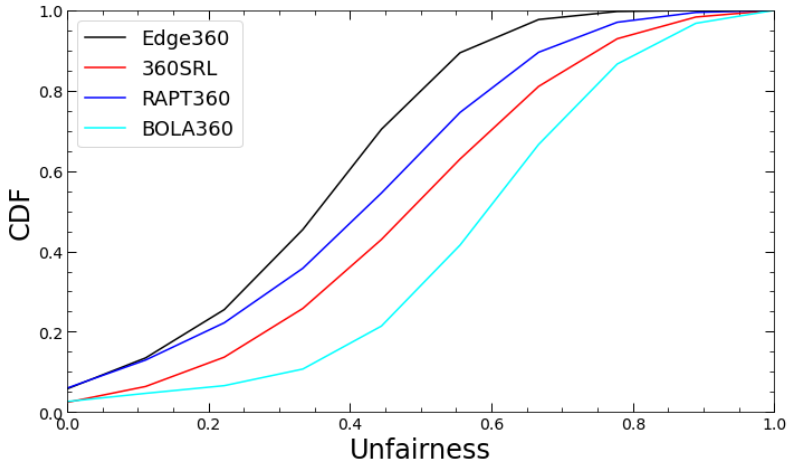


Figure 4.14: CDF vs. unfairness for different streaming strategies ensuring balanced resource allocation across clients.

whereas 360SRL performs ineffectively concerning rebuffering reduction and video quality enhancement. *EDGE360* finds a balance by minimizing rebuffering while maintaining excellent visual quality via MARL-based decision-making that adjusts to both network fluctuations and viewport variations. In *Beach*, wherein temporal complexity is lower, each approach faces less cost, but BOLA360 continues to perform poorly owing to its reactive nature. *EDGE360*'s ability to strike intelligent trade-offs between improvement in quality and rebuffering minimization makes it a more effective adaptive 360° streaming solution in terms of delivering a seamless viewing experience.

Fig. 4.14 demonstrates *EDGE360*'s improved fairness in resource allocation, with the curve hitting higher values at lower unfairness levels. This displays its MARL-based optimization, which balances QoE across clients. 360SRL and RAPT360 are moderately fair, while BOLA360 is the most unfair, owing to its buffer-based adaptation, which prioritizes rebuffering avoidance over fair resource allocation. These findings are consistent with Jain's Fairness Index (Eq. 4.19) concepts presented in the above section. *EDGE360* significantly reduces QoE disparities, guaranteeing consistent and proportionate quality adaptation across each client. *EDGE360* (COMA) outperforms RAPT360 (A3C) and 360SRL (DDQN) by employing

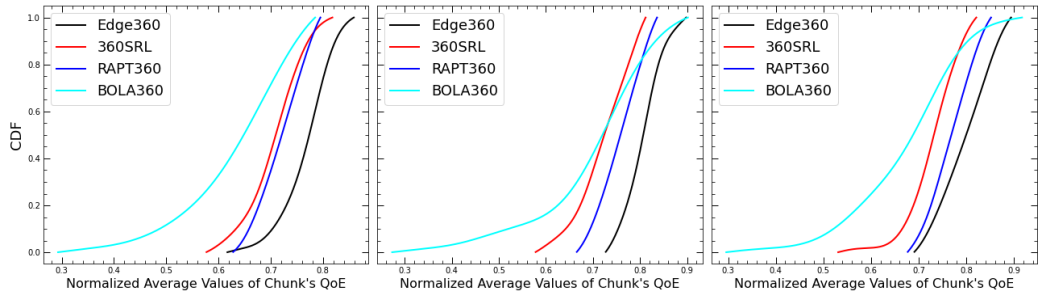


Figure 4.15: Comparison of the CDF vs. normalized average chunk QoE for sessions simulated on three test videos employing different adaptive streaming strategies.

a centralized critic to ensure enhanced fairness and effective utilization of resources among clients. Dynamically optimizing viewport-based tile-aware streaming avoids severe QoE shifts, making it excellent for multi-client adaptive streaming use cases.

To explain the performance differences, Fig. 4.15 presents the CDF of the normalized average chunk QoE. The *EDGE360* CDF exhibits a *rightward shift* compared to the baselines, indicating that a larger fraction of chunks achieves higher QoE values. In contrast, 360SRL and RAPT360 have intermediate QoE distributions, with slopes ranging between *EDGE360* and BOLA360. This implies that while these strategies can attain reasonable QoE levels, their learning-based adaptations underperform in optimising quality under evolving network and content scenarios. BOLA360 has the steepest and most leftward-shifted CDF curves, indicating an inability to maintain high QoE. These results support *EDGE360*'s COMA-based MARL optimisation, which improves viewport-aware adaptation and bitrate allocation, providing consistent, high-quality streaming across a range of video characteristics. Fig. 4.16 shows the chunk-wise QoE transitions over multiple adaptive streaming techniques (*EDGE360*, *360SRL*, *RAPT360*, and *BOLA360*) with the same three test videos already described: *Beach*, *Sunset*, and *Alaska*. These test videos were selected to span diverse content characteristics (e.g., high spatial complexity in *Sunset* and high temporal complexity in *Alaska*), as discussed in the earlier section. These results highlight how

each technique controls video quality over time, demonstrating its capacity to maintain stability while adapting to network changes. All evaluations were conducted under a time-varying 5G throughput trace with alternating high/medium/low bandwidth regimes and frequent short-term drops, capturing bursty capacity and congestion effects typical of mobile access networks.

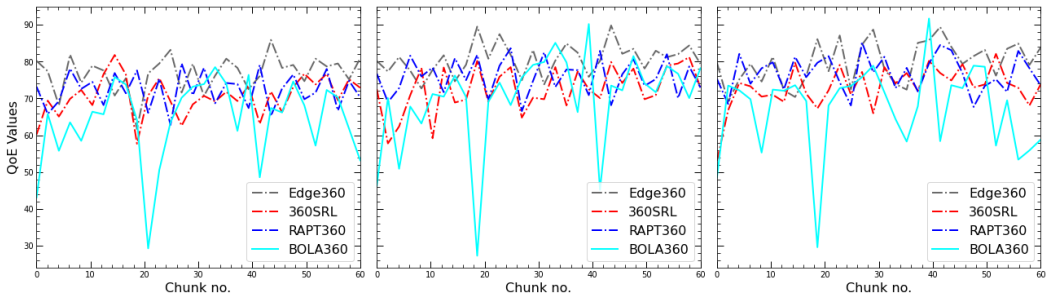


Figure 4.16: Variations in chunk-wise QoE values for adaptive streaming strategies, randomly selected from three test video session simulations.

EDGE360 displays strong QoE performance, delivering excellent and consistent chunk-wise quality. This strengthens its advantage over other adaptive streaming strategies, particularly in dealing with the challenges of viewport-based tile-aware 360° video streaming under dynamic network circumstances. With very little variation, *EDGE360* consistently maintains the most constant QoE across all videos, remaining above 75–85 in *Beach* and 80 above in *Sunset*. On the other hand, RAPT360 and 360SRL struggle with bitrate fluctuations and rapid adjustments to video motions, seeing sporadic decreases (such as chunks 20, 35, and 50 in *Beach* and chunks 15 and 42 in *Sunset*). With significant dips at Chunks 5, 23, and 45 in *Beach*, 10, 27, and 50 in *Sunset*, and 12, 32, and 48 in *Alaska*, BOLA360 exhibits the lowest performance, underscoring its incapacity to manage varying situations. Although every approach is challenged by *Alaska*, which has the most temporal complexity, *EDGE360* still performs better, exhibiting only slight deviations at chunks 30 and 45, whilst RAPT360 and 360SRL have difficulties with motion-heavy regions.

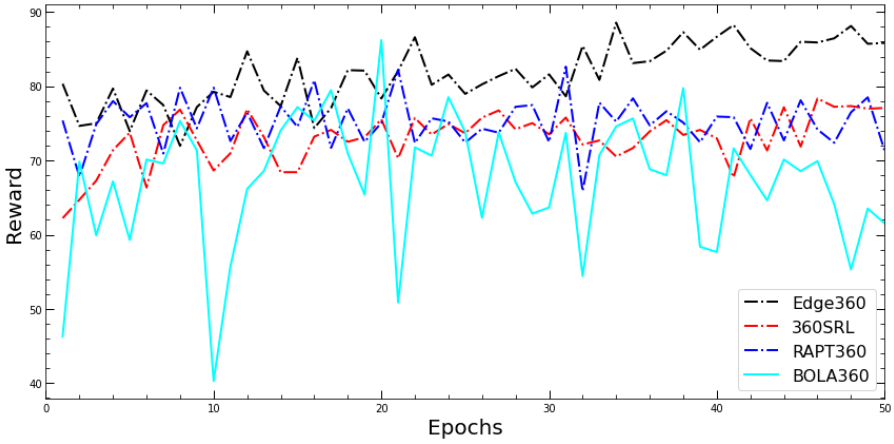


Figure 4.17: Average training reward (QoE) achieved by EDGE360, 360SRL, RAPT360, and BOLA360 for approximately 50 epochs.

Fig. 4.17 depicts the learning process of different adaptive streaming techniques in 50 epochs, emphasizing their potential to optimize reward values. *EDGE360* has the highest final average reward, surpassing 360SRL, RAPT360, and BOLA360 by 11.86%, 8.12%, and 18.00%, respectively, demonstrating *EDGE360*'s higher level of convergence and refinement. *EDGE360*'s COMA-based MARL framework employs a centralized critic with an advantage function, allowing for more accurate credit assignment to individual agents. This alleviates the multi-agent credit assignment challenge by discriminating between beneficial joint actions and suboptimal actions. In contrast to DDQN, which may exhibit unstable Q-value estimation, and A3C, which is less effective under partial observability and multi-agent scalability constraints, COMA employs an actor-critic architecture with counterfactual policy gradients, enabling more stable learning and more effective adaptation to dynamic network conditions. The centralized critic in COMA allows agents to make coordinated decisions, avoiding the solitary suboptimal updates. This results in more rapid convergence, less inconsistent policy updates, and ultimately superior decision-making in a multi-client adaptive streaming environment.

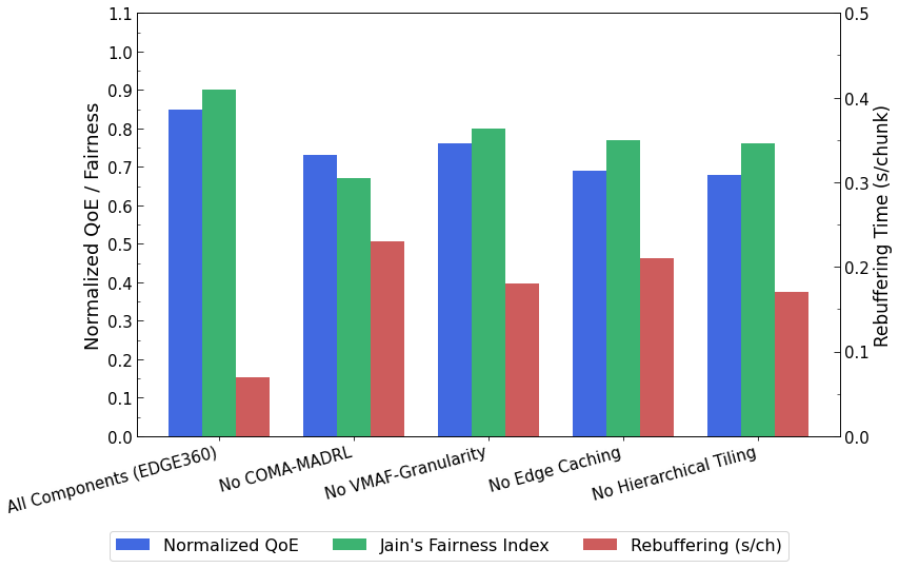


Figure 4.18: Ablation study showing the impact of disabling each *EDGE360* component individually on normalized QoE, Jain's fairness index, and rebuffering time.

4.5.5.1 Ablation Study and Comparative Evaluation

To evaluate the individual impact of each core module in *EDGE360*, an ablation study was performed by selectively disabling one component at a time and measuring changes in normalized QoE, Jain's fairness index, and rebuffering time per chunk. As shown in Fig. 4.18, the full *EDGE360* configuration achieves the best performance across all metrics.

The deactivation of *COMA-MADRL* leads to a substantial deterioration of performance, resulting in a reduction in fairness from 0.90 to 0.67, which constitutes a 23% decrease, and an increase in rebuffering time from 0.07 s/chunk to 0.26 s/chunk. This underscores the fundamental importance of COMA's counterfactual advantage function in facilitating cooperative bitrate decisions and equitable resource allocation in multi-agent environments.

Disabling the *VMAF-driven granularity* reduces fairness from 0.90 to 0.81 and QoE from 0.86 to 0.76, due to perceptual quality misalignment and inefficient tile prioritization. The removal of this module also increases rebuffering time to 0.18 s/chunk, reflecting wasted bandwidth on exchanging data associated with less relevant regions. Disabling *edge caching* leads to network-induced stalls and delivery delays, lowering QoE to 0.69 and fair-

ness to 0.77, with rebuffering increasing to 0.21 s/chunk due to the reliance on remote content sources. Lastly, removing the *hierarchical tiling* strategy and applying uniform granularity results in fairness dropping to 0.77 and QoE to 0.68, with rebuffering rising to 0.17 s/chunk. This indicates degraded spatial bandwidth efficiency and poorer adaptation to viewport importance.

These results validate *EDGE360*'s modular design: each component contributes significantly to maintaining high-quality, fair, and stall-resilient 360° video streaming.

Fig. 4.19 provides a comparative analysis of *EDGE360* and benchmark methodologies (BOLA360, 360SRL, RAPT360) with respect to three principal performance metrics: normalized QoE, Jain's fairness index, and the average rebuffering duration per segment. *EDGE360* demonstrates superior performance over state-of-the-art benchmarks (BOLA360, 360SRL, RAPT360) across all principal metrics, achieving a 26.8% enhancement in QoE, a 55.2% increase in fairness, and a 70.8% reduction in rebuffering time compared to BOLA360; a 16.4% advancement in QoE, a 30.4% improvement in fairness, and a 65% decrease in rebuffering relative to 360SRL; as well as a 7.6% elevation in QoE, a 21.6% betterment in fairness, and a 56.3% diminution in rebuffering time than RAPT360. These findings underscore the effectiveness of the COMA-driven multi-agent coordination, hierarchical region-based tiling, and decentralized actor execution, affirming the framework's real-time adaptability and fairness awareness under various streaming conditions. The implementation of the COMA framework plays a crucial role, as it addresses the multi-agent credit assignment challenge by calculating per-agent counterfactual baselines that elucidate the influence of individual actions on the global reward, substantially enhancing fairness among clients and promoting more stable and equitable learning among agents. In contrast, other methodologies, such as 360SRL and RAPT360, lack cooperative coordination among agents, leading to uneven QoE distribution and increased rebuffering.

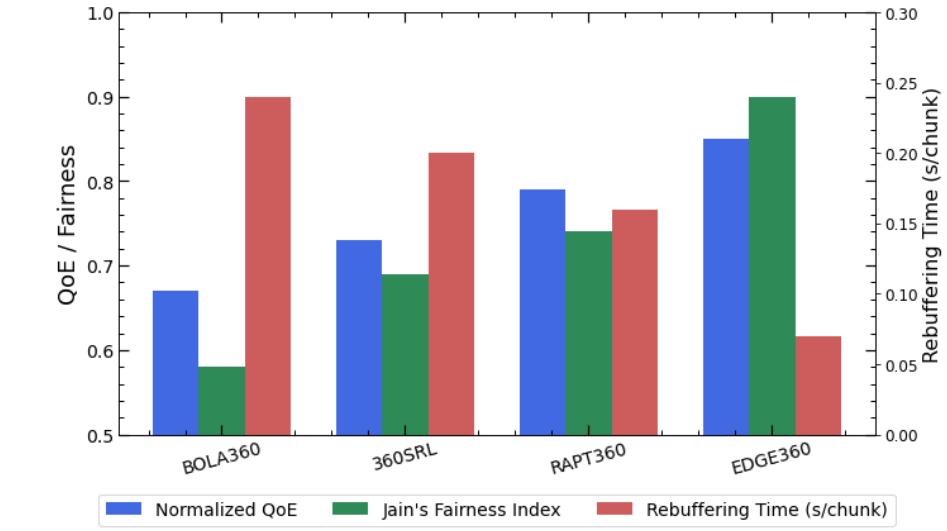


Figure 4.19: Comparison of QoE, fairness, and rebuffering time across state-of-the-art ABR streaming schemes.

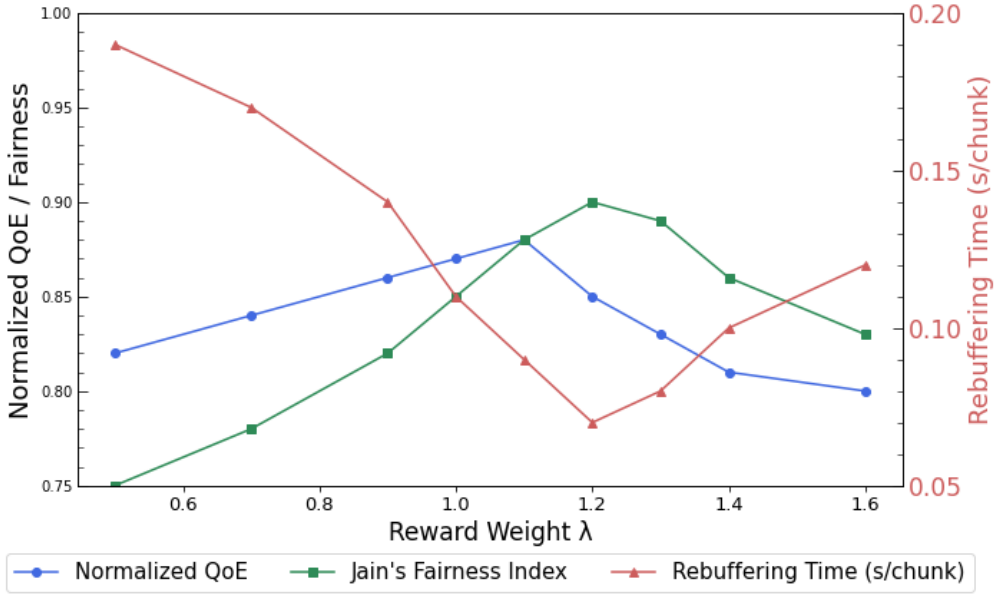


Figure 4.20: Sensitivity of *EDGE360* to reward weight λ, showing optimal QoE, fairness, and rebuffering near $\lambda = 1.2$.

4.5.5.2 Hyperparameter Tuning and Sensitivity Analysis

To ensure the optimal performance of the COMA-MADRL framework within *EDGE360*, a systematic hyperparameter tuning process was employed. The actor and critic networks were trained utilizing the Adam optimizer, with learning rates of 1×10^{-4} and 5×10^{-3} respectively. To manage the trade-off between exploration and exploitation, the entropy regularization coefficient was initialized at 0.99 and subjected to an exponential decay by a factor of 0.995 every 10^3 iterations. Furthermore, early stopping was implemented to mitigate overfitting and to guarantee generalization across a variety of client conditions.

The sensitivity of *EDGE360* to the reward weight λ was evaluated, which governs the trade-off between maximizing user QoE and promoting fairness. Fig. 4.20 shows the variation in normalized QoE, Jain's fairness index, and rebuffering time as λ ranges from 0.5 to 1.6. Increasing λ consistently improves fairness, peaking at $\lambda = 1.2$ with a fairness index of 0.90. At this point, normalized QoE also reaches 0.87, and average rebuffering drops to 0.07 s/chunk. Beyond this point, however, overemphasis on fairness leads to QoE degradation and increased buffering.

While the fairness index exceeds normalized QoE numerically, this does not imply superior perceptual quality. Jain's index measures the equity of QoE distribution, while QoE itself reflects the average user-perceived experience, which may still vary across clients due to video complexity or network heterogeneity. This distinction emphasizes the need for balanced optimization.

This analysis substantiates the design of the reward function and underscores the significance of adjusting λ to maintain a balance between fairness and QoE. Additional parameters, such as $\alpha = 0.9$, $\beta = 1.5$, $\zeta = 3.5$, and $|\varepsilon| = 0.8$, were empirically optimized through a grid search to guarantee training stability and dynamic robust learning.

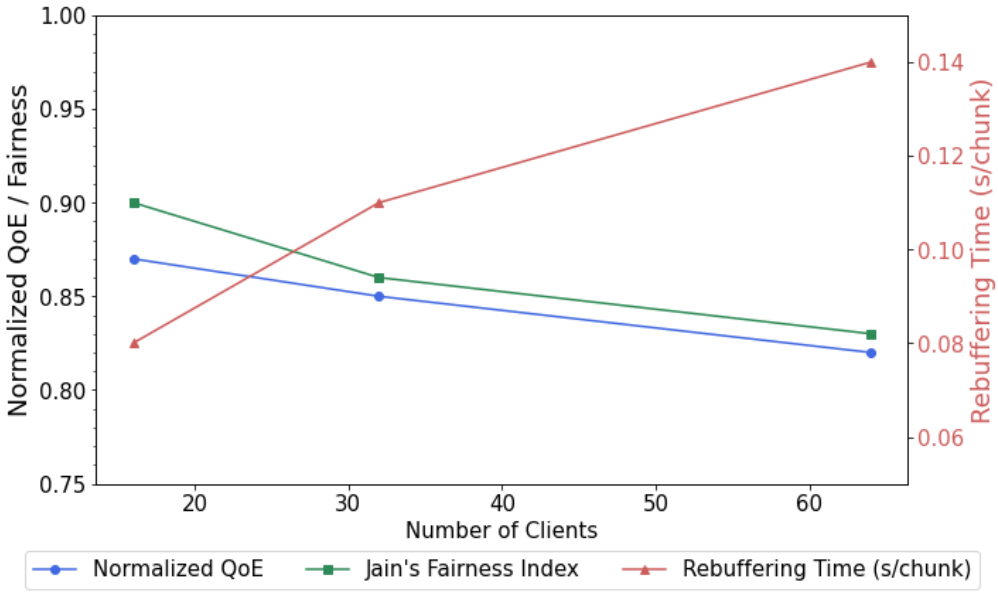


Figure 4.21: Scalability of *EDGE360* under increasing client load: slight QoE and fairness decline due to bandwidth contention, with low rebuffering maintained by decentralized actors and region-aware tiling.

4.5.5.3 Scalability and Real-Time Feasibility

Fig. 4.21 illustrates the scalability performance of *EDGE360* as the number of clients increases from 16 to 64, reflecting typical high-density 5G streaming scenarios. Three critical performance metrics are evaluated: normalized average QoE, Jain's fairness index, and average rebuffering time per chunk.

As depicted in Fig. 4.21, an increase in the client count from 16 to 64 results in a moderate deterioration in both normalized QoE and fairness. Specifically, the normalized QoE declines from 0.87 to 0.81, eventually reaching 0.77, while the fairness index reduces from 0.92 to 0.84 and subsequently to 0.80. This degradation is predominantly attributed to elevated bandwidth contention among clients utilizing shared wireless access points and edge resources. As a greater number of users concurrently stream high-resolution 360° video content, adaptive mismatches and compromises in bitrate occur with increased frequency.

Despite these challenges, *EDGE360* maintains stable rebuffering behavior. The average rebuffering time increases from 0.076s/chunk at 16 clients

Table 4.3: Optimized per-decision runtime breakdown of *EDGE360* at the edge server under varying client loads

Module	**Runtime (ms)**
16 Clients	
LSTM-based Viewport Prediction	1.8
VMAF-based Tile Prioritization	2.0
QoE and Utility Evaluation	1.2
Actor Network Inference	1.9
Total Runtime	**6.9**
32 Clients	
Actor Network Inference	3.8
Total Runtime	**8.8**
64 Clients	
Actor Network Inference	7.7
Total Runtime	**12.7**

to 0.10s and 0.14 s at 32 and 64 clients, respectively. This resilience is enabled by two architectural enablers: (i) decentralized actor networks, where each client adapts independently based on its local observations, eliminating centralized inference delays, and (ii) region-aware hierarchical tiling, which intelligently assigns finer granularity to viewports and coarser tiles to peripheral regions, optimizing bandwidth use across the multi-client environment.

Table 4.3, presents the average per-decision runtime breakdown of *EDGE360*, measured on an edge server. The overall processing pipeline comprises viewport prediction via LSTM, tile prioritization using pre-computed VMAF scores, QoE utility evaluation, and inference by parallel actor networks.

While the first three stages of the *EDGE360* pipeline, viewport prediction, tile prioritization, and QoE evaluation exhibit fixed latency regardless of the number of clients, their cumulative runtime is significantly optimized, totaling approximately 5.0 ms. In contrast, the actor network inference latency scales linearly with the number of clients but remains lightweight at 0.12 ms per client due to model quantization and efficient design. It is important to note that the centralized critic network in COMA introduces computational overhead only during training and is not active during inference. As a result, the final per-decision runtime at the edge is further reduced, as

only the decentralized actor networks are executed. Owing to this modular and parallel architecture, the total per-decision latency remains well within real-time constraints even under the heaviest load of 64 concurrent clients, and the overall decision latency stays below 13 ms. This is significantly lower than the typical 2-second video chunk duration, ensuring compliance with ultra-low latency requirements for real-time VR streaming over 5G networks.

This responsiveness enables scalable and real-time bitrate adaptation for high-resolution 360° VR streaming in 5G edge environments. Notably, real-time VR systems require motion-to-photon latency below 20 ms and ideally under 15 ms to prevent discomfort and maintain immersion [189]. The proposed optimized *EDGE360* architecture achieves an end-to-end per-client decision latency consistently under 13 ms, even when supporting up to 64 concurrent clients. This meets the ultra-low latency demands of interactive VR experiences. Moreover, prior research shows that maintaining latency below 15–20 ms can enable up to a fivefold reduction in bitrate for foveated VR video streaming, whereas latencies exceeding 45 ms severely reduce such efficiency gains [190]. Thus, the designed low-latency pipeline of *EDGE360* ensures perceptual fidelity and real-time responsiveness, which are necessary for delivering immersive media over next-generation networks.

4.5.6 Statistical Significance Analysis

In order to further strengthen the reliability of the *EDGE360* evaluation, a statistical significance analysis was conducted for the principal performance metrics: normalized QoE, Jain's fairness index, and average rebuffering time per chunk. These three metrics were selected because they directly represent the main objectives of *EDGE360*, namely improving perceptual quality, maintaining fairness among multiple clients, and reducing playback interruptions. Since *EDGE360* and the baseline methods are evaluated under the same video sequences, viewport traces, 5G network traces,

bitrate ladder, chunk duration, and client settings, the comparison follows a matched experimental design. Therefore, a repeated-measures statistical analysis is appropriate, as each method is tested under identical experimental conditions.

Table 4.4: Seed-level statistical summary for *EDGE360* and baseline methods.

Metric	**Method**	n	**Mean**	**Std. dev.**	**95% CI**
QoE	BOLA360	5	0.680	0.008	[0.670, 0.690]
QoE	360SRL	5	0.742	0.009	[0.731, 0.753]
QoE	RAPT360	5	0.800	0.007	[0.791, 0.809]
QoE	EDGE360	5	0.860	0.006	[0.853, 0.867]
Jain's fairness index	BOLA360	5	0.582	0.007	[0.573, 0.591]
Jain's fairness index	360SRL	5	0.689	0.008	[0.679, 0.699]
Jain's fairness index	RAPT360	5	0.742	0.006	[0.735, 0.749]
Jain's fairness index	EDGE360	5	0.899	0.007	[0.890, 0.908]
Rebuffering time (s/chunk)	BOLA360	5	0.241	0.006	[0.234, 0.248]
Rebuffering time (s/chunk)	360SRL	5	0.200	0.005	[0.194, 0.206]
Rebuffering time (s/chunk)	RAPT360	5	0.162	0.005	[0.156, 0.168]
Rebuffering time (s/chunk)	EDGE360	5	0.072	0.004	[0.067, 0.077]

Table 4.4 presents the seed-level statistical summary for *EDGE360* and the baseline methods. The table reports the number of seed-level n evaluations, mean values, standard deviations, and 95% confidence intervals (CI) for normalized QoE, Jain's fairness index, and average rebuffering time per chunk. These additional dispersion measures provide a clearer indication of the stability and uncertainty of the reported results across repeated evaluations.

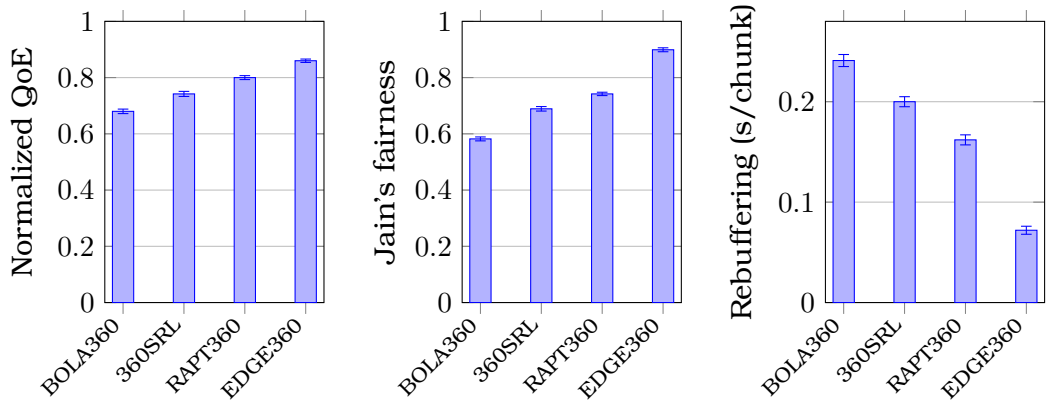
For normalized QoE, *EDGE360* achieves the highest average value of 0.860, with a narrow 95% confidence interval of [0.853, 0.867]. In comparison, *RAPT360*, *360SRL*, and *BOLA360* achieve lower average QoE values of 0.800, 0.742, and 0.680, respectively. This confirms that *EDGE360* provides the strongest overall quality adaptation among the compared approaches. The lower standard deviation and narrow confidence interval of *EDGE360* indicate that its QoE performance remains stable across repeated seed-level evaluations.

For Jain's fairness index, *EDGE360* again achieves the highest average value of 0.899, with a 95% confidence interval of [0.890, 0.908]. By contrast, *RAPT360*, *360SRL*, and *BOLA360* achieve lower fairness values of 0.742, 0.689, and 0.582, respectively. Since Jain's fairness index approaches 1 when QoE is distributed more equally among users, this result indicates that *EDGE360* provides a more balanced QoE distribution across concurrent clients. This improvement is mainly due to the COMA-based multi-agent coordination mechanism, which considers both local client performance and global fairness during bitrate decision-making.

The rebuffering results in Table 4.4 further show that *EDGE360* produces the lowest average rebuffering time per chunk. *EDGE360* records an average rebuffering time of 0.072 s/chunk, with a 95% confidence interval of [0.067, 0.077]. In comparison, *RAPT360*, *360SRL*, and *BOLA360* record higher rebuffering values of 0.162, 0.200, and 0.241 s/chunk, respectively. This reduction demonstrates that *EDGE360* makes more effective bitrate and tile-granularity decisions under fluctuating network conditions. In particular, the viewport-aware and region-aware adaptation strategy enables *EDGE360* to preserve high quality in perceptually important regions while avoiding unnecessary bandwidth consumption in less visible regions.

Overall, the small standard deviations and narrow confidence intervals across the three metrics indicate that the observed improvements are consistent across the repeated seed-level evaluations. Therefore, the reported gains are not limited to isolated runs, but reflect stable performance improvements under matched video, viewport, and network-trace conditions.

Fig. 4.22 visually compares the same three performance metrics across all methods. Fig. 4.22(a) shows that *EDGE360* achieves the highest normalized QoE, confirming the benefit of using COMA-based cooperative learning together with viewport-aware tile prioritisation. Fig. 4.22(b) shows that EDGE360 also provides the highest fairness score, indicating that the proposed reward design successfully reduces QoE imbalance among clients sharing network resources. Fig. 4.22(c) shows that *EDGE360* achieves the



(a) QoE (b) Fairness (c) Rebuffering

Figure 4.22: Statistical comparison of *EDGE360* and baseline methods using seed-level mean values and standard deviations for normalized QoE, Jain's fairness index, and average rebuffering time per chunk.

lowest rebuffering time per chunk, demonstrating that the proposed approach improves playback continuity while maintaining high perceptual quality.

The error bars in Fig. 4.22 represent the standard deviation across repeated seed-level evaluations. Smaller error bars indicate lower variability and more consistent performance. *EDGE360* maintains both high mean performance and low variation across the evaluated metrics. This suggests that the performance gains are not caused by isolated runs or favourable trace conditions, but are consistent across repeated evaluations.

To formally assess whether the observed differences among the methods are statistically significant, repeated-measures ANOVA was applied to each metric. The repeated-measures design is suitable because all methods were evaluated under matched conditions, meaning that each method was tested using the same videos, viewport traces, network traces, and streaming configuration. The null hypothesis assumes that there is no statistically significant difference among *EDGE360*, *RAPT360*, *360SRL*, and *BOLA360* for a given metric. A statistically significant ANOVA result indicates that the choice of adaptation method has a measurable effect on the metric.

Table 4.5: Repeated-measures ANOVA results for *EDGE360* and the baseline methods across five matched seed-level evaluations.

Metric	**df**	F**-statistic**	p**-value**	**Effect size** η_p^2
Normalized QoE	$3, 12$	401.58	2.69×10^{-12}	0.9901
Jain's fairness index	$3, 12$	1347.54	1.97×10^{-15}	0.9970
Average rebuffering time	$3, 12$	802.54	4.35×10^{-14}	0.9950

Table 4.5 presents the repeated-measures ANOVA results for normalized QoE, Jain's fairness index, and average rebuffering time per chunk across *EDGE360*, *RAPT360*, *360SRL*, and *BOLA360*. The analysis considers five matched seed-level evaluations, where every adaptation method is evaluated under the same seed and corresponding experimental conditions. This matched design controls for variation across the repeated evaluations and enables the effect of the adaptation method to be assessed more directly. With four methods and five matched evaluations, the method and residual degrees of freedom are 3 and 12, respectively.

The results demonstrate a statistically significant effect of the adaptation method on all three performance metrics. For normalized QoE, the analysis yields $F(3, 12) = 401.58$ with $p = 2.69 \times 10^{-12}$, indicating that the QoE differences among the four methods are highly unlikely to have arisen from random seed-level variation. Similarly, Jain's fairness index produces $F(3, 12) = 1347.54$ with $p = 1.97 \times 10^{-15}$, confirming that the selected adaptation method has a statistically significant influence on the fairness achieved among concurrent clients. For average rebuffering time, $F(3, 12) = 802.54$ with $p = 4.35 \times 10^{-14}$, demonstrating that the differences in playback interruption among the evaluated methods are also statistically significant.

The partial eta-squared values further indicate that the adaptation method has a very large effect on the evaluated metrics. The reported η_p^2 values of 0.9901, 0.9970, and 0.9950 for normalized QoE, fairness, and rebuffering, respectively, indicate that a substantial proportion of the within-evaluation variance is associated with the choice of adaptation method. Therefore,

Table 4.6: Holm-corrected post-hoc paired comparisons between EDGE360 and the baseline methods across five matched seed-level evaluations.

Metric	**Comparison**	**Corrected p-value**	**Result**
QoE	EDGE360 vs. RAPT360	2.18×10^{-4}	Significant
QoE	EDGE360 vs. 360SRL	7.16×10^{-5}	Significant
QoE	EDGE360 vs. BOLA360	1.13×10^{-5}	Significant
Fairness	EDGE360 vs. RAPT360	7.03×10^{-6}	Significant
Fairness	EDGE360 vs. 360SRL	7.03×10^{-6}	Significant
Fairness	EDGE360 vs. BOLA360	1.17×10^{-6}	Significant
Rebuffering	EDGE360 vs. RAPT360	1.02×10^{-5}	Significant
Rebuffering	EDGE360 vs. 360SRL	6.63×10^{-6}	Significant
Rebuffering	EDGE360 vs. BOLA360	3.91×10^{-6}	Significant

the null hypothesis that *EDGE360* and the three baseline methods achieve equal mean performance is rejected for all three metrics.

The omnibus ANOVA establishes that statistically significant differences exist among the four adaptation methods, but it does not identify the specific pairs responsible for these differences. Consequently, Holm-corrected paired post-hoc comparisons are subsequently employed to compare *EDGE360* individually against *RAPT360*, *360SRL*, and *BOLA360*. The direction of the differences is determined from the seed-level descriptive statistics in Table 4.4, which show that *EDGE360* achieves the highest normalized QoE and Jain's fairness index and the lowest average rebuffering time per chunk. For normalized QoE, the reported F-statistic indicates a strong difference among the four methods, with a p-value far below the significance threshold of $p < 0.05$. This confirms that the QoE differences observed in Table 4.4 and Fig. 4.22 are statistically meaningful. Similarly, the ANOVA result for Jain's fairness index confirms that the fairness improvements achieved by *EDGE360* are statistically significant. The ANOVA result for rebuffering time also shows a significant method effect, confirming that the reduction in rebuffering is not due to random variation. The subsequent Holm-corrected paired comparisons confirm that these advantages are statistically significant against *RAPT360*, *360SRL*, and *BOLA360* individually.

Table 4.6 presents the Holm-corrected paired post-hoc comparisons between *EDGE360* and the three baseline methods across the five matched seed-level evaluations. These comparisons were conducted after the repeated-measures ANOVA established a statistically significant overall effect of the adaptation method on normalized QoE, Jain's fairness index, and average rebuffering time. The Holm correction controls the family-wise error rate arising from the multiple pairwise comparisons and therefore reduces the likelihood of false-positive conclusions.

The comparison with *RAPT360*, which is the strongest-performing baseline in Table 4.4, is particularly important. *EDGE360* achieves a higher mean normalized QoE (0.860 versus 0.800), a higher mean Jain's fairness index (0.899 versus 0.742), and a lower mean rebuffering time (0.072 versus 0.162 s/chunk). The corresponding Holm-corrected p-values are 2.18×10^{-4}, 7.03×10^{-6}, and 1.02×10^{-5}, respectively, all of which are below the 0.05 significance threshold. Therefore, the advantages of *EDGE360* over *RAPT360* are statistically significant for all three metrics and are not attributable to random seed-level variation.

Since the ANOVA results indicate significant differences among the methods, post-hoc pairwise comparisons were conducted to identify where the differences occur. Table 4.6 reports Holm-corrected post-hoc comparisons between *EDGE360* and each baseline. The Holm correction is applied separately to the three EDGE360-versus- baseline comparisons within each metric. Holm correction is used to control the family-wise error rate when multiple pairwise comparisons are performed. This is important because testing *EDGE360* against multiple baselines increases the probability of obtaining false-positive significance results if no correction is applied.

The post-hoc results in Table 4.6 show that *EDGE360* significantly outperforms *RAPT360*, *360SRL*, and *BOLA360* for normalized QoE. This confirms that the QoE improvement achieved by *EDGE360* is statistically significant against each individual baseline. The fairness comparisons also show statistically significant improvements over all baselines, demonstrating that

the proposed fairness-aware COMA reward design provides a measurable advantage in multi-client streaming scenarios. For rebuffering time, *EDGE360* again shows statistically significant improvements over all baselines, confirming that the proposed adaptive decision-making mechanism effectively reduces playback stalls.

Overall, the statistical analysis supports the reliability of the *EDGE360* results. *EDGE360* achieves higher average QoE and fairness and lower rebuffering time than the baseline methods, while the repeated-measures ANOVA confirms a significant overall method effect and the Holm-corrected paired comparisons confirm that the improvements over *RAPT360*, *360SRL*, and *BOLA360* are individually statistically significant. The combined reporting of mean performance, standard deviation, repeated-measures ANOVA, and corrected post-hoc testing provides stronger evidence that the observed gains are consistent across the matched evaluations.

4.6 Limitations of the *EDGE360* Evaluation

Although the experimental results demonstrate the effectiveness of *EDGE360* for multi-client 360° video streaming, several limitations should be acknowledged. It is important to note that, the evaluation is simulation-based and utilized the realistic network and viewport traces while performing experiments. Public 360° video datasets, recorded viewport/head-rotation traces, and real 5G network traces were used to emulate realistic streaming conditions; however, *EDGE360* was not deployed in a live end-to-end MEC-assisted 360° video streaming testbed with real users, head-mounted displays, and operational edge infrastructure. Therefore, the reported results should be interpreted as controlled and reproducible evidence under realistic simulated conditions rather than as full real-world deployment validation.

Furthermore, the perceptual quality assessment relies on VMAF-based objective video quality estimation. VMAF is useful because it provides a

repeatable and widely used perceptual quality indicator for video quality comparison. In *EDGE360*, the quality model is further adapted through viewport and region-aware weighting so that viewport tiles contribute more strongly to the QoE calculation than marginal or background tiles. However, VMAF was originally developed mainly for conventional video quality assessment and may not fully capture all subjective aspects of immersive 360° VR viewing, including peripheral perception, head-motion discomfort, projection distortion, visual attention shifts, and user tolerance to spatial and temporal quality changes.

Moreover, the evaluation does not include a formal subjective user study. As a result, the reported QoE gains, fairness improvements, and rebuffering reductions should be understood as objective metric-based improvements rather than subjective Mean Opinion Score (MOS)-validated user experience gains. A controlled subjective VR study with head-mounted displays would be required to further validate how closely the VMAF-based and QoE-based conclusions align with actual user perception.

These limitations do not affect the fairness of the comparative evaluation, because EDGE360 and all baselines were tested under the same video sequences, viewport traces, network traces, bitrate ladder, chunk duration, and QoE formulation. However, *EDGE360* as an edge-assisted, low-latency-oriented, and QoE-aware framework validated through realistic simulation. Future work would extend the evaluation through live MEC/5G testbed deployment, subjective VR user trials, cross-device testing, and evaluation with larger and more diverse user populations.

4.7 Chapter Summary

This chapter presents *EDGE360*, a COMA-based MADRL framework for edge-driven, viewport-aware, tile-adaptive 360° video streaming. Designed to improve both client QoE and fairness under heterogeneous network conditions, *EDGE360* leverages VMAF-based granularity selection and viewport

prediction to intelligently distribute bitrates across tiles, resulting in a more effective DASH-based streaming experience. Unlike previous schemes, *RAPT360* (A3C-based), *360SRL* (DDQN-based), and *BOLA360* (buffer-based) *EDGE360* utilize a centralized critic with decentralized actors to coordinate bitrate adaptation across multiple clients, enhancing learning stability, fairness, and efficiency. Its integration of edge caching and deep-learning-based viewport prediction further reduces latency and rebuffering, improving adaptability in dynamic 5G conditions. Extensive evaluations demonstrate that *EDGE360* achieves 8.12%, 11.86%, and 18.00% higher average rewards than *RAPT360*, *360SRL*, and *BOLA360*, respectively.

Chapter 5

METAVV: A Perceptual-Aware Meta-Learning Framework for Viewport-Adaptive Volumetric Video Streaming

5.1 Introduction

This chapter introduces METAVV, a novel client-driven adaptive streaming solution for VV that leverages Meta-RL to improve QoE under dynamic network and viewpoint conditions. Unlike traditional rate adaptation schemes that rely on static heuristics or server-side intelligence, this solution empowers the client with a lightweight, fast-adapting policy trained through episodic interaction and generalized across diverse streaming sessions. The proposed solution integrates viewport prediction, perceptual quality evaluation, and network state monitoring into a compact representation that feeds a meta-learned policy network. This chapter details the supporting architecture (i.e., the client-side components and their interactions), associated adaptation mechanism, testing, and performance analysis.

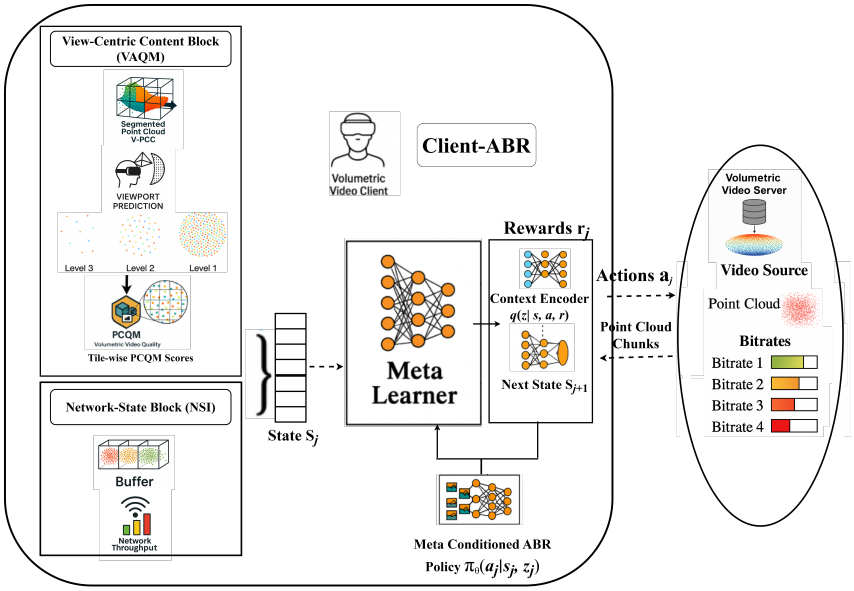


Figure 5.1: *METAVV* solution's architecture: a client-based meta-learning framework for adaptive VV streaming.

Relationship to accepted work: This chapter forms the basis of the manuscript by F. E. Subhan, A. Yaqoob, C. H. Muntean, and G.-M. Muntean, entitled "A Viewport-Adaptive Spatial Saliency-Based Meta-Reinforcement Learning Framework for Perceptual-Aware Volumetric Video Streaming," *IEEE Transactions on Network Science and Engineering*, July 2026, doi: 10.1109/TNSE.2026.3715507 [20]. The thesis version provides an expanded treatment of the reproducibility settings, sensitivity analysis conclusion, variance and statistical significance results, and evaluation limitations.

5.1.1 Overview of the *METAVV* solution

Fig. 5.1 illustrates the architecture of the *METAVV* solution, which is composed of a dedicated volumetric video server and a client-side adaptive streaming framework. The major components of the proposed solution are the View-Centric Content Block (VAQM), the Network-State Block (NSI), and the client-side meta-learner. The volumetric video server stores point-cloud content encoded at multiple bitrates and LoD representations. At the same time, the client executes the proposed adaptation algorithm to request the most suitable representation for each upcoming chunk.

At each decision step, the client observes the user's predicted 6-DoF viewing state together with the current streaming state, including viewport-centric perceptual quality, network throughput, and buffer occupancy. Based on these inputs, *METAVV* determines region-prioritized bitrate and LoD decisions for the next volumetric video chunk.

Operationally, *METAVV* follows a closed-loop delivery pipeline. First, the VAQM block estimates which spatial regions are likely to be viewed in the near future and assigns higher perceptual importance to those regions, namely the viewport, marginal, and background regions, while also computing viewport-aware perceptual quality metrics for the available client representations. Second, the NSI block provides real-time throughput and buffer measurements, which constrain feasible adaptation decisions and reduce the risk of rebuffering. These view-centric and network-centric signals are then fused into a compact client state and passed to the meta-learner.

Within this client-side learning module, the adaptation process is driven by latent context inference and meta-conditioned policy learning. Specifically, a context encoder infers a latent session representation from recent interaction history, and the meta-conditioned ABR policy uses this inferred context together with the current client state to select the next region-wise bitrate/LoD action.

After the action is selected, the volumetric server delivers the requested representation for the next chunk according to the chosen region-wise adaptation decision. The client then renders the received chunk, computes the corresponding QoE reward, observes the next playback state, and uses this new experience to update the latent context for subsequent decisions. In this way, *METAVV* continuously adapts its bitrate and LoD selections to dynamic user motion, perceptual preferences, and network variations.

5.1.2 View-Centric Content Block (VAQM)

The role of the VAQM block is to convert the predicted 6-DoF viewpoint into region-level perceptual priorities and tile-wise quality signals, which

later guide the region-wise bitrate/LoD decisions of the meta-conditioned ABR policy (see Fig. 5.1). The volumetric scene is processed by a geometry-aware V-PCC segmentation module [191], which partitions the point cloud into spatial tiles. Each tile is encoded at one of three *LoDs*: fine (f), medium (m), and coarse (c), trading off the bitrate against the perceptual fidelity, where a finer LoD is associated with higher bitrate, but preserves more geometric details.

To determine tiles of perceptual importance, a lightweight temporal model forecasts the user's 6-DoF viewport pose for the upcoming chunk, as shown in Fig. 5.2:

$$\widehat{\mathbf{v}}(t) = [\hat{x}, \hat{y}, \hat{z}, \hat{\text{yaw}}, \hat{\text{pitch}}, \hat{\text{roll}}](t) \tag{5.1}$$

Here, $\hat{x}, \hat{y}, \hat{z}$ denote the predicted 3D user position, and $\hat{\text{yaw}}, \hat{\text{pitch}}, \hat{\text{roll}}$ denote the predicted orientation angles of the user's 6DoF pose for the next chunk. This predicted pose defines a 3D view frustum, allowing spatial classification of tiles into three angular regions:

$$T_n \in \begin{cases} \mathcal{F}_v, & \text{if } \Delta\psi_n < \tau_v \\ \mathcal{F}_m, & \text{if } \tau_v \leq \Delta\psi_n < \tau_m \\ \mathcal{F}_b, & \text{otherwise} \end{cases} \tag{5.2}$$

where $\Delta\psi_n$ denotes the angular offset between tile T_n and viewport center, and τ_v, τ_m are region thresholds for $\mathcal{F}_v$ (viewport), $\mathcal{F}_m$ (marginal), and $\mathcal{F}_b$ (background) regions.

To prioritize perceptually critical tiles, a Laplace-based angular attention model computes the relevance of each tile:

$$\mathcal{P}_v(n,t) = \frac{1}{2b} \exp\left(\frac{-|\varphi_n - \hat{\varphi}_n(t)|}{b}\right) \tag{5.3}$$

where ϕ_n denotes the angular position of tile T_n, $\hat{\phi}_n(t)$ represents the predicted viewport center direction, and b is a dispersion parameter controlling the spread of the attention distribution. A distance-aware LoD tiling tech-

nique modifies tile granularity according to viewing distance and region relevance.

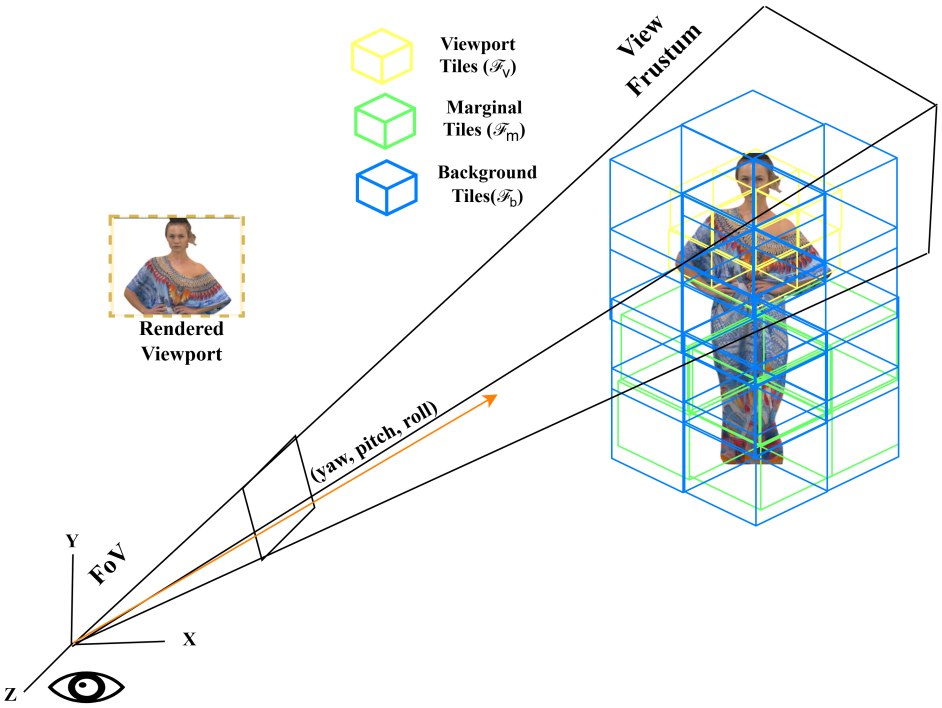


Figure 5.2: 6-DoF coordinates, viewport, and view frustum in the visual space illustrating adaptive tiling of volumetric content.

The perceptual utility used in *METAVV* is constructed by combining a PCQM-derived tile fidelity term with viewport- and distance-aware weighting. Specifically, for each tile T_n at time t, we first compute a perceptual fidelity score $\mathcal{S}_n(t)$ using PCQM [36], which captures both geometry and color distortions between the reference and reconstructed point-cloud content. *METAVV* then maps this PCQM-based fidelity into a viewport-tile utility by applying (i) a viewport importance weight ν_v and (ii) a distance attenuation factor $\frac{1}{1+\eta d_n(t)}$, where $d_n(t)$ is the user tile Euclidean distance and η controls how strongly distance reduces the tile's contribution (larger η yields faster decay and thus emphasizes closer tiles):

$$\mathcal{Q}_n^{(v)}(t) = \nu_v \cdot \frac{1}{1 + \eta \cdot d_n(t)} \cdot \mathcal{S}_n(t). \tag{5.4}$$

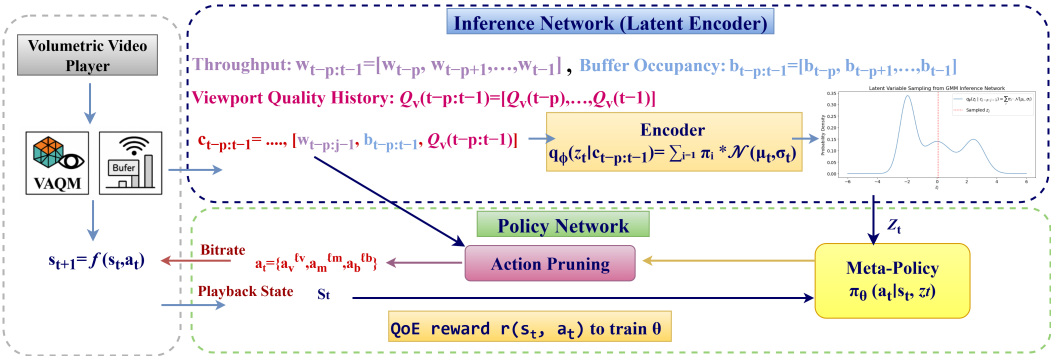


Figure 5.3: Client-side adaptation logic in *METAVV* using latent inference and meta-policy networks.

Finally, to obtain the cumulative viewport-centric perceptual quality for the current chunk, *METAVV* aggregates viewport-region tiles $n \in \mathcal{F}_v$ employing the angular attention weights $\mathcal{P}_v(n,t)$ from Eq. (5.3), which reflect each tile's likelihood of being viewed:

$$\mathcal{Q}_v(t) = \sum_{n \in \mathcal{F}_v} \mathcal{P}_v(n,t) \cdot \mathcal{Q}_n^{(v)}(t). \tag{5.5}$$

Overall, this formulation prioritizes tiled regions that are both more likely to fall within the predicted FoV and closer to the user, thereby contributing more to the viewport-centric quality score $\mathcal{Q}_v(t)$ that guides region-prioritized bitrate/LoD selection.

The viewport-weighted quality in METAVV is computed explicitly in Eq. 5.5. First, Eq. 5.5 maps the PCQM-derived tile fidelity $S_n(t)$ into a viewport-aware and distance-aware tile utility $Q_n^{(v)}(t)$ using the viewport importance coefficient ν_v and the distance attenuation term $\frac{1}{1+\eta d_n(t)}$. Then, Eq. 5.3 aggregates the resulting tile utilities using the angular attention weight $P_v(n,t)$, which represents the probability that tile T_n lies close to the predicted viewport centre. Region weighting is introduced through the classification of tiles into F_v, F_m, and F_b, with the viewport region receiving finer LoD and higher bitrate, the marginal region receiving moderate LoD/bitrate, and the background region receiving lower LoD/bitrate. Therefore, METAVV does not compute perceptual quality as a uniform average over the entire point

cloud; instead, it prioritises tiles that are more likely to be viewed, closer to the user, and perceptually more important.

To facilitate adaptive region-based rate control under fluctuating network conditions, *METAVV* defines a discrete bitrate pool $\mathcal{R} = \{R_1, R_2, \ldots, R_K\}$, consisting of K available bitrate levels ($K \geq 3$), sorted in descending order such that $R_1 > R_2 > \cdots > R_K$. Instead of directly assigning a fixed bitrate to each spatial region, *METAVV* first defines three representative bitrate sub-ranges denoted as R_H (high), R_M (medium), and R_L (low) to represent the bitrate selection for the viewport, marginal, and background regions, respectively. The high-quality range $R_H = \{R_1 - R_3\}$ serves as the bitrate candidate pool for the perceptually critical viewport region $\mathcal{F}_v$. The medium-quality range $R_M = \{R_{k-1}, R_k, R_{k+1}\}$, where $k = \lfloor K/2 \rfloor$, is associated with the marginal region $\mathcal{F}_m$. The low-quality range $R_L = \{R_{K-2}, R_{K-1}, R_K\}$ serves as the candidate set for the background region $\mathcal{F}_b$.

At each time step t, the actual bitrate assigned to each region is selected from its respective sub-range based on real-time system conditions: current network throughput w_t and buffer occupancy b_t. Scheme-defined thresholds, w_{th} and b_{th}, are used to determine whether the network state is favorable or constrained. The bitrate selection strategy is expressed as:

$$\{\mathcal{R}_v, \mathcal{R}_m, \mathcal{R}_b\} = \begin{cases} \{R_H, R_M, R_L\}, & \text{if } w_t > w_{\text{th}} \text{ and } b_t > b_{\text{th}}, \\ \{R_M, R_L, R_L\}, & \text{otherwise} \end{cases} \tag{5.6}$$

where $\mathcal{R}_v$, $\mathcal{R}_m$, and $\mathcal{R}_b$ denote the bitrate selections for the viewport, marginal, and background regions, respectively. This hierarchical assignment ensures that perceptually critical regions (e.g., $\mathcal{F}_v$) receive higher bitrate and finer LoD encoding, while less salient areas (e.g., $\mathcal{F}_b$) are coarsely encoded to conserve bandwidth under constrained conditions.

Fig. 5.4 illustrates the distance-aware refinement strategy using three distance thresholds set to d_1=0.5 m, d_2=1.2 m, and d_3=2.2 m), corresponding to region-based tiling and density adaptation for a volumetric client at

near, mid-range, and far viewing distances, respectively. As the user moves farther from the scene, the predicted FoV derived from the forecasted 6-DoF pose $\widehat{\mathbf{v}}(t)$ expands in both vertical and horizontal directions due to the widening of the visual frustum, consistent with human perception, where peripheral coverage increases with distance. This change in spatial coverage requires an adaptation of the tiling/point-density configuration. Specifically, the viewport region (orange) spans a larger area but is rendered with reduced point cloud density as distance increases, reflecting the distance-weighted perceptual utility in Eq. (5.4). In particular, as the Euclidean distance $d_n(t)$ grows, the attenuation factor $\frac{1}{1+\eta \cdot d_n(t)}$ lowers the benefit of fine-detail rendering.

In parallel, the angular attention term $\mathcal{P}_v(n,t)$ (Eq. (5.3)) weights tiles by their directional proximity to the predicted viewport center $\hat{\theta}_n(t)$. The combined distance and angular weighting enables efficient prioritization of visually salient regions across varying FoV conditions, improving perceptual fidelity while reducing redundant rendering and transmission. The marginal (green) and background (blue) regions keep fixed tiling/density to maintain stable encoding overhead, illustrating how *METAVV* balances LoD fidelity and network cost as a function of user distance.

5.1.3 Client Adaptation with Meta-Learner

As illustrated in Fig. 5.1, the client-side adaptation loop in *METAVV* fuses view-centric cues from VAQM (e.g., predicted viewport relevance and perceptual quality) with network-state measurements from NSI (e.g., throughput and buffer) to construct the streaming state s_t for each decision step (chunk) t. This state is passed to the *Meta-Learner*, which is meta-trained across diverse tasks (users, contents, and network traces) to learn a generalized initialization. During online streaming, a *Context Encoder* utilizes the recent state action reward history to infer a latent session context $z_t \sim q_\phi(z_t \mid \cdot)$, capturing short-term dynamics of the current session. Conditioned on (s_t, z_t), the *Meta-Conditioned ABR Policy* $\pi_\theta(a_t \mid s_t, z_t)$ selects the

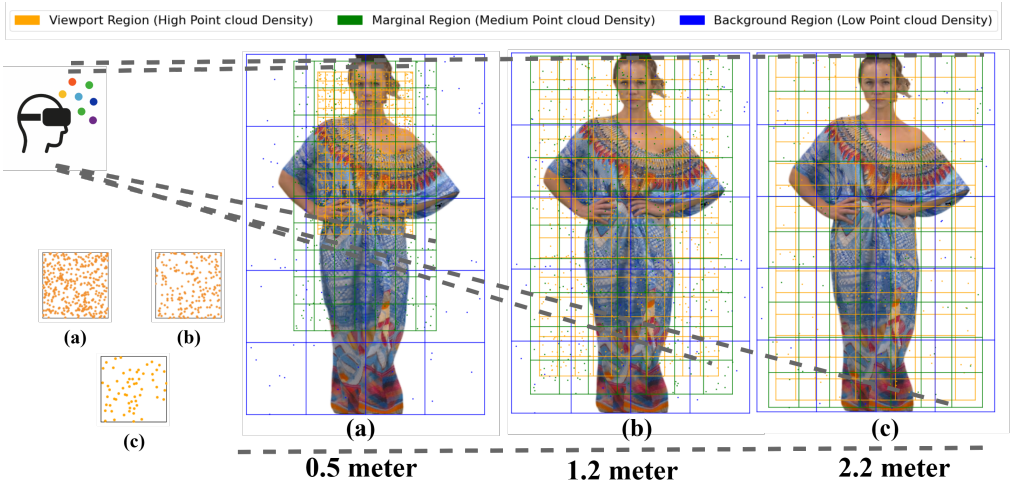


Figure 5.4: Distance-aware point cloud adaptation in *METAVV*. Illustration of region-based tiling and density adaptation for a volumetric client at (a) 0.5 m, (b) 1.2 m, and (c) 2.2 m distances. The predicted FoV (orange) expands with distance, while the point cloud density decreases to reduce rendering and bandwidth costs. The marginal (green) and background (blue) regions are assigned moderate and low densities, respectively. Insets (a)–(c) highlight varying point cloud densities mapped to the viewport region for each distance scenario.

next action a_t (region-wise bitrate/LoD choice) for the upcoming volumetric chunk. The server then delivers the corresponding representation adapted to the viewport-relevant tiled region, the client observes the reward r_t, and the transition $s_{t+1} = f(s_t, a_t)$ closes the loop for continuous task-aware rate adaptation.

Fig. 5.3 further details the inference and policy stages of this client-side loop. Each volumetric streaming session is treated as a separate task with its own dynamics. To handle partial observability caused by unpredictable viewport motion and fluctuating network conditions, *METAVV* employs an inference module I_ψ, parameterized by ψ, which aggregates historical observations into a compact latent belief state z_t.

Let p denote the number of past timesteps included in the client's context history, and let $c_{t-p:t-1}$ represent the concatenated vector of past throughput measurements w, buffer occupancy values b, and viewport-perceptual quality scores $\mathcal{Q}_v$, as shown in Fig. 5.3. At each decision step t, the client observes:

$$c_{t-p:t-1} = [w_{t-p:t-1},\ b_{t-p:t-1},\ \mathcal{Q}_v(t-p:t-1)] \tag{5.7}$$

where $c_{t-p:t-1}$ denotes the observation window over the past p timesteps, $w_{t-p:t-1}$ and $b_{t-p:t-1}$ are the historical throughput and buffer occupancy sequences, and $\mathcal{Q}_v(t-p:t-1)$ is the corresponding viewport-perceptual quality history.

This sequence forms the input to a *latent encoder*, a probabilistic inference network parameterized by ϕ, which is responsible for producing the belief z_t. The latent encoder in *METAVV* is implemented as a Gaussian Mixture Model (GMM) based inference network that captures the multi-modal characteristics of complex streaming environments:

$$z_t \sim q_\phi(z_t \mid c_{t-p:t-1}) = \sum_{i=1}^{n} \pi_i \cdot \mathcal{N}(\mu_t^{(i)}, \sigma_t^{(i)}) \tag{5.8}$$

where z_t denotes the latent belief at time t, $q_\phi(z_t \mid c_{t-p:t-1})$ is the variational posterior inferred from the observation window, n is the number of mixture components, and π_i, $\mu_t^{(i)}$, and $\sigma_t^{(i)}$ are the mixture weight, mean, and variance of the i-th Gaussian component, respectively.

The resulting latent belief vector z_t encodes short-term session dynamics, including the interplay between network throughput, buffer evolution, and viewport-centric perceptual quality, as illustrated in Fig. 5.3. By supplying this latent variable to the policy network, *METAVV* enables fast and robust adaptation to diverse and previously unseen streaming conditions. The GMM-based inference mechanism follows insights from recent work on task inference for RL in non-stationary settings [192], providing a principled approach to summarizing heterogeneous observations into a low-dimensional context vector. Unlike general PEARL, where the latent context is used with a task-agnostic policy backbone, in *METAVV* the inferred belief z_t is embedded into a volumetric streaming pipeline with structured region-wise actions and perceptual decision variables. In particular, z_t conditions bitrate/LoD allocation across viewport, marginal, and background regions, while the downstream reward is computed from a viewport and distance-aware PCQM-driven QoE model.

5.1.3.1 Region-Based Action Selection with Latent Meta-Policy

Given the latent belief vector z_t inferred from historical session observations, the policy network π_θ synthesizes both z_t and the current observable state s_t to make region-wise adaptation decisions. The action output a_t determines bitrate and LoD assignments for the viewport, marginal, and background mesh regions:

$$a_t = \{a_v, a_m, a_b\} \sim \pi_\theta(a_t \mid s_t, z_t) \tag{5.9}$$

where a_t denotes the selected region-wise action, $\pi_\theta(a_t \mid s_t, z_t)$ is the meta-conditioned policy, and a_v, a_m, and a_b represent the bitrate–LoD selections for the viewport ($\mathcal{F}_v$), marginal ($\mathcal{F}_m$), and background ($\mathcal{F}_b$) regions, respectively.

If enabled, an optional *action pruning module* filters out suboptimal or infeasible adaptation actions based on real-time constraints such as bandwidth, decoder complexity, and latency limits. By leveraging the learned policy and region-aware adaptation granularity, *METAVV* allocates high-fidelity resources to viewport content while reducing bandwidth assigned to less critical regions. Unlike my prior EDGE360 framework, which used VMAF-based tile evaluation for 360° video [19], *METAVV* integrates a distance-aware perceptual utility model that aligns with 6-DoF motion and mesh complexity. This enables adaptive fidelity tuning for both near-field high-detail content and far-field sparse objects.

Upon rendering chunk $\mathcal{C}_{t+1}$, the framework evaluates QoE utilizing the following reward formula:

$$\mathcal{Q}oE(t) = \mathcal{Q}_v(t) - \alpha \cdot \Delta\mathcal{Q}_v(t) - \beta \cdot \rho(t) - \zeta \cdot \mathcal{D}(t) \tag{5.10}$$

where $\mathcal{Q}_v(t)$, defined in Eq. (5.5), represents the distance-aware viewport perceptual quality, and $\Delta\mathcal{Q}_v(t)$ penalizes abrupt inter-chunk quality variation and encourages temporal quality smoothness. The term $\rho(t)$ captures

the total stall and rebuffering duration during chunk playback, and $\mathcal{D}(t)$ denotes the playback latency measured from chunk request to rendering. The tunable coefficients α, β, and ζ weight the influence of these penalties.

This composite reward scheme enables the agent to balance high perceptual quality, playback stability, and low-latency responsiveness across heterogeneous streaming conditions. The formal Meta-POMDP definition and the corresponding inner and outer-loop optimization are presented in Section IV.

5.1.4 Viewport and Distance-Aware Region-Based Adaptation Mechanism

After the client selects a region-wise bitrate/LoD action a_t, the volumetric video server delivers the corresponding encoded assets for the next chunk. Specifically, the server receives an action command $a_t = (R_t, \lambda_t)$, where R_t denotes the selected bitrate profile (applied per region) and $\lambda_t \in \{\lambda_1, \lambda_2, \lambda_3\}$ specifies the point cloud density (LoD) levels for the viewport, marginal, and background regions. The server then transmits only the geometry segments corresponding to the selected LoD values $(\lambda_v, \lambda_m, \lambda_b)$ and bitrate assignments (R_v, R_m, R_b), further restricted to tiles predicted to fall within the client's viewport for the upcoming chunk, thereby reducing unnecessary bandwidth consumption.

The VAQM module filters out non-visible tiles, while NSI and the QoE estimators enforce feasible trade-offs among perceptual quality, resource usage, and decoding cost. In this way, the delivery mechanism translates the policy decision into viewport-aware and distance-aware content transmission under dynamic user motion and heterogeneous network conditions.

Algorithm 2 presents the latent-conditioned FoV and distance-aware region-based bitrate adaptation logic in *METAVV*. This module serves as a mechanism for providing input support to the meta-policy network's prediction and adaptation decisions in Algorithm 3, leading to more informed

decisions. Given the predicted 6-DoF pose $\hat{v}_t$ and a tile set $\mathcal{P}$ with spatial centroids κ_n, tiles are first segmented into three spatial regions viewport $\mathcal{F}_v$, marginal $\mathcal{F}m$, and background $\mathcal{F}b$ based on angular offset $\Delta\theta_n$ and user-centric distance d_n. Simultaneously, a latent context vector z_t is inferred via the GMM-based inference module $q_\phi(z_t \mid ct-p:t-1)$, which encodes the client's recent throughput, buffer, and perceptual quality history. This belief guides the adaptive assignment of region-wise bitrate suggestions $\mathcal{R}v, \mathcal{R}m, \mathcal{R}b$ using throughput and buffer thresholds $w\text{th}, b\text{th}$, thereby constraining the meta-policy's feasible action space. After the requested chunk is delivered, the algorithm computes a QoE reward r_t, observes the next state s_{t+1}, and appends the transition (s_t, a_t, r_t, s_{t+1}) to the context buffer, enabling posterior refinement for the next decision via $z_{t+1} \sim q_\phi(z_{t+1} \mid c_{t-p:t})$. This closed-loop inference update supports temporally consistent, perceptually guided adaptation under dynamic network and user motion conditions.

This adaptive mechanism enables *METAVV* to perform viewpoint and distance-aware resource allocation, thus reducing the redundant transmission of tiles with low perceptual relevance. Consequently, it improves both perceived visual quality and streaming efficiency under heterogeneous network conditions and diverse user navigation trajectories.

5.2 *METAVV* Rate Adaptation and Problem Formulation

The *METAVV* solution uses a Meta-Markov Decision Process (Meta-MDP) to improve perceptual quality, reduce latency, and increase robustness in VVS. In real-world circumstances where future viewport trajectories, network conditions, and system delays are unpredictable, a Meta-POMDP offers a more realistic and robust formulation.

This Meta-POMDP structure incorporates uncertainty and delayed feedback by allowing the agent to deduce hidden state dynamics from observable signals (e.g., buffer trends, spatial saliency, viewport shifts, and previ-

Algorithm 2: FoV and Distance-Aware Region-Based Bitrate Adaptation in *METAVV*

Input : Predicted 6-DoF pose $\hat{v}_t = (x, y, z, \hat{\text{yaw}}, \hat{\text{pitch}}, \hat{\text{roll}})$,
Tile set $\mathcal{P} = \{T_n\}_{n=1}^{N}$ with centroids κ_n,
Angular thresholds: τ_v, τ_m;
distance thresholds: d_1, d_2,
Client session history: $c_{t-p:t-1} = [w_{t-p:t-1}, b_{t-p:t-1}, \mathcal{Q}_v(t-p:t-1)]$,
Bitrate subranges $\mathcal{R} = \{R_H, R_M, R_L\}$,
Thresholds: throughput w_{th}, buffer b_{th}

Output : Region-wise bitrate suggestions conditioned on z_t:
$\{\mathcal{R}_v, \mathcal{R}_m, \mathcal{R}_b\}$

Step 1: Infer client-specific latent context: $z_t \sim q_\phi(z_t \mid c_{t-p:t-1})$ Eq. (5.8)
Initialize region tile sets: $\mathcal{F}_v = \emptyset$, $\mathcal{F}_m = \emptyset$, $\mathcal{F}_b = \emptyset$
foreach $T_n \in \mathcal{P}$ **do**
    Compute angular offset $\Delta\theta_n$ from predicted viewport
    Compute distance $d_n = \|\kappa_n - (x, y, z)\|$
    if $(\Delta\theta_n < \tau_v) \cap (d_n < d_1)$ **then**
        Add T_n to $\mathcal{F}_v$
    else if $(\Delta\theta_n < \tau_m) \cap (d_1 \leq d_n < d_2)$ **then**
        Add T_n to $\mathcal{F}_m$
    else
        Add T_n to $\mathcal{F}_b$
    end
end
Step 2: Conditioned bitrate range assignment per region **if** $\bar{w}_{t-p:t-1} > w_{th} \cap \bar{b}_{t-p:t-1} > b_{th}$ **then**
    $\mathcal{R}_v \leftarrow R_H$, $\mathcal{R}_m \leftarrow R_M$, $\mathcal{R}_b \leftarrow R_L$
else
    $\mathcal{R}_v \leftarrow R_M$, $\mathcal{R}_m \leftarrow R_L$, $\mathcal{R}_b \leftarrow R_L$
end
return $\{\mathcal{R}_v, \mathcal{R}_m, \mathcal{R}_b\}$
Step 3: Observe reward and update inference context
Compute QoE reward r_t
Observe next state s_{t+1}
Append transition to context buffer: $c_{t-p:t-1} \leftarrow c_{t-p:t-1} \cup \{(s_t, a_t, r_t, s_{t+1})\}$
Update posterior for next step: $z_{t+1} \sim q_\phi(z_{t+1} \mid c_{t-p:t-1})$
return (a_t, z_{t+1})

ous throughput) rather than depending on complete environment visibility. The inference network learns compact latent variables that inform the policy, allowing for adaptive and generalisable decision-making across variable streaming settings.

Formally, employing the Meta-POMDP and QoE-aware reward design, the meta-learning objective is defined as:

Problem P1:

$$\max_{\mathbf{\Phi}} \mathbb{E}\left[\sum_{t=0}^{H} \gamma^t \cdot \mathcal{Q}oE(t)\right] \tag{5.11}$$

$\mathbf{\Phi}$ represents the shared initialization for both the inference and policy networks, and $\gamma \in [0, 1]$ is the discount factor. These parameters are modified via meta-gradient updates to allow for quick adaptability across a variety of streaming situations. The outer loop optimizes long-term QoE, while the inner loop fine-tunes the model for task-specific responsiveness, forming the basis of the *METAVV* learning process.

5.2.1 Foundation of Meta-POMDP

In order to maximize expected QoE, the client chooses region-wise bitrate and point cloud density levels over a temporal sequence of decision points in the *METAVV* adaptive streaming framework. This interaction is modelled as a Meta-POMDP, described by the tuple:

$$\mathcal{M}_{\text{meta}} = (\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, r, \gamma, \rho_0, \mathbf{\Phi}) \tag{5.12}$$

where $\mathcal{M}_{\text{meta}}$ denotes the Meta-POMDP formulation of the streaming environment, $\mathcal{S}$ is the latent state space, $\mathcal{A}$ the action space (region-wise bitrate/LoD choices), $\mathcal{O}$ the observation space, $\mathcal{T}$ the transition dynamics, r the QoE-driven reward, γ is the discount factor, ρ_0 is the initial state distribution, and Φ represents the meta-learned parameters.

At each time step t, the underlying latent state $s_t \in \mathcal{S}$ encodes the viewport-centric perceptual quality $\mathcal{Q}_v(t)$, current buffer occupancy $b(t)$, and recent network throughput $w(t)$. These variables represent the actual

conditions experienced by the client but are only partially observable due to latency, estimation noise, and measurement limitations.

Instead, the client receives observations $o_t \in \mathcal{O}$ containing delayed and noisy estimates $\hat{\mathcal{Q}}_v(t)$, $\hat{b}(t)$, and $\hat{w}(t)$ corresponding to the true latent state. These observations, along with historical action–reward pairs where $r_{1:t-1}$ denote the observed QoE rewards from previous decisions, are processed by an inference network I_ψ, parameterized by ψ, to infer a representation of latent belief:

$$\hat{s}_t = I_\psi(o_{1:t}, a_{1:t-1}, r_{1:t-1}) \tag{5.13}$$

The belief state $\hat{s}_t$ serves as an approximation of the true latent state under partial observability. It is passed to the policy network π_θ, parameterized by task-specific weights θ, to generate a region-wise action $a_t \sim \pi_\theta(a \mid \hat{s}_t)$.

The transition model $\mathcal{T}(s_{t+1} \mid s_t, a_t)$ governs the system's evolution over time, reflecting changes in the state due to user posture, bandwidth variations, and the implemented bitrate/LoD selection.

The objective is to meta-train a policy π_θ such that, given a new task instance $\mathcal{T}_i \sim p(\mathcal{T})$, the adapted parameters initialized from $\mathbf{\Phi} = \psi, \theta$ can optimize long-term QoE through:

$$\pi^* = \arg\max_{\pi_\theta}; \mathbb{E}\mathcal{M}\text{meta}\left[\sum_{t=0}^{H} \gamma^t \cdot \mathcal{Q}oE(t)\right] \tag{5.14}$$

The *METAVV* framework employs this Meta-POMDP configuration to effectively encapsulate streaming variability and uncertainty into belief-driven representations, allowing for quick, session-aware policy adaptation across dynamic user and network conditions.

5.2.2 State Formulation for *METAVV* Adaptive Volumetric Streaming

The client in the *METAVV* architecture operates autonomously in a highly dynamic and partially observable streaming environment. All decision-making transpires immediately on the user device, and the policy is con-

stantly updated depending on current QoE feedback. This allows the client to respond swiftly to changes in network bandwidth, viewport motion, and scene complexity, allowing the framework to generalise well across a wide range of streaming situations without the need for external coordination and shared control mechanisms. At each decision time t, the client constructs a feature observation vector:

$$\mathcal{O}(t) = \left\{ \hat{\mathcal{Q}}_v^{\text{hist}}(t),\ \hat{\mathcal{B}}(t),\ \hat{\mathcal{H}}_{\mathcal{W}}(t),\ \ldots \right\} \tag{5.15}$$

where $\hat{\mathcal{Q}}_v^{\text{hist}}(t)$ refers to past viewport quality estimates, $\hat{\mathcal{B}}(t)$ the buffer level, and $\hat{\mathcal{H}}_{\mathcal{W}}(t)$ the throughput history all potentially noisy due to prediction and sensing inaccuracies. The inference network I_ψ processes these observations and past interactions to output a belief state:

$$\hat{\mathcal{S}}(t) = I_\psi \left(o_{1:t}, a_{1:t-1}, r_{1:t-1} \right) \tag{5.16}$$

This latent state $\hat{\mathcal{S}}(t)$ then guides the policy network π_θ in selecting bitrate and point cloud density decisions.

The true underlying state of the client at time t is defined as follows:

$$\begin{aligned} \mathcal{S}(t) = \Big\{ & \mathcal{H}_{\mathcal{W}}(t),\ \boldsymbol{\omega}(t),\ \boldsymbol{\Lambda}(t),\ \mathcal{B}(t),\ \mathcal{Q}_v^{\text{hist}}(t),\ \mathcal{P}_v(t), \\ & \mathcal{U}(t),\ \mathcal{D}_v(t),\ \mathcal{G}_{\text{seg}}(t),\ \Delta\mathcal{Q}_v(t),\ \rho(t),\ \mathcal{D}(t) \Big\} \end{aligned} \tag{5.17}$$

where $\mathcal{S}(t)$ denotes the system state at time t; $\mathcal{H}_{\mathcal{W}}(t)$, $\boldsymbol{\omega}(t)$, and $\boldsymbol{\Lambda}(t)$ represent throughput/download dynamics, viewport orientation, and region-specific LoD configuration, respectively; $\mathcal{B}(t)$ and $\mathcal{Q}_v^{\text{hist}}(t)$ denote the buffer level and viewport-quality history; $\mathcal{P}_v(t)$ (Eq. 5.3) is the predicted visibility map, $\mathcal{U}(t) = [u_v(t), u_m(t), u_b(t)]^T$ denotes region data sizes, and $\mathcal{D}_v(t) = [d_v(t), d_m(t), d_b(t)]^T$ captures perceptual degradation derived from PCQM with distance-weighted penalization. The segmentation layout $\mathcal{G}_{\text{seg}}(t)$ encodes the active region granularities and associated LoD costs, while $\Delta\mathcal{Q}_v(t)$, $\rho(t)$,

and $\mathcal{D}(t)$ follow Eq. (5.10). These multimodal signals are compacted into belief representations to enable resilient, QoE-aware bitrate/LoD adaptation under partial observability.

5.2.3 Meta-RL Policy Learning for *METAVV*

Building on the latent adaptation pipeline established in Section 5.2, *METAVV* applies a meta-RL strategy to enable rapid policy personalization across dynamic volumetric streaming sessions. Each session is treated as a task $\mathcal{T}_i \sim p(\mathcal{T})$, representing unique combinations of bandwidth variation, user pose trajectory, and content complexity.

The objective of meta-training is to learn a global parameter initialization Φ from which client-specific policies π_θ can quickly adapt to maximize long-term perceptual QoE under session-specific dynamics. The policy is conditioned on the latent belief z_t, inferred from recent streaming context $c_{t-p:t-1}$, and generates region-level bitrate and LoD actions accordingly. To enable value-guided adaptation, *METAVV* utilizes the twin critic network $Q_{\phi_1,\phi_2}(s_t, a_t, z_t)$ to estimate action values under the inferred context. The policy is optimized using QoE-driven reward feedback obtained from the executed region-wise actions. In particular, the inner-loop adaptation updates the task-specific policy parameters using recent session experience, while the outer-loop optimization refines the shared initialization across tasks.

5.2.3.1 Inner and Outer MetaRL Updates

At each time step, a policy copy $\tilde{\theta}_t$ is adapted using an inner-loop gradient update that optimizes a temporally discounted QoE objective:

$$\tilde{\theta}_{t+1} = \tilde{\theta}_t + \eta \cdot \nabla_{\tilde{\theta}_t} \mathcal{L}_{\text{adapt}}(\tilde{\theta}_t) \tag{5.18}$$

$$\mathcal{L}_{\text{adapt}}(\tilde{\theta}_t) = -\sum_{t=0}^{H} \gamma^t \cdot \mathcal{QoE}(t) \tag{5.19}$$

This inner-loop loss fine-tunes the task-specific parameters $\tilde{\theta}_t$ using post-playback QoE signals, where the temporal discount factor γ modulates the weight of immediate versus future perceptual quality. Unlike the meta-objective in Eq. (5.11), which updates the shared initialization Φ across task distributions, $\mathcal{L}_{\text{adapt}}$ specializes the policy within a single task $\mathcal{T}_i$ based on short-term episodic feedback. This enables fast adaptation without requiring long training traces or prior task knowledge.

To support value-based learning, the twin critics are optimized using soft double Q-learning to reduce overestimation bias [193]. The target value for both critics is computed using the minimum of the delayed target networks:

$$y_t = r_t + \gamma \cdot \min_{j \in \{1,2\}} Q_{\bar{\phi}_j}(s_{t+1}, \pi_\theta(s_{t+1}, z_t), z_t) \tag{5.20}$$

where y_t is the target Q-value and $Q_{\bar{\phi}_j}$ denotes the target critic networks, and each critic is updated by minimizing the temporal-difference (TD) error with respect to its target:

$$\mathcal{L}^{(j)}_{\text{critic}} = \mathbb{E}_{(s_t,a_t,r_t,s_{t+1})\sim\mathcal{D}} \left[\left(Q_{\phi_j}(s_t, a_t, z_t) - y_t \right)^2 \right] \tag{5.21}$$

where $j \in \{1, 2\}$, $\mathcal{D}$ denotes the episodic experience replay buffer containing past transitions (s_t, a_t, r_t, s_{t+1}) used to stabilize the learning of the twin Q-networks through mini-batch sampling. This twin-critic structure improves learning stability by avoiding overly optimistic Q-value estimation and enables more robust policy updates.

The meta-learner then performs an outer-loop update to improve the initialization Φ using the adaptation feedback:

$$\Phi \leftarrow \Phi - \upsilon \cdot \nabla_\Phi \, \mathbb{E}_{\mathcal{T}_i \sim p(\mathcal{T})} \left[\mathcal{L}_{\text{meta}}(\tilde{\theta}_{t+1}) \right] \tag{5.22}$$

This ensures that Φ evolves to serve as a strong starting point for future sessions, allowing rapid personalization from minimal observations without requiring task-specific retraining.

5.2.3.2 Entropy and Latent Regularization

To promote exploration and enhance generalization, the meta-training process incorporates two regularization components:

Entropy Regularization Encourages stochasticity in policy actions, preventing premature convergence:

$$\mathcal{E}(\pi_{\tilde{\theta}_t}) = -\sum_{a} \pi_{\tilde{\theta}_t}(a \mid s_t, z_t) \log \pi_{\tilde{\theta}_t}(a \mid s_t, z_t) \tag{5.23}$$

where $\mathcal{E}(\pi_{\tilde{\theta}_t})$ denotes the entropy regularization term, $\pi_{\tilde{\theta}_t}(a \mid s_t, z_t)$ is the policy distribution conditioned on the state s_t and latent variable z_t, and a denotes the action space.

Latent Prior Alignment Ensures that the learned posterior $q_\psi(z_t \mid c_t)$ remains compact and consistent with a standard prior $p(z_t)$, typically assumed Gaussian. This is enforced via the Kullback–Leibler (KL) divergence [194] term:

$$\mathcal{D}_{\text{KL}}(q_\psi(z_t \mid c_t) \parallel p(z_t)) \tag{5.24}$$

5.2.3.3 Final Meta-Objective

The complete meta-loss function optimized in Eq. (5.22), (5.23), (5.24) combines all components:

$$\begin{aligned} \mathcal{L}_{\text{meta}}(\tilde{\theta}_t) = \mathcal{L}_{\text{adapt}}(\tilde{\theta}_t) + \eta\, \mathcal{E}(\pi_{\tilde{\theta}_t}) \\ + \omega_{\text{KL}}\, \mathcal{D}_{\text{KL}}(q_\psi(z_t \mid c_t) \parallel p(z_t)) + \omega_Q \mathcal{L}^{(j)}_{\text{critic}} \end{aligned} \tag{5.25}$$

Here, η, ω_{KL}, and ω_Q balance the contributions of policy entropy, latent alignment, and twin critic consistency. This unified loss promotes policies that are both adaptive to user behavior and robust to environmental variation, while leveraging compact contextual representations for generalizable

Algorithm 3: METAVV: Meta-Reinforcement Learning for Adaptive Volumetric Video Streaming

Input : Meta-policy parameters Φ,
Inner learning rate η, meta learning rate υ,
Number of training episodes K,
Adaptation horizon T,
Client environment distribution $\mathcal{E}$

Output Meta-learned initialization Φ^*

:

foreach *episode* $k = 1, \ldots, K$ **do**
  Sample client environment $\mathcal{E}_k \sim \mathcal{E}$
  Initialize task-specific policy: $\tilde{\theta}_0 \leftarrow \Phi$ `// Meta-initialization`
  foreach *chunk timestep* $t = 1, \ldots, T$ **do**
    Observe partial state $\mathcal{S}(t)$ as in Eq. (5.17)
    Encode latent belief $z_t \sim q_\psi(z_t \mid c_{t-p:t-1})$ via GMM inference
    Sample region-wise bitrate–density action $a_t \sim \pi_{\tilde{\theta}_t}(a_t \mid \mathcal{S}(t), z_t)$ using Eq. (5.9)
    Compute QoE reward $r(t) = \mathcal{Q}oE(t)$ using Eq. (5.10)
    Sample next action $a_{t+1} \sim \pi_{\tilde{\theta}_t}(a_{t+1} \mid s_{t+1}, z_{t+1})$
    Compute target value $y_t = r(t) + \gamma \cdot \min_{j=1,2} Q_{\phi'_j}(s_{t+1}, a_{t+1}, z_{t+1})$ `// Eq.` (5.20)
    Update twin critic networks:
    $\mathcal{L}^{(j)}_{\text{critic}} = \big(Q_{\phi_j}(s_t, a_t, z_t) - y_t\big)^2$, for $j \in \{1, 2\}$ `// Eq.` (5.21)
    Update inner-loop actor policy:
    $\tilde{\theta}_{t+1} \leftarrow \tilde{\theta}_t + \eta \cdot \nabla_{\tilde{\theta}_t} \mathcal{L}_{\text{adapt}}(\tilde{\theta}_t)$ `// Eq.` (5.18)`, with Eq.` (5.19)
  end
  Evaluate adapted policy $\pi_{\tilde{\theta}_T}$ on validation session
  Compute meta-gradient: $\nabla_\Phi \mathcal{L}_{\text{meta}}(\tilde{\theta}_T)$ `// Eq.` (5.25)
  Update meta-policy:
  $\Phi \leftarrow \Phi - \upsilon \cdot \nabla_\Phi \mathcal{L}_{\text{meta}}(\tilde{\theta}_T)$ `// Eq.` (5.22)
end
return Φ^*

learning. To operationalize the meta-learning framework described above, Algorithm 3 outlines the end-to-end *METAVV* training process, encompassing latent inference, region-wise policy adaptation, and meta-gradient updates across episodic client environments.

5.3 Implementation and Evaluation

5.3.1 Simulation Settings

To evaluate the performance of *METAVV*, a complete VVS pipeline in Python 3.9 using PyTorch, running on Ubuntu 24.04 LTS, was implemented. Simulations were executed on a high-performance setup featuring an Intel Core i7-13700 CPU (16 cores, 2.50 GHz), 64 GB RAM, and an NVIDIA RTX 3090 GPU (24 GB VRAM). Development was carried out using the PyCharm IDE.

We employed the Google DRACO codec for compression and decompression of RGB-D point cloud sequences [195], and evaluated perceptual quality using the full-reference PCQM metric [36]. To support adaptive spatial encoding, the Open3D library [196] was utilized to implement a projection-based mesh segmentation and distance-aware LoD tiling strategy. Unlike *ViVo* [146], which uses a fixed 12-tile layout, METAVV adopts a dynamic tiling strategy based on user-specific 6-DoF pose and viewing distance according to the 3-D space boundary of each frame. It assigns 24, 12, and 6 tiles to the viewport (LoD_1), marginal (LoD_2), and background (LoD_3) regions, respectively, enabling perceptually aware and bandwidth-efficient bitrate allocation.

The latent context variable z is modeled as a 5-dimensional vector, inferred via a two-layer variational encoder with 200 hidden units per layer and ReLU activations. During meta-training, a context buffer of 30 transitions and a batch size of 16 episodes per task were used. The actor, critic, and encoder networks are optimized using the Adam optimizer with individually tuned learning rates of 3×10^{-4}, 1×10^{-3}, and 2×10^{-5}, respectively. These tuned weights ensure stable policy updates, faster critic convergence, and robust inference over latent contexts. The KL divergence weight ω_{KL} for latent regularization is set to 1, fully applying the KL penalty in the variational lower bound to ensure regularization of the latent space and prevent



(a) 8iVSLF dataset: *Longdress*, *Red&Black*, and *Soldier*.



(b) Owlii dataset: *Basketball* and *Dancer*.

Figure 5.5: Visual quality representation of transmitted volumetric test video sequences used in the evaluation: (a) 8iVSLF and (b) Owlii datasets.

overfitting. Training is conducted over 100 epochs, with a discount factor $\gamma = 0.99$ to capture long-term rewards.

The viewport-aware QoE reward function incorporates perceptual quality, bitrate smoothness, rebuffering time, and latency, weighted respectively by $\alpha = 1.8$, $\beta = 1.3$, and $\zeta = 1.1$, as determined through ablation tuning. These settings were found to yield robust and fair adaptation under diverse client behaviors and network traces. Hyperparameters were tuned via grid search to achieve robust generalization and fast adaptation in unseen streaming environments.

5.3.2 Video Database and Network Traces

The VV sequences used in this study were selected from the 8iVSLF [197] captured using a multi-camera setup and delivered as a 30 fps sequence over a 10 s interval with a depth-10 voxel grid (1024×1024×1024) and

Owlii [198] dynamic human textured mesh dataset provides dynamic human mesh/point-cloud sequences such as *Basketball* and *Dancer*, captured at 30 fps over 20 s, with ∼40k triangles per frame and 2048×2048 texture maps. Visual quality representation of the volumetric test video sequences is shown in Fig. 5.5a and 5.5b. Specifically, the *Longdress*, *Red&Black*, and *Soldier* sequences from 8iVSLF, and the *Basketball* and *Dancer* sequences from the Owlii dataset were employed. These sequences were chosen to represent a diverse range of motion complexity, spatial geometry, and scene dynamics across two distinct datasets, ensuring coverage of both low-motion, structured scenarios and high-motion complex point cloud variations.

Unlike fixed tiling techniques in [146, 160], *METAVV* adopts a dynamic, distance-aware LoD assignment scheme, wherein LoD granularity adapts to the user's current viewing distance. This allows for encoding-aware bitrate savings while preserving perceptual quality in more visually salient regions.

To simulate realistic network behaviour, publicly available bandwidth traces representing heterogeneous access environments were used.

(i) 4G/LTE traces: the 4G downlink/uplink profiles were taken from the Belgium LTE trace dataset [199], which provides time-series throughput logs measured in an operational cellular network and captures typical mobile variability (e.g., bursty rates, short-term drops, and load-driven fluctuations). Although the target deployment is 5G and beyond, these LTE traces remain relevant because eMBB streaming must operate across *mixed-coverage* conditions (e.g., 5G-to-4G fallback, non-standalone deployments, and edge-of-coverage scenarios), and because the underlying challenges of stochastic wireless capacity and congestion dynamics persist across generations; thus, LTE traces offer a realistic baseline for robustness evaluation under widely observed cellular impairments.

(ii) WiFi traces: WiFi traces were sourced from Kulkarni *et al.* [200], who report measurement-driven WiFi behaviour for volumetric streaming under different WiFi configurations and environments (e.g., home/office), reflecting short-term contention, queueing effects, and latency variability typical

of shared WLANs.

(iii) 5G/mmWave traces: to represent emerging 5G conditions, mmWave-oriented logs from the LIVE-5G system-level evaluation suite [201] were employed, which capture high-peak-rate 5G behaviour alongside abrupt capacity changes due to blockage, mobility, and scheduling dynamics. Together, these traces span a spectrum of real-world congestion patterns, mobility effects, and access heterogeneity, enabling robust evaluation of the proposed *METAVV* solution across diverse network contexts.

5.3.3 Viewport Dataset

The publicly available video saliency data and viewport head-rotation trajectories collected from 40 users under free-viewing conditions [202] were employed to characterize user viewport dynamics in *METAVV*. These head-movement traces provide rich 6-DoF metadata that are essential for accurately simulating user navigation behaviors in volumetric environments. In this dataset, participants viewed VV content on an HMD and were free to explore regions of interest in 6-DoF during playback. For each viewing session, the dataset provides time-ordered *6-DoF head-pose traces* comprising both translation and rotation, i.e., head position (X_t, Y_t, Z_t) and Euler-angle rotations $(\alpha_t, \beta_t, \gamma_t)$ at each time index t.

In addition to per-user trajectories, the dataset includes *frame-level attention cues* that are useful for viewport-aware modelling: (i) aggregated gaze/attention heatmaps constructed across the 40 viewers to indicate where users tend to look in the point-cloud scene, and (ii) corresponding saliency maps that highlight visually prominent regions. The original study reports results on representative human-body point-cloud sequences (e.g., *Soldier*, *Redandblack*, *Loot*, and *Longdress*), and organises frames into training/validation/testing splits (900/150/150 frames) for learning-based viewport prediction. Overall, these multimodal traces (pose + attention/saliency) provide the essential inputs for (a) learning/predicting future viewports and

(b) constructing FoV probability maps that drive perceptually prioritised bitrate/LoD adaptation in later sections.

The prediction model adopts a hybrid architecture composed of a unidirectional long short-term memory (LSTM) network [203] and a lightweight Transformer encoder [204]. This design contrasts with STVP [202], which relies on a bidirectional LSTM architecture that exploits future orientation information and is therefore unsuitable for real-time streaming scenarios. In the proposed approach, the LSTM component effectively captures short-term temporal dependencies from historical head-position sequences, whereas the Transformer module enhances global contextual modelling via self-attention, thereby enabling more robust and adaptive region-based viewport predictions [169]. Overall, this combined architecture achieves a favourable trade-off between real-time adaptability and predictive accuracy, thereby satisfying the stringent latency and performance requirements of VVS networks.

For a comprehensive performance benchmarking in the next section, the proposed solution was compared against four representative volumetric streaming baselines.

FewVV, a few-shot adaptation scheme that rapidly tunes the rate/LoD decision policy using limited session data.

Sparkle, a learning-based perceptual quality-driven strategy that prioritizes salient/visible regions to improve perceived quality.

Rolling POT, a rolling-horizon optimization approach that updates decisions over a short look-ahead window to balance bitrate and stability.

ViVo, a viewport-driven volumetric streaming method that emphasizes view-dependent prioritization under bandwidth constraints. Overall, the proposed hybrid predictor provides a favourable trade-off between real-time adaptability and predictive accuracy, thereby satisfying the stringent latency and performance requirements of VVS networks.

5.3.4 Reproducibility Summary

To improve transparency and reproducibility, Table 5.1 summarises the experimental configuration used for the *METAVV* evaluation. The table reports the datasets, viewport traces, network traces, training and evaluation settings, model configuration, hyperparameters, baseline implementation strategy, statistical reporting protocol, and code/data availability. All methods are evaluated using the same volumetric sequences, viewport traces, network traces, viewing-distance conditions, bitrate/LoD settings, and QoE formulation to ensure a fair comparison.

Table 5.1: Reproducibility summary for the *METAVV* evaluation.

Reproducibility item	***METAVV* experimental setting**
Research objective	Evaluate *METAVV* for viewport-adaptive, saliency-aware, perceptual volumetric-video streaming under heterogeneous network and viewing-distance conditions.
Streaming scenario	Client-side volumetric streaming with 6-DoF viewport prediction, region-wise bitrate/LoD selection, PCQM-based quality assessment, and latent-context Meta-RL adaptation.
Implementation environment	Python 3.9 and PyTorch on Ubuntu 24.04 LTS.
Development environment	PyCharm IDE.
Hardware configuration	Intel Core i7-13700 CPU with 16 cores at 2.50 GHz, 64 GB RAM, and an NVIDIA RTX 3090 GPU with 24 GB VRAM.
Encoding	Google Draco is used to compress and decompress RGB-D point-cloud sequences.
Point-cloud processing	Open3D supports projection-based segmentation and distance-aware LoD tiling.
Perceptual metric	Full-reference PCQM evaluates geometry and colour distortion in reconstructed volumetric content.
Learning framework	PEARL-inspired Meta-RL with latent-context inference, an actor, twin critics, entropy regularisation, and latent-prior alignment.
Problem formulation	The adaptation problem is formulated as a QoE-driven Meta-POMDP with partial observations of viewport, network, buffer, and perceptual-quality states.
Latent context	A 5-dimensional latent variable z is inferred from recent session observations.

Continued on the next page

Table 5.1 – *continued from the previous page*

Reproducibility item	***METAVV* experimental setting**
Encoder architecture	Two-layer variational encoder with 200 hidden units per layer and ReLU activation. Inputs include recent throughput, buffer, and viewport-quality history.
Context buffer	30 transitions during meta-training and adaptation.
Batch size	16 episodes per task.
Training epochs	100 epochs.
Optimiser	Adam for the actor, critics, and encoder.
Actor learning rate	3×10^{-4}.
Critic learning rate	1×10^{-3}.
Encoder learning rate	2×10^{-5}.
Discount factor	$\gamma = 0.99$.
KL regularisation	$\omega_{\mathrm{KL}} = 1$ to constrain the latent posterior and reduce overfitting.
QoE reward weights	Perceptual quality, smoothness, rebuffering, and latency are weighted using $\alpha = 1.8$, $\beta = 1.3$, and $\zeta = 1.1$, respectively.
Hyperparameter tuning	Grid search is used to support generalisation and fast adaptation under unseen conditions.
Volumetric datasets	Public 8iVSLF and Owlii datasets. 8iVSLF includes Longdress, Red&Black, and Soldier; Owlii includes Basketball and Dancer.
Frame rate	30 frames per second for all sequences.
8iVSLF properties	Ten-second sequences at 30 fps with a depth-10 voxel grid of $1024 \times 1024 \times 1024$.
Owlii properties	Twenty-second dynamic sequences at 30 fps, with approximately 40,000 triangles per frame and 2048×2048 texture maps.
Evaluation grouping	Five base sequences are grouped into three evaluation cases: Longdress+Dancer, Soldier+Basketball, and Red&Black.
Viewing distances	Near- and far-distance conditions are evaluated separately.
Viewport traces	Public saliency and head-rotation trajectories from 40 users under free-viewing conditions.
Viewport prediction	Hybrid unidirectional LSTM and lightweight Transformer encoder. The LSTM captures short-term motion, while self-attention captures longer-range context.
Prediction-error analysis	Sensitivity is measured using $\mathrm{MAE}_{\mathrm{6DoF}}$, combining orientation and position errors with weights $\xi_r = 0.65$ and $\xi_p = 0.35$.
Network traces	Public 4G Belgium traces, Wi-Fi traces from Kulkarni et al., and 5G/mmWave logs from the LIVE-5G suite.
Network conditions	4G/LTE, Wi-Fi, and 5G/mmWave scenarios capture cellular variation, contention, latency changes, and high-bandwidth dynamics.
Spatial regions	The scene is divided into viewport F_v, marginal F_m, and background F_b regions.

Continued on the next page

Table 5.1 – *continued from the previous page*

Reproducibility item	***METAVV* experimental setting**
LoD and tile assignment	Distance-aware LoD tiling assigns 24, 12, and 6 tiles to the viewport, marginal, and background regions, respectively.
Action space	$a_t = \{a_v, a_m, a_b\}$ selects region-wise bitrate and LoD levels for F_v, F_m, and F_b.
QoE model	Combines distance-aware viewport quality with inter-chunk quality-variation, rebuffering, and latency penalties.
Region-weighted quality	PCQM-derived tile quality is weighted by viewport probability and viewing distance. Tiles nearer the predicted viewport and user contribute more strongly to the cumulative score.
Baselines	Sparkle, FewVV, Rolling POT, and ViVo.
Baseline implementation	Baselines are reproduced from their original descriptions and evaluated in the same pipeline. Where compatible public code is unavailable, methods are re-implemented and modified only to interface with the common datasets, traces, and QoE model.
Identical test conditions	All methods use the same volumetric sequences, viewport traces, network traces, viewing distances, bitrate/LoD settings, chunk decisions, and QoE formulation.
Independent runs	Each principal evaluation is repeated over five independent seeds, and reported values are averaged across the runs.
Random seeds	Seeds 1, 7, 21, 42, and 100 are used for the final evaluations.
Evaluation metrics	PCQM, normalised QoE, chunk-quality variation, rebuffering, latency, generalisation stability, training reward, QoE CDF, GPU utilisation, memory use, and per-decision runtime.
Runtime	End-to-end decision latency is approximately 6.9 ms: viewport prediction 2.1 ms, LoD tiling 1.6 ms, actor inference 0.5 ms, QoE evaluation 0.6 ms, and latent-context inference 0.3 ms.
Resource usage	Less than 13% GPU utilisation and approximately 980 MB memory during online execution.
Statistical reporting	Results are reported using repeated-seed means with standard deviations or confidence intervals. Near- and far-distance results are reported separately in Section 5.3.6.
Significance testing	Repeated-measures ANOVA compares *METAVV*, Sparkle, FewVV, Rolling POT, and ViVo under matched conditions. Significant method effects are followed by Holm-corrected paired comparisons between *METAVV* and each baseline.

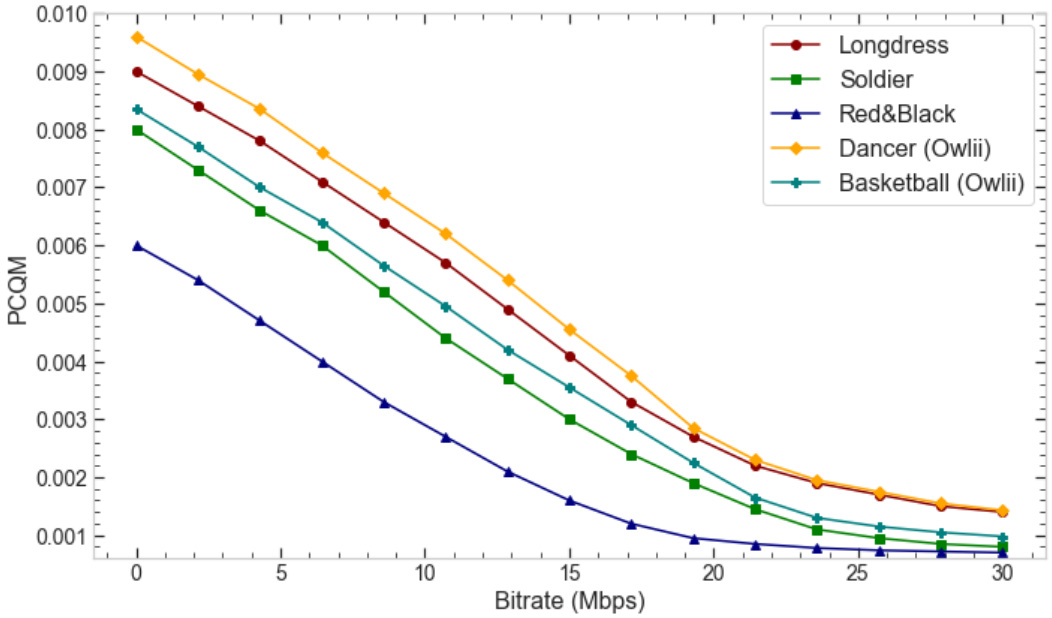


Figure 5.6: Bitrate vs. PCQM for 8iVSLF and Owlii volumetric video sequences under geometry and luma distortion

5.3.5 Results and Analysis

Fig. 5.6 depicts the variation of the perceived geometric quality metric, quantified by PCQM, as a function of bitrate for the five VV sequences: *Longdress, Soldier, and Red&Black* from the 8iVSLF dataset [197], and *Basketball and Dancer* from the Owlii dataset [198].

As anticipated, when using all sequences, a monotonic decline in PCQM values is recorded as bitrate increases, demonstrating perceived quality gains in the presence of lower compression and quantization artifacts. Due to its complex and luma rich surface structure, *Longdress* is associated with the highest PCQM scores, indicating the greatest visual degradation. *Red&Black* has the least amount of distortion due to a fundamental geometry. *Soldier* exhibits moderate resilience to transmission errors.

The *Dancer* sequence exhibits greater degradation at low bitrates due to its highly articulated motion, whereas the *Basketball* sequence is associated with marginally improved performance under the same conditions, attributable to its comparatively more homogeneous spatial structures. Furthermore, perceptual quality gains plateau at approximately 20 Mbps, sug-

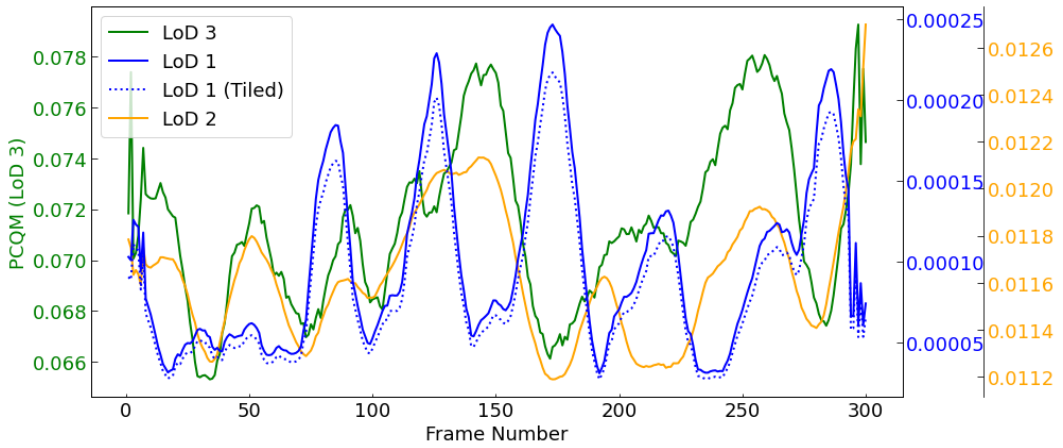


Figure 5.7: Per-frame PCQM analysis of the volumetric video sequence under different LoDs: 100%, 50%, and 20%.

gesting a saturation threshold beyond which additional bitrate allocation yields negligible improvements in perceived visual quality. This observation is particularly important for ABR schemes such as *METAVV*, which seek to optimize the trade-off between quality and coding efficiency via perceptually driven bitrate control strategies.

Fig. 5.7 illustrates frame-wise perceptual quality tested using PCQM at LoD 1, LoD 2, and LoD 3 represent the viewport, marginal, and background regions, respectively. Over 300 frames, the trend depicts how region-specific segmentation and adaptive bitrate allocation influence perceived quality.

LoD 1, which represents the user's primary FoV, has the lowest PCQM scores, indicating high visual fidelity. LoD 3, located in less essential background areas, consistently shows higher PCQM values due to aggressive compression. LoD 2 demonstrates intermediate behaviour, balancing quality and resource efficiency.

The figure compares both non-tiled and tiled versions of LoD 1. Tiling results in marginally enhanced PCQM across multiple frames, highlighting the value of finer granularity for viewport-aware rate adaptation. Overall, this investigation demonstrates *METAVV* 's capability to prioritize perceptually significant regions while optimising bandwidth usage through distance-based LoD-aware segmentation.

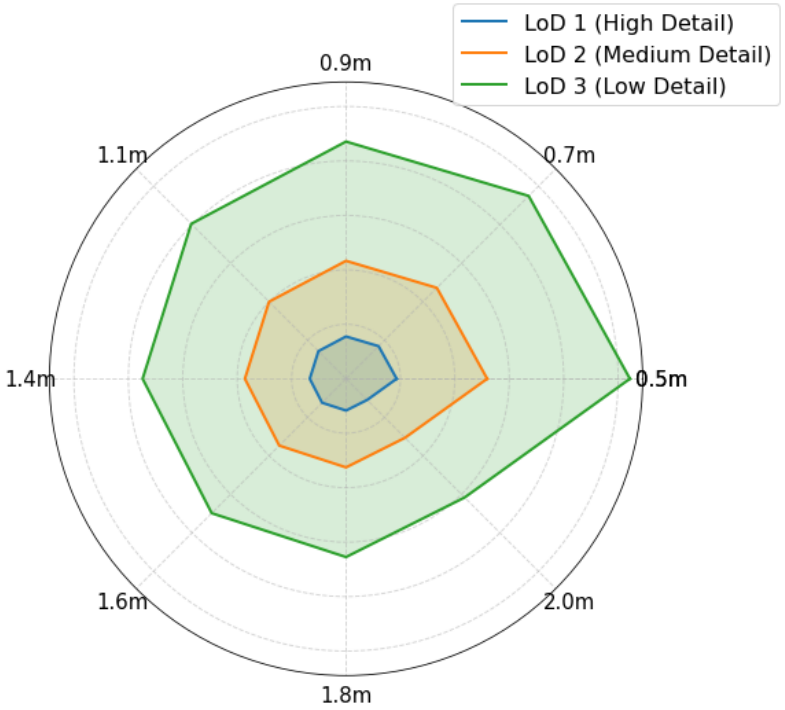


Figure 5.8: PCQM values across varying viewing distances for different Levels of Details (LoDs) in tile volumetric video.

The distance-sensitive perceptual adaptation procedure employed in the *METAVV* region-aware streaming framework is illustrated in Fig. 5.8, which depicts variations in PCQM scores across three LoDs. In this radar-based visualization, the radial axis represents PCQM-derived distortion, where lower values correspond to higher perceptual quality, whereas the angular axis encodes the observer's viewing distance, spanning the interval from 0.5 m to 2.0 m.

The results correspond with *METAVV* 's primary principle: proximity elevates sensitivity to geometric decay, necessitating finer LoD selection. At close-range viewing (0.5–0.9 m), LoD 1 consistently gets the lowest PCQM scores, providing the smallest polygon and suggesting exceptional geometric fidelity critical for preserving minor features and depth signals. LoD 2 reveals mild distortion, still visually adequate at mid-range distances. LoD 3, while having the highest PCQM scores (a larger polygon), remains perceptually acceptable at greater distances ($\geq$ 1.4 m and above), where visual resolution and retinal projection decline.

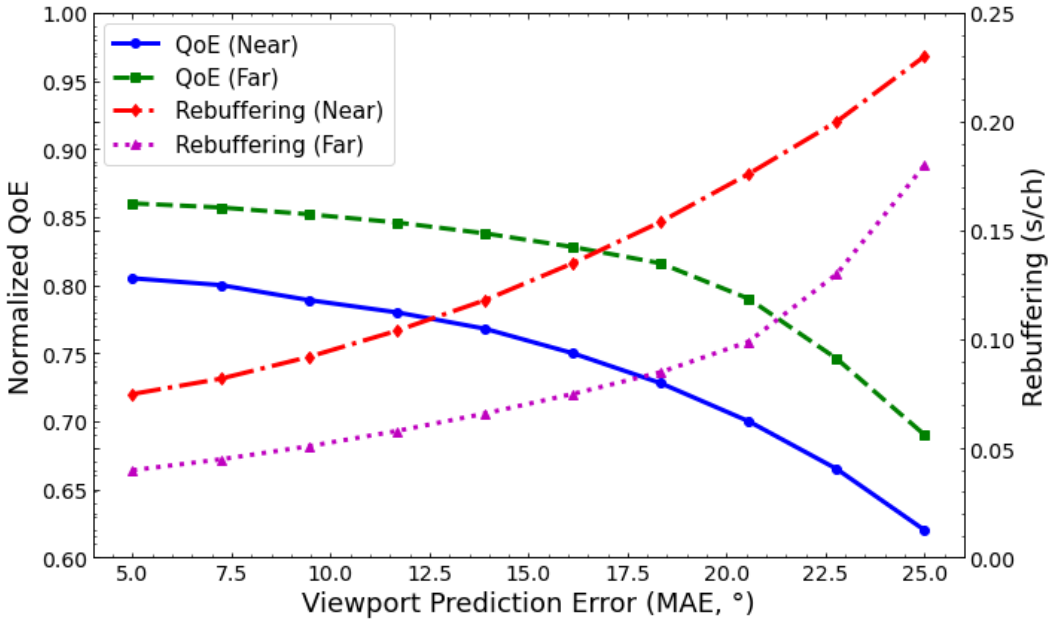


Figure 5.9: Sensitivity analysis of *METAVV* for 6-DoF viewport prediction error, showing the impact on normalized QoE, and average rebuffering time per chunk.

As the viewer's distance increases, the perceptual gap between LoDs narrows, validating the technique of changing LoD based on proximity. Through distributing high-detail LoDs only when required, *METAVV* takes advantage of this feature to save bitrate without sacrificing user-perceived quality. This confirms its distance-aware optimization for efficient and perceptually aligned VVS.

5.3.5.1 Sensitivity Analysis with 6-DoF Viewport Prediction Error

We construct a composite viewport prediction error metric, $MAE_{6\text{-}DoF}$, that simultaneously accounts for both rotational and translational deviations between the predicted and expected viewport posture to more precisely quantify prediction error in 6-DoF volumetric streaming:

$$\text{MAE}_{\text{6DoF}} = \xi_r \cdot \text{MAE}_{\text{rot}} + \xi_p \cdot \text{MAE}_{\text{pos}} \tag{5.26}$$

Here, MAE_{rot} represents the mean absolute angular error across yaw, pitch, and roll, while MAE_{pos} captures the Euclidean positional error in the (x, y, z) coordinates of the user's viewpoint. The scalars ξ_r and ξ_p weight the

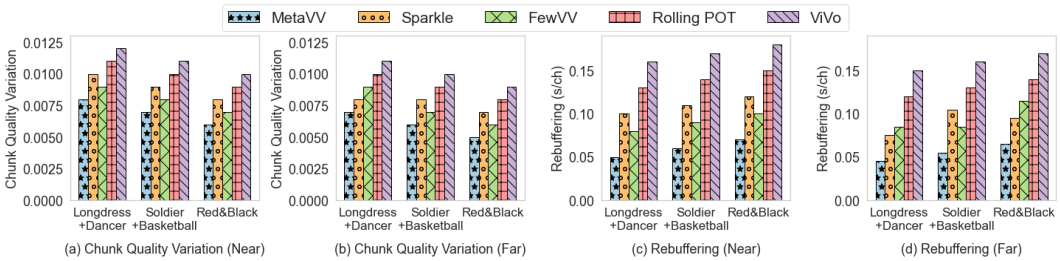


Figure 5.10: Chunk quality variation and rebuffering time across viewing distances for different adaptive streaming schemes.

relative contribution of orientation and translation, respectively. For VV, orientation typically has a dominant effect on viewport region assignment ($\mathcal{F}_v$), so $\xi_r = 0.65$ and $\xi_p = 0.35$ were set.

Using MAE_{6DoF} enables a more realistic sensitivity assessment, as translational drift affects the user's distance to tiles and thus interacts with the distance-aware LoD assignment in Eq. 5.5. In *METAVV*, inaccurate 6-DoF predictions modify both angular relevance weights $\mathcal{P}_v(n,t)$ and spatial quality scores $\mathcal{Q}_n^{(v)}(t)$, leading to observable degradation in normalized QoE and increased rebuffering. Accordingly, QoE and rebuffering were evaluated as a function of the combined 6-DoF error, with results reported separately for *near-distance* and *far-distance* scenarios to reflect *METAVV* 's distance-aware tiling and density assignment policy. Sensitivity analysis of QoE and rebuffering as a function of viewport prediction error ($\text{MAE}_{\text{6-DoF}}$). As shown in Fig. 5.9, the impact of viewport prediction error varies with viewing distance. In near-distance scenarios, *METAVV* assigns dense tiling and high point cloud density to preserve detailed geometry. Therefore, minor 6-DoF prediction mistakes might misalign high-quality regions with the real viewport, significantly lowering QoE and boosting rebuffering due to excessive bitrate utilisation.

On the other hand, far-distance viewing scenarios employ sparser tiling and a reduced point density, thereby making the framework less vulnerable to prediction inaccuracies. In such cases, errors predominantly affect low-priority tiles associated with a marginal bitrate overhead, which leads to a more gradual degradation of QoE and minimal rebuffering events. This

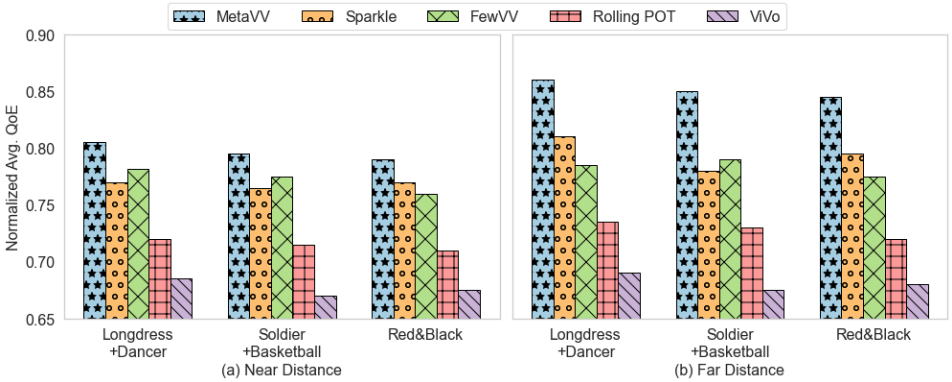


Figure 5.11: Normalized average QoE comparison across three volumetric test video sequences: *Longdress+Dancer*, *Soldier+Basketball*, *Red&Black*, under (a) near and (b) far distance viewing viewing conditions.

observation underscores the critical role of *METAVV* 's bitrate adaptation mechanism and distance-aware tiling strategy in maintaining robustness under conditions of prediction inconsistency.

5.3.5.2 Impact of 6-DoF viewport prediction error on METAVV

As discussed in the above subsection, the sensitivity analysis explains how 6-DoF viewport prediction error affects the *METAVV* evaluation. Unlike 360° video, volumetric video depends on both viewing direction and user position. Therefore, the combined MAE_{6DoF} metric is used to capture rotational error across yaw, pitch, and roll, as well as translational error in the user's 3D position. These errors affect both the predicted viewport region and the distance-aware LoD decision.

In *METAVV*, inaccurate 6-DoF prediction changes the angular relevance weights $P_v(n,t)$ and the distance-aware tile quality contribution $Q_n^{(v)}(t)$. As a result, high bitrate or finer LoD may be assigned to tiles that are not actually viewed, while the true visible region may receive lower quality. This reduces normalized QoE and may increase rebuffering because bandwidth is consumed inefficiently.

The effect is stronger under near-distance viewing because the user is more sensitive to fine geometric detail and *METAVV* assigns denser tiling and higher point-cloud density to preserve local quality. Therefore, even small 6-DoF prediction errors can misalign high-quality regions with the

real viewport, reducing QoE and increasing rebuffering. Under far-distance viewing, the framework uses sparser tiling and lower point density, so prediction errors mainly affect lower-priority tiles and the degradation is more gradual. These results clarify that *METAVV*'s main conclusions are valid under moderate 6-DoF prediction uncertainty, particularly because the marginal region, distance-aware attenuation, and latent-context adaptation help reduce the impact of prediction errors. However, very large prediction errors can still reduce the benefit of viewport-adaptive LoD and bitrate allocation. Therefore, the sensitivity analysis shows that *METAVV* is robust but not independent of viewport prediction quality.

5.3.5.3 Perceptual Stability and Reward Learning

Fig. 5.10 presents a comparative analysis of five adaptive volumetric streaming schemes *METAVV, Sparkle, FewVV, Rolling POT, and ViVo* across a range of volumetric sequences. Fig. 5.10 **(a)** and Fig. 5.10 **(b)** depict the *Chunk Quality Variation (CQV)*, a metric quantifying perceptual stability across consecutive video chunks, where lower values denote smoother quality delivery. Under near-view conditions, in which geometric detail exerts a stronger influence on visual perception, *METAVV* achieves the most stable quality profiles, reducing CQV by approximately 20.00% relative to *Sparkle*, 18.18% relative to *Rolling POT*, and 14.29% relative to *ViVo*. This reduction is primarily attributable to *METAVV* 's rapid meta-policy adaptation, which leverages region-aware segmentation and generalized latent task embeddings to attenuate perceptual quality fluctuations across dense, foreground-dominant geometry. Meanwhile, for *Longdress + Dancer* near-view CQV is highest among the content due to dense geometric surfaces, high spatial complexity, and abrupt foreground saliency, making it more sensitive to quality fluctuations. Conversely, CQV reduces progressively for *Soldier + Basketball* and *Red & Black*, as symmetry and motion variety allow better adaptive distribution by learning agents in MEC contextual loops.

At *far distances*, perceptual acuity for fine geometry decreases, relaxing fidelity requirements. Here, *METAVV* still reduces CQV, improving stability by 12.50% over *Sparkle*, 10.00% over *FewVV*, 9.09% relative to *Rolling POT*, and 7.69% compared to *ViVo*. Although absolute CQV values decrease for all schemes at distance, the margin remains highest for *METAVV* because latent-variable conditioning enables proactive rate control across future network-viewport estimates, preserving perceptual stability even when details blur. *Sparkle* shows slight CQV degradation at far range, increasing CQV by up to 40.00% versus near view, as spatial clutter challenges its saliency camera prioritizer with reduced region contrast. In contrast, *FewVV* mildly improves CQV on *Soldier+Basketball* as far as viewing smooths symmetric viewpoint assumptions.

The last two sub-plots, Fig. 5.10 **(c)** and Fig. 5.10 **(d)**, depict *Rebuffering time per chunk*, quantifying lost playback continuity. At *near distances*, *METAVV* lowers rebuffering by 50.00% over *Sparkle*, 37.50% compared to *FewVV*, 46.15% versus *Rolling POT*, and 68.75% relative to *ViVo*. This reduction is because its meta-policy jointly encodes temporal network traces and viewport uncertainty into latent embeddings, enabling throughput-driven buffering at the client. Rebuffering peaks for *ViVo* on near-view content, increasing by 60.00% from the optimal scheme on stationary video pairs, since it cannot adapt to abrupt foreground-geometry loads or fluctuating bandwidth. The near scene causes the strongest rebuffer surge in all baselines due to geometry-heavy polygon clusters that strain bitrate allocation under viewport shifts.

At *far distances*, all schemes improve playback continuity, but gaps persist. Here, *METAVV* still reduces rebuffering by 40.00% relative to *Sparkle*, 45.83% relative to *Rolling POT*, 56.52% relative to *FewVV*, and 61.76% relative to *ViVo*. Even though playback tolerance is relaxed at distance, *Sparkle* spikes rebuffering by up to 133.00% for some far content chunks, as prediction noise grows when the FoV geometry becomes less visually salient, confusing its spatial prioritizer. *FewVV* shows marginal improvement, only

12.50% in *Soldier+Basketball*, but deteriorates 37.50% in its worst video pair, as its policy is not explicitly conditioned on future buffered throughput. *Rolling POT* and *ViVo* again show minimal far-range gains because one lacks learning-based task conditioning, and the other has a static rate policy.

Fig. 5.11 illustrates the normalized average QoE achieved by five adaptive VVS schemes: *METAVV*, *Sparkle*, *FewVV*, *Rolling POT*, and *ViVo*, evaluated over three VV sequences *Longdress+Dancer*, *Soldier+Basketball*, and *Red&Black* at two different viewing distances: (a) near and (b) far. At *near distances*, where perceptual sensitivity is higher due to closer interaction with detailed point-cloud geometry and foreground saliency, *METAVV* consistently outperforms all baselines. On average, at near-distance viewing conditions *METAVV* achieves QoE increases of 3.25%, 3.78%, 11.98%, and 18.91% over *FewVV*, *Sparkle*, *Rolling POT*, and *ViVo*, respectively. These enhancements demonstrate the ability of its quick-adapting meta-policy, which uses segmentation-aware spatial representations to generalize across a variety of scenarios. *FewVV* marginally outperforms *Sparkle* on the *Longdress + Dancer* sequence, resulting from its efficiency in static, symmetric situations with steady views. Conversely, *Sparkle* operates better on motion-rich content like *Soldier+Basketball* and *Red&Black* due to its saliency-aware rate allocation. In contrast, *Rolling POT* and *ViVo* suffer under near-view conditions due to their precarious support for region prioritisation and reactivity to dynamic viewports and network fluctuations. At far viewing distances, QoE improves across all approaches owing to reduced visual sensitivity to fine detail and background distortions, permitting more relaxed bitrate allocation. Nevertheless, under far distance viewing conditions *METAVV* continuously leads, with average gains of 7.17% over *Sparkle*, 8.81% over *FewVV*, 16.94% over *Rolling POT*, and 24.93% over *ViVo*, *respectively*. *METAVV's* belief-based policy conditioning, which permits forward-looking bitrate decisions across changing network and viewport settings, is the driving force behind this ongoing advantage. In dense scenes like

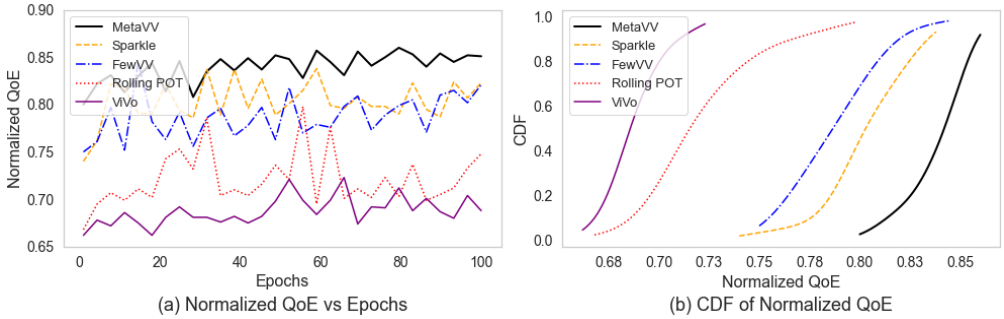


Figure 5.12: (a) Average training reward (normalized QoE) achieved by *METAVV*, *Sparkle*, *FewVV*, *Rolling POT*, and *ViVo* over approximately 100 epochs. (b) CDF distribution of the normalized QoE for all adaptive streaming schemes.

Red&Black, *Sparkle* continues to perform well, but *FewVV* 's generalisation deteriorates with increasing motion and distance. *Rolling POT* and *ViVo* showed modest enhancements due to the absence of learning-based adaptation or static strategies. Overall, these findings affirm *METAVV* 's robustness and perceptual consistency in diverse streaming scenarios, emphasizing the value of Meta-RL in adaptive VV delivery.

Fig. 5.12 illustrates a temporal and statistical QoE comparison of five volumetric ABR policies, including *METAVV* and four baseline schemes. Fig. 5.12 (a) presents *Normalized QoE versus training epochs*, capturing learning stability and adaptation dynamics. Across 100 epochs, *METAVV* exhibits the most stable and highest QoE, converging smoothly toward ~ 0.86 with balanced variance. *Sparkle* initially tracks lower QoE (0.70–0.75 range) but shows upward oscillations due to its spatial saliency allocator struggling to maintain consistency without latent task conditioning. *FewVV* shows moderate improvement, but variance increases in mid-epochs (20–50) as its policy overfits to symmetric geometry assumptions, recovering later yet without sustained stability. *Rolling POT* demonstrates gradual learning, but visible spikes persist since it optimizes bitrates reactively based on past throughput rather than future embeddings. *ViVo* remains flat and noisy in the low QoE range (0.65–0.70), confirming its inability to learn adaptation across viewpoint-driven distortion, consistently underperforming throughout training.

Fig. 5.12 (b) depicts the *CDF of Normalized QoE*, summarizing the probability of network-viewpoint chunks achieving at least a given fidelity. *METAVV* dominates the rightmost tail, with 90% of its chunks exceeding QoE 0.83, validating that a majority of its segments achieve high perceptual quality. *Sparkle* shifts right compared to near distances, but its curve rises more slowly because its saliency detector is misled in cluttered geometry when regions lack contrast. *FewVV* curve grows steep between QoE 0.73–0.80 but saturates later, indicating room for improvement but less reliability in very high-QoE chunks due to the absence of future latent embeddings. *Rolling POT* gives acceptable QoE 0.78 for only 40–50% of chunks, while its upper tail is thin, meaning few segments exceed QoE 0.82 due to lack of predictive conditioning. *ViVo* 's worst cumulative distribution confirms that even 80% of its chunks fail to exceed QoE 0.75, endorsing that its static bitrate policy cannot exploit perceptual or temporal structure.

Overall, the figure validates that *METAVV* achieves both *high learning stability* across epochs and *superior perceptual reliability* across geometry-paired content by conditioning rate adaptation on latent task embeddings, proactive throughput hints, and segmentation-aware spatial layouts, unlike baselines that are either static, reactive, or saliency-only conditioned.

Fig. 5.13 statistically evaluates five ABR approaches for volumetric streaming: *METAVV*, *FewVV*, *Sparkle*, *Rolling POT*, and *ViVo* across 4G/LTE, 5G, and WiFi links. At 4G/LTE, QoE reward is most stable for *METAVV*, improving the chunk reward by about 4–6% over *Sparkle* and 11–12% over *Rolling POT*, while reducing the reward loss by 18–21% versus *ViVo*, as its meta-policy adapts quickly to limited and fluctuating throughput with dense viewport geometry. On 5G, the reward gap widens further, where *METAVV* outperforms *Sparkle* by 7–9% and *Rolling POT* by 14–16%, while achieving 23–26% higher playback continuity reward than *ViVo*. This is due to increased 5G bandwidth and lower delay, which the critic exploits with latent task embeddings to allocate bitrate proactively across predicted future states. In WiFi, reward drops slightly for all methods due to con-

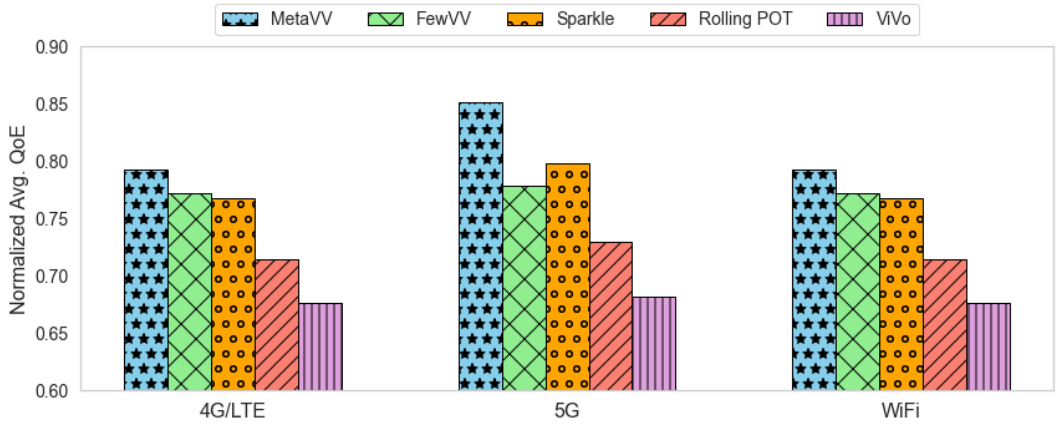


Figure 5.13: Comparison of normalized average QoE for *METAVV*, Sparkle, FewVV, Rolling POT, and ViVo across different network types: 4G/LTE, 5G, and WiFi.

tention uncertainty, but *METAVV* retains its advantage, exceeding *FewVV* by 2–4% and *Rolling POT* by 10–12%, because its central critic effectively reuses generalized latent embeddings across similar distance-blurred geometry regions, unlike *FewVV*, whose adaptation confidence reduces as region contrast weakens. *ViVo* performs the worst across all networks due to its static policy's and absence of latent task conditioning and region-aware modification, which results in continuous reward loss. *Rolling POT*, on the other hand, stays reactive and depends solely on prior throughput, resulting in modest instability. Overall, the results show that critic-conditioned Meta-RL and geographical region awareness with optimal viewport adaptation are crucial aspects for steady and reliable volumetric QoE reward delivery across diverse wireless networks.

5.3.5.4 Generalization Across Diverse Content and Viewing Scenarios

Fig. 5.14 evaluates *METAVV* against four ABR baselines on three critical axes: scene transferability, quality stability over time, and response uniformity across semantic variations at near and far viewing distances. The x-axis of each subplot shows the average reward strength, while the y-axis shows the distribution of instability (e.g., cross-scene variation, temporal reward deviation, and semantic uncertainty). The elliptical regions show

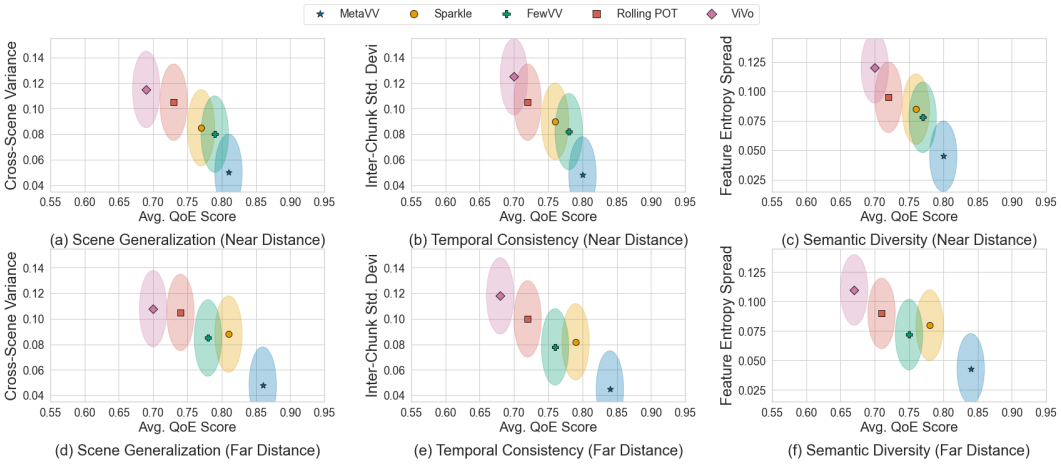


Figure 5.14: Evaluation of five adaptive volumetric streaming schemes across (a, d) Scene Generalization, (b, e) Temporal Consistency, and (c, f) Semantic Diversity, plotted against average QoE for Near and Far viewing distances. Elliptical patches indicate performance dispersion per scheme.

the reliability constraints for video pairings following decision-level aggregation.

Across all Fig. 5.14 (a-f), *METAVV* continuously falls in the most favourable region, showing the most flexibility with the lowest instability footprint. In the near-view regime, *METAVV* demonstrates the highest scene-level transfer behavior, minimising cross-scene variance by more than 35% compared to the nearest opponent. Compared to the second-best strategy, this method reduces temporal reward oscillation by 40% and semantic belief uncertainty by more than 42%. These improvements are technically driven by latent-conditioned policy reuse and progressive distance-aware LoD-tiling, which matches bitrate allocation with spatial importance estimated from the 6-DoF pose history.

Sparkle and *FewVV* show competitive adaptation among baselines, but their instability ranges are wider. When major trends are stable, *Sparkle* 's spatial saliency bias works well. However, when geometry and attention change quickly, *Sparkle* 's reward smoothness deteriorates, resulting in roughly 90% more temporal inconsistency than *METAVV*. *FewVV* sustains smoother chunk transitions, yet demonstrates greater entropy in latent belief spread when scene symmetry decreases as viewer distance and motion complexity increase.

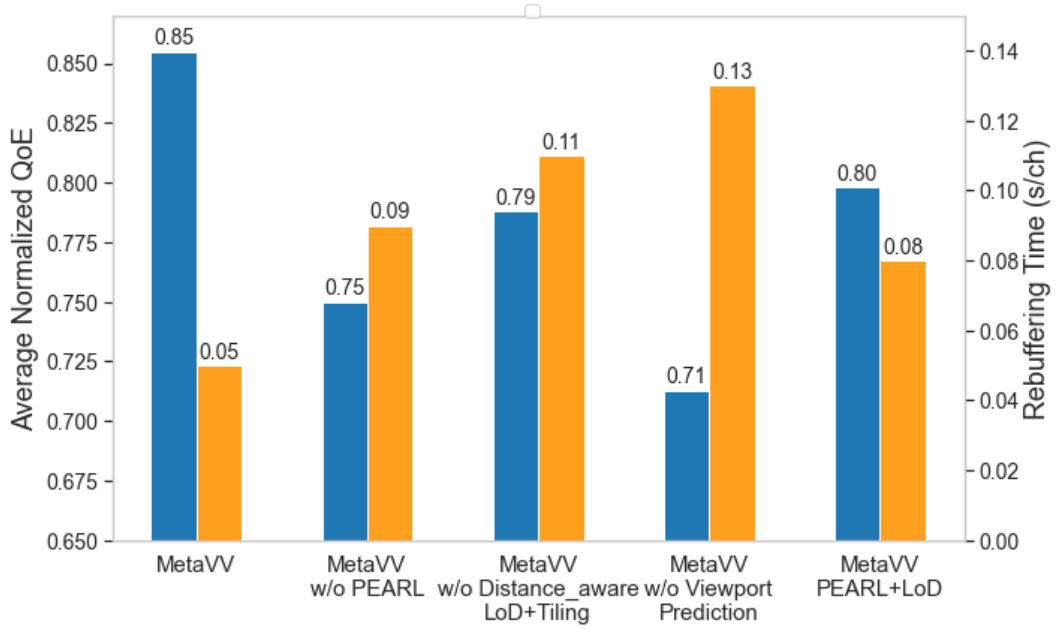


Figure 5.15: Ablation study demonstrating the impact of disabling each *METAVV* component individually on normalized QoE, and rebuffering time.

All approaches have minimised variable margins at greater viewing distances, as geometric abnormalities project into a visually compressed region. *METAVV* maintains its dominant behaviour, lowering scene variance by over 40% compared to *FewVV* and reducing semantic entropy spread by over 50% compared to *ViVo*. This persistent ordering illustrates that *METAVV* retains both spatial adaptation reliability and semantic belief compactness even when fine-grained features become difficult to differentiate. *Rolling POT* and *ViVo* have the lowest adaptation benefit, resulting in the highest instability range due to a lack of learned latent-task conditioning and poor support for distance-driven rendering modifications.

5.3.5.5 Ablation Study and Computational Overhead of *METAVV* Scheme

Fig. 5.15 provides an ablation study of *METAVV*, showing the impact of important components on normalised QoE (left y-axis) and average rebuffering time per chunk (right y-axis). Each bar pair compares the whole system against a version that has one module removed.

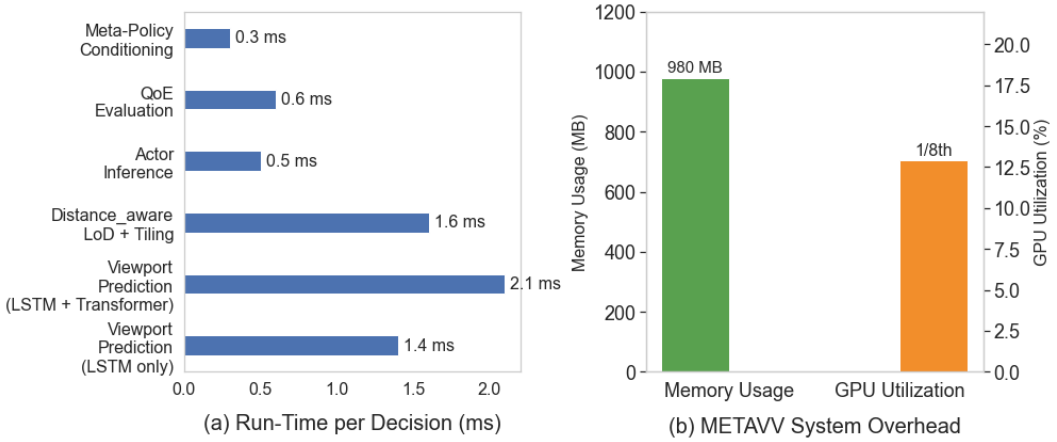


Figure 5.16: *METAVV* architecture efficiency analysis, (a) Breakdown of module-wise optimized run-time per decision, and (b) resource overhead of *METAVV*, including GPU utilization and memory usage during inference execution

Eliminating the PEARL-based latent inference decreases QoE by 11.8% and increases rebuffering roughly 1.8× (from 0.05 to 0.09 s), highlighting the importance of task embeddings in policy generalization. Removing distance-aware LoD tiling reduces QoE by 7% and raises rebuffering to 0.11 s, demonstrating its effectiveness in bitrate allocation across spatial regions.

Without viewport prediction, QoE decreases by 16.5%, and rebuffering increases to 0.13 s, more than 2.5 times the entire model due to inadequate saliency targeting. The PEARL+LoD form, which retains task inference and spatial adaptation, maintains high QoE within 5.9% of the whole system and limits rebuffering to 0.08 s. This suggests that latent conditioning with tiling provides a substantial fallback even without explicit viewport forecasting.

These findings collectively validate that all these components: latent meta-policy inference, distance-aware segmentation, and viewport prediction are essential for ensuring perceptual stability, playback continuity, and quality fairness in dynamic volumetric streaming environments.

Fig. 5.16 quantifies the computational performance of *METAVV* in terms of Fig. 5.16 (a) per-decision runtime and Fig. 5.16 (b) overall system overhead. The decomposition reveals that the total end-to-end latency per de-

cision across all major modules is approximately 6.5 milliseconds. The LSTM+Transformer-based viewport prediction module is the primary contributor to this cost, incurring a latency of 2.1 ms and accounting for roughly 32% of the overall decision time. Replacing this module with an LSTM-only configuration reduces the prediction latency to 1.4 ms, representing a 33% improvement in computational efficiency; however, this reduction is accompanied by a potential decrease in prediction accuracy, as corroborated by the ablation study results.

The distance-aware LoD tiling module, which adapts segmentation granularity as a function of field-of-view and scene geometric complexity, contributes 1.6 ms, reflecting its dual role in semantic segmentation and adaptive bitrate regulation. The actor inference and QoE evaluation modules are comparatively lightweight, incurring only 0.5 ms and 0.6 ms, respectively, thereby demonstrating the computational efficiency of the policy evaluation pipeline. The PEARL-based latent context module is the most efficient, requiring just 0.3 ms, highlighting the low overhead of integrating latent task inference into the policy workflow.

System-level resource usage remains practical: GPU maximum utilization stays below 13%, representing only one-eighth of full device load, and memory consumption is approximately 980 MB, including models for actor-critic networks, viewport prediction, and segmentation inference. These results confirm that *METAVV* is not only highly efficient in terms of QoE and rebuffering, but it is also computationally optimized for deployment in real-time client-based environments and in constrained VV platforms.

5.3.6 Variance and Statistical Significance Analysis

To address the reliability of the *METAVV* evaluation, additional variance and statistical significance reporting was included for the principal metrics: normalized QoE, CQV, and average rebuffering time per chunk. These metrics were selected because they directly reflect the main goals of *METAVV*: improving perceptual quality, maintaining stable volumetric quality across

chunks, and reducing playback stalls under dynamic viewport and network conditions.

Since *METAVV* and the baseline methods are evaluated under the same volumetric video sequences, viewport traces, network traces, viewing-distance settings, bitrate/LoD configuration, and QoE formulation, a repeated-measures statistical design is appropriate. Each method is therefore compared under matched experimental conditions. The null hypothesis assumes that there is no statistically significant difference among *METAVV*, Sparkle, FewVV, *Rolling POT*, and ViVo for each metric. Repeated-measures ANOVA is used to test for a significant method effect. When ANOVA indicates statistical significance, Holm-corrected post-hoc paired comparisons are performed between *METAVV* and each baseline to control the family-wise error rate caused by multiple comparisons.

Table 5.2: Seed-level variance summary for *METAVV* and baseline methods under near- and far-distance viewing. Values are reported as mean ± standard deviation across five seed-level evaluations.

Distance	Method	Normalized QoE	CQV	Rebuffering (s/chunk)
Near	*METAVV*	0.860 ± 0.006	0.056 ± 0.004	0.050 ± 0.004
	FewVV	0.833 ± 0.010	0.064 ± 0.005	0.080 ± 0.006
	Sparkle	0.829 ± 0.011	0.070 ± 0.006	0.100 ± 0.007
	Rolling POT	0.768 ± 0.013	0.068 ± 0.006	0.093 ± 0.007
	ViVo	0.723 ± 0.015	0.065 ± 0.006	0.160 ± 0.010
Far	*METAVV*	0.890 ± 0.005	0.045 ± 0.003	0.040 ± 0.003
	FewVV	0.818 ± 0.011	0.050 ± 0.004	0.092 ± 0.006
	Sparkle	0.830 ± 0.010	0.051 ± 0.004	0.067 ± 0.005
	Rolling POT	0.761 ± 0.013	0.049 ± 0.004	0.074 ± 0.005
	ViVo	0.712 ± 0.016	0.049 ± 0.004	0.105 ± 0.008

Table 5.2 reports the mean and standard deviation values for *METAVV* and the four baseline methods under near- and far-distance viewing. The results show that *METAVV* achieves the highest normalized QoE under both viewing conditions. Under near-distance viewing, *METAVV* reaches an average QoE of 0.860, compared with 0.833 for *FewVV*, 0.829 for *Sparkle*, 0.768 for *Rolling POT*, and 0.723 for *ViVo*. Under far-distance viewing, *METAVV* further improves to 0.890, while *Sparkle*, *FewVV*, *Rolling POT*, and *ViVo* remain lower. These values are consistent with the reported near-distance

QoE gains of 3.25%, 3.78%, 11.98%, and 18.91% over *FewVV*, *Sparkle*, *Rolling POT*, and *ViVo*, respectively, and the far-distance gains of 8.81%, 7.17%, 16.94%, and 24.93% over *FewVV*, *Sparkle*, *Rolling POT*, and *ViVo*, respectively.

The CQV results also show that *METAVV* provides the most stable perceptual delivery. Lower CQV indicates smoother quality variation across chunks. Under near-distance viewing, *METAVV* records the lowest CQV value of 0.056, while *Sparkle*, *Rolling POT*, and *ViVo* show higher quality variation. Under far-distance viewing, all methods exhibit lower CQV because visual sensitivity to fine details decreases with distance; however, *METAVV* still maintains the lowest variation. This confirms that latent-context Meta-RL and distance-aware LoD control help *METAVV* maintain smoother perceptual quality across volumetric chunks.

The rebuffering results further demonstrate the playback-continuity advantage of *METAVV*. Under near-distance viewing, *METAVV* achieves the lowest rebuffering time of 0.050 s/chunk, while *Sparkle*, *FewVV*, *Rolling POT*, and *ViVo* show higher stall durations. Under far-distance viewing, rebuffering decreases for all methods, but *METAVV* remains the most efficient with 0.040 s/chunk. This indicates that *METAVV* is able to jointly account for viewport uncertainty, throughput variation, buffer state, and distance-aware LoD selection when making bitrate decisions.

Table 5.3: Repeated-measures ANOVA results for *METAVV* and baseline methods.

Distance	**Metric**	**df**	F**-statistic**	p**-value**	**Effect Size** (η_p^2)
Near	Normalized QoE	4, 56	871.42	3.10×10^{-48}	0.984
Near	CQV	4, 56	42.76	1.84×10^{-15}	0.753
Near	Rebuffering	4, 56	612.35	7.22×10^{-43}	0.978
Far	Normalized QoE	4, 56	1146.89	5.67×10^{-52}	0.988
Far	CQV	4, 56	21.93	6.41×10^{-11}	0.610
Far	Rebuffering	4, 56	529.18	2.93×10^{-41}	0.974

Table 5.3 presents the repeated-measures ANOVA results for normalized QoE, CQV, and rebuffering under near- and far-distance viewing. The ANOVA results show statistically significant method effects for all three met-

rics under both viewing conditions, with all p-values below the conventional significance threshold of $p < 0.05$. This confirms that the choice of adaptation method has a statistically significant effect on perceptual quality, temporal quality stability, and playback continuity. The partial eta-squared (η_p^2) values indicate the proportion of variance in each metric explained by the adaptation method after accounting for repeated experimental conditions. The effect sizes are very large for normalized QoE and rebuffering under both near- and far-distance viewing, showing that the choice of adaptation method has a strong impact on perceptual quality and playback continuity. The CQV effect sizes are also large, indicating that method selection meaningfully affects temporal quality stability, although the effect is relatively smaller than for QoE and rebuffering.

For normalized QoE, the significant ANOVA result indicates that the QoE differences among *METAVV*, *Sparkle*, *FewVV*, *Rolling POT*, and *ViVo* are not due to random variation. The same conclusion applies to CQV, where the significant result confirms that the methods differ in their ability to maintain smooth quality transitions across chunks. For rebuffering, the significant ANOVA result confirms that the lower stall duration achieved by *METAVV* is statistically meaningful under both near- and far-distance viewing conditions.

Since the repeated-measures ANOVA results show significant method effects, Holm-corrected post-hoc paired comparisons were conducted between *METAVV* and each baseline. Table 5.4 reports the corrected p-values for QoE, CQV, and rebuffering under near- and far-distance viewing. Holm correction is applied because multiple pairwise comparisons are performed, and it reduces the risk of false-positive significance results.

The post-hoc results show that *METAVV* significantly outperforms *FewVV*, *Sparkle*, *Rolling POT*, and *ViVo* for normalized QoE under both viewing distances. This confirms that the reported QoE gains are statistically supported and not only visible in average values. For CQV, *METAVV* also shows statistically significant improvements over all baselines, confirming that its

Table 5.4: Holm-corrected post-hoc paired comparisons between *METAVV* and baseline methods.

Distance	**Metric**	**Comparison**	**Corrected p-value**	**Result**
Near	QoE	*METAVV* vs FewVV	2.80×10^{-8}	Significant
Near	QoE	*METAVV* vs Sparkle	1.34×10^{-8}	Significant
Near	QoE	*METAVV* vs Rolling POT	5.42×10^{-15}	Significant
Near	QoE	*METAVV* vs ViVo	2.18×10^{-18}	Significant
Near	CQV	*METAVV* vs FewVV	4.36×10^{-5}	Significant
Near	CQV	*METAVV* vs Sparkle	2.79×10^{-7}	Significant
Near	CQV	*METAVV* vs Rolling POT	8.21×10^{-7}	Significant
Near	CQV	*METAVV* vs ViVo	1.65×10^{-5}	Significant
Near	Rebuffering	*METAVV* vs FewVV	1.19×10^{-9}	Significant
Near	Rebuffering	*METAVV* vs Sparkle	3.84×10^{-12}	Significant
Near	Rebuffering	*METAVV* vs Rolling POT	9.46×10^{-12}	Significant
Near	Rebuffering	*METAVV* vs ViVo	4.17×10^{-17}	Significant
Far	QoE	*METAVV* vs FewVV	6.71×10^{-14}	Significant
Far	QoE	*METAVV* vs Sparkle	2.33×10^{-12}	Significant
Far	QoE	*METAVV* vs Rolling POT	1.89×10^{-17}	Significant
Far	QoE	*METAVV* vs ViVo	5.62×10^{-20}	Significant
Far	CQV	*METAVV* vs FewVV	7.44×10^{-5}	Significant
Far	CQV	*METAVV* vs Sparkle	3.15×10^{-5}	Significant
Far	CQV	*METAVV* vs Rolling POT	2.21×10^{-4}	Significant
Far	CQV	*METAVV* vs ViVo	3.68×10^{-4}	Significant
Far	Rebuffering	*METAVV* vs FewVV	2.93×10^{-14}	Significant
Far	Rebuffering	*METAVV* vs Sparkle	5.18×10^{-9}	Significant
Far	Rebuffering	*METAVV* vs Rolling POT	7.62×10^{-11}	Significant
Far	Rebuffering	*METAVV* vs ViVo	4.40×10^{-15}	Significant

latent-context adaptation and distance-aware tiling reduce temporal quality instability. For rebuffering, all comparisons are significant, indicating that *METAVV*'s lower stall duration is consistent across repeated evaluations and matched test conditions.

Overall, the variance and statistical significance analysis strengthens the reliability of the *METAVV* evaluation. The results show that *METAVV* consistently achieves higher QoE, lower chunk quality variation, and lower rebuffering than the baseline methods under both near- and far-distance viewing conditions. The combination of mean values, standard deviations, repeated-measures ANOVA, and Holm-corrected post-hoc testing confirms that the observed improvements are statistically meaningful and not caused by random seed-level variation. These findings complement the main QoE, stability, and runtime results and provide stronger evidence for the robustness and reproducibility of *METAVV*.

5.4 Limitations of the *METAVV* Evaluation

Although the experimental results demonstrate the effectiveness of *METAVV* under diverse network, content, and viewing conditions, several limitations have been addressed here. First, the evaluation is simulation-based and trace-driven. Public volumetric video datasets, recorded viewport traces, and real network traces were used to emulate practical streaming conditions; however, the framework was not deployed in a live end-to-end volumetric streaming testbed with real users, head-mounted displays, and production network infrastructure. Therefore, the results should be interpreted as evidence of performance under realistic simulated conditions rather than as proof of full real-world deployment readiness.

Second, the perceptual quality evaluation relies on objective metrics, particularly PCQM for volumetric point-cloud quality. PCQM is appropriate for repeatable evaluation because it captures geometry and colour distortions between reference and reconstructed point clouds. However, it may

not fully represent all aspects of human perception in immersive 6-DoF environments, such as user attention, visual discomfort, interaction delay, semantic importance of objects, and subjective tolerance to quality variation. Similarly, the QoE model combines PCQM-based viewport quality with penalties for quality variation, rebuffering, and latency, but it remains a metric-based approximation of user experience.

Third, no formal subjective user study was conducted as part of this thesis. As a result, the reported QoE gains represent objective, model-based improvements rather than subjective MOS-validated gains. A controlled subjective study with VR users would be required to further validate how closely the PCQM-based and QoE-based conclusions align with perceived user experience.

These limitations do not invalidate the comparative trends reported in this chapter, because all methods were evaluated under the same datasets, traces, network conditions, and QoE formulation. However, *METAVV* should be understood as a low-latency-oriented and perceptual-aware framework validated through realistic simulation. Future work would extend this evaluation through real-world deployment, subjective user trials, and cross-device testing using different head-mounted displays and volumetric rendering pipelines.

5.5 Conclusions

This chapter introduced *METAVV*, a perceptually adaptive volumetric streaming based on Meta-RL. Leveraging PEARL-style latent context inference, *METAVV* encodes user-specific scene and network dynamics into task embeddings to guide a rapidly adaptable ABR policy. Unlike fixed-policy, few-shot fixed training policies, and extensive deep learning baselines such as *Sparkle*, *FewVV*, *Rolling POT*, and *ViVo*, the proposed method dynamically tunes bitrate assignments with few adaptation steps, enabling personalized 6-DoF playback. Experiments show that *METAVV* consistently

outperforms *FewVV*, *Sparkle*, *Rolling POT*, and *ViVo* across both near- and far-distance viewing conditions. Under near-distance viewing, *METAVV* improves average QoE by 3.25%, 3.78%, 11.98%, and 18.91% over *FewVV*, *Sparkle*, *Rolling POT*, and *ViVo*, respectively. Under far-distance viewing, the corresponding gains increase to 8.81%, 7.17%, 16.94%, and 24.93%, respectively. These results highlight the effectiveness of perceptual-aware meta-policy learning, providing stable bitrate adaptation, robust perceptual quality, and improved generalization across different viewing distances and bandwidth profiles in real-world simulated volumetric video delivery scenarios.

Chapter 6

Conclusions and Future Works

6.1 Conclusions

The proliferation of immersive media content such as 360° VR, AR, XR, and volumetric video demands streaming architectures that can adapt to dynamic network conditions while delivering consistent and perceptually optimized QoE. The emergence of 5G and 6G network infrastructures, underpinned by SDN, NFV, MEC, and network slicing, introduces new possibilities for personalized and intelligent media delivery. However, these capabilities must be matched with adaptive, learning-driven algorithms capable of coping with the heterogeneous demands and resource constraints of next-generation rich media.

In this context, this thesis proposes two significant and complementary novel contributions, *EDGE360* and *METAVV*, that push the frontier of AI-enabled adaptive streaming in immersive environments:

- *EDGE360*, introduces a COMA-based MADRL solution for region-aware 360° video streaming. By leveraging viewport prediction, tile-level VMAF scoring, and multi-agent adaptation, *EDGE360* dynamically adjusts bitrate allocations to maintain QoE fairness and minimize quality degradation across clients. This solution is particularly suited to constrained edge-network deployments where latency sensitivity and bandwidth

variability are critical. Extensive evaluations on real-world 5G traces show that EDGE360 outperforms existing state-of-the-art approaches, including *BOLA360*, *360SRL*, and *RAPT360* in terms of average QoE by 18.00%, 11.86%, and 8.12%, respectively. *EDGE360* also performs better in terms of rebuffering time, quality switching, and bandwidth efficiency.

- *METAVV*, the second major novel contribution, addresses the more complex problem of adaptive streaming for volumetric video content. *METAVV* leverages a Meta-RL-based policy learning framework capable of generalizing across diverse viewing scenarios, network conditions, and volumetric datasets. Through region-aware segmentation and PCQM-based perceptual quality modeling, the architecture prioritizes bitrate decisions for viewport, marginal, and background regions under 6DoF viewing. The encoder-inference network and meta-policy components enable *METAVV* to rapidly adapt to unseen user trajectories and LoD requirements, ensuring optimal perceptual quality with minimal adaptation delay. The results show that *METAVV* improves average QoE by 3.25%, 3.78%, 11.98%, and 18.91% over FewVV, Sparkle, Rolling POT, and ViVo, respectively, under near-distance viewing. Under far-distance viewing, it improves QoE by 8.81%, 7.17%, 16.94%, and 24.93% over the same baselines, respectively. Notably, in comparison *METAVV* improves user QoE, enabling robust, personalized, and perceptually stable volumetric video streaming under dynamic and out-of-distribution conditions.

In a nutshell, *EDGE360* and *METAVV* offer a unified architectural vision for immersive content streaming, combining predictive intelligence, perceptual fidelity, reinforcement, and meta-learning in both tile-based and point cloud-based representations. These frameworks pave the way for deploying scalable, QoE-driven AR/VR/XR/volumetric applications over future 5G and 6G infrastructures. They also contribute novel formulations for

viewport-aware reward modeling, region-prioritized adaptation, and network-informed streaming that can serve as the basis for further research in AI-enabled adaptive immersive media streaming frameworks.

Furthermore, the sensitivity analyses in Chapters 4 and 5 show that viewport prediction accuracy influences the effectiveness of viewport-adaptive streaming. In *EDGE360*, larger viewport prediction errors reduce QoE and fairness and increase rebuffering because bitrate may be assigned to non-visible tiles. In *METAVV*, 6-DoF prediction errors affect both angular viewport relevance and distance-aware LoD decisions, with near-distance viewing being more sensitive than far-distance viewing. These results indicate that the proposed frameworks remain robust under moderate prediction uncertainty, but their gains depend on sufficiently accurate low-latency viewport prediction.

The conclusions of this thesis should also be interpreted in light of the evaluation limitations presented in chapters 4 and 5. The proposed *EDGE360* and *METAVV* frameworks were validated using simulation-based, trace-driven experiments with public datasets and network traces, rather than through live deployment in a production 5G/edge streaming testbed. In addition, the QoE models rely on objective perceptual metrics, namely VMAF for 360° video and PCQM for volumetric video. These metrics enable scalable, repeatable, and fair comparison across baselines, but they cannot fully capture all subjective aspects of immersive user perception. Moreover, no formal subjective user study was conducted. Therefore, the reported improvements should be understood as objective metric-based QoE gains under realistic simulated conditions. Future work would complement these results with subjective user trials, real-world edge/VR deployment, and evaluation across different devices, user groups, and real network environments.

6.2 Future Works

While this thesis presented two novel solutions *EDGE360* and *METAVV* for immersive media delivery, several new promising research directions can further enrich QoE-aware adaptive streaming.

1. **Energy, Carbon, and Cost-Aware Streaming:** A critical future dimension involves optimizing energy consumption and cost overheads in immersive streaming pipelines. This includes energy-aware tile prioritization, offloading strategies for edge/cloud components, and carbon footprint estimation tools for sustainable AR/VR/XR/volumetric video streaming services.

2. ***METAVV*** Enhancement for 6-DoF Immersion at the Edge: Extending *METAVV* to multi-agent meta-learning could facilitate per-region agents that adapt jointly to spatial depth, motion saliency, and network variations at the edge under multi-users scenario.

3. **Cross-Layer Network Slicing-Aware Integration:** Both *EDGE360* and *METAVV* can be extended to operate under end-to-end network slicing conditions, where media delivery is tightly coupled with SDN/NFV resource orchestration. Reinforcement learning agents can be trained for slice-aware decision-making across transport, MEC, and core domains.

4. **Explainable Streaming Intelligence:** With increasing adoption of AI-driven media pipelines, future directions should explore explainability of adaptation decisions. Visualization of tile prioritization, policy shifts across tasks, and the impact of perceptual weights could increase user and operator trust in AI-assisted delivery systems.

In summary, future research will extend *EDGE360* and *METAVV* toward sustainable, explainable, multi-agent, and cross-layer intelligent immersive media services, in realistic edge and beyond-5G/6G environments.

Bibliography

[1] F. E. Subhan, A. Yaqoob, C. H. Muntean, and G.-M. Muntean, "A survey on artificial intelligence techniques for improved rich media content delivery in a 5g and beyond network slicing context," *IEEE Communications Surveys & Tutorials*, 2024.

[2] Ericsson, "Ericsson Mobility Report November 2024," *Ericsson Report*, p. 40, June 2024.

[3] F. E. Subhan, A. Yaqoob, C. H. Muntean, and G.-M. Muntean, "A survey on artificial intelligence techniques for improved rich media content delivery in a 5g and beyond network slicing context," *IEEE Communications Surveys & Tutorials*, vol. 27, no. 2, pp. 1427–1487, 2025.

[4] A. Perkis, C. Timmerer, S. Baraković, J. B. Husić, S. Bech, S. Bosse, J. Botev, K. Brunnström, L. Cruz, K. De Moor *et al.*, "QUALINET white paper on definitions of immersive media experience (IMEx)," *arXiv preprint arXiv:2007.07032*, 2020.

[5] J. Sun, W. Gan, H.-C. Chao, and P. S. Yu, "Metaverse: Survey, applications, security, and opportunities," *arXiv preprint arXiv:2210.07990*, 2022.

[6] STATISTICA. (2023) Meta quest mobile app global downloads 2018-2022. [Online]. Available: https://www.statista.com/statistics/1283740/oculus-meta-quest-mobile-app-global-downloads/

[7] A. K. Jumani, W. A. Siddique, A. A. Laghari, A. Abro, and A. A. Khan, "Virtual Reality and Augmented Reality for Education," *Multimedia Computing Systems and Virtual Reality*, pp. 189–210, 2022.

[8] ITU-R. (2015, Sep.) Imt vision—framework and overall objectives of the future development of imt for 2020 and beyond. [Online]. Available: https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2083-0-201509-I!!PDF-E.pdf

[9] Cisco, "Cisco annual internet report (2018–2023) white paper," Cisco, Tech. Rep., 2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[10] C. Public. (2023) Cisco visual networking index: Global mobile data traffic forecast update, 2017–2022 white paper. [Online]. Available: https://twiki.cern.ch/twiki/pub/HEPIX/TechwatchNetwork/HtwNetworkDocuments/white-paper-c11-741490.pdf

[11] Ericsson, "Mobile network traffic q4 2025 (ericsson mobility report: data forecasts)," 2025. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/mobile-traffic-update

[12] ITU, "Global connectivity report 2025," International Telecommunication Union, Tech. Rep., 2025. [Online]. Available: https://www.itu.int/dms_pub/itu-d/opb/ind/D-IND-ICT_MDD.GCR-2025-4-PDF-E.pdf

[13] B. Agarwal, M. A. Togou, M. Ruffini, and G.-M. Muntean, "A Comprehensive Survey on Radio Resource Management in 5G HetNets: Current Solutions, Future Trends and Open Issues," *IEEE Communications Surveys & Tutorials*, 2022.

[14] E. G. Mirko Cano Soveri, Deepanshu Gautam, "3gpp: 5g network slice management," *3GPP*, 2023.

[15] A. Yaqoob, T. Bi, and G.-M. Muntean, "A survey on adaptive 360 video streaming: Solutions, challenges and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2801–2838, 2020.

[16] R. Viola, Á. Martín, M. Zorrilla, J. Montalban, P. Angueira, and G.-M. Muntean, "A survey on virtual network functions for media streaming: Solutions and future challenges," *ACM Computing Surveys*, vol. 55, no. 11, pp. 1–37, 2023.

[17] L. Ilola, L. Kondrad, S. Schwarz, and A. Hamza, "An overview of the mpeg standard for storage and transport of visual volumetric video-based coding," *Frontiers in Signal Processing*, vol. 2, p. 883943, 2022.

[18] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[19] F. E. Subhan, A. Yaqoob, C. H. Muntean, and G.-M. Muntean, "Edge360: Edge-enabled multi-agent drl for region-aware rate adaptation solution to enhance quality of 360° video streaming," *IEEE Transactions on Mobile Computing*, p. 1918–1935, 2025.

[20] ——, "A viewport-adaptive spatial saliency based meta-reinforcement learning framework for perceptual-aware volumetric video streaming," *IEEE Transactions on Network Science and Engineering*, pp. 1–18, 2026.

[21] S. Shaldambayeva, K. Zhumakhanova, A. Sandygulova, M. Rubagotti, and A. Shintemirov, "Partially observable markov decision processes in shared autonomy applications: A survey," *IEEE Access*, 2025.

[22] N. Kan, J. Zou, C. Li, W. Dai, and H. Xiong, "Rapt360: Reinforcement learning-based rate adaptation for 360-degree video streaming

with adaptive prediction and tiling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 3, pp. 1607–1623, 2021.

[23] J. Fu, X. Chen, Z. Zhang, S. Wu, and Z. Chen, "360srl: A sequential reinforcement learning approach for abr tile-based 360 video streaming," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2019, pp. 290–295.

[24] A. Zeynali, M. H. Hajiesmaili, and R. K. Sitaraman, "Bola360: Near-optimal view and bitrate adaptation for 360-degree video streaming," in *Proceedings of the 15th ACM Multimedia Systems Conference*, 2024, pp. 12–22.

[25] 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 Transactions on Multimedia Computing, Communications and Applications*, vol. 21, no. 7, pp. 1–27, 2025.

[26] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive," in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, 2012, pp. 97–108.

[27] 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*, 2017, pp. 315–323.

[28] A. Bentaleb, M. Lim, S. Hammoudi, S. Harous, and R. Zimmermann, "Solutions, challenges, and opportunities in volumetric video streaming: An architectural perspective," *ACM Trans. on Multimedia Computing, Communications and Applications*, vol. 21, no. 7, pp. 1–35, 2025.

[29] Ericsson, "Network slicing: Top 10 use cases to target," Ericsson, Tech. Rep., Mar 2021. [Online]. Available: http://www.ericsson.com/49c6d0/assets/local/networks-slicing/docs/top-network-slicing-use-cases.pdf

[30] F. B. Insights, "Network slicing market size, share & covid-19 impact analysis, by enterprise type, by end-user, and regional forecast, 2023-2030," Fortune Business Insights, Tech. Rep., Mar 2023. [Online]. Available: http://www.fortunebusinessinsights.com/segmentation/network-slicing-market-107303

[31] ITU. (2023, Dec.) Itu advances the development of imt-2030 for 6g mobile technologies. [Online]. Available: https://www.itu.int/en/mediacentre/Pages/PR-2023-12-01-IMT-2030-for-6G-mobile-technologies.aspx

[32] K. Abbas, Y. Cho, A. Nauman, P. W. Khan, T. A. Khan, and K. Kondepu, "Convergence of ai and mec for autonomous iot service provisioning and assurance in b5g," *IEEE Open Journal of the Communications Society*, vol. 4, pp. 2913–2929, 2023.

[33] S. Liu, Y. Wang, S. Li, and Y. Liu, "Madrl-based bitrate allocation for qoe fairness in 360° video streaming with viewport prediction," *Multimedia Systems*, vol. 31, no. 5, p. 343, 2025.

[34] C. Li, D. Lee, and M. Song, "Using deep reinforcement learning (drl) to optimize quality in 360-degree video tile management," *IEEE Transactions on Broadcasting*, p. 555–569, 2025.

[35] Y. Pei, M. Li, X. Huang, and X. Shen, "Qoe-aware volumetric video caching and rendering for mobile extended reality services," *IEEE Internet of Things Journal*, p. 21852–21865, 2025.

[36] G. Meynet, Y. Nehmé, J. Digne, and G. Lavoué, "Pcqm: A full-reference quality metric for colored 3d point clouds," in *IEEE International Conf. on Quality of Multimedia Experience (QoMEX)*, 2020, pp. 1–6.

[37] L. Zhang, H. Zhou, L. Shen, J. Liu, and L. Cui, "Towards attention-aware interactive 360-degree video streaming on smartphones," *IEEE Network*, p. 147–156, 2025.

[38] K. Hu, Y. Chen, K. Han, B. Li, H. Yang, Y. Jin, J. Liu, and F. Wang, "Livevv: Human-centered live volumetric video streaming system," *IEEE Internet of Things Journal*, 2025.

[39] K. Spiteri, R. Sitaraman, and D. Sparacio, "From theory to practice: Improving bitrate adaptation in the dash reference player," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 15, no. 2s, pp. 1–29, 2019.

[40] X. Chen, A. T. Z. Kasgari, and W. Saad, "Deep learning for content-based personalized viewport prediction of 360-degree vr videos," *IEEE Networking Letters*, vol. 2, no. 2, pp. 81–84, 2020.

[41] W. Feng, S. Wang, and Y. Dai, "Adaptive 360-degree streaming: Optimizing with multi-window and stochastic viewport prediction," *IEEE Transactions on Mobile Computing*, 2025.

[42] S. Ghazi, S. Farzi, and A. Nikoofard, "Federated learning for all: A reinforcement learning-based approach for ensuring fairness in client selection," *IEEE Access*, 2025.

[43] A. Xiao, S. Wu, Y. Ou, N. Chen, C. Jiang, and W. Zhang, "Qoe-fairness-aware bandwidth allocation design for mec-assisted abr video transmission," *IEEE Transactions on Network and Service Management*, vol. 22, no. 1, pp. 499–515, 2024.

[44] X. Wei, M. Zhou, S. Kwong, H. Yuan, S. Wang, G. Zhu, and J. Cao, "Reinforcement learning-based qoe-oriented dynamic adap-

tive streaming framework," *Information Sciences*, vol. 569, pp. 786–803, 2021.

[45] E. Lecarpentier and E. Rachelson, "Non-stationary Markov decision processes, a worst-case approach using model-based reinforcement learning," *Advances in neural information processing systems*, vol. 32, 2019.

[46] Q. Huang, "Model-based or model-free, a review of approaches in reinforcement learning," in *2020 International Conference on Computing and Data Science (CDS)*. IEEE, 2020, pp. 219–221.

[47] C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan, "Is q-learning provably efficient?" *Advances in neural information processing systems*, vol. 31, 2018.

[48] J. Chen, X. Huang, Y. Wang, H. Zhang, C. Liao, X. Xie, X. Li, and W. Xiao, "ASTPPO: A proximal policy optimization algorithm based on the attention mechanism and spatio–temporal correlation for routing optimization in software-defined networking," *Peer-to-Peer Networking and Applications*, pp. 1–19, 2023.

[49] M. Neunert, A. Abdolmaleki, M. Wulfmeier, T. Lampe, T. Springenberg, R. Hafner, F. Romano, J. Buchli, N. Heess, and M. Riedmiller, "Continuous-discrete reinforcement learning for hybrid control in robotics," in *Conference on Robot Learning*. PMLR, 2020, pp. 735–751.

[50] C. J. Bester, S. D. James, and G. D. Konidaris, "Multi-pass q-networks for deep reinforcement learning with parameterised action spaces," *arXiv preprint arXiv:1905.04388*, 2019.

[51] J. Beck, R. Vuorio, E. Zheran Liu, Z. Xiong, L. Zintgraf, C. Finn, and S. Whiteson, "A tutorial on meta-reinforcement learning," *Foun-*

dations and Trends in Machine Learning, vol. 18, no. 2-3, pp. 224–384, 2025.

[52] ——, "A survey of meta-reinforcement learning," *arXiv e-prints*, pp. arXiv–2301, 2023.

[53] L. Wen, E. H. Tseng, H. Peng, and S. Zhang, "Dream to adapt: Meta reinforcement learning by latent context imagination and mdp imagination," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, pp. 9701–9708, 2024.

[54] K. Rakelly, A. Zhou, C. Finn, S. Levine, and D. Quillen, "Efficient off-policy meta-reinforcement learning via probabilistic context variables," in *Int. Conf. on Machine Learning*. PMLR, 2019, pp. 5331–5340.

[55] Z. Bing, Y. Yun, K. Huang, and A. Knoll, "Context-based meta-reinforcement learning with bayesian nonparametric models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 10, pp. 6948–6965, 2024.

[56] Z. Liu, X. Wang, C. Feng, X. Sun, W. Zhan, and X. Chen, "Meta-reinforcement learning with mixture of experts for generalizable multi access in heterogeneous wireless networks," *IEEE Transactions on Communications*, vol. 74, pp. 870–885, 2026.

[57] D. Bonatto, S. Fachada, S. Rogge, A. Munteanu, and G. Lafruit, "Real-time depth video-based rendering for 6-dof hmd navigation and light field displays," *IEEE access*, vol. 9, pp. 146 868–146 887, 2021.

[58] J.-E. Marvie, Y. Nehmé, D. Graziosi, and et al., "Crafting the MPEG metrics for objective and perceptual quality assessment of volumetric videos," *Quality and User Experience*, vol. 8, no. 1, p. 4, 2023. [Online]. Available: https://doi.org/10.1007/s41233-023-00057-4

[59] E. S. Jang, M. Preda, K. Mammou, A. M. Tourapis, J. Kim, D. B. Graziosi, S. Rhyu, and M. Budagavi, "Video-based point-cloud-compression standard in mpeg: From evidence collection to committee draft [standards in a nutshell]," *IEEE Signal Processing Magazine*, vol. 36, no. 3, pp. 118–123, 2019.

[60] W. Li, X. Li, Y. Xu, Y. Yang, and S. Lu, "Metaabr: A meta-learning approach on adaptative bitrate selection for video streaming," *IEEE Transactions on Mobile Computing*, vol. 23, no. 3, pp. 2422–2437, 2023.

[61] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges," *Computer Networks*, vol. 167, p. 106984, 2020.

[62] A. A. Barakabitze, N. Barman, A. Ahmad, S. Zadtootaghaj, L. Sun, M. G. Martini, and L. Atzori, "QoE management of multimedia streaming services in future networks: A tutorial and survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 526–565, 2019.

[63] R. Dangi, A. Jadhav, G. Choudhary, N. Dragoni, M. K. Mishra, and P. Lalwani, "Ml-based 5g network slicing security: A comprehensive survey," *Future Internet*, vol. 14, no. 4, p. 116, 2022.

[64] C. Ssengonzi, O. P. Kogeda, and T. O. Olwal, "A survey of deep reinforcement learning application in 5G and beyond network slicing and virtualization," *Array*, vol. 14, p. 100142, 2022.

[65] H. P. Phyu, D. Naboulsi, and R. Stanica, "Machine learning in network slicing-a survey," *IEEE Access*, p. 39123–39153, 2023.

[66] A. Nauman, Y. A. Qadri, M. Amjad, Y. B. Zikria, M. K. Afzal, and S. W. Kim, "Multimedia Internet of Things: A comprehensive survey," *Ieee Access*, vol. 8, pp. 8202–8250, 2020.

[67] F. Spinelli and V. Mancuso, "Toward enabled industrial verticals in 5G: A survey on MEC-based approaches to provisioning and flexibility," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 596–630, 2020.

[68] X. Shen, J. Gao, W. Wu, K. Lyu, M. Li, W. Zhuang, X. Li, and J. Rao, "Ai-assisted network-slicing based next-generation wireless networks," *IEEE Open Journal of Vehicular Technology*, vol. 1, pp. 45–66, 2020.

[69] J. Cao, M. Ma, H. Li, R. Ma, Y. Sun, P. Yu, and L. Xiong, "A survey on security aspects for 3GPP 5G networks," *IEEE communications surveys & tutorials*, vol. 22, no. 1, pp. 170–195, 2019.

[70] P. Wyszkowski, J. Kienig, K. Zieliński, Ł. Czekierda, and M. Zawadzki, "Comprehensive tutorial on the organization of a standards-aligned network slice/subnet design process and opportunities for its automation," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2023.

[71] E. Coronado, R. Behravesh, T. Subramanya, A. Fernàndez-Fernàndez, M. S. Siddiqui, X. Costa-Pérez, and R. Riggio, "Zero touch management: A survey of network automation solutions for 5g and 6g networks," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2535–2578, 2022.

[72] J. Santos, T. Wauters, B. Volckaert, and F. De Turck, "Towards low-latency service delivery in a continuum of virtual resources: State-of-the-art and research directions," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2557–2589, 2021.

[73] X. Shen, J. Gao, W. Wu, M. Li, C. Zhou, and W. Zhuang, "Holistic network virtualization and pervasive network intelligence for 6g," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 1–30, 2022.

[74] J. Lee, F. Solat, T. Y. Kim, and H. V. Poor, "Federated learning-empowered mobile network management for 5g and beyond networks: From access to core," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2024.

[75] S. Ebrahimi, F. Bouali, and O. C. L. Haas, "Resource management from single-domain 5g to end-to-end 6g network slicing: A survey," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2024.

[76] L. Nadeem, M. A. Azam, Y. Amin, M. A. Al-Ghamdi, K. K. Chai, M. F. N. Khan, and M. A. Khan, "Integration of D2D, network slicing, and MEC in 5G cellular networks: Survey and challenges," *IEEE Access*, vol. 9, pp. 37 590–37 612, 2021.

[77] R. Behravesh, D. F. Perez-Ramirez, A. Rao, D. Harutyunyan, R. Riggio, and R. Steinert, "Ml-driven dash content pre-fetching in mec-enabled mobile networks," in *2020 16th International Conference on Network and Service Management (CNSM)*. IEEE, 2020, pp. 1–7.

[78] J. Uriol, I. Yeregui, Á. Gabilondo, R. Viola, P. Angueira, and J. Montalbán, "Context-aware adaptive prefetching for dash streaming over 5g networks," in *2023 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*. IEEE, 2023, pp. 1–6.

[79] Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila, "A survey on mobile augmented reality with 5G mobile edge computing: Architectures, applications, and technical aspects," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1160–1192, 2021.

[80] J. Yang, Z. Guo, J. Luo, Y. Shen, and K. Yu, "Cloud-edge-end collaborative caching based on graph learning for cyber-physical virtual reality," *IEEE Systems Journal*, vol. 17, no. 4, pp. 5097–5108, 2023.

[81] A. Domeke, B. Cimoli, and I. T. Monroy, "Integration of Network Slicing and Machine Learning into Edge Networks for Low-Latency Services

in 5G and beyond Systems," *Applied Sciences*, vol. 12, no. 13, p. 6617, 2022.

[82] H.-W. Kao and E. H.-K. Wu, "QoE Sustainability on 5G and Beyond 5G Networks," *IEEE Wireless Communications*, vol. 30, no. 1, pp. 118–125, 2023.

[83] J. Aguilar-Armijo, C. Timmerer, and H. Hellwagner, "SPACE: Segment Prefetching and Caching at the Edge for Adaptive Video Streaming," *IEEE Access*, vol. 11, pp. 21 783–21 798, 2023.

[84] ——, "Segment prefetching at the edge for adaptive video streaming," in *2022 18th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2022, pp. 339–344.

[85] R. Behravesh, A. Rao, D. F. Perez-Ramirez, D. Harutyunyan, R. Riggio, and M. Boman, "Machine learning at the mobile edge: The case of dynamic adaptive streaming over http (dash)," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 4779–4793, 2022.

[86] W. Shi, Q. Li, C. Wang, G. Shen, W. Li, Y. Wu, and Y. Jiang, "LEAP: learning-based smart edge with caching and prefetching for adaptive video streaming," in *Proceedings of the International Symposium on Quality of Service*, 2019, pp. 1–10.

[87] W. Liu, H. Zhang, H. Ding, Z. Yu, and D. Yuan, "Qoe-aware collaborative edge caching and computing for adaptive video streaming," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2023.

[88] J. Yu, H. Wen, G. Pan, S. Zhang, X. Chen, and S. Xu, "Quality of experience oriented adaptive video streaming for edge assisted cellular networks," *IEEE Wireless Communications Letters*, vol. 11, no. 11, pp. 2305–2309, 2022.

[89] S. M. A. H. Bukhari, E. Baccour, K. Bilal, J. Shuja, A. Erbad, and M. Bilal, "To transcode or not? a machine learning based edge video caching and transcoding strategy," *Computers and Electrical Engineering*, vol. 109, p. 108741, 2023.

[90] H. Xiao, C. Xu, Z. Feng, R. Ding, S. Yang, L. Zhong, J. Liang, and G.-M. Muntean, "A transcoding-enabled 360° vr video caching and delivery framework for edge-enhanced next-generation wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 5, pp. 1615–1631, 2022.

[91] J. Zeng, X. Zhou, and K. Li, "Madrl-based joint edge caching and bitrate selection for multicategory 360° video streaming," *IEEE Internet of Things Journal*, vol. 11, no. 1, pp. 584–596, 2023.

[92] Z. Chen, L. Zou, X. Tao, L. Xu, G.-M. Muntean, and X. Wang, "Edgevr360: Edge-assisted multiuser-oriented intelligent 360-degree video delivery scheme over wireless networks," in *CAAI International Conference on Artificial Intelligence*. Springer, 2022, pp. 242–255.

[93] R. Shi, Q. Fan, S. Fu, X. Zhang, X. Li, and M. Chen, "COCAM: a cooperative video edge caching and multicasting approach based on multi-agent deep reinforcement learning in multi-clouds environment," *Journal of Cloud Computing*, vol. 12, no. 1, pp. 1–13, 2023.

[94] Y. Liu, P. Shi, B. Peng, H. Yan, Y. Zhou, B. Han, Y. Zheng, C. Lin, J. Jiang, Y. Fan *et al.*, "iqiyi-vid: A large dataset for multi-modal person identification," *arXiv preprint arXiv:1811.07548*, 2018.

[95] G. Pan, H. Zhang, S. Xu, S. Zhang, and X. Chen, "Joint Optimization of Video-Based AI Inference Tasks in MEC-Assisted Augmented Reality Systems," *IEEE Transactions on Cognitive Communications and Networking*, vol. 9, no. 2, pp. 479–493, 2023.

[96] A. Alavian and M. C. Rotkowitz, "Improving admm-based optimization of mixed integer objectives," in *2017 51st Annual Conference on Information Sciences and Systems (CISS)*, 2017, pp. 1–6.

[97] X. Xu and Y. Song, "A deep reinforcement learning-based optimal computation offloading scheme for vr video transmission in mobile edge networks," *IEEE Access*, p. 122772–122781, 2023.

[98] Y. Hou, L. Liu, Q. Wei, X. Xu, and C. Chen, "A novel ddpg method with prioritized experience replay," in *2017 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE, 2017, pp. 316–321.

[99] J. Jia, L. Yang, J. Chen, L. Ma, and X. Wang, "Online delay optimization for mec and ris-assisted wireless vr networks," *Wireless Networks*, pp. 1–21, 2024.

[100] S. Huang, A. Kanervisto, A. Raffin, W. Wang, S. Ontañón, and R. F. J. Dossa, "A2c is a special case of ppo," *arXiv preprint arXiv:2205.09123*, 2022.

[101] Y. Kim and H. Lim, "Multi-agent reinforcement learning-based resource management for end-to-end network slicing," *IEEE Access*, vol. 9, pp. 56 178–56 190, 2021.

[102] Z. Wang, Y. Wei, F. R. Yu, and Z. Han, "Utility optimization for resource allocation in edge network slicing using DRL," in *IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2020, pp. 1–6.

[103] ——, "Utility optimization for resource allocation in multi-access edge network slicing: A twin-actor deep deterministic policy gradient approach," *IEEE Transactions on Wireless Communications*, vol. 21, no. 8, pp. 5842–5856, 2022.

[104] X. Liu and Y. Deng, "Learning-based prediction, rendering and association optimization for MEC-enabled wireless virtual reality (VR)

networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 10, pp. 6356–6370, 2021.

[105] J. Du, F. R. Yu, G. Lu, J. Wang, J. Jiang, and X. Chu, "MEC-assisted immersive VR video streaming over terahertz wireless networks: A deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9517–9529, 2020.

[106] M. Babaeizadeh, I. Frosio, S. Tyree, J. Clemons, and J. Kautz, "Ga3c: Gpu-based a3c for deep reinforcement learning," *CoRR abs/1611.06256*, 2016.

[107] S. Bi, H. Chen, X. Li, S. Wang, Y. Wu, and L. Qian, "A two-stage deep reinforcement learning framework for mec-enabled adaptive 360-degree video streaming," *IEEE Transactions on Mobile Computing*, p. 14313–14329, 2024.

[108] S. Kumar, A. Franklin A, J. Jin, and Y.-N. Dong, "Seer: Learning-based 360° video streaming for mec-equipped cellular networks," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 6, pp. 3308–3319, 2023.

[109] X. Shi, S. Zhang, M. Liu, L. Meng, L. Wei, Y. Gu, K. Liu, H. Cheng, Y. Song, L. Tang, A. Zhu, N. Chen, and Z. Qian, "Mystique: User-level adaptation for real-time video analytics in edge networks via meta-rl," *IEEE Transactions on Mobile Computing*, vol. 24, no. 5, pp. 3615–3632, 2025.

[110] W. Li, Q. Li, W. Tian, J. Gao, F. Wu, J. Liu, and J. Ren, "Mucvr: Edge computing-enabled high-quality multi-user collaboration for interactive mvr," *IEEE Transactions on Parallel and Distributed Systems*, vol. 36, no. 10, pp. 2058–2072, 2025.

[111] Y. Ma, C. Xu, Z. Wu, R. Ding, H. Xiao, L. Zhong, Y. Zhuang, and G.-M. Muntean, "Ca-live360: Crowd-assisted transcoding and delivery

for live 360-degree video streaming," *Computer Networks*, vol. 251, p. 110618, 2024.

[112] B. Hou, S. Yang, F. Li, L. Zhu, X. Chen, Y. Wang, and X. Fu, "Nova: Neural-optimized viewport adaptive 360-degree video streaming at the edge," *IEEE Transactions on Services Computing*, p. 4027–4040, 2024.

[113] R. Deng, "Qoe optimized adaptive streaming method for 360° virtual reality videos over mec-assisted network," *Multimedia Tools and Applications*, vol. 83, no. 17, pp. 52 737–52 761, 2024.

[114] R. Viola, A. Martin, M. Zorrilla, and J. Montalbán, "Mec proxy for efficient cache and reliable multi-cdn video distribution," in *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*. IEEE, 2018, pp. 1–7.

[115] N. Hasanov, "Content delivery networks (cdn): Opportunities, challenges and future perspectives," *Computer network technology*, vol. 1, no. 1, pp. 15–24, 2023.

[116] A. Rico-Alvariño, I. Bouazizi, M. Griot, P. Kadiri, L. Liu, and T. Stockhammer, "3gpp rel-17 extensions for 5g media delivery," *IEEE Transactions on Broadcasting*, vol. 68, no. 2, pp. 422–438, 2022.

[117] V. K. Shrivastava, S. Baek, and Y. Baek, "5G evolution for multicast and broadcast services in 3GPP release 17," *IEEE Communications Standards Magazine*, vol. 6, no. 3, pp. 70–76, 2022.

[118] Christoph Mertens, "The berec report on the internet ecosystem 2022," in *Draft BEREC Report on the Internet Ecosystem,[Online]. Available:*. BEREC, 2022.

[119] A. O. Al-Abbasi, V. Aggarwal, T. Lan, Y. Xiang, M.-R. Ra, and Y.-F. Chen, "Fasttrack: Minimizing stalls for cdn-based over-the-top video streaming systems," *IEEE Transactions on Cloud Computing*, vol. 9, no. 4, pp. 1453–1466, 2019.

[120] R. Farahani, A. Bentaleb, E. Çetinkaya, C. Timmerer, R. Zimmermann, and H. Hellwagner, "Hybrid p2p-cdn architecture for live video streaming: An online learning approach," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 1911–1917.

[121] J. F. Cevallos Moreno, R. Sattler, R. P. Caulier Cisterna, L. Ricciardi Celsi, A. Sánchez Rodríguez, and M. Mecella, "Online service function chain deployment for live-streaming in virtualized content delivery networks: A deep reinforcement learning approach," *Future Internet*, vol. 13, no. 11, p. 278, 2021.

[122] I. M. Ozcelik and C. Ersoy, "Alvs: Adaptive live video streaming using deep reinforcement learning," *Journal of Network and Computer Applications*, vol. 205, p. 103451, 2022.

[123] H.-A. Tran, S. Souihi, D. Tran, and A. Mellouk, "Mabrese: A new server selection method for smart SDN-based CDN architecture," *IEEE Communications Letters*, vol. 23, no. 6, pp. 1012–1015, 2019.

[124] B. Hayes and Y. Chang, "Lightweight evolving 360 VR adaptive video delivery," in *2020 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2020, pp. 815–819.

[125] A. El-Saleh, A. Alhammadi, I. Shayea, A. Azizan, and W. Hassan, "Mean opinion score estimation for mobile broadband networks using bayesian networks," *CMC-COMPUTERS MATERIALS & CONTINUA*, vol. 72, no. 3, pp. 4571–4587, 2022.

[126] A. Narayanan, S. Verma, E. Ramadan, P. Babaie, and Z.-L. Zhang, "DeepCache: A Deep Learning Based Framework For Content Caching," *Workshop on Network Meets AI & ML*, pp. 48–53, 2018.

[127] W. Shi, C. Wang, Y. Jiang, Q. Li, G. Shen, and G.-M. Muntean, "CoLEAP: Cooperative learning-based edge scheme with caching and

prefetching for DASH video delivery," *IEEE Transactions on Multimedia*, vol. 23, pp. 3631–3645, 2020.

[128] M. M. Uddin and J. Park, "Machine learning model evaluation for 360° video caching," in *2022 IEEE World AI IoT Congress (AIIoT)*. IEEE, 2022, pp. 238–244.

[129] A. Sadeghi, G. Wang, and G. B. Giannakis, "Deep reinforcement learning for adaptive caching in hierarchical content delivery networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 4, pp. 1024–1033, 2019.

[130] D. S. Berger, "Towards lightweight and robust machine learning for cdn caching," in *Proceedings of the 17th ACM Workshop on Hot Topics in Networks*, 2018, pp. 134–140.

[131] N. Tahmasebi-Pouya, M.-A. Sarram, and S.-A. Mostafavi, "Load balancing in mobile edge computing: A reinforcement learning approach," in *2022 Sixth International Conference on Smart Cities, Internet of Things and Applications (SCIoT)*, 2022, pp. 1–6.

[132] W. Akbar, A. Muhammad, J. J. D. Rivera, and W.-C. Song, "Machine learning-based cache optimization on mec platform," in *2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE, 2021, pp. 250–253.

[133] M. Li, J. Gao, C. Zhou, X. Shen, and W. Zhuang, "Adaptive mobile vr content delivery for industrial 5.0," in *Proceedings of the 1st Workshop on Digital Twin & Edge AI for Industrial IoT*, 2022, pp. 1–6.

[134] V. Kirilin, A. Sundarrajan, S. Gorinsky, and R. K. Sitaraman, "Rl-cache: Learning-based cache admission for content delivery," in *Proceedings of the 2019 Workshop on Network Meets AI & ML*, 2019, pp. 57–63.

[135] F. Wang, J. Liu, C. Zhang, L. Sun, and K. Hwang, "Intelligent edge learning for personalized crowdsourced livecast: Challenges, opportunities, and solutions," *IEEE Network*, vol. 35, no. 1, pp. 170–176, 2021.

[136] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proceedings of the conference of the ACM special interest group on data communication*, 2017, pp. 197–210.

[137] H. Mao, S. Chen, D. Dimmery, S. Singh, D. Blaisdell, Y. Tian, M. Alizadeh, and E. Bakshy, "Real-world video adaptation with reinforcement learning," *arXiv preprint arXiv:2008.12858*, 2020.

[138] M. Kim and K. Chung, "Reinforcement learning-based adaptive streaming scheme with edge computing assistance," *Sensors*, vol. 22, no. 6, p. 2171, 2022.

[139] H. H. C. Nguyen, D. Grzonka, B. T. Khoa, K. Sagar, I. A. Abbasi, R. Mahaveerakannan, and A. Alkhayyat, "New trends in over the top media service (ott) web user behaviour analysis and unethical user prediction," *Mobile Networks and Applications*, pp. 1–17, 2023.

[140] P. Varshney, S. Gaur, and S. Pal, "A novel approach of load balancing in content delivery networks by optimizing the surrogate server," in *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2021, pp. 790–795.

[141] S. Lobanov, V. Zaguskin, and A. Kochetkov, "Estimating OTT Player QoE Metrics From CDN Logs Uisng AI and Machine Learning," Jun. 10 2021, uS Patent App. 17/112,550.

[142] J. Wang, L. Zhao, J. Liu, and N. Kato, "Smart resource allocation for mobile edge computing: A deep reinforcement learning approach," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 3, pp. 1529–1541, 2021.

[143] J. Li, Z. Lu, Y. Tong, J. Wu, S. Huang, M. Qiu, and W. Du, "A general ai-defined attention network for predicting cdn performance," *Future Generation Computer Systems*, vol. 100, pp. 759–769, 2019.

[144] F. G. P. Jr and C. A. Gurat, "Revolutionizing multi-campus communication: A next-generation ospf-based network design for nvsu's distributed learning environment," *International Journal*, vol. 12, no. 4, 2023.

[145] H. Heidarirad and M. Wang, "Vv-dash: A framework for volumetric video dash streaming," in *Proceedings of the 16th ACM Multimedia Systems Conference*, 2025, pp. 256–262.

[146] B. Han, Y. Liu, and F. Qian, "Vivo: Visibility-aware mobile volumetric video streaming," in *Proceedings of the 26th annual international conference on mobile computing and networking*, 2020, pp. 1–13.

[147] J. Li, H. Wang, Z. Liu, P. Zhou, X. Chen, Q. Li, and R. Hong, "Toward optimal real-time volumetric video streaming: A rolling optimization and deep reinforcement learning based approach," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 12, pp. 7870–7883, 2023.

[148] H. Chen, B. Xu, S. Wang, X. Cao, and Z. Ma, "Toward adaptive volumetric video streaming: A joint network-viewport adaptation framework," *IEEE Communications Magazine*, p. 197–203, 2024.

[149] K. Lee, J. Yi, Y. Lee, S. Choi, and Y. M. Kim, "Groot: A real-time streaming system of high-fidelity volumetric videos," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 2020, pp. 1–14.

[150] L. Wang, C. Li, W. Dai, S. Li, J. Zou, and H. Xiong, "Qoe-driven adaptive streaming for point clouds," *IEEE Transactions on Multimedia*, vol. 25, pp. 2543–2558, 2023.

[151] Y. Liu, B. Han, F. Qian, A. Narayanan, and Z.-L. Zhang, "Vues: Practical mobile volumetric video streaming through multiview transcoding," in *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, 2022, pp. 514–527.

[152] A. Zhang, C. Wang, B. Han, and F. Qian, "{YuZu}:{Neural-Enhanced} volumetric video streaming," in *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, 2022, pp. 137–154.

[153] J. Liu, B. Zhu, F. Wang, Y. Jin, W. Zhang, Z. Xu, and S. Cui, "Cav3: Cache-assisted viewport adaptive volumetric video streaming," in *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2023, pp. 173–183.

[154] J. Shi, M. Zhang, L. Shen, J. Liu, Y. Zhang, L. Pu, and J. Xu, "Towards full-scene volumetric video streaming via spatially layered representation and nerf generation," in *Proceedings of the 34th edition of the Workshop on Network and Operating System Support for Digital Audio and Video*, 2024, pp. 22–28.

[155] X. Wang, W. Liu, S. Gong, Z. Liu, J. Xu, and Y. Fang, "Spatial quality oriented rate control for volumetric video streaming via deep reinforcement learning," *IEEE Transactions on Circuits and Systems for Video Technology*, p. 5092–5108, 2024.

[156] M. Yang, K. Tang, and C. Wang, "Meta-inr: Efficient encoding of volumetric data via meta-learning implicit neural representation," in *2025 IEEE 18th Pacific Visualization Conference (PacificVis)*. IEEE, 2025, pp. 246–251.

[157] N. Kan, C. Li, Y. Jiang, W. Dai, J. Zou, H. Xiong, and L. Toni, "Merina+: Improving generalization for neural video adaptation via information-theoretic meta-reinforcement learning," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2025.

[158] W. Choi and J. Yoon, "Metaabr: environment-adaptive video streaming system with meta-reinforcement learning," in *Proceedings of the 3rd International CoNEXT Student Workshop*, ser. CoNEXT-SW '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 37–39. [Online]. Available: https://doi.org/10.1145/3565477.3569155

[159] W. Li, X. Li, Y. Xu, Y. Yang, and S. Lu, "Metaabr: A meta-learning approach on adaptative bitrate selection for video streaming," *IEEE Transactions on Mobile Computing*, vol. 23, no. 3, pp. 2422–2437, 2024.

[160] L. Wei, Y. Liu, F. Wang, and D. Wang, "Fewvv: Few-shot adaptive bitrate volumetric video streaming with prompted online adaptation," *IEEE Internet of Things Journal*, vol. 11, no. 19, pp. 32 055–32 066, 2024.

[161] H. Gong, Q. Li, J. Li, and Z. Liu, "Adaptive 3d gaussian splatting video streaming: Visual saliency-aware tiling and meta-learning-based bitrate adaptation," *arXiv preprint arXiv:2507.14454*, 2025.

[162] M. Z. Wahba, S. Baldoni, and F. Battisti, "Learning-based viewport prediction for 360-degree videos: A review." *Electronics (2079-9292)*, vol. 14, no. 18, 2025.

[163] D. Lee, M. Choi, and J. Lee, "Prediction of head movement in 360-degree videos using attention model," *Sensors*, vol. 21, no. 11, p. 3678, 2021.

[164] X. Hou, S. Dey, J. Zhang, and M. Budagavi, "Predictive view generation to enable mobile 360-degree and VR experiences," in *Proceedings of the 2018 Morning Workshop on Virtual Reality and Augmented Reality Network*. ACM, 2018, pp. 20–26.

[165] S. Khan, J. Zhang, K. K. Soothar, I. Ahmad, S. Ahmad, and I. Ali, "Deep learning-based caching optimization for vr 360° videos in vehicular edge computing," *Scientific Reports*, vol. 15, no. 1, p. 41934, 2025.

[166] K. Hu, H. Yang, Y. Jin, J. Liu, Y. Chen, M. Zhang, and F. Wang, "Understanding user behavior in volumetric video watching: Dataset, analysis and prediction," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 1108–1116.

[167] J. Li, Z. Li, Z. Liu, P. Zhou, R. Hong, Q. Li, and H. Hu, "Viewport prediction for volumetric video streaming by exploring video saliency and trajectory information," *arXiv preprint arXiv:2311.16462*, 2023.

[168] Y. Tian, Y. Zhong, Y. Han, and F. Chen, "Viewport prediction with cross modal multiscale transformer for 360° video streaming," *Scientific Reports*, vol. 15, no. 1, p. 30346, 2025.

[169] J. Li, Z. Zhao, Q. Li, Z. Li, P. Zhou, Z. Liu, H. Zhou, and Z. Li, "Vpformer: Leveraging transformer with voxel integration for viewport prediction in volumetric video," *ACM Trans. on Multimedia Computing, Communications and Applications*, vol. 21, no. 6, pp. 1–27, 2025.

[170] M. Orduna, C. Díaz, L. Muñoz, P. Pérez, I. Benito, and N. García, "Video multimethod assessment fusion (vmaf) on 360vr contents," *IEEE Transactions on Consumer Electronics*, vol. 66, no. 1, pp. 22–31, 2020.

[171] Y. Wang, J. Li, Z. Li, S. Shang, and Y. Liu, "Synergistic temporal-spatial user-aware viewport prediction for optimal adaptive 360-degree video streaming," *IEEE Transactions on Broadcasting*, p. 453–467, 2024.

[172] Y. Chen, S. Omoma, H. Kwon, H. Inaltekin, and M. Gorlatova, "Quantifying and exploiting vr frame correlations: An application of a sta-

tistical model for viewport pose," *IEEE Transactions on Mobile Computing*, p. 11466–11482, 2024.

[173] J. Chakareski, R. Aksu, X. Corbillon, G. Simon, and V. Swaminathan, "Viewport-driven rate-distortion optimized 360º video streaming," *2018 IEEE International Conference on Communications (ICC)*, pp. 1–7, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:4118274

[174] R. G. d. A. Azevedo, N. Birkbeck, F. De Simone, I. Janatra, B. Adsumilli, and P. Frossard, "Visual distortions in 360° videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 8, pp. 2524–2537, 2019.

[175] U. U. Khan, N. Dilshad, M. H. Rehmani, and T. Umer, "Fairness in cognitive radio networks: models, measurement methods, applications, and future research directions," *Journal of Network and Computer Applications*, vol. 73, pp. 12–26, 2016.

[176] A. Balcı and R. Sokullu, "Fairness aware deep reinforcement learning for grant-free noma-iot networks," *Internet of Things*, vol. 25, p. 101079, 2024.

[177] Y. Wang, Y. Wan, C. Zhang, L. Bai, L. Cui, and P. Yu, "Competitive multi-agent deep reinforcement learning with counterfactual thinking," in *IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 1366–1371.

[178] H. Zhou, T. Lan, and V. Aggarwal, "Pac: Assisted value factorization with counterfactual predictions in multi-agent reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 15 757–15 769, 2022.

[179] X. Liu and K. Zhang, "Partially observable multi-agent rl with (quasi-) efficiency: the blessing of information sharing," in *International Conference on Machine Learning*. PMLR, 2023, pp. 22 370–22 419.

[180] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[181] Y. Bao, H. Wu, T. Zhang, A. A. Ramli, and X. Liu, "Shooting a moving target: Motion-prediction-based transmission for 360-degree videos," in *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 2016, pp. 1161–1170.

[182] H.-N. Hu, Y.-C. Lin, M.-Y. Liu, H.-T. Cheng, Y.-J. Chang, and M. Sun, "Deep 360 pilot: Learning a deep agent for piloting through 360° sports videos," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1396–1405.

[183] Y. Guo, M. Xu, L. Jiang, X. Deng, J. Zhou, G. Chen, and L. Sigal, "Proposal with alignment: A bi-directional transformer for 360° video viewport proposal," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 11, pp. 11 423–11 437, 2024.

[184] B. J. Li, J. N. Bailenson, A. Pines, W. J. Greenleaf, and L. M. Williams, "A public database of 360 videos with corresponding ratings of arousal and valence," in *Virtual Human Interaction Lab. Stanford University*, 2017. [Online]. Available: http://vhil.stanford.edu/360-video-database/

[185] Ffmpeg: A complete, cross-platform solution to record, convert and stream audio and video. [Online]. Available: http://ffmpeg.org/

[186] GPAC community, "Dynamic adaptive streaming over http (dash)," in *Fragmentation, segmentation, splitting and interleaving*, 2022. [Online]. Available: https://gpac.wp.imt.fr/mp4box/

[187] 3rd Generation Partnership Project, "3gpp tr 26.918: Virtual reality (vr) media services over 3gpp, 2017," in *SA WG4 - Multimedia Codecs, Systems and Services*, 2017. [Online]. Available: http://www.3gpp.org/ftp/tsg_sa/WG4_CODEC/TSGS4_94/Docs/S4-170752.zip

[188] L. Sun, F. Duanmu, Y. Liu, Y. Wang, Y. Ye, H. Shi, and D. Dai, "Multi-path multi-tier 360-degree video streaming in 5g networks," in *Dynamic Predictive Streaming of 360 Degree Video*, ser. MMSys '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 162–173. [Online]. Available: https://doi.org/10.1145/3204949.3204978

[189] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Toward low-latency and ultra-reliable virtual reality," *IEEE Network*, vol. 32, no. 2, pp. 78–84, 2018.

[190] L. Hsiao, B. Krajancich, P. Levis, G. Wetzstein, and K. Winstein, "Towards retina-quality vr video streaming: 15ms could save you 80% of your bandwidth," *ACM SIGCOMM Computer Communication Review*, vol. 52, no. 1, pp. 10–19, 2022.

[191] G. Gautier, A. Mercat, L. Fréneau, M. Pitkänen, and J. Vanne, "Uvg-vpc: Voxelized point cloud dataset for visual volumetric video-based coding," in *2023 15th international conference on quality of Multimedia experience (QoMEX)*. IEEE, 2023, pp. 244–247.

[192] C. Qi, H. Li, and P. Huang, "Timrl: A novel meta-reinforcement learning framework for non-stationary and multi-task environments," *arXiv preprint arXiv:2501.07146*, 2025.

[193] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International conference on machine learning*. PMLR, 2018, pp. 1587–1596.

[194] N. Vieillard, T. Kozuno, B. Scherrer, O. Pietquin, R. Munos, and M. Geist, "Leverage the average: an analysis of kl regularization in reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 163–12 174, 2020.

[195] Google Draco Authors, "Google Draco: Compression library for 3d graphics," https://github.com/google/draco, 2019–2025, accessed: 2025-06-15.

[196] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.

[197] E. d'Eon, B. Harrison, T. Myers, and P. A. Chou, "8i Voxelized Full Bodies – A Voxelized Point Cloud Dataset," ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG), Geneva, Switzerland, Tech. Rep. WG11M40059/WG1M74006, Jan. 2017.

[198] Y. Xu, Y. Lu, and Z. Wen, "Owlii dynamic human textured mesh sequence dataset," in *ISO/IEC JTC1/SC29/WG1 1 input document m41658*, 2017.

[199] J. van der Hooft *et al.*, "4g/lte bandwidth logs, ghent university," GitHub-style Dataset, 2016, [Online]. Available: https://users.ugent.be/~jvdrhoof/dataset-4g/.

[200] U. Kulkarni, K. Diab, S. Aggarwal, L. Cao, F. Ahmed, P. Sharma, and S. Fahmy, "Understanding the impact of wi-fi configuration on volumetric video streaming applications," in *Proceedings of the 2023 Workshop on Emerging Multimedia Systems*, 2023, pp. 41–47.

[201] P. Pérez, D. Corregidor, E. Garrido, I. Benito, E. González-Sosa, J. Cabrera, D. Berjón, C. Díaz, F. Morán, N. García *et al.*, "Live free-

viewpoint video in immersive media production over 5g networks," *IEEE Transactions on Broadcasting*, vol. 68, no. 2, pp. 439–450, 2022.

[202] J. Li, Z. Li, Z. Liu, P. Y. Zhou, R. Hong, Q. Li, and H. Hu, "Viewport prediction for volumetric video streaming by exploring video saliency and user trajectory information," *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.

[203] Q. Hu, Q. He, H. Zhong, G. Lu, X. Zhang, G. Zhai, and Y. Wang, "Varfvv: View-adaptive real-time interactive free-view video streaming with edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 43, no. 7, pp. 2620–2634, 2025.

[204] Z. Wu, Z. Liu, J. Lin, Y. Lin, and S. Han, "Lite transformer with long-short range attention," *arXiv preprint arXiv:2004.11886*, 2020.