

# Digital Agent-Based Resource Management for Short Video Streaming in Multicast Networks

by

Xinyu Huang

A thesis  
presented to the University of Waterloo  
in fulfillment of the  
thesis requirement for the degree of  
Doctor of Philosophy  
in  
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2025

© Xinyu Huang 2025

## **Examining Committee Membership**

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:      Vojislav Misic  
Professor  
Department of Computer Science  
Toronto Metropolitan University

Supervisor:              Xuemin (Sherman) Shen  
University Professor  
Department of Electrical and Computer Engineering  
University of Waterloo

Internal Member:        Zhou Wang  
Professor  
Department of Electrical and Computer Engineering  
University of Waterloo

Internal Member:        Kshirasagar Naik  
Professor  
Department of Electrical and Computer Engineering  
University of Waterloo

Internal-External Member: Yaoliang Yu  
Associate Professor  
David R. Cheriton School of Computer Science  
University of Waterloo

### **Author's Declaration**

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

## Abstract

As fifth-generation (5G) networks approach maturity and widespread deployment, both industry and academia are turning their attention to sixth-generation (6G) networks. It is anticipated that 6G networks will support an unprecedented diversity of services with heterogeneous user requirements, accelerating the shift from service-oriented to experience-centric resource management. Among these services, short video streaming has become one of the majority of users' daily mobile traffic consumptions due to its highly engaging content, but this also leads to substantial traffic increase, especially in densely populated areas. Considering the popularity-based and user similarity-driven recommendation principles in short video platforms, multicast transmission over the air can effectively relieve traffic pressure by delivering the same video data to a group of users with similar characteristics and locations. Quality of experience (QoE), as a subjective performance metric in experience-centric resource management, can reflect the user satisfaction level on multicast short video streaming, which usually consists of rebuffer time, video quality, and video quality variation. To achieve experience-centric resource management, digital agent (DA), as a cutting-edge technology in 6G networks, owns advanced status emulation, data analytics, and decision-making capabilities, which can perceive network dynamics, abstract hidden behavior patterns or QoE models, and solve complex optimization problems. The interesting issue is maximizing user QoE in multicast short video streaming under limited radio and computing resources within dynamic network environments. However, the main technical challenges are: (1) how DAs abstract user swipe behavior patterns for large-timescale resource reservation to enhance resource utilization and improve long-term user QoE; (2) how DAs characterize multicast buffer dynamics for real-time resource allocation to alleviate buffer length overestimation and improve real-time user QoE; (3) how to adaptively select appropriate DA models to assist resource management and timely update them to further improve user QoE.

In this thesis, we develop an efficient DA-based resource management framework to enhance user QoE for multicast short video streaming, including swipe behavior-aware resource reservation, multicast buffer-aware resource allocation, and network dynamics-aware DA management. First, we propose a DA-based resource reservation scheme by considering dynamic user swipe behaviors to enhance resource utilization and large-timescale user QoE. Particularly, user DAs are constructed for individual users, which store users' historical data for updating multicast groups and abstracting useful information. The swipe probability distributions and recommended video lists are abstracted from user DAs to predict bandwidth and computing resource demands. Parameterized sigmoid functions are leveraged to characterize multicast groups' user QoE. A joint non-convex bandwidth

and computing resource reservation problem is formulated and transformed into a convex piecewise problem by utilizing a tangent function to approximately substitute the concave part. A low-complexity scheduling algorithm is developed to find the optimal resource reservation decisions. Simulation results based on the real-world dataset demonstrate that the proposed scheme outperforms benchmark schemes in terms of user QoE and resource utilization. Second, we propose a DA-based resource allocation scheme by considering multicast buffer dynamics to enhance real-time user QoE. In specific, user statuses emulated by DAs are utilized to estimate the transmission capabilities and watching probability distributions of sub-multicast groups for adaptive segment buffering. The sub-multicast groups' buffers are aligned to the unique virtual buffers managed by DAs for fine-grained buffer updates. A multicast QoE model consisting of multicast rebuffer time, video quality, and quality variation is developed by considering the mutual influence of segment buffering among sub-multicast groups. A joint optimization problem of segment version selection and slot division is formulated to maximize user QoE. To efficiently solve the problem, a data-model-driven algorithm is proposed by integrating a convex optimization method and a deep reinforcement learning (DRL) algorithm. Simulation results based on the real-world dataset demonstrate that the proposed DA-based resource allocation scheme outperforms benchmark schemes in terms of user QoE improvement. Third, we develop an adaptive DA-based resource management scheme to enhance long-term user QoE. Particularly, DAs consist of user status data and data-based models, which can update multicast groups and abstract user swipe features. An adaptive DA management mechanism for DA data processing model selection and update is developed to adapt to user status dynamics. A fine-grained QoE model is established by considering the impact of resource constraints and DA model accuracy. A joint optimization problem of bandwidth and computing resource management is formulated to maximize long-term user QoE. To efficiently solve this problem, a diffusion-based DRL algorithm is proposed, which utilizes the denoising technique to improve the action exploration capabilities of DRL. Simulation results based on a real-world dataset demonstrate that the proposed adaptive DA-based resource management scheme outperforms benchmark schemes in terms of user QoE, with improvements of 18.4% and 20.5% under low and high user dynamics, respectively.

In summary, we have investigated DA-based radio and computing resource management from the perspectives of large-timescale resource reservation, real-time resource allocation, and adaptive DA management. The proposed approaches and theoretical results provide valuable insights and practical guidelines for experience-centric resource management in future 6G networks.

## Acknowledgements

Pursuing a PhD has been one of the most challenging yet rewarding journeys of my life, and I am deeply grateful to those who have supported, guided, and encouraged me throughout this transformative experience.

First and foremost, I would like to express my heartfelt gratitude to my supervisor, Professor Xuemin (Sherman) Shen. His insightful guidance, rigorous academic standards, and continuous encouragement have been instrumental to my growth as a researcher. He always gave me the freedom to explore my ideas and guided me in the right direction when I needed help. His patience and dedication set a great example for me, and his belief in my potential gave me the confidence to keep moving forward, even during difficult times. Meanwhile, I would like to express my gratitude to Professor Weihua Zhuang for all her help in my PhD study. Her insightful thoughts and valuable instructions have greatly improved my research skills and helped me develop a rigorous academic attitude.

I am sincerely thankful to the members of my doctoral committee—Professor Zhou Wang, Professor Kshirasagar Naik, and Professor Yaoliang Yu—for their invaluable time, thoughtful feedback, and critical insights. Their suggestions have greatly enriched the depth and clarity of my research.

I also wish to acknowledge my labmates in BBCR, such as Dr. Wen Wu, Dr. Cheng Huang, Dr. Conghao Zhou, Dr. Mushu Li, Dr. Huaqing Wu, Dr. Liang Xue, Dr. Dongxiao Liu, Dr. Mingcheng He, Dr. Xiaodan Shao, Dr. Yilong Hui, Yingying Pei, Shisheng Hu, Lingshuang Liu, Xue Qin, Yixiao Zhang, Zhixuan Tang, and Yiren Zhou, at the University of Waterloo. Thank you for the engaging discussions, collaborative spirit, and all shared moments that made the lab feel like a second home. The challenges we faced were made easier and even enjoyable because of your support and companionship.

A very special thank goes to my fiancée, Yueyue Wang, whom I met and fell in love with at the University of Waterloo. Thank you for walking beside me through the highs and lows of this journey, for your understanding during the long nights and stressful deadlines, and for filling my life with warmth, joy, and unwavering support. Your love has been my greatest comfort and motivation, and I am endlessly grateful for everything we have shared.

Finally, I owe my deepest and most profound gratitude to my parents, Aiqin Kai and Bin Huang. Your unconditional love, unwavering support, and enduring faith in me have been my strongest source of strength.

Xinyu Huang  
Apr. 26, 2025  
*Waterloo, Ontario, Canada*

## Dedication

*This PhD thesis is dedicated to my beloved parents, Aiqin Kai and Bin Huang.*

# Table of Contents

|                                                                                                |            |
|------------------------------------------------------------------------------------------------|------------|
| <b>List of Figures</b>                                                                         | <b>xii</b> |
| <b>List of Tables</b>                                                                          | <b>xiv</b> |
| <b>List of Abbreviations</b>                                                                   | <b>xv</b>  |
| <b>1 Introduction</b>                                                                          | <b>1</b>   |
| 1.1 Digital Agents in 6G . . . . .                                                             | 1          |
| 1.2 Multicast Networks for Short Video Streaming . . . . .                                     | 3          |
| 1.3 Overview of Digital Agent-Empowered Multicast Networks for Short Video Streaming . . . . . | 4          |
| 1.4 Research Motivations and Challenges . . . . .                                              | 7          |
| 1.4.1 Resource Wastage and Low User Satisfaction in Resource Reservation                       | 7          |
| 1.4.2 Overestimated Multicast Buffer Length in Resource Allocation . . .                       | 7          |
| 1.4.3 Massive Network Data and Data Drift in Digital Agent Management                          | 8          |
| 1.5 Our Approaches and Contributions . . . . .                                                 | 9          |
| 1.5.1 Digital Agent-Based Behavior Analysis for Resource Reservation . .                       | 9          |
| 1.5.2 Digital Agent-Based Buffer Management for Resource Allocation . .                        | 10         |
| 1.5.3 Adaptive Digital Agent Model Selection and Update . . . . .                              | 10         |
| 1.6 Thesis Outline . . . . .                                                                   | 11         |

|          |                                                                                                |           |
|----------|------------------------------------------------------------------------------------------------|-----------|
| <b>2</b> | <b>Background and Literature Review</b>                                                        | <b>12</b> |
| 2.1      | Multicast Short Video Streaming . . . . .                                                      | 12        |
| 2.2      | Digital Agent-Based Resource Management . . . . .                                              | 14        |
| 2.2.1    | Large-Timescale Resource Reservation . . . . .                                                 | 14        |
| 2.2.2    | Small-Timescale Resource Allocation . . . . .                                                  | 16        |
| 2.3      | Digital Agent Management . . . . .                                                             | 18        |
| 2.3.1    | Digital Agent Model Selection . . . . .                                                        | 18        |
| 2.3.2    | Digital Agent Model Update . . . . .                                                           | 19        |
| 2.4      | Summary . . . . .                                                                              | 20        |
| <b>3</b> | <b>Digital Agent-Based Resource Reservation for QoE-Driven Multicast Short Video Streaming</b> | <b>21</b> |
| 3.1      | Background and Motivations . . . . .                                                           | 21        |
| 3.2      | System Model . . . . .                                                                         | 24        |
| 3.2.1    | Digital Agent-Based Resource Reservation Framework . . . . .                                   | 25        |
| 3.2.2    | Digital Agent Construction . . . . .                                                           | 26        |
| 3.2.3    | Digital Agent-Based Multicast Group Management . . . . .                                       | 27        |
| 3.2.4    | User Satisfaction-Based QoE Model . . . . .                                                    | 32        |
| 3.3      | Problem Formulation . . . . .                                                                  | 34        |
| 3.4      | Proposed Solution . . . . .                                                                    | 35        |
| 3.4.1    | Problem Transformation . . . . .                                                               | 36        |
| 3.4.2    | Fast Resource Reservation Algorithm . . . . .                                                  | 38        |
| 3.4.3    | Computational Complexity Analysis . . . . .                                                    | 40        |
| 3.5      | Simulation Results . . . . .                                                                   | 40        |
| 3.5.1    | Simulation Setup . . . . .                                                                     | 40        |
| 3.5.2    | Clustering Performance Evaluation . . . . .                                                    | 43        |
| 3.5.3    | System Utility Performance Evaluation . . . . .                                                | 46        |
| 3.6      | Summary . . . . .                                                                              | 46        |

|          |                                                                                               |           |
|----------|-----------------------------------------------------------------------------------------------|-----------|
| <b>4</b> | <b>Digital Agent-Based Resource Allocation for QoE-Driven Multicast Short Video Streaming</b> | <b>49</b> |
| 4.1      | Background and Motivations . . . . .                                                          | 49        |
| 4.2      | System Model . . . . .                                                                        | 52        |
| 4.2.1    | Digital Agent-Based Resource Allocation Framework . . . . .                                   | 53        |
| 4.2.2    | Digital Agent Construction . . . . .                                                          | 55        |
| 4.2.3    | Digital Agent-Based Segment Buffering . . . . .                                               | 56        |
| 4.2.4    | Digital Agent-Based Buffer Update . . . . .                                                   | 60        |
| 4.2.5    | Multicast QoE Model . . . . .                                                                 | 61        |
| 4.3      | Problem Formulation . . . . .                                                                 | 63        |
| 4.4      | Proposed Solution . . . . .                                                                   | 63        |
| 4.4.1    | Deep Reinforcement Learning-Based Segment Version Selection . .                               | 64        |
| 4.4.2    | Optimization-Based Slot Division . . . . .                                                    | 66        |
| 4.4.3    | Computational Complexity Analysis . . . . .                                                   | 68        |
| 4.5      | Simulation Results . . . . .                                                                  | 70        |
| 4.5.1    | Simulation Setup . . . . .                                                                    | 70        |
| 4.5.2    | Convergence Analysis . . . . .                                                                | 73        |
| 4.5.3    | Performance Evaluation of QoE Under Different Settings . . . . .                              | 73        |
| 4.6      | Summary . . . . .                                                                             | 76        |
| <b>5</b> | <b>Adaptive Digital Agent Management for QoE-Driven Multicast Short Video Streaming</b>       | <b>77</b> |
| 5.1      | Background and Motivations . . . . .                                                          | 77        |
| 5.2      | System Model . . . . .                                                                        | 80        |
| 5.2.1    | Adaptive Digital Agent-Based Resource Management Framework .                                  | 80        |
| 5.2.2    | Digital Agent Construction . . . . .                                                          | 83        |
| 5.2.3    | Digital Agent Data Processing . . . . .                                                       | 84        |
| 5.2.4    | Digital Agent Model Update . . . . .                                                          | 86        |

|          |                                                                       |            |
|----------|-----------------------------------------------------------------------|------------|
| 5.2.5    | Adaptive Digital Agent Management . . . . .                           | 89         |
| 5.2.6    | Multicast Buffer Control . . . . .                                    | 91         |
| 5.2.7    | Fine-Grained QoE Model . . . . .                                      | 92         |
| 5.3      | Problem Formulation . . . . .                                         | 94         |
| 5.4      | Proposed Solution . . . . .                                           | 95         |
| 5.4.1    | Problem Transformation . . . . .                                      | 95         |
| 5.4.2    | Diffusion-Based Deep Reinforcement Learning Algorithm . . . . .       | 96         |
| 5.4.3    | Time Complexity Analysis . . . . .                                    | 100        |
| 5.5      | Simulation Results . . . . .                                          | 100        |
| 5.5.1    | Simulation Setup . . . . .                                            | 100        |
| 5.5.2    | Convergence Analysis . . . . .                                        | 104        |
| 5.5.3    | The Influence of DA Data Processing and Model Update on QoE . . . . . | 104        |
| 5.5.4    | Performance Evaluation of QoE Components . . . . .                    | 105        |
| 5.5.5    | QoE Evaluation Under Different Network Settings . . . . .             | 107        |
| 5.6      | Summary . . . . .                                                     | 109        |
| <b>6</b> | <b>Conclusions and Future Works</b>                                   | <b>110</b> |
| 6.1      | Main Research Contributions . . . . .                                 | 110        |
| 6.2      | Future Research Directions . . . . .                                  | 112        |
|          | <b>References</b>                                                     | <b>114</b> |
|          | <b>Appendix A</b>                                                     | <b>127</b> |

# List of Figures

|     |                                                                                                        |    |
|-----|--------------------------------------------------------------------------------------------------------|----|
| 1.1 | An illustration of digital agent. . . . .                                                              | 2  |
| 1.2 | An illustration of multicast transmission for short video streaming. . . . .                           | 3  |
| 1.3 | DA-empowered multicast network framework for short video streaming. . . . .                            | 5  |
| 3.1 | DA-based resource reservation framework for multicast short video streaming. . . . .                   | 25 |
| 3.2 | User DA data collection procedure. . . . .                                                             | 26 |
| 3.3 | The improved Double Deep Q-Network (DDQN) architecture for clustering number determination. . . . .    | 28 |
| 3.4 | User satisfaction over different buffer lengths and rebuffering time. . . . .                          | 32 |
| 3.5 | The simulation scene, where BSs and users are represented by red and gray icons, respectively. . . . . | 41 |
| 3.6 | Clustering number comparison. . . . .                                                                  | 44 |
| 3.7 | User density in each multicast group. . . . .                                                          | 44 |
| 3.8 | A comparison between DDQN training with autoencoder and without autoencoder. . . . .                   | 45 |
| 3.9 | The performance comparison of each component in the system utility. . . . .                            | 47 |
| 4.1 | DA-based resource allocation framework for multicast short video streaming. . . . .                    | 53 |
| 4.2 | Proposed DA-assisted multicast short video streaming workflow. . . . .                                 | 54 |
| 4.3 | Watching probability analysis. . . . .                                                                 | 57 |
| 4.4 | Multicast transmission mechanism among sub-multicast groups. . . . .                                   | 59 |
| 4.5 | DA-assisted sub-multicast group's buffer management. . . . .                                           | 60 |

|      |                                                                                                           |     |
|------|-----------------------------------------------------------------------------------------------------------|-----|
| 4.6  | The proposed data-model-driven algorithm structure. . . . .                                               | 64  |
| 4.7  | The constructed BDQN architecture. . . . .                                                                | 65  |
| 4.8  | The simulation scene, where BSs and users are represented by red and gray<br>icons, respectively. . . . . | 71  |
| 4.9  | Convergence performance of proposed DA-based network management scheme.                                   | 73  |
| 4.10 | QoE comparison under different users, bandwidths, and computing capacities.                               | 74  |
| 4.11 | QoE comparison under different sensitivity degrees. . . . .                                               | 75  |
| 5.1  | Adaptive DA-based resource management framework. . . . .                                                  | 81  |
| 5.2  | DA data processing procedure. . . . .                                                                     | 84  |
| 5.3  | Data fitting on DA data processing accuracy. . . . .                                                      | 86  |
| 5.4  | The procedure of DA model update process. . . . .                                                         | 87  |
| 5.5  | The procedure of adaptive DTP and DTU. . . . .                                                            | 89  |
| 5.6  | The proposed diffusion-based TD3 algorithm framework. . . . .                                             | 96  |
| 5.7  | The simulation scene, where BSs and users are represented by red and gray<br>icons, respectively. . . . . | 101 |
| 5.8  | Convergence performance. . . . .                                                                          | 104 |
| 5.9  | Real-time QoE in each step. . . . .                                                                       | 105 |
| 5.10 | The performance comparison of different QoE components. . . . .                                           | 106 |
| 5.11 | The performance of QoE under different settings. . . . .                                                  | 108 |

# List of Tables

|     |                                                           |     |
|-----|-----------------------------------------------------------|-----|
| 3.1 | Simulation parameters . . . . .                           | 40  |
| 3.2 | Autoencoder parameters . . . . .                          | 43  |
| 3.3 | DDQN parameters . . . . .                                 | 43  |
| 4.1 | Notations and definitions . . . . .                       | 55  |
| 4.2 | Simulation parameters . . . . .                           | 70  |
| 4.3 | BDQN parameters . . . . .                                 | 71  |
| 5.1 | Notations and definitions . . . . .                       | 83  |
| 5.2 | Communication network parameters . . . . .                | 102 |
| 5.3 | DFFTD3 algorithm parameters . . . . .                     | 103 |
| 5.4 | QoE performance under different user velocities . . . . . | 107 |

# List of Abbreviations

**6G** Sixth-generation 1, 3, 14, 18

**A2C** Actor-to-Critic 15

**AI** Artificial Intelligence 18, 19

**AP** Access Point 53, 55, 56, 59

**ARIMA** Autoregressive Integrated Moving Average 20

**BDQN** Branching Dueling Q-Network 64

**BS** Base Station 25, 80, 81

**CDF** Cumulative Distribution Function 105, 107

**CPU** Central Processing Unit 15, 18

**CSI** Channel State Information 3, 15

**DA** Digital Agent 1–6, 10, 12, 14–18, 21, 22, 34, 50–52, 54, 56, 78, 80, 82–87, 90, 91, 110

**DAP** DA Data Processing 79, 82–85, 89–93, 102–104, 109

**DAU** DA Model Update 80, 82, 83, 87, 89–91, 93, 95, 99, 102–105, 109

**DDPG** Deep Deterministic Policy Gradient 15

**DDQN** Double Deep Q-Network xii, 14, 23, 28, 82, 84, 85, 87

**DFFTD3** Diffusion-Based TD3 98

**DRL** Deep Reinforcement Learning 9–11, 15–17, 27, 52, 77, 79, 80, 111

**DT** Digital Twin 1

**ES** Edge Server 25

**GAI** Generative AI 78

**GNN** Graph Neural Network 16, 20

**IoT** Internet of Things 14, 15, 19

**ISAC** Integrated Sensing and Communication 14

**LSTM** Long Short-Term Memory 14, 20, 55, 82, 84, 86–90

**MDP** Markov Decision Process 15, 95

**MIMO** Multiple-Input and Multiple-Output 17

**MINLP** Mixed Integer Nonlinear Programming 15

**MSE** Mean Square Error 98

**NFV** Network Function Virtualization 14

**NOMA** Non-Orthogonal Multiple Access 15

**QoE** Quality of Experience 2–5, 8, 16, 33, 50–52, 56, 63, 78, 94–96

**QoS** Quality of Service 22

**RAN** Radio Access Network 3, 4

**RMSE** Root Mean Square Error 86

**RSU** Roadside Unit 15

**SDN** Software Defined Network 14

**SLA** Service Level Agreement 20

**SLSQP** Sequential Least Squares Quadratic Programming 64, 66–68

**SNR** Signal-to-Noise Ratio 13

**SSIM** Structural Similarity Index Measure 52, 62, 93

**SVC** Scalable Video Coding 13, 58, 82, 92

**TD3** Twin Delayed Deep Deterministic Policy Gradient 96, 97

**TSN** Time-Sensitive Networking 19

**URLLC** Ultra Reliable Low Latency Communications 15, 16

**VM** Virtual Machine 25, 33

**VNF** Virtualized Network Function 14

# Chapter 1

## Introduction

### 1.1 Digital Agents in 6G

Digital Agents (DAs) represent an advanced paradigm in network management, by serving as intelligent proxies for physical entities within Sixth-generation (6G) communication networks. Existing automation capabilities in communication networks, such as analytics services and decision-making functions, typically operate under the assumption that information about network behaviors is readily available within these functions [1]. To alleviate this issue, DAs are anticipated to provide new information on how the network behaves or will behave, enabling a deeper understanding and modeling of network dynamics. Compared with Digital Twins (DTs), which primarily emphasize constructing real-time digital replicas of physical entities to facilitate the monitoring and perception of dynamic network states [2], DAs are more focused on intelligent reasoning and decision-making. DTs usually enable real-time awareness of physical networks for timely resource allocation [3], while DAs embed cognitive capabilities such as prediction, analysis, and autonomous decision-making. In this sense, DAs can be regarded as goal-driven extensions of DTs. By integrating advanced data analysis and processing methods, DAs can provide enhanced capabilities for network performance optimization through three primary functions, including accurate network behavior emulation, intrinsic feature extraction, and customized network decision-making [4], as shown in Fig. 1.1.

Firstly, DAs enable precise emulation of network behaviors, which can reflect potential real-world network status. Network behaviors consist of user behaviors, such as user interaction frequency, locations, mobility speed, preferences, and network operational behaviors, such as load balance, topology configuration, and data routing. Through detailed

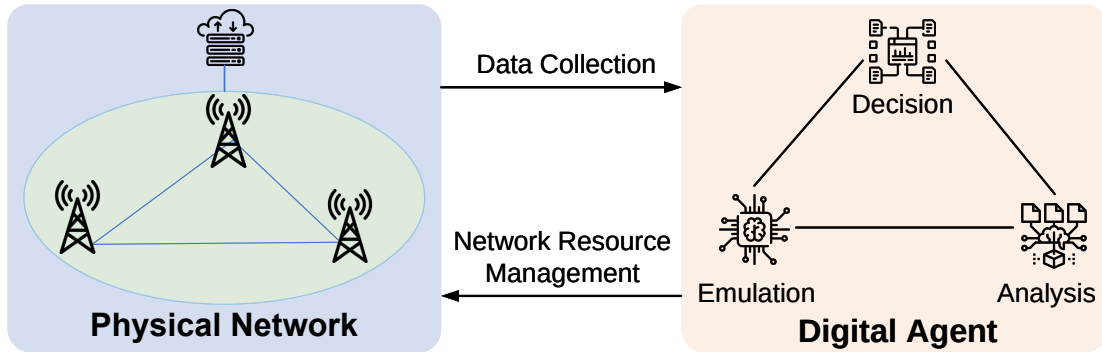


Figure 1.1: An illustration of digital agent.

emulation and predictive modeling, DAs anticipate the network’s responses to varying configurations and operational strategies without disrupting the actual infrastructure. This capability allows network operators to validate configurations proactively and adjust network management policies. Furthermore, data collection costs from physical networks can also be effectively reduced by generating network status data through data analysis of spatiotemporal correlations in historical data.

Secondly, DAs facilitate hidden feature extraction by analyzing extensive network data. Since DA data are usually high-dimensional and time-series, such as varying channel conditions and user locations in a high-density network, it is hard to directly use them to guide efficient network management [5]. Therefore, accurate and efficient data feature abstraction is necessary. For instance, autoencoders can encode the high-dimensional DA status data into a low-dimensional representation to capture the network traffic patterns and user behavioral patterns. Furthermore, transformer models can accurately analyze spatiotemporal correlations from DA data through its advanced attention mechanism [6]. The abstracted features can simplify the network management problem and provide tailored network management information.

Thirdly, DAs provide customized network decision-making capabilities tailored to user-specific requirements. By integrating predictive analytics with Quality of Experience (QoE) models, DAs can support personalized network management decisions to improve user satisfaction. For example, in the network orchestration process, users can be clustered into different groups based on their QoE model structures, where tailored resource scheduling algorithms are designed to solve specific QoE optimization problems [4]. Since the network management problem is usually complex and non-convex, it is difficult to directly use model-based or data-based methods to solve the problem and obtain a near-optimal solution. Therefore, data-model-driven algorithms can be developed in the DAs to improve

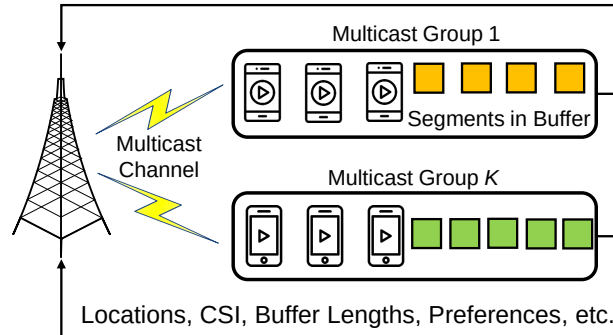


Figure 1.2: An illustration of multicast transmission for short video streaming.

QoE.

In summary, DAs in 6G networks offer a transformative solution that bridges the gap between network dynamics modeling and customized resource management. Through advanced behavior emulation, hidden feature extraction, and tailored decision-making, DAs are expected to substantially enhance network performance and ultimately shape the future landscape of 6G networks.

## 1.2 Multicast Networks for Short Video Streaming

With the explosive growth of short video applications, efficient streaming methods are critical to ensure low latency and optimal bandwidth utilization. Multicast transmission within Radio Access Networks (RANs) has emerged as a promising solution for addressing the challenges inherent in delivering short video content to large-scale audiences simultaneously [7]. As shown in Fig. 1.2, multicast transmission in RAN enables a single stream of data to be efficiently delivered to multiple user devices without replicating individual data packets for each group at the air interface. The group's information, such as locations, Channel State Information (CSI), buffer lengths, and preferences, are usually reported to base stations to adjust multicast transmission strategies. This significantly enhances radio resource utilization compared to traditional unicast streaming, where each user requires an independent wireless connection. In the context of short video streaming, characterized by brief durations, high-frequency requests, and bursty traffic patterns [8], multicast solutions at the radio access level provide substantial benefits in terms of scalability, spectrum efficiency, and reduced latency.

However, deploying multicast mechanisms in the RANs for short videos introduces

several challenges. Highly dynamic user behaviors observed in platforms, such as TikTok or Instagram Reels, where users frequently initiate, pause, and switch videos, necessitate sophisticated RAN multicast management strategies. These strategies must quickly adapt to fluctuating user demands and rapidly changing wireless channel conditions, to ensure a seamless streaming experience. Adaptive RAN multicast techniques, such as dynamic multicast group management [9], radio resource scheduling [10], and proactive caching at base stations [11], have been proposed and evaluated in recent studies.

Recent research has highlighted adaptive multicast frameworks within RANs that dynamically adjust multicast groups based on predictive analytics and real-time user behavior modeling. Machine learning algorithms have been utilized to anticipate content popularity, user access patterns, and radio channel quality, which can proactively cache and efficiently distribute popular short videos directly at the base station level [12, 13]. This approach achieves lower start-up delays and significantly enhances QoE. Furthermore, integrating multicast strategies in RANs with edge computing infrastructure significantly enhances performance by caching popular content at network edge nodes or base stations, which can reduce latency and resource consumption and alleviate congestion in densely populated areas. By leveraging edge multicast capabilities, network operators can efficiently serve high-density user clusters, such as urban hotspots and event venues, where concurrent viewing of trending short videos is common.

In summary, multicast networks offer a scalable and efficient approach to short video distribution, particularly in scenarios involving large user populations. The continued advancement in adaptive RAN multicast technologies, combined with predictive analytics, dynamic group management, and edge computing, presents the substantial potential to optimize short video streaming for user experience improvement.

### 1.3 Overview of Digital Agent-Empowered Multicast Networks for Short Video Streaming

As shown in Fig. 1.3, we propose a DA-empowered multicast network framework for short video streaming, which integrates content providers, core networks, cloud and edge servers, multicast groups, and DAs in a scalable network infrastructure. Centralized and local controllers are responsible for resource reservation and allocation by leveraging DA capabilities to improve network performance and user experience. In the proposed framework, content from various popular providers, including social media platforms and streaming services, is initially stored in cloud and edge servers. The content delivery network utilizes multicast

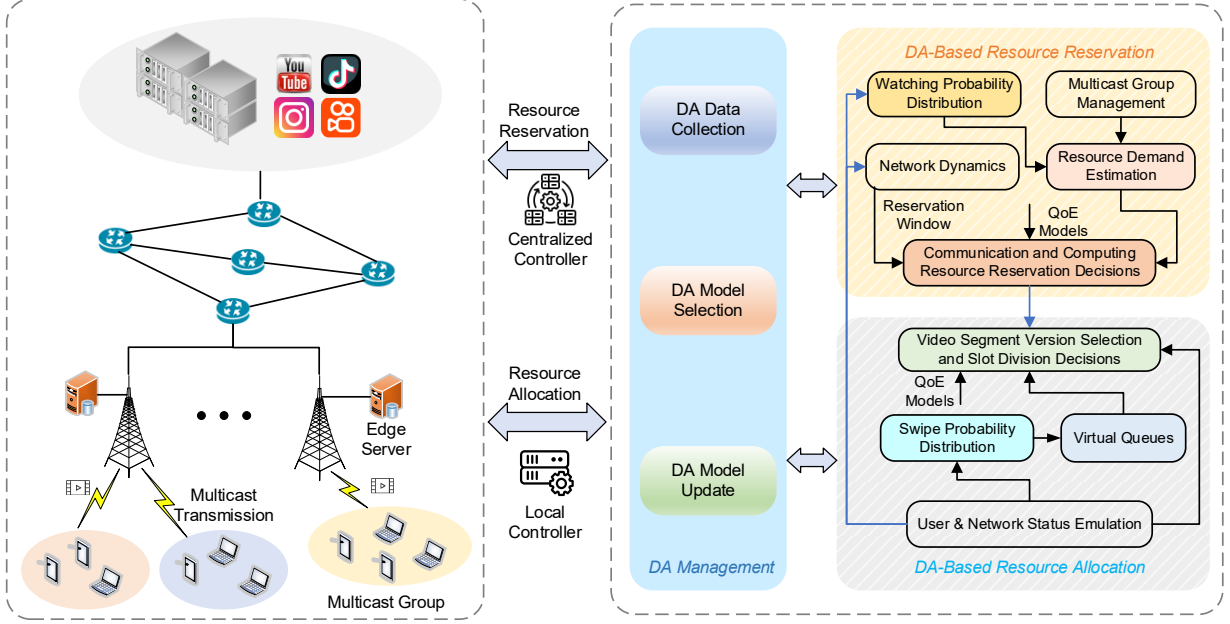


Figure 1.3: DA-empowered multicast network framework for short video streaming.

techniques to transmit video data from base stations directly to multicast groups, which can significantly reduce network redundancy and bandwidth consumption.

Effective communication and computing resource management in multicast networks involves two critical mechanisms: resource reservation and resource allocation, each operating over distinct timescales. Resource reservation, executed over a large timescale, aims to reserve resources based on long-term user demand patterns, content popularity, and network dynamics. By proactively reserving sufficient communication bandwidth and computing resources, network operators can guarantee service quality and overall user QoE, particularly during peak demand periods.

At a small timescale, resource allocation dynamically distributes available resources to individual multicast groups or users, which can react swiftly to short-term fluctuations in network conditions and user requests. Efficient resource allocation requires real-time decision-making mechanisms that consider current network states, instantaneous user behaviors, and immediate performance metrics. Adaptive resource scheduling algorithms and predictive analytics can be utilized to maintain smooth and high-definition video streaming performance.

DAs play a critical role in optimizing network resource management through three

core management components, i.e., DA management, DA-based resource reservation, and DA-based resource allocation.

- DA management mainly involves efficient data collection, appropriate data processing model selection, and timely model updates. For example, advanced data emulation methods can be developed to aware user status in real time on the network side and effectively reduce data collection frequencies. Since DA data are usually multidimensional and dynamic, adaptively selecting an appropriate data processing model to abstract network features can facilitate tailored resource management. Due to data drift existing in DA data, DA models need to be timely updated to guarantee accuracy. By developing efficient data collection, data processing, and model update methods, DAs can provide more useful and accurate information for network resource management.
- The DA-based resource allocation can emulate user status, such as locations, swipe timestamps and buffer lengths, and network status, such as channel conditions and available bandwidth and computing resources. Swipe probability distribution is an important feature that needs to be abstracted, which can help DAs capture user buffer dynamics to maintain accurate virtual queues. With the virtual queues and user-specific QoE models, the network resource allocation problem of video segment version selection and slot division can be effectively solved.
- The DA-based resource reservation can acquire user status and network status information from the DA-based resource allocation module. The acquired information can be utilized to abstract watching probability distribution and network dynamics. The front is integrated with clustered multicast groups to estimate resource demands, while the latter can be used to determine the resource reservation window size. Based on the estimated resource demands, resource reservation window size, and QoE models, tailored resource reservation algorithms can be developed to make communication and computing resource reservation decisions.

Overall, DA-empowered multicast networks can effectively improve multicast efficiency, resource utilization, and user QoE, which can satisfy the growing demands of short video streaming services.

## 1.4 Research Motivations and Challenges

In this section, we explore research motivations and challenges in multicast networks for short video streaming from the following three aspects.

### 1.4.1 Resource Wastage and Low User Satisfaction in Resource Reservation

Since user status usually differs from one user to another one, inaccurate user clustering can lead to low transmission rates and resource wastage due to various channel conditions and frequent swipe behaviors on uninterested video content, respectively. Therefore, it is essential to develop an accurate and efficient user clustering scheme. Furthermore, due to differentiated user satisfaction with service quality, multicast groups may have different levels of satisfaction with the reserved bandwidth and computing resources. Therefore, it is critical to establish the user QoE model to characterize the impact of resource reservation on each multicast group's satisfaction.

Designing an effective resource reservation scheme in multicast networks for short video streaming needs to address the following challenges: 1) incorporating swipe behaviors in resource reservation decision-making; 2) establishing the mathematical model between multicast groups' user satisfaction and reserved resources. Specifically, users' swipe behaviors are stochastic and spatiotemporally varied, which are difficult to be predicted in real time. Therefore, how to conduct effective data abstraction to obtain the distilled swipe feature and utilize it to predict bandwidth and computing resource demands is challenging. Furthermore, due to the dynamics of users' personalized preferences and sensitivities to service quality, the same amount of reserved resources to one multicast group can lead to different user satisfaction at different time. Therefore, how to establish and update each multicast group's QoE model in each resource reservation window is challenging.

### 1.4.2 Overestimated Multicast Buffer Length in Resource Allocation

Due to users' stochastic swipe behaviors, users' viewing sequences are usually nonsequential. The existing sequential buffering principle can cause the segments to be swiped to not being buffered in time, resulting in playback lags. Moreover, users' buffer lengths are usually overestimated due to multicast segment buffering, which can bring inaccurate

rebuffering time estimation. Therefore, it is essential to develop an efficient segment buffering scheme that adapts well to users' swipe behaviors. Furthermore, user QoE model is typically composed of multiple factors. Due to the impact of multicast transmission and segment buffering priority, different QoE factors among sub-multicast groups influence each other, collectively impacting the multicast group QoE. Therefore, it is paramount to establish a QoE model specifically tailored for multicast short video streaming. Finally, since network management is usually a multivariable decision-making problem, such as the segment version selection and resource scheduling, the interplay among the variables makes the optimization problem complex and difficult to solve. Therefore, how to design an efficient algorithm to solve it for users' high QoE is important.

Designing an efficient resource allocation scheme in multicast networks for short video streaming needs to address the following challenges: 1) incorporating the impact of users' swipe behaviors in multicast segment buffering; 2) establishing an accurate multicast QoE model; 3) designing an efficient algorithm to solve the complex multi-variable decision-making problem. Specifically, users' swipe behaviors are stochastic and spatiotemporally varied, which are difficult to accurately predict in real time. Therefore, how to conduct effective data abstraction to obtain the distilled swipe feature and utilize it to facilitate accurate multicast segment buffering is challenging. Furthermore, since the lagging sub-multicast group can still receive segments from other leading sub-multicast groups in its scheduling slot, this interactivity results in complex QoE estimation. Therefore, how to characterize the impact of multicast segment buffering among sub-multicast groups on the QoE model is challenging. Finally, since the resource allocation problem is usually a mixed-integer non-convex problem, directly using model-based or data-driven algorithms can lead to a loss in the system performance. Therefore, how to ingeniously combine the advantages of model-based and data-driven algorithms to efficiently solve the formulated problem is challenging.

### **1.4.3 Massive Network Data and Data Drift in Digital Agent Management**

Highly dynamic user behaviors, characterized by stochasticity and notable spatiotemporal variations, have caused substantial difficulties in efficiently processing and accurately modeling user behaviors. Therefore, it is essential to select appropriate DA data processing models and timely update them for accurate and efficient user behavior data processing. Furthermore, conventional general QoE models, which inadequately characterize buffer dynamics and rebuffer time, can lead to frequent playback interruptions and degrade the user

experience. Therefore, constructing a fine-grained QoE model that accurately captures the interplay between resource constraints, buffer management, and multicast transmission delays thus becomes essential. Finally, the inherent complexity of DA management, typically formulated as a mixed-integer nonlinear programming problem, often exceeds the capability of traditional Deep Reinforcement Learning (DRL) algorithms, which struggle in dynamic environments with varying user behaviors. Therefore, developing an adaptive DA management approach that leverages improved DRL techniques is necessary.

Designing an efficient DA management scheme for short video streaming in multicast networks faces the following challenges: 1) processing highly dynamic user behavior data; 2) constructing a fine-grained QoE model; 3) developing an efficient algorithm for the mixed-integer nonlinear programming problem. Specifically, due to the stochastic nature and spatiotemporal differences in users' swipe behaviors, designing an efficient DA management mechanism to characterize and process the highly dynamic and multi-dimensional user behavior data is challenging. Furthermore, using a general QoE model that lacks accurate buffer control and rebuffer time estimation can cause frequent playback lags. It is challenging to construct a fine-grained QoE model by considering the effect of resource constraints on buffer update and DA model accuracy on multicast transmission delay estimation. Finally, since the DA management problem is usually complex mixed-integer nonlinear programming, traditional DRL algorithms usually struggle with highly dynamic environments due to spatiotemporal variation in user behaviors, leading to suboptimal performance. Therefore, an improved DRL algorithm should be developed to solve the complex DA management problem.

## 1.5 Our Approaches and Contributions

In this thesis, we investigate the DA-based resource management schemes in multicast networks for short video streaming to tackle the above-mentioned challenges. The objective is to develop efficient resource management strategies to improve user QoE within the limited communication and computing resources. In specific, we focus on the following three research topics.

### 1.5.1 Digital Agent-Based Behavior Analysis for Resource Reservation

We investigate a DA-based user behavior analysis scheme for resource reservation, which aims to enhance user QoE with minimal resource consumption. Firstly, we design a DA-

assisted resource reservation framework. Specifically, DAs are constructed for individual users, which store users' historical data for updating multicast groups and abstracting useful information. The swipe probability distributions and recommended video lists are abstracted from user DAs to predict bandwidth and computing resource demands. Parameterized sigmoid functions are leveraged to characterize multicast groups' user satisfaction. Secondly, we formulate a joint non-convex bandwidth and computing resource reservation problem which is transformed into a convex piecewise problem by utilizing a tangent function to approximately substitute the concave part. A low-complexity scheduling algorithm is then developed to find the optimal resource reservation decisions. Simulation results based on the real-world dataset demonstrate that the proposed scheme outperforms benchmark schemes in terms of user QoE and resource consumption.

### **1.5.2 Digital Agent-Based Buffer Management for Resource Allocation**

We propose a DA-based buffer management scheme for resource allocation to enhance user QoE. Firstly, user status emulated by the DA is utilized to estimate the transmission capabilities and watching probability distributions of sub-multicast groups for an adaptive segment buffering. The sub-multicast groups' buffers are aligned to the unique virtual buffers managed by the DA for a fine-grained buffer update. Then, a multicast QoE model consisting of rebuffering time, video quality, and quality variation is developed, by considering the mutual influence of segment buffering among sub-multicast groups. Finally, a joint optimization problem of segment version selection and slot division is formulated to maximize user QoE. To efficiently solve the problem, a data-model-driven algorithm is proposed by integrating a convex optimization method and a DRL algorithm. Simulation results based on the real-world dataset demonstrate that the proposed DA-based resource allocation scheme outperforms benchmark schemes in terms of user QoE improvement.

### **1.5.3 Adaptive Digital Agent Model Selection and Update**

We investigate an adaptive DA management scheme to enhance user QoE. Particularly, DAs consist of user status data and data-based models, which can update multicast groups and abstract users' swipe features. A precise measurement model of DA data processing is constructed to characterize the relationship among DA model size, user dynamics, and user clustering accuracy, which can facilitate flexible DA model selection. The proposed DA management module can adaptively update based on the error and entropy of predicted

user status and user clustering result. Then, a fine-grained QoE model is established by considering the impact of resource constraints and DA model accuracy, leading to accurate buffer control. Finally, a joint optimization problem of resource management is formulated to maximize long-term QoE. To efficiently solve this problem, a diffusion-based DRL algorithm is proposed, which utilizes the denoising technique to improve the action exploration capabilities of DRL. Simulation results based on a real-world dataset demonstrate that the proposed DA-assisted resource management scheme outperforms benchmark schemes in terms of user QoE.

## 1.6 Thesis Outline

The remainder of the thesis is organized as follows: In Chapter 2, we provide a comprehensive review of the state-of-the-art resource management strategies in multicast networks for short video streaming. In Chapter 3, we propose a DA-based resource reservation scheme by incorporating the impact of user swipe behaviors to enhance user QoE. In Chapter 4, we develop a DA-based resource allocation scheme by emulating user buffer dynamics for tailored segment buffering to enhance user QoE. In Chapter 5, we investigate an adaptive DA management scheme to adapt to user behavior dynamics to enhance resource utilization for user QoE enhancement. Finally, we conclude the thesis and discuss future research works in Chapter 6.

# Chapter 2

## Background and Literature Review

In this chapter, we review the existing work from three aspects, i.e., multicast short video streaming, DA-based resource management, and DA management, respectively.

### 2.1 Multicast Short Video Streaming

Multicast streaming over wireless networks has attracted extensive research interest due to its potential to reduce packet error rates and optimize resource allocation. Early investigations highlighted the importance of reliable data transmission in multicast scenarios, where feedback from multiple users can lead to collisions [14]. To address these challenges, researchers introduced various error correction techniques and adapted unicast transmission methods for multicast applications [15]. Moreover, resource allocation has emerged as a critical research area, particularly in wireless environments characterized by limited bandwidth and heterogeneous user demands [16].

Resource management is a cornerstone in the research of multicast short video streaming. Although traditional unicast methods are inefficient for large-scale video delivery, multicast streaming offers significant bandwidth savings while introducing new complexities in resource allocation. A novel resource management framework has been proposed to dynamically optimize resource allocation, which effectively balances trade-offs among cost, efficiency, and user satisfaction. This framework implements optimization algorithms that account for diverse network conditions and user requirements, substantially reducing both bandwidth usage and startup delays [17]. Another approach focuses on loss-tolerant multicast video streaming, which permits controlled packet loss while still maintaining

acceptable user quality. This method reduces dependency on the weakest network links and achieves lower peak Signal-to-Noise Ratio (SNR) degradation compared to existing schemes [18]. Furthermore, optimal resource allocation problems have been formulated with the objective of minimizing bandwidth usage while ensuring target multicast rates, with solutions leveraging iterative algorithms for enhanced computational efficiency [19].

Adaptive streaming techniques have been integrated with multicast strategies to further enhance video delivery. For instance, a multicast adaptive logic has been developed to support multicast constraints at the client side, demonstrating superior performance under multicast conditions while maintaining robust performance in unicast scenarios [20]. Additionally, in-network caching has been proposed to reduce transit traffic by redirecting content requests to edge routers, thereby combining the advantages of content delivery networks with multicast schemes [21]. Recent research has also explored cooperative video edge caching and multicasting approaches based on multi-agent deep reinforcement learning, which minimize transmission counts in multi-cloud environments by integrating caching and multicasting strategies [22, 23].

In cellular networks, multi-connectivity has been investigated as a way to enhance multicast streaming performance. By enabling users to receive content from multiple base stations, multi-connectivity improves both resource utilization and connectivity, particularly for cell-edge users [24]. Empirical results indicate that multi-connectivity can support up to 60% more users than single-connectivity multicast systems under identical resource constraints [25]. Nonetheless, the optimal resource allocation problem for multi-connectivity multicast is NP-hard, which has motivated the development of greedy approximation algorithms with guaranteed approximation factors.

In broadband wireless networks, the combination of Scalable Video Coding (SVC) and adaptive modulation and coding has been employed to optimize multicast streaming. These techniques enhance wireless resource allocation by assigning appropriate modulation and coding schemes to different video layers [26]. A dynamic programming-based algorithm has been proposed to maximize system utility while reducing computational overhead in both single-session and multi-session scenarios [27]. Fair bandwidth allocation is essential for supporting multicast applications that require guaranteed performance, such as online video streaming. To this end, a credit-based multicast fair scheduling algorithm has been developed to achieve low latency and equitable bandwidth distribution by tracking the discrepancy between reserved and actual bandwidth and supporting simultaneous transmission requests across multiple output ports [28].

In summary, the research on multicast short video streaming over wireless networks has significantly advanced our ability to manage resources and maintain high-quality ser-

vice. The integration of adaptive streaming, in-network caching, multi-connectivity, and advanced coding techniques has not only improved network performance but also provided promising directions for addressing emerging challenges in 6G networks.

## 2.2 Digital Agent-Based Resource Management

In this section, we review the existing DA-based resource management from two aspects: large-timescale resource reservation and small-timescale resource allocation.

### 2.2.1 Large-Timescale Resource Reservation

DAs facilitate the construction of virtual models of physical networks, which allows for real-time monitoring, simulation, and prediction of network states. This capability can empower resource reservation by leveraging advanced data analytics and processing techniques to anticipate network behaviors and reserve resources, thereby enhancing both service quality and network reliability [29, 30, 31].

In Integrated Sensing and Communication (ISAC) networks, continuously changing environmental conditions can lead to predictive model drift, thereby affecting the accuracy of resource reservation. To mitigate this issue, researchers have proposed methods that use DAs to construct dual predictive spatial models. Through ongoing network emulation to assess model performance, these methods enable the selection of optimal resource reservation decisions that ensure service quality while minimizing resource consumption [29]. Within the realm of Internet of Things (IoT), supported by Software Defined Network (SDN) and Network Function Virtualization (NFV), DA is applied to facilitate the migration of Virtualized Network Functions (VNFs). This strategy uses convolutional neural networks, gated recurrent units, and attention mechanisms to forecast VNF resource demands [32]. By adjusting the synchronization granularity of the digital twin and integrating a multi-agent algorithm that combines Long Short-Term Memory (LSTM) with a DDQN, VNF migration can be executed based on predicted resource requirements, thereby preventing node overload and maintaining service quality [30].

In multi-tier computing frameworks, DAs are utilized to collect user status data and group users based on mobility patterns, thereby enabling personalized resource reservation. By incorporating meta-learning methods, the system can tailor resource reservation strategies for state-aware applications, such as task execution in 6G networks, while balancing network resource utilization with reconfiguration costs in accordance with the specific needs

of different user groups [31]. To address the challenges inherent in edge computing networks with high user mobility and dynamic network conditions, DA frameworks provide real-time information on both computing and communication resources. By leveraging CSI for Roadside Unit (RSU) selection and integrating Non-Orthogonal Multiple Access (NOMA) technology, a model based on a Markov Decision Process (MDP) is constructed. The Actor-to-Critic (A2C) algorithm is then employed to optimize task completion delays and computational rates [33].

For adaptive service provisioning, the conversion of physical networks into digital models enables flexible resource aggregation and orchestration. An architecture based on network DAs is capable of real-time coordination of resource reservation, utilizing optimal substructures to reduce computational complexity and provide customized services to users with heterogeneous requirements [34]. In Ultra Reliable Low Latency Communications (URLLC) edge networks, task offloading must account for potential asynchronies between DAs and their physical counterparts. To address this challenge, a robust task offloading scheme has been proposed. This scheme combines decision networks with deviation networks to ensure low latency and low energy consumption, while simultaneously enhancing system robustness [35]. Furthermore, in Industrial IoT environments, the integration of DAs with NOMA is used to improve edge intelligence and spectral efficiency. By decomposing the resource reservation problem into sub-problems, such as computational resource allocation, subchannel allocation, and offloading decision-making, and developing a joint solution based on DRL alongside DA strategies, researchers have achieved reductions in task completion delays and significant improvements in system throughput [36].

To address task dependency issues in URLLC edge networks, a dual-sampling-based stochastic solution enhancement algorithm has been proposed to jointly optimize task offloading and resource allocation. This algorithm iteratively refines decisions to achieve optimal Central Processing Unit (CPU) frequency and transmission power while reducing energy consumption and time costs [37]. Additionally, a joint optimization model for user association, task offloading, and resource allocation based on DAs has been developed for Industrial IoTs environments. This model, solved using iterative decomposition and alternating optimization techniques, minimizes overall delay [38].

In 6G air-ground integrated industrial IoT networks, DAs are leveraged for real-time state prediction and adaptive resource utilization. This approach models the resource reservation problem as a Mixed Integer Nonlinear Programming (MINLP) problem and utilizes the Deep Deterministic Policy Gradient (DDPG) algorithm to achieve an effective balance between high energy savings and strict task delay constraints [39]. In the field of multimedia short video streaming, DAs have been used to establish user-centric resource management schemes. By constructing user DAs to predict bandwidth and computational

requirements, and applying low-complexity scheduling algorithms to solve non-convex resource reservation challenges, these approaches have significantly enhanced user experience while reducing resource consumption [40].

Other research directions include the integration of DAs with federated learning and deep neural networks for real-time monitoring and scheduling [41]; the use of blockchain and metaheuristic algorithms for resource reservation optimization in 6G IoT networks [42]; and the application of network digital twins for end-to-end network slicing management [43]. Additionally, QoE-driven resource reservation schemes that combine DAs with reinforcement learning, the trinity equilibrium algorithm, and Graph Neural Network (GNN)-based power allocation and user association strategies in the terahertz band have provided novel insights into the application of DAs for resource reservation [44, 45, 46, 47].

Overall, the application of DAs in communication network resource reservation encompasses a diverse range of areas, from adaptive model drift correction and virtual network function migration to multi-tier computing network planning, vehicular edge computing, URLLC, industrial IoT, air-ground integrated networks, and high-frequency communications. Although these research directions vary in focus, their common objective is to enhance the real-time, accurate, and flexible nature of resource reservation by constructing virtual replicas of physical networks, which can effectively address the evolving demands of network environments and service requirements.

### 2.2.2 Small-Timescale Resource Allocation

DA has emerged as a pivotal tool in optimizing small-timescale resource allocation, particularly within dynamic and complex network environments. By creating virtual replicas of physical systems, DAs enable real-time monitoring, prediction, and decision-making that are crucial for efficient resource management. The integration of DAs with advanced computational techniques, such as DRL and edge computing, further enhances the ability to allocate resources effectively to improve overall system performance. This approach is particularly beneficial in contexts characterized by high mobility and unpredictable network dynamics, such as vehicular edge computing and heterogeneous cellular networks [48].

Recent studies have demonstrated the potential of two-timescale frameworks enabled by DAs, where decision-making leverages both statistical state information over longer periods and real-time data in shorter intervals. For instance, Liu et al. illustrate how DRL techniques can optimize computational resource allocation and user association, thereby reducing long-term system delays [49]. In the realm of vehicular edge computing, Jeremiah et al. have proved that DAs provide essential and real-time insights into computing and

communication resources, which are critical for optimizing offloading decisions and sub-channel allocation [33].

Beyond the two-timescale approach, DAs significantly advance edge computing resource allocation by creating virtual representations of network nodes. This capability allows for accurate, real-time estimation and prediction of network conditions, which is particularly useful in scenarios where collaboration between roadside units and small cells is required to enhance network performance. Moreover, the integration of DAs with mobile edge computing in heterogeneous cellular networks enables dynamic user-device association and more efficient resource allocation, which can effectively reduce both transmission delays and energy consumption [50].

Advanced optimization algorithms have also been incorporated into DA-assisted resource allocation models to improve convergence speed and overall performance. Algorithms such as the whale optimization algorithm and DRL have been applied to reduce power consumption and mitigate resource allocation imbalances [51]. In the context of vehicular networks, DAs facilitate intelligent and delay-sensitive task offloading schemes that employ multi-network DRL algorithms. This approach not only maximizes network utility but also reduces energy consumption and overall service delay [52].

In addition to dynamic network management, DAs contribute substantially to energy-efficient resource allocation. For example, DAs can optimize wireless energy transfer through techniques like fractional planning and block coordinate descent algorithms, thereby enhancing energy efficiency and reducing interference in large-scale Multiple-Input and Multiple-Output (MIMO) systems [53]. Similarly, DAs enable the precise matching of dynamic resources between vehicles and edge servers through predictive analysis, which can effectively increase the utilization rate of idle resources [54].

Despite these promising advances, several challenges remain in the practical implementation of DA-based resource allocation. Issues related to computational complexity and the development of robust optimization models continue to present significant hurdles. Moreover, the successful integration of DAs into existing network infrastructures necessitates careful consideration of factors such as data privacy, network congestion, and resource constraints. The effectiveness of DA-based strategies is inherently dependent on the accuracy and reliability of the virtual models, which must be continuously updated to reflect real-time physical network status accurately.

## 2.3 Digital Agent Management

In this section, we review the existing DA management from two aspects: DA model selection and DA model update.

### 2.3.1 Digital Agent Model Selection

DAs have emerged as a transformative technology in communication networks, enabling real-time monitoring, emulation, and optimization of network operations. The selection of appropriate DA models is critical to ensure accuracy, efficiency, and scalability in various network environments [55].

Model accuracy is a fundamental criterion for DA model selection. The accuracy of a DA model is determined by its ability to reflect the real-time state of the physical system it represents. High-fidelity models are essential for tasks such as network optimization, traffic prediction, and resource allocation. For instance, in wireless edge networks, the accuracy of DA depends on the amount of input data, update frequency, and computational capacity at the edge server [56]. Similarly, in 6G networks, generative Artificial Intelligence (AI)-driven DAs are used to generate high-fidelity data, enabling accurate modeling of network elements such as mobile users, base stations, and wireless environments [57]. Energy efficiency is a growing concern in communication networks, and DAs can play a crucial role in optimizing energy consumption. For example, the use of digital twins in mobile networks enables energy-efficient resource allocation and reduces the carbon footprint of network operations [58].

Resource constraints, such as computational power, memory, and communication bandwidth, play a significant role in DA model selection. In edge computing environments, the selection of DA models must consider the limited resources available at the edge server. For example, the problem of model selection in wireless edge networks is formulated as an NP-complete integer program, where the goal is to maximize the minimum feature accuracy while respecting resource constraints [56]. Similarly, in large-scale networks, resource utilization metrics such as CPU usage, memory consumption, and network latency must be carefully evaluated to ensure efficient operation [59].

The integration of DAs with advanced technologies such as generative AI is another critical criterion. For instance, the integration of generative AI with DAs enables the generation of high-fidelity data, which can be used to enhance model accuracy and flexibility [60]. Additionally, the use of AI-native DAs in 6G networks facilitates intelligent network management by enabling network status prediction, pattern abstraction, and

decision-making [61]. A data-driven approach is essential for digital twin model selection. DAs rely on real-time data from physical systems to maintain accuracy and fidelity. For example, in the context of traffic efficiency and software-defined vehicular networks, DAs leverage real-time data to optimize traffic flow and reduce waiting times at roundabouts [62].

The ability of DAs to adapt to dynamic network conditions is essential for real-time optimization. In wireless networks, automatic model selection using DAs can optimize network performance by leveraging contextual information such as network conditions, topology, and traffic statistics [63]. The cost and complexity of deploying DAs are also important factors. While DAs offer significant benefits, their deployment can be resource-intensive. For example, the use of DAs in IoT network deployments requires careful planning to minimize deployment costs while maximizing coverage and performance [64]. Scalability and flexibility are critical for DAs in communication networks. DAs must be able to handle the increasing complexity and dynamism of 6G networks, which require sophisticated management solutions [65]. The use of DAs in industrial networking environments also requires support for mixed-criticality workloads and application-driven services [66]. Additionally, the ability of DAs to coexist as both a DA of the network and a network of DAs allows for greater flexibility in network management. Compatibility with existing network architectures and standards is another important criterion. DAs must integrate seamlessly with the current network infrastructure to ensure smooth operation. For instance, the integration of DAs with Time-Sensitive Networking (TSN) in 6G smart cities enables deterministic networking and low-latency communication [67].

Overall, DA model selection depends on accuracy, resource constraints, AI integration, and scalability. High-fidelity and AI-driven DAs can enhance performance in 6G networks, while efficient deployment ensures adaptability and cost-effectiveness.

### 2.3.2 Digital Agent Model Update

The concept of DA has gained attention in communication networks due to its potential to support predictive and autonomous network management. Early works primarily focused on constructing static digital replicas for network planning and simulation purposes [68]. However, as networks become more dynamic, such as frequent topology changes, fluctuating traffic patterns, and increasing heterogeneity, static models cannot guarantee accuracy. This led to a growing body of research emphasizing the need for real-time or near-real-time model updates to ensure that DAs accurately reflect the physical network state.

Several studies have explored data-driven methods for model synchronization. These

typically involve the continuous ingestion of telemetry data (e.g., latency, throughput, packet loss) and the use of machine learning or statistical estimation techniques to infer and update the DA state [69]. For instance, time-series prediction models such as Autoregressive Integrated Moving Average (ARIMA) and LSTM have been used to forecast network parameters and anticipate future states, thereby allowing the DA to preemptively adapt [70]. In more recent approaches, GNNs and federated learning have been applied to model the spatial and temporal dynamics of large-scale, distributed networks [71]. These methods allow for scalable and decentralized updates but often struggle with model drift and data inconsistency.

Model update in the context of network slicing and 5G/6G infrastructures has also received considerable attention. Researchers have proposed feedback loops and RL-based agents to align the DA model with network control policies and Service Level Agreements (SLAs) [72]. In these settings, DA updates are often triggered by threshold violations or SLA breaches, allowing the virtual model to evolve in sync with reconfigurations in the underlying network. Nevertheless, ensuring low-latency updates and minimizing the overhead associated with constant monitoring remain open challenges. Another line of work has investigated semantic and knowledge-driven updates to DAs using ontology-based representations and reasoning engines [73]. These approaches aim to make the DAs not only data-synchronized but also context-aware, enabling intelligent decision-making. However, such frameworks often suffer from scalability limitations and are difficult to integrate with real-time telemetry systems.

Despite the progress, several gaps persist. Most existing solutions assume ideal conditions for data availability and transmission between the physical and digital layers, which is often unrealistic in practice. Issues such as data sparsity, out-of-sync measurements, and device failures can cause significant divergence between the DA and its physical counterpart. Moreover, few studies address the computational cost of continuous updates, which can be prohibitive in large-scale networks with limited edge resources. Lastly, while some work has begun exploring transfer learning and meta-learning for model adaptation, robust and generalizable update mechanisms remain an open research area.

## 2.4 Summary

In this chapter, we have surveyed the existing literature for the DA-empowered multicast networks for short video streaming. With the literature review, we aim to identify research gaps and opportunities for methodological innovation.

## Chapter 3

# Digital Agent-Based Resource Reservation for QoE-Driven Multicast Short Video Streaming

In this chapter, we propose a user-centric resource management scheme based on the DA technique, which aims to enhance user satisfaction as well as reduce resource consumption. Firstly, we design a user DA-assisted resource reservation framework. Specifically, user DAs are constructed for individual users, which store users' historical data for updating multicast groups and abstracting useful information. The swipe probability distributions and recommended video lists are abstracted from user DAs to predict bandwidth and computing resource demands. Parameterized sigmoid functions are leveraged to characterize multicast groups' user satisfaction. Secondly, we formulate a joint non-convex bandwidth and computing resource reservation problem which is transformed into a convex piecewise problem by utilizing a tangent function to approximately substitute the concave part. A low-complexity scheduling algorithm is then developed to find the optimal resource reservation decisions. Simulation results based on the real-world dataset demonstrate that the proposed scheme outperforms benchmark schemes in terms of user satisfaction and resource consumption.

### 3.1 Background and Motivations

The proliferation of short video platforms, such as TikTok, Instagram Reels, and YouTube Shorts, enabled by the ubiquity of smartphones and high-speed wireless networks, has

ushered in a new era of digital entertainment [74, 75]. According to a recent report from TikTok, the number of daily active global users has reached 766 million in 2024 and will continue to maintain significant growth <sup>1</sup>. However, this short video streaming surge puts a significant burden on the existing wireless network infrastructures. Particularly, the ultra-definition and panoramic short videos pose new requirements on the transmission and computing capabilities of wireless networks, especially on higher bandwidth (300 Mbps) and transcoding speed (30.2 FPS), for providing immersive user experience [76, 77]. Multicast transmission, as an important technology in wireless networks, allows a single data stream to be disseminated to numerous users in a group simultaneously. By applying multicast transmission to short video streaming, the bandwidth utilization and network throughput can be effectively enhanced [78, 79].

To support multicast short video streaming, network resources, including bandwidth and computing resources, are required. Specifically, spectrum bandwidths are required for video delivery to each multicast group [80], and computing resources are required for video transcoding for each multicast group [81]. For consistent Quality of Service (QoS) provisioning, such network resources need to be reserved in advance [82]. Nevertheless, existing resource reservation schemes are mainly based on aggregated video requests while neglecting the unique users' behaviors in watching short videos, i.e., swipe behaviors. Without considering the swiping behaviors that can make videos not completely played by users, the resource demands can be overestimated and the reserved resources can be underutilized. As the swipe behaviors are user-specific [83], resource reservation can incorporate the user-specific swipe behaviors, i.e., user-centric resource reservation, is desired for efficient resource utilization in the multicast short video streaming.

DA is a potential technology to realize the user-centric resource reservation. DA is an intelligent network proxy that can emulate network status, abstract network features, and make tailored resource management [4]. To update current end users' statuses, such as network conditions, data traffic, and mobility trajectories, user DAs are constructed for individual users. The data stored in user DAs can be leveraged to analyze users' transmission rates and behavior patterns for resource reservation. In the multicast short video streaming scenario, user DAs can store users' historical networking and personal information, and analyze their video traffic patterns and watching behaviors. Based on the distilled coarse- and fine-grained user information from user DAs, users' bandwidth and computing resource demands can be accurately predicted to facilitate the effective user-centric resource reservation. In this work, the precise role of constructed user DAs is to accurately cluster users into different multicast groups, and abstract swipe probability distributions and recommended video lists for resource reservation.

---

<sup>1</sup><https://www.businessofapps.com/data/tik-tok-statistics/>

The motivation of the DA-based resource reservation for multicast short video streaming includes three aspects. Firstly, since user status differs from each other, inaccurate user clustering can lead to low transmission rate and resource wastage due to various channel conditions and frequent swipe behaviors on uninterested video content, respectively. Therefore, it is essential to develop an accurate and efficient user clustering scheme. Secondly, due to users' differentiated user satisfaction of service quality, multicast groups may have different levels of satisfaction with the reserved bandwidth and computing resources. Therefore, it is critical to establish the user satisfaction model to characterize the impact of resource reservation on each multicast group's satisfaction. Thirdly, since the joint bandwidth and computing resource reservation is usually a complex and non-convex problem, how to design a low-complexity algorithm to solve it with near-optimal results is an important issue.

Designing an effective DA-based resource reservation scheme needs to address the following challenges: 1) incorporating swipe behaviors in resource reservation decision-making; 2) establishing the mathematical model between multicast groups' user satisfaction and reserved resources. Specifically, users' swipe behaviors are stochastic and spatiotemporally varied, which are difficult to be predicted in real time. Therefore, how to conduct effective data abstraction to obtain the distilled swipe feature and utilize it to predict bandwidth and computing resource demands is challenging. Furthermore, due to the dynamics of users' personalized preferences and sensitivities to service quality, the same amount of reserved resources to one multicast group can lead to different user satisfaction at different time. Therefore, how to establish and update each multicast group's user satisfaction model in each resource reservation window is challenging.

In this chapter, we propose a DA-assisted resource reservation scheme, which can effectively enhance user satisfaction and reduce resource consumption. The main contributions are summarized as follows:

- Firstly, we propose a novel DA-assisted resource reservation framework to incorporate the impact of swipe behaviors. Specifically, we construct user DAs to store users' historical data, including channel conditions, locations, swipe timestamps, and preferences. A novel user clustering algorithm is proposed to analyze user DA data for multicast group construction. The proposed user clustering algorithm consists of three parts, i.e., autoencoders, a DDQN, and the K-means++ method, which are responsible for user DA data compression, clustering number determination, and fast user clustering, respectively. Then, the group-level information, i.e., the swipe probability distribution and recommended video list, for each multicast group is abstracted. Based on that, multicast groups' average engagement time, video traffic,

and computing consumption can be analyzed to predict bandwidth and computing resource demands for resource reservation.

- Secondly, we establish a user satisfaction model to quantify the impact of bandwidth and computing resource reservation on user satisfaction. Specifically, we adopt a real-world user satisfaction dataset to explore the relationship between resource demand, reserved resources, and user satisfaction. Based on the observation of experiments, we find that user satisfaction is positively and negatively exponential with buffer length and rebuffering time. By using the buffer length and rebuffering time as a linkage, we establish an exponential mathematical model for bandwidth reservation satisfaction. Similarly, video quality is selected as the intermediate variable to build an exponential mathematical model for computing resource reservation satisfaction.
- Thirdly, our objective is to maximize the system utility consisting of user satisfaction and resource consumption in each resource reservation window. Since the formulated problem is non-convex and difficult to solve, we first transform it into two independent subproblems regarding bandwidth and computing resource reservation, respectively. Then, we perform continuous processing on these two subproblems and utilize the tangent function to approximately substitute the convex parts, which transforms the original subproblems into convex problems. Finally, we design a linear approximation method to make the near-optimal resource reservation decisions for the transformed convex problems. The extensive simulation results on real-world short video streaming datasets show that the proposed scheme can effectively improve system utility as compared with the state-of-the-art resource reservation schemes.

The remainder of this paper is organized as follows. The system model and problem formulation are presented in Sections 3.2 and 3.3, respectively. A fast resource reservation scheme is developed to solve the problem in Section 3.4, validated by simulation results in Section 3.5. Finally, we summarize this work in Section 3.6.

## 3.2 System Model

In this section, we propose a DA-based resource reservation framework. Within the framework, we introduce how to construct DAs, how to use DAs for multicast group management, and build a user satisfaction model, respectively.

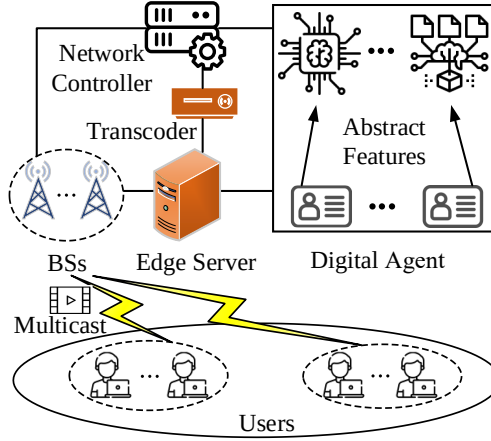


Figure 3.1: DA-based resource reservation framework for multicast short video streaming.

### 3.2.1 Digital Agent-Based Resource Reservation Framework

As shown in Fig. 3.1, we consider a DA-assisted multicast short video streaming scenario, which consists of multiple Base Stations (BSs), an Edge Server (ES), users, and user DAs.

- **BSs:** BSs utilize multicast technology to transmit short videos to different multicast groups. The set of multicast groups is denoted by  $\mathcal{G} = \{1, \dots, G\}$ . The set of bandwidths for all BSs is denoted by  $\mathcal{M} = \{1, \dots, M\}$ .
- **ES:** The ES connects to all BSs and stores popular short videos with the highest version (bitrate) to avoid frequent video retrievals from content providers. The stored short videos can be transcoded to a lower version in the transcoder to adapt to users' dynamic channel conditions and swipe behaviors. The set of Virtual Machine (VM) instances for the transcoders is denoted by  $\mathcal{N} = \{1, \dots, N\}$ , and the computing capability of each VM instance is denoted by  $\omega$ .
- **Users:** Users with similar statuses, such as preferences, swipe behaviors, locations, and channel conditions, are clustered into one multicast group, and the corresponding user set is denoted by  $\mathcal{K}_g = \{1, \dots, K_g\}$ . Users in the same multicast group are recommended the same short video list and receive them by the multicast transmission.
- **User DAs:** There are multiple user DAs deployed at the ES. Each user DA corresponds to a user and stores users' statuses. BSs are responsible for collecting real-time

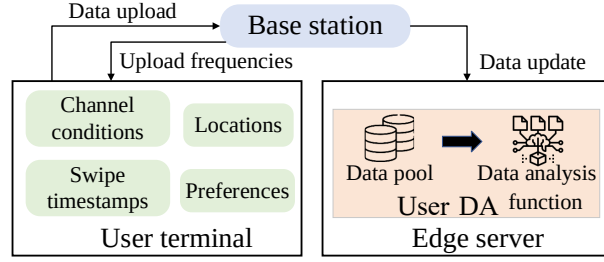


Figure 3.2: User DA data collection procedure.

users' statuses to update user DAs. User DA data are utilized to abstract some useful information, such as data similarity and swipe probability distribution, which can help the network controller realize an accurate multicast group construction and resource demand prediction.

The DA-assisted resource reservation framework for multicast short video streaming operates as follows. In each resource reservation window, user statuses are uploaded to the ES to update corresponding user DAs and adjust multicast groups. In each multicast group, user DAs' swipe timestamps and preferences are used to abstract the swipe probability distribution and recommended video list for accurate bandwidth and computing resource demand prediction. Based on the predicted information, the network controller reserves appropriate bandwidths and VM instances for each multicast group.

### 3.2.2 Digital Agent Construction

User DA consists of a finite data pool and a data analysis function as shown in Fig. 3.2. The data pool records a user's recent status through diversified data collection frequencies. The data analysis function investigates a user's swipe timestamps to obtain a swipe probability distribution for each video type. The construction of a user DA can be summarized into two phases.

- BS collects a user's data from two aspects, i.e., networking-related data and behavior-related data. The networking-related data include a user's real-time channel conditions and locations, which are utilized to estimate the transmission rate. The behavior-related data consist of a user's swipe timestamps and preferences. The swipe timestamps reflect a user's swipe behavior [84], which are used to abstract swipe probability distribution for engagement time prediction. The preferences are

leveraged for the recommended video list update. By integrating these two kinds of data, user DAs can well reflect users' real-time statuses and predict bandwidth and computing resource demands. Data collection number can be different for various user DA data attributes in each resource reservation window. Given that a low-speed mobility scenario where users' channel conditions usually vary on a small timescale [85], the data collection number of a user's channel condition and location in one resource reservation window,  $T$ , is denoted by  $F_1$ . Since the swipe probability needs to be calculated over several short videos, the corresponding data collection number of a user's swipe timestamps in one resource reservation window is denoted by  $F_2$ . A user's preference is an essential metric to determine which videos should be recommended. Since the user's preference is relatively dynamic data that is updated based on the user's like, share, and swipe frequency [86], the data collection number remains consistent with that of the swipe timestamps to ensure the data freshness.

- A user DA's functionality consists of two parts, i.e., a user's swipe behavior analysis module and a data interaction module. The former analyzes the collected swipe timestamps to abstract swipe probability distributions. The latter provides stored data and the abstracted swipe probability distributions to the network controller to update multicast groups and predict resource demands. In current resource reservation window  $t$ , user  $k$ 's swipe probability distribution,  $p_{k,c}^t(e)$ , for video type  $c$  is updated by

$$p_{k,c}^t(e) = \lambda p_{k,c}^{t-1}(e) + (1 - \lambda) \frac{A_{k,c}^{t-1}(e)}{\widehat{A}_{k,c}^{t-1}}, \quad (3.1)$$

where  $e$  is the video timestamp ranging from 1 to 15. Parameter  $\lambda$  is a weighting factor ranging from 0 to 1. A smaller  $\lambda$  indicates that a user's current swipe probability distribution is more related to recent swipe behaviors. In Eq. (3.1),  $A_{k,c}^{t-1}(e)$  represents user  $k$ 's swipe numbers for video type  $c$  in previous window  $t - 1$ , which is counted by swipe timestamps, and  $\widehat{A}_{k,c}^{t-1}$  denotes the number of transmitted videos of type  $c$  to user  $k$  in previous window  $t - 1$ .

### 3.2.3 Digital Agent-Based Multicast Group Management

Based on stored data in user DAs, we can analyze users' similarities to divide users with similar statuses into the same multicast group. A three-step method is proposed to realize fast and accurate multicast group construction. Specifically, autoencoders are first employed to compress time-series DA data for dimension reduction. Then, a DRL network is adopted to determine the clustering number by mining users' intrinsic correlation. Finally,

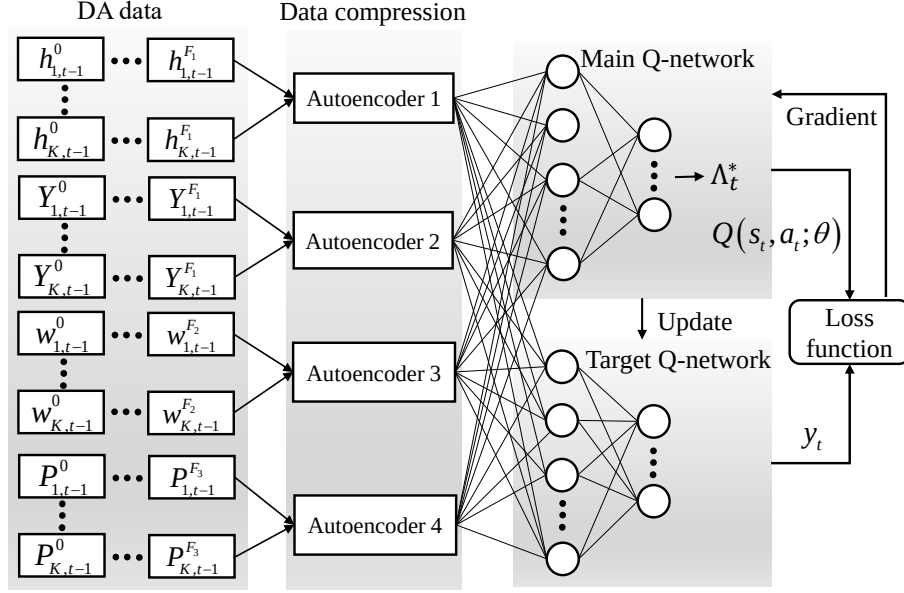


Figure 3.3: The improved DDQN architecture for clustering number determination.

the K-means++ algorithm [87] is utilized to realize a fast multicast group construction based on the determined clustering number. After multicast groups are constructed, we can abstract the swipe probability distribution and recommended video list for each multicast group.

### Clustering Number Determination

User DA data consist of four dimensions and each dimension corresponds to time-series data, directly using a DRL method to analyze user DA data may suffer from the curse of input dimension. Therefore, we add four Autoencoders to the existing DDQN [88] to compress user DA data, which can effectively reduce the input dimension [89]. The compressed data are further input to the actor-critic network for determining an appropriate clustering number, as shown in Fig. 3.3.

As shown in Fig. 3.3, the processed user DA data consist of users' historical channel conditions,  $\{h_{k,t-1}^f\}_{k \in \mathcal{K}, f \in \mathcal{F}_1}$ , locations,  $\{Y_{k,t-1}^f\}_{k \in \mathcal{K}, f \in \mathcal{F}_1}$ , swipe timestamps,  $\{w_{k,t-1}^f\}_{k \in \mathcal{K}, f \in \mathcal{F}_2}$ , and preferences,  $\{P_{k,t-1}^f\}_{k \in \mathcal{K}, f \in \mathcal{F}_2}$ , in window  $t-1$ . Since each user DA data attribute has a temporal correlation, we utilize Autoencoders to process these data to reduce input dimen-

sion and abstract temporal features. The processed user DA data are further input into the DDQN, consisting of one target and one main Q-network, for model training to obtain the appropriate clustering number  $\Lambda_t^*$ . Specifically, the main Q-network with network parameter  $\theta$  takes the current processed user DA data as input and outputs a set of Q-values. The target Q-network with network parameter  $\theta'$  is a copy of the main Q-network that is periodically updated to match the parameters of the main Q-network. The target value,  $y_t$ , of DDQN is given by

$$y_t = r_{t+1} + \gamma Q(s_{t+1}, \arg \max Q(s_{t+1}, a; \theta); \theta'), \quad (3.2)$$

where main Q-network  $Q(s, a; \theta)$  is utilized for the clustering number determination and target Q-network  $Q(s, a; \theta')$  is used for the value evaluation. Here,  $s$  and  $a$  represent processed user DA data and the clustering number, respectively. Since the objective of clustering is to reserve appropriate resources for different multicast groups to enhance system utility, we select the system utility that will be introduced in Section 3.3 as reward  $r$ . During the training process, the main Q-network is updated to minimize the difference between the predicted and true Q-values. The target Q-network is periodically updated to match the parameters of main Q-network to help stabilize the training process.

## User Clustering

When the clustering number of multicast groups is determined, we can classify users into different multicast groups based on user DA data similarities. We adopt the K-means++ algorithm to complete the user clustering process. Specifically, the Euclidean distance,  $D(i, j)$ , between User DAs  $i$  and  $j$  is first calculated as follows:

$$D(i, j) = \|\tilde{h}_i - \tilde{h}_j\|_2 + \|\tilde{Y}_i - \tilde{Y}_j\|_2 + \|\tilde{w}_i - \tilde{w}_j\|_2 + \|\tilde{P}_i - \tilde{P}_j\|_2, \quad (3.3)$$

where  $\tilde{h}_i, \tilde{Y}_i, \tilde{w}_i, \tilde{P}_i$  represent the compressed user DA data attributes output from autoencoders. Then, the Euclidean distance between the sampling user DA and the current clustering center is used to calculate the selection probability. Since the longer distance causes a higher selection probability as the new clustering center, the selection probability,  $\Theta(i)$ , of user DA  $i$  is depicted as

$$\Theta(i) = \frac{D(i, \tau)^2}{\sum_j D(j, \tau)^2}, \quad (3.4)$$

where  $\tau$  denotes the current clustering center. After several iterations, users with similar statuses are clustered into one multicast group. The set of constructed multicast group is denoted by  $\Omega_t = \{1, \dots, \Lambda_t^*\}$ .

## Abstracted Information

After constructing multicast groups, we need to abstract some useful information from each of them to predict resource demands in each resource reservation window. Recommended video lists and corresponding swipe probability distributions can well achieve this requirement. To obtain a good recommended video list, a good video recommendation mechanism not only needs to consider video popularity distributions but also users' preferences [86]. The former can be directly obtained by analyzing view counts and engagement time from content providers. In window  $t - 1$ , the popular video set is denoted by  $\mathcal{V} = \{1, \dots, V\}$  and the corresponding popularity distribution is  $\{E_v\}_{v \in \mathcal{V}}$ . The latter is time-series data and exists a relatively strong temporal correlation in each multicast group. The collected preference,  $P_{k,t-1}^f$ , in user DA  $k$  is a  $1 \times C$  matrix, i.e.,  $\{P_{k,t-1}^{f,1}, \dots, P_{k,t-1}^{f,C}\}$ , reflecting user  $k$ 's preferences of  $C$  video types. To accurately estimate multicast group  $g$ 's preference matrix,  $\{\hat{P}_{t,g}^c\}_{c \in \mathcal{C}}$ , in current window  $t$ , we first calculate the users' average preference on each kind of video in previous window  $t - 1$ , and then integrate it with the discounted preference,  $\tilde{\lambda}\hat{P}_{t-1,g}^c$ , in previous window  $t - 1$ , as follows:

$$\hat{P}_{t,g}^c = \tilde{\lambda}\hat{P}_{t-1,g}^c + \frac{1 - \tilde{\lambda}}{K_g F_2} \sum_{k=1}^{K_g} \sum_{f=1}^{F_2} P_{k,t-1}^{f,c}, \forall c \in \mathcal{C}, \quad (3.5)$$

where  $\tilde{\lambda}$  is a parameter ranging from 0 to 1. The recommended ranking,  $\mathfrak{R}_{t,g,v}$ , of video  $v$  for multicast group  $g$  in window  $t$  combines the video popularity distribution and preference matrix, which is depicted as  $\mathfrak{R}_{t,g,v} = E_v \sum_{c \in \mathcal{C}} I_c(v) \hat{P}_{t,g}^c$ . Here,  $I_c(v)$  is an indicator function, representing if video  $v$  belongs to type  $c$ ,  $I_c(v) = 1$ ; otherwise,  $I_c(v) = 0$ .

In each resource reservation window, the recommended video list  $\tau_{g,t}$  for multicast group  $g$  is constituted based on the recommended rankings in popular video set  $\mathcal{V}$ , and the corresponding list length is denoted by  $\rho_t$ . In addition, each video in the recommend list has a unique swipe probability distribution,  $p_{g,t}^v(e)$ , which can be abstracted by accumulating users' swipe probability distributions in a multicast group, as follows:

$$p_{g,t}^v(e) = \sum_{k \in \mathcal{K}_g} \sum_{c \in \mathcal{C}} I_c(v) p_{k,c}^t(e). \quad (3.6)$$

## DA-Assisted Resource Demand Prediction

Based on the abstracted grouping information, i.e., swipe probability distributions and recommended video lists, each multicast group's bandwidth and computing resource de-

mands can be predicted to help the network controller to facilitate an efficient resource reservation scheme.

The average engagement time,  $W_g^t$ , of multicast group  $g$  at window  $t$  is estimated based on the abstracted swipe probability distribution, which is expressed as

$$W_g^t = \sum_{v \in \tau_g} \int_0^{\varphi(v)} (1 - p_{g,t}^v(e)) e de. \quad (3.7)$$

We need to predict the video traffic of each multicast group. Since one video can be transmitted to all users by multicast transmission in a multicast group, the video traffic of each user should not be accumulated. In addition, since the multicast video version adaptively changes with users' dynamic channel conditions and buffer lengths, we select the average multicast video version,  $\bar{l}$ , in previous window  $t - 1$ , to approximate the multicast video version in current window  $t$ . Based on the above analysis, the video traffic,  $Y_g^t$ , of multicast group  $g$  at window  $t$  is predicted by

$$Y_g^t = \sum_{v \in \tau_g} \int_0^{\varphi(v)} (1 - p_{g,t}^v(e)) \sum_{l=1}^{\bar{l}} z_v^l(e) de, \quad (3.8)$$

where  $z_v^l(e)$  is the file size of segment layer  $l$  of video  $v$  at video timestamp  $e$ .

Based on the estimated average engagement time and video traffic, the bandwidth resource demand,  $R_g^t$ , is given by

$$R_g^t = W_g^t / Y_g^t. \quad (3.9)$$

Since the caching capacity of an edge server is limited, we only cache the basement layer of each recommended segment in the edge server to guarantee users' basic watching requirements. Enhancement layers can be obtained by transcoding and then jointly transmitted with the basic layer to users to enhance the video quality. In this process, the computing consumption is predicted by

$$Z_g^t = \mu \left( Y_g^t - \sum_{v \in \tau_g} \int_0^{\varphi(v)} (1 - p_{g,t}^v(e)) z_v^0(e) de \right), \quad (3.10)$$

where  $\mu$  is the computing density for video transcoding, and  $z_v^0(e)$  is the file size of the basement layer of video  $v$  at video timestamp  $e$ .

Based on the estimated average engagement time and computing consumption, the computing resource demand,  $O_g^t$ , is given by

$$O_g^t = Z_g^t / Y_g^t. \quad (3.11)$$

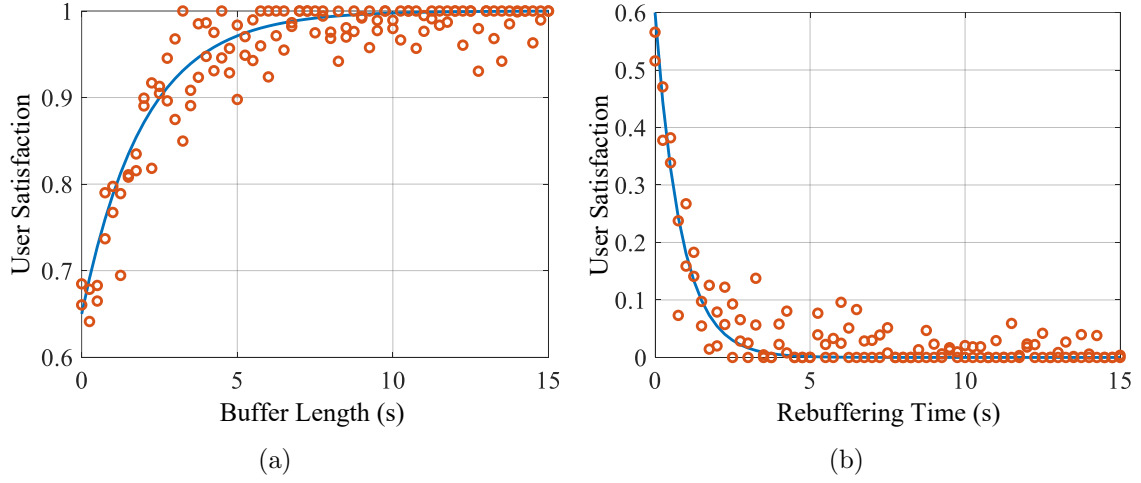


Figure 3.4: User satisfaction over different buffer lengths and rebuffering time.

### 3.2.4 User Satisfaction-Based QoE Model

Since the system bandwidth and computing resources are limited, multicast groups' resource demands may not be always satisfied. In addition, users usually have different sensitive degrees of rebuffering time and video quality [90], which indicates that the same reserved resources for various multicast groups can lead to different user satisfactions. Therefore, we analyze the relationship between reserved bandwidth and computing resources and user satisfaction for each multicast group, which can help the network controller to make a better resource reservation scheme to improve user satisfaction/QoE.

First, we analyze how reserved bandwidths affect user satisfaction based on observed users' buffers. Specifically, if the downlink transmission rate estimated by reserved bandwidths is larger than the bandwidth resource demand, users' buffer lengths can increase; otherwise, users' buffers will gradually become empty and rebuffering time will increase. Therefore, the gap between the downlink transmission capability and the bandwidth resource demand is positively correlated with users' buffers. To measure how users' buffers can affect user satisfaction, we select a part of users' watching and rating records from video dataset<sup>2</sup>, and use the fitting function<sup>3</sup> to form a fitted curve from scattered points, as shown in Fig. 3.4.

It can be observed that user satisfaction is positively and negatively exponential with

<sup>2</sup>Video quality assessment: <https://live.ece.utexas.edu/research/LIVESTallStudy/liveMobile.html>

<sup>3</sup>Mathworks: <https://www.mathworks.com/help/optim/ug/lsqcurvefit.html>

buffer length and rebuffering time, respectively. To unify these two factors, we first take the negative of rebuffering time, with lower values indicating longer rebuffering time, and then use an approximately exponential function to express the relationship between user satisfaction, buffer length, and rebuffering time. By using the buffer length and rebuffering time as a linkage, we can construct a parameterized sigmoid function to characterize the exponential relationship between bandwidth resource reservation and user satisfaction, as follows:

$$U_{B,g}^t = \frac{1}{1 + \exp \left\{ -\xi_g^t \left( m_g^t B \log (1 + \ell_g^t) - R_g^t \right) \right\}}, \quad (3.12)$$

where  $U_{B,g}^t$  denotes multicast group  $g$ 's bandwidth reservation satisfaction in window  $t$ . Here,  $\ell_g^t$  and  $\xi_g^t$  are the average signal-to-interference plus noise ratio and the sensitivity of bandwidth reservation for multicast group  $g$  in window  $t$ , respectively. Here,  $m_g^t$  and  $B$  are the reserved bandwidth number and size for multicast group  $g$  in window  $t$ , respectively. When the downlink transmission rate of reserved bandwidths exceeds the bandwidth resource demand, user satisfaction will quickly increase and gradually tend to the highest value, and vice versa.

Second, we analyze how reserved VM instances affect user satisfaction. Specifically, if the transcoding capability estimated by reserved VM instances exceeds the computing resource demand, the video transcoding process can be quickly completed to guarantee that users can watch videos with satisfied video quality; otherwise, users will watch videos with low video quality and experience frequent quality variation. Since the video quality also has a similar exponential relationship with user satisfaction [91], we also adopt a parameterized sigmoid function to characterize the exponential relationship between VM instance reservation and user satisfaction as follows:

$$U_{V,g}^t = \frac{1}{1 + \exp \left\{ -\vartheta_g^t \left( n_g^t \omega - O_g^t \right) \right\}}, \quad (3.13)$$

where  $U_{V,g}^t$  denotes multicast group  $g$ 's computing reservation satisfaction in window  $t$ . Here,  $\vartheta_g^t$  and  $n_g^t$  represent the sensitivity of VM instance reservation and the number of reserved VM instances for multicast group  $g$  in window  $t$ , respectively. Here,  $\omega$  is the computing capacity of one VM instance.

Compared with the standard QoE model [92], our user satisfaction-based QoE model has two main differences. Firstly, our user satisfaction-based QoE model is established for characterizing users' long-term experience in each resource reservation process instead of real-time experience. The traditional QoE model is especially effective when we conduct real-time resource allocation. However, it cannot be directly applied to the long-term

resource reservation process because real-time performance metrics hardly reflect long-term performance. In our constructed user satisfaction-based QoE model, we establish the mathematical relationship between resource reservation, resource demand, and user satisfaction, which can efficiently guide the resource reservation process. Secondly, our user satisfaction-based QoE model includes the bandwidth reservation satisfaction model and the VM instance reservation satisfaction model. The front mainly considers the impact of buffer length and rebuffering time on long-term user satisfaction. The latter mainly considers the impact of video quality on long-term user satisfaction.

### 3.3 Problem Formulation

In this section, we construct a system utility function to evaluate the performance of the DA-assisted resource reservation scheme, including operation cost, reconfiguration cost, and user QoE.

The operation cost,  $U_O^t$ , refers to the cost of reserved bandwidths and VM instances for all multicast groups [93] in window  $t$ , which can be expressed as

$$U_O^t = \varpi_1 \sum_{g=1}^{\Lambda_t^*} m_g^t + \varpi_2 \sum_{g=1}^{\Lambda_t^*} n_g^t, \quad (3.14)$$

where  $\varpi_1$  and  $\varpi_2$  represent the unit cost of using a bandwidth and a VM instance, respectively.

In each resource reservation window, if the previous resource reservation configuration for a multicast group cannot satisfy the current dramatically increasing resource demands, the network controller needs to configure bandwidths and VM instances for the multicast group, which incurs reconfiguration cost. The resource release cost is omitted since this process can be quickly completed with negligible cost. Based on the above analysis, the reconfiguration cost,  $U_R^t$ , can be expressed as

$$U_R^t = \sum_{g=1}^{\Lambda_t^*} \varpi_3 [m_g^t - m_g^{t-1}]^+ + \varpi_4 [n_g^t - n_g^{t-1}]^+, \quad (3.15)$$

where  $\varpi_3$  and  $\varpi_4$  denote the unit cost of bandwidth and VM instance reconfigurations, respectively. Since the clustering number in the previous resource reservation window may

be less than that in the current window, a part of elements in vectors  $\mathcal{M}^{t-1}$  and  $\mathcal{N}^{t-1}$  is denoted by 0 to guarantee dimension match, which can be expressed as

$$m_g^{t-1} = 0, n_g^{t-1} = 0, \forall g \in [\Lambda_{t-1}^* + 1, \Lambda_t^*]. \quad (3.16)$$

The objective of system utility is to achieve low resource operation and reconfiguration costs and high user QoE. With Eq. (3.12)-(3.15), the system utility is defined as

$$U^t = \frac{1}{\Lambda_t^*} \sum_{g=1}^{\Lambda_t^*} (U_{B,g}^t + \delta_1 U_{V,g}^t) - \delta_2 U_O^t - \delta_3 U_R^t, \quad (3.17)$$

where  $\delta_1$ ,  $\delta_2$ , and  $\delta_3$  represent the weighted parameters to balance each system utility component.

Since multicast groups' bandwidth and computing resource demands are dynamic due to users' diversified swipe behaviors and channel conditions, it is paramount to reserve the appropriate bandwidth and computing resources for each multicast group to guarantee a satisfied service quality. In each resource reservation window, our objective is to maximize the system utility within the limited bandwidth and computing resources, and the corresponding optimization problem is formulated as

$$\mathbf{P}_0 : \quad \max_{\{m_g^t, n_g^t\}_{g \in \Omega_t}} U^t \quad (3.18a)$$

$$\text{s.t.} \quad \sum_{g=1}^{\Lambda_t^*} m_g^t \leq M, \forall t \in \mathcal{T}, \quad (3.18b)$$

$$\sum_{g=1}^{\Lambda_t^*} n_g^t \leq N, \forall t \in \mathcal{T}, \quad (3.18c)$$

$$m_g^t \in Z^+, \forall g \in \Omega_t, t \in \mathcal{T}, \quad (3.18d)$$

$$n_g^t \in Z^+, \forall g \in \Omega_t, t \in \mathcal{T}, \quad (3.18e)$$

and (3.16).

Constraints (3.18b) and (3.18c) are resource capacity constraints, which guarantee that the total reserved bandwidths and VM instances cannot exceed the system capacity.

### 3.4 Proposed Solution

In this section, we first transform the optimization problem into two convex subproblems. Then, we develop a fast resource reservation algorithm and analyze its computational complexity.

### 3.4.1 Problem Transformation

Problem  $\mathbf{P}_0$  is a nonlinear integer programming problem. To solve this problem, we first transform it into two subproblems. Specifically, since optimization variable  $m_g^t$  is independent of optimization variable  $n_g^t$ , the problem can be decoupled into two independent minimization subproblems regarding bandwidth and VM instance reservation. The decoupled subproblems  $\mathbf{P}_1$  and  $\mathbf{P}_2$  are given by

$$\mathbf{P}_1 : \min_{\{m_g^t\}_{g \in \Omega_t}} \frac{1}{\Lambda_t^*} \sum_{g \in \Omega_t} \delta_3 \varpi_3 [m_g^t - m_g^{t-1}]^+ + \delta_2 \varpi_1 m_g^t - \frac{1}{1 + \exp \{ \xi_g^t (m_g^t B \log (1 + \ell_g^t) - R_g^t) \}} \quad (3.19a)$$

$$\text{s.t. } \sum_{g \in \Omega_t} m_g^t \leq M, \forall t \in \mathcal{T}, \quad (3.19b)$$

$$m_g^t \in Z^+, \forall g \in \Omega_t, t \in \mathcal{T}, \quad (3.19c)$$

$$m_g^{t-1} = 0, \forall g \in [\Lambda_{t-1}^* + 1, \Lambda_t^*], t \in \mathcal{T}, \quad (3.19d)$$

$$\mathbf{P}_2 : \min_{\{n_g^t\}_{g \in \Omega_t}} \frac{1}{\Lambda_t^*} \sum_{g \in \Omega_t} \delta_3 \varpi_4 [n_g^t - n_g^{t-1}]^+ + \delta_2 \varpi_2 n_g^t - \frac{\delta_1}{1 + \exp \{ \vartheta_g^t (n_g^t \omega - O_g^t) \}} \quad (3.20a)$$

$$\text{s.t. } \sum_{g \in \Omega_t} n_g^t \leq N, \forall t \in \mathcal{T}, \quad (3.20b)$$

$$n_g^t \in Z^+, \forall g \in \Omega_t, t \in \mathcal{T}, \quad (3.20c)$$

$$n_g^{t-1} = 0, \forall g \in [\Lambda_{t-1}^* + 1, \Lambda_t^*], t \in \mathcal{T}. \quad (3.20d)$$

Then, we perform continuous processing on variables  $m_g^t$  and  $n_g^t$ . If the discrete feasible set is the set of integer points in a convex polytope and the objective is convex, the discrete problem can be solved as a convex program [94]. The objective function of subproblem  $\mathbf{P}_1$  consists of two parts, i.e., one related to the exponential term and one related to the approximately linear term. The former is expressed by  $\Psi_g^t = -1/(1 + \exp \{ -\xi_g^t (m_g^t B \log (1 + \ell_g^t) - R_g^t) \})$ , and its convexity is related to the range of independent variable  $m_g^t$ . Specifically, the second derivative of  $\Psi_g^t$  is shown in Eq. (3.21).

$$\begin{aligned} \frac{\partial^2 \Psi_g^t}{\partial^2 m_g^t} &= \frac{\exp \{ -\xi_g^t (m_g^t B \log (1 + \ell_g^t) - R_g^t) \} (\xi_g^t B \log (1 + \ell_g^t))^2}{(1 + \exp \{ -\xi_g^t (m_g^t B \log (1 + \ell_g^t) - R_g^t) \})^3} \\ &\quad \times (1 - \exp \{ -\xi_g^t (m_g^t B \log (1 + \ell_g^t) - R_g^t) \}). \end{aligned} \quad (3.21)$$

When  $m_g^t \geq R_g^t/B \log(1 + \ell_g^t)$ , we can have  $\partial^2 \Psi_g^t / \partial^2 m_g^t \geq 0$  and function  $\Psi_g^t$  is convex; otherwise, function  $\Psi_g^t$  is concave. The latter is expressed by  $\Gamma_g^t = \delta_3 \varpi_3 [m_g^t - m_g^{t-1}]^+ + \delta_2 \varpi_1 m_g^t$ , and it is convex due to the convexity of  $[\cdot]^+$  function [95].

Since function  $\Psi_g^t$  is not always convex in the whole range of independent variable  $m_g^t$ , we make an approximation to the concave part of  $\Psi_g^t$  to transform it into a convex function. Specifically, when  $m_g^t < R_g^t/B \log(1 + \ell_g^t)$ , we utilize a tangent to approximately substitute the concave part. The slope of tangent is the first derivative of  $\Psi_g^t$  when  $m_g^t$  equals to  $R_g^t/B \log(1 + \ell_g^t)$ , which can be expressed by

$$\kappa_g^t = -\frac{1}{4} \xi_g^t B \log(1 + \ell_g^t). \quad (3.22)$$

The intersection of the tangent and the horizontal axis is given by

$$b_g^t = \frac{\xi_g^t R_g^t - 2}{\xi_g^t B \log(1 + \ell_g^t)}. \quad (3.23)$$

To guarantee function  $\Psi_g^t$  is always negative, independent variable  $m_g^t$  needs to satisfy  $m_g^t > b_g^t$ . Based on this transformation, approximate function,  $\tilde{\Psi}_g^t$ , can be expressed as a convex piecewise function, i.e.,

$$\tilde{\Psi}_g^t = \begin{cases} \Psi_g^t, & m_g^t \geq R_g^t/B \log(1 + \ell_g^t), \\ \kappa_g^t m_g^t - 0.5, & b_g^t < m_g^t < R_g^t/B \log(1 + \ell_g^t). \end{cases} \quad (3.24)$$

Since the addition of convex functions  $\tilde{\Psi}_g^t$  and  $\Gamma_g^t$  is still convex and constraints are linear, subproblem  $\mathbf{P}_1$  is transformed into a convex optimization subproblem  $\mathbf{P}'_1$ , i.e.,

$$\mathbf{P}'_1 : \min_{\{m_g^t\}_{g \in \Omega_t}} \frac{1}{\Lambda_t^*} \sum_{g \in \Omega_t} \tilde{\Psi}_g^t + \Gamma_g^t \quad (3.25a)$$

$$\text{s.t. } m_g^t > b_g^t, \quad (3.25b)$$

(3.19b), (3.19c), and (3.19d).

Similarly, we can transform the exponential term,  $\Xi_g^t$ , in the objective function of subproblem  $\mathbf{P}_2$  into a convex piecewise function, which can be expressed as

$$\tilde{\Xi}_g^t = \begin{cases} \Xi_g^t, & n_g^t \geq O_g^t/\omega, \\ \hat{\kappa}_g^t n_g^t - 0.5, & \hat{b}_g^t < n_g^t < O_g^t/\omega, \end{cases} \quad (3.26)$$

where  $\hat{\kappa}_g^t = -\frac{1}{4}\vartheta_g^t\omega$  and  $\hat{b}_g^t = \frac{\vartheta_g^t O_g^t - 2}{\vartheta_g^t \omega}$ . The approximately linear term,  $\hat{\Gamma}_g^t$ , in the objective function P2 is also convex due to the max function. Since the addition of convex functions  $\tilde{\Xi}_g^t$  and  $\hat{\Gamma}_g^t$  is still convex and constraints are linear, subproblem  $\mathbf{P}_2$  is transformed into a convex optimization subproblem  $\mathbf{P}'_2$ , which can be expressed as

$$\mathbf{P}'_2 : \min_{\{n_g^t\}_{g \in \Omega_t}} \frac{1}{\Lambda_t^*} \sum_{g \in \Omega_t} \tilde{\Xi}_g^t + \hat{\Gamma}_g^t \quad (3.27a)$$

$$\text{s.t. } n_g^t > \hat{b}_g^t, \quad (3.27b)$$

(3.20b), (3.20c), and (3.20d).

Based on the linear approximation on the concave part, the original optimization problem  $\mathbf{P}_0$  is transformed into a convex optimization problem.

### 3.4.2 Fast Resource Reservation Algorithm

Since the local optimal point is also the global optimal point for a convex optimization problem, we design a FS algorithm to find the local optimal bandwidth and VM instance reservation variables, i.e.,  $\{m_g^t\}_{g \in \Omega_t}$  and  $\{n_g^t\}_{g \in \Omega_t}$ . Specifically, each multicast group is first assigned  $\lceil b_g^t \rceil$  bandwidths and  $\lceil \hat{b}_g^t \rceil$  VM instances. Then, in each iteration, the unassigned bandwidths and VM instances are sequentially assigned to the multicast group that can obtain the highest values of objective functions in subproblems  $\mathbf{P}'_1$  and  $\mathbf{P}'_2$ , respectively. If the objective function value in the previous iteration is higher than that in the current iteration, the iteration process is terminated and local optimal resource reservation variables are the variables in the previous iteration. The specific algorithm is presented in Algorithm 1.

---

**Algorithm 1:** Fast Scheduling (FS) Algorithm

---

```

1 Initialize the objective functions  $U^t(\mathbf{P}'_1)$  and  $U^t(\mathbf{P}'_2)$ .
2 Input  $\Omega_t$ ,  $\{\ell_g^t\}_{g \in \Omega_t}$ ,  $\{R_g^t\}_{g \in \Omega_t}$ ,  $\{O_g^t\}_{g \in \Omega_t}$ ,  $\lceil b_g^t \rceil$ ,  $\lceil \widehat{b}_g^t \rceil$ ,  $M$ ,  $N$ ,  $\Lambda_{t-1}^*$ , and all weighted
   parameters.
3 Output  $\{m_g^t\}_{g \in \Omega_t}$  and  $\{n_g^t\}_{g \in \Omega_t}$ .
4 for  $g \in \Omega_t$  do
5     Multicast group  $g$  is assigned  $\lceil b_g^t \rceil$  bandwidths and  $\lceil \widehat{b}_g^t \rceil$  VM instances;
6     Variables  $m_g^t$  and  $n_g^t$  take the values of  $\lceil b_g^t \rceil$  and  $\lceil \widehat{b}_g^t \rceil$ , respectively;
7 end
8 Calculate the number of unassigned bandwidths and VM instances, i.e.,
    $\widetilde{M} = M - \sum_{g \in \Omega_t} \lceil b_g^t \rceil$  and  $\widetilde{N} = N - \sum_{g \in \Omega_t} \lceil \widehat{b}_g^t \rceil$ ;
9 for  $m = 1 : \widetilde{M}$  do
10    for  $g \in \Omega_t$  do
11        Assign one bandwidth to multicast group  $g$ , and calculate  $U^t(\mathbf{P}'_{1,g})$ ;
12    end
13    Update variable  $m_g^t$  by  $m_g^t = m_g^t + 1$  with the minimum value of the objective
       function, i.e.,  $g^* = \arg \min_g U^t(\mathbf{P}'_{1,g})$ ;
14    if  $U^t(\mathbf{P}'_1)^{(m)} \geq U^t(\mathbf{P}'_1)^{(m-1)}$  then
15        Stop the iteration and return  $\{m_g^t\}_{g \in \Omega_t}$ ;
16    end
17 end
18 for  $n = 1 : \widetilde{N}$  do
19    for  $g \in \Omega_t$  do
20        Assign one VM instance to multicast group  $g$ , and calculate  $U^t(\mathbf{P}'_{2,g})$ ;
21    end
22    Update variable  $n_g^t$  by  $n_g^t = n_g^t + 1$  with the maximum value of the objective
       function, i.e.,  $g^* = \arg \max_g U^t(\mathbf{P}'_{2,g})$ ;
23    if  $U^t(\mathbf{P}'_2)^{(n)} \geq U^t(\mathbf{P}'_2)^{(n-1)}$  then
24        Stop the iteration and return  $\{n_g^t\}_{g \in \Omega_t}$ ;
25    end
26 end

```

---

Table 3.1: Simulation parameters

| Parameter | Value         | Parameter       | Value | Parameter  | Value |
|-----------|---------------|-----------------|-------|------------|-------|
| $M$       | 15            | $T$             | 5 min | $\varpi_1$ | 0.5   |
| $B$       | 2 MHz         | $\rho$          | 50    | $\varpi_2$ | 0.5   |
| $N$       | 10            | $C$             | 8     | $\varpi_3$ | 0.7   |
| $\omega$  | 2 G Cycles/s  | $F_1$           | 150   | $\varpi_4$ | 1     |
| $K$       | 60            | $F_2$           | 5     | $\delta_1$ | 1.5   |
| $V$       | 1000          | $\lambda$       | 0.4   | $\delta_2$ | 0.3   |
| $\mu$     | 2 G Cycles/Mb | $\hat{\lambda}$ | 0.3   | $\delta_3$ | 0.3   |

### 3.4.3 Computational Complexity Analysis

The proposed FS algorithm needs to find the local optimal points for bandwidth and VM instance reservation. The analysis of computational complexity is as follows. First, the computational complexity of initial resource reservation for each multicast group is  $O(\Lambda_t^*)$ . Then, the computational complexity of assigning the rest bandwidths is  $O(\tilde{m}\Lambda_t^*)$ , where  $\tilde{m}$  is a positive value less than  $\tilde{M}$  since the bandwidth assignment can be momentarily terminated before all bandwidths are completely assigned. Next, the computational complexity of assigning the rest VM instances is  $O(\tilde{n}\Lambda_t^*)$ , where  $\tilde{n}$  is a positive value less than  $N$  since the VM instance assignment can be momentarily terminated before all VM instances are completely assigned. Finally, the overall computational complexity of FS algorithm is  $O(\Lambda_t^* + \tilde{m}\Lambda_t^* + \tilde{n}\Lambda_t^*)$ .

## 3.5 Simulation Results

In this section, we conduct extensive experiments to validate the performance of DA-based resource reservation scheme.

### 3.5.1 Simulation Setup

The main simulation parameters are presented in Table 3.1. The key components of the simulation are introduced as follows.

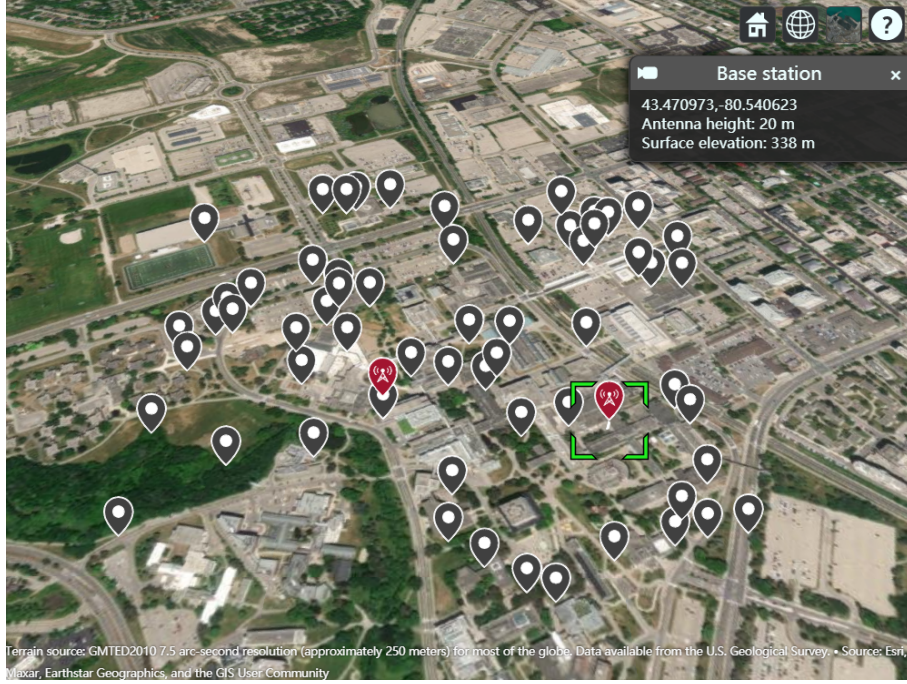


Figure 3.5: The simulation scene, where BSs and users are represented by red and gray icons, respectively.

We adopt the short video streaming dataset<sup>4</sup> to obtain users' swipe behaviors and the user satisfaction dataset<sup>1</sup> to fit the user satisfaction function. We sample 1000 short videos from the YouTube 8M dataset<sup>5</sup>, which includes 8 video types, i.e., Entertainment, Games, Food, Sports, Science, Dance, Travel, and News. Each video has a duration of 15 *sec* and is encoded into four versions by the H. 265 encoder. We consider the scenario where two BSs are deployed at the University of Waterloo (UW) campus and users' initial positions are randomly and uniformly generated around two BSs, as shown in Fig. 3.5. Each user moves along a prescribed path within the UW campus at a speed of 2~5 *km/h*, and the corresponding channel path loss is obtained by the propagationModel at Matlab. The transmission power and noise power are set to 27 dBm and -174 dBm, respectively.

Since the dimensions of different user DA data attributes are different, we construct four Autoencoder models for data compression. Take the Autoencoder model for the

<sup>4</sup>ACM MM Grand Challenges: <https://github.com/AItransCompetition/Short-Video-Streaming-Challenge/tree/main/data>

<sup>5</sup>YouTube 8M dataset: <https://research.google.com/youtube8m/index.html>

compression of locations as an example, the encoder is composed of two Conv2D layers, each using a ReLU activation function and a 'same' padding. The first Conv2D layer has 32 filters with a kernel size of (1, 3), while the second Conv2D layer has 64 filters with the same kernel size. After the Conv2D layers, a Flatten layer is applied to transform the multi-dimensional tensor into a one-dimensional tensor. The flattened tensor is then passed through a Dense layer with 60 neurons and a linear activation function, resulting in a compressed representation of input data in the latent space. The decoder is designed to reconstruct input data from the latent space representation. It starts with a Dense layer having  $60 \times 150 \times 64$  neurons, followed by a Reshape layer to convert the tensor back to the shape (60, 150, 64). Then, a Conv2DTranspose layer with 32 filters, kernel size (1, 3), ReLU activation function, and 'same' padding is applied. Finally, the output layer is another Conv2DTranspose layer with 2 channels (matching the input data), kernel size (1, 3), linear activation function, and 'same' padding. The compressed DA data is then input to DDQN to determine the clustering number. The detailed Autoencoder and DDQN parameters are shown in Tables 3.2 and 3.3, respectively.

We compare the proposed DA-assisted resource reservation scheme with the following benchmark schemes:

- Without DA (WDA): Multicast groups are constructed and updated based on users' preferences and locations. The resource demand prediction relies on historical video traffic distribution, while the user satisfaction model selects the same model structure as the proposed DA-assisted resource reservation scheme. Bandwidth and computing resource reservations are based on predicted resource demands without considering users' swipe behaviors.
- Density-based spatial clustering of applications with noise [96] with FS algorithm (DBSCAN-FS): User DAs are first constructed to store users' historical data, but the user clustering is based on the DBSCAN scheme. The resource demand prediction, user satisfaction model, and resource scheduling algorithm are the same as the proposed DA-assisted resource reservation scheme.
- DA-based user clustering, and branch- and bound-based scheduling algorithm (DA-BBS): User DAs are first constructed to store users' historical data. The user clustering, resource demand prediction, and user satisfaction model are the same as the proposed DA-assisted resource reservation scheme. However, the resource scheduling is based on the 0-1 branch and bound method [97] that can obtain the optimal scheduling decisions but with high computational complexity and storage consumption.

Table 3.2: Autoencoder parameters

| Model                                                                                | Layer name      | NN units         | Activation | Padding |
|--------------------------------------------------------------------------------------|-----------------|------------------|------------|---------|
| Compression<br>for channel<br>conditions,<br>swipe<br>timestamps,<br>and preferences | Conv1D          | 32, 3            | ReLu       | same    |
|                                                                                      | Conv1D          | 64, 3            | ReLu       | same    |
|                                                                                      | Dense           | 60               | Linear     | /       |
|                                                                                      | Dense           | $60 \times 64$   | ReLu       | /       |
|                                                                                      | Conv1DTranspose | 32, 3            | ReLu       | same    |
|                                                                                      | Conv1DTranspose | 150/9e3/160, 3   | Linear     | same    |
| Compression<br>for locations                                                         | Conv1D          | 32, $1 \times 3$ | ReLu       | same    |
|                                                                                      | Conv1D          | 64, $1 \times 3$ | ReLu       | same    |
|                                                                                      | Dense           | 60               | Linear     | /       |
|                                                                                      | Dense           | 5.76e5           | ReLu       | /       |
|                                                                                      | Conv1DTranspose | 32, $1 \times 3$ | ReLu       | same    |
|                                                                                      | Conv1DTranspose | 2, $1 \times 3$  | Linear     | same    |

Table 3.3: DDQN parameters

| Parameter          | Value | Parameter                | Value                                           |
|--------------------|-------|--------------------------|-------------------------------------------------|
| Memory size        | 2000  | Initial exploration rate | 1                                               |
| Discount rate      | 0.95  | Exploration decay rate   | 0.995                                           |
| Episode length     | 90    | NN layer connection      | FC                                              |
| Number of Episodes | 300   | Number of hidden layers  | 3                                               |
| Learning rate      | 0.001 | Activation function      | ReLU                                            |
| Mini-batch size    | 32    | Number of neurons        | $512 \times 256 \times 128 \times 64 \times 10$ |

- DA-based user clustering, and branching dueling Q-network-based scheduling algorithm (DA-BDQN): User DAs are first constructed to store users' historical data. The user clustering, resource demand prediction, and user satisfaction model are the same as the proposed DA-based resource reservation scheme. However, the resource scheduling is based on the BDQN algorithm [98] that can efficiently solve the high-dimensional resource scheduling problem by splitting the action space.

### 3.5.2 Clustering Performance Evaluation

In this section, we will compare the clustering number, the user density in each multicast group, and the convergence performance of proposed clustering algorithm, respectively.

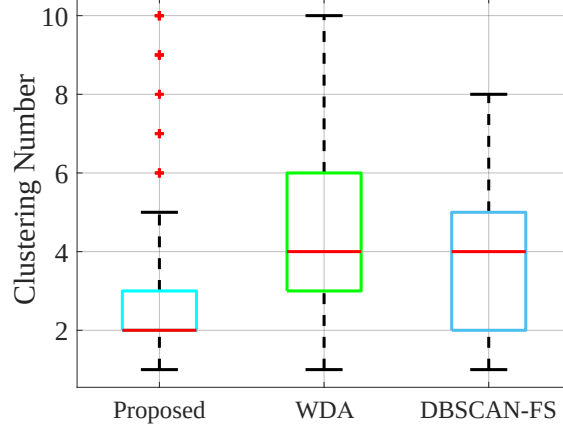


Figure 3.6: Clustering number comparison.

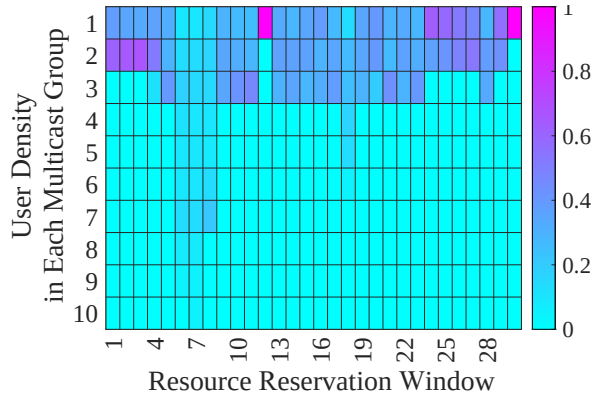


Figure 3.7: User density in each multicast group.

As shown in Fig. 3.6, we present the clustering number distribution in the boxplot for 90 resource reservation windows. It can be observed that the proposed scheme can achieve lower first-quartile, median, third-quartile, and maximum values compared with other schemes. Our proposed scheme demonstrates superior performance with relatively minimal variations, while the WDA scheme exhibits a larger fluctuation. This can be attributed to the unique capability of DAs to effectively extract users' swipe probability distributions. Consequently, our proposed scheme empowered by DAs can well adapt to changes in swipe behaviors and network conditions. Overall, the proposed scheme offers a more robust and efficient solution for managing multicast groups in the face of dynamic changes in user statuses.

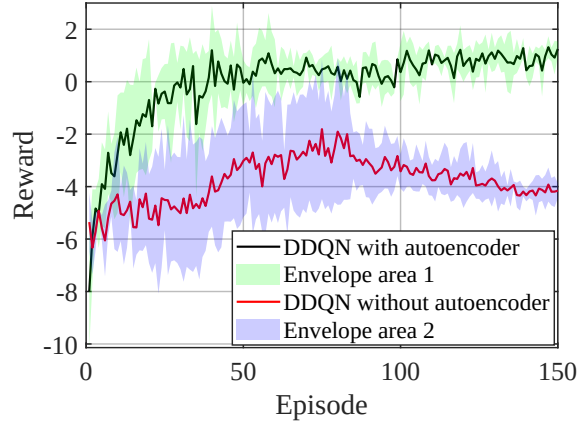


Figure 3.8: A comparison between DDQN training with autoencoder and without autoencoder.

As illustrated in Fig. 3.7, we present the user density of proposed DA-assisted clustering scheme during each resource reservation window. The darker the color, the higher the user density, and vice versa. Overall, the variation trend of user density in each multicast group is relatively gradual. This can be attributed to two primary reasons. First, reconfiguring bandwidth and computing resources for a multicast group entails additional network overheads. Second, to accommodate diverse user demands and preferences, the number of users within a multicast group is inherently limited. Furthermore, we observe that certain multicast groups exhibit significantly higher user density than others. This phenomenon arises because the majority of users own similar characteristics. By clustering the similar users for multicast transmission and video transcoding, network traffic burden can be effectively relieved.

As shown in Fig. 3.8, we present the convergence curve of the DDQN-based clustering algorithm with and without autoencoder for determining the number of multicast groups. We conducted three trials of training to draw the corresponding envelope area and mean curve. Each episode consists of 90 steps, and the corresponding reward is the average reward for all steps within an episode. It can be observed that as the number of episodes increases, the reward of DDQN with autoencoder gradually grows larger. When the number of episodes approaches nearly 70, the reward of DDQN with autoencoder converges to a stable state, indicating that the DDQN-based clustering algorithm with autoencoder can effectively extract user similarity from user statuses to determine the number of multicast groups. The reason is that the autoencoder can compress high-dimensional user DA data into a low-dimensional latent space through neural network training, and minimize recon-

struction error through a loss function. The lower-dimensional latent space can capture essential intrinsic features of user DA data, which can help DDQN algorithms efficiently mine user similarity for user clustering.

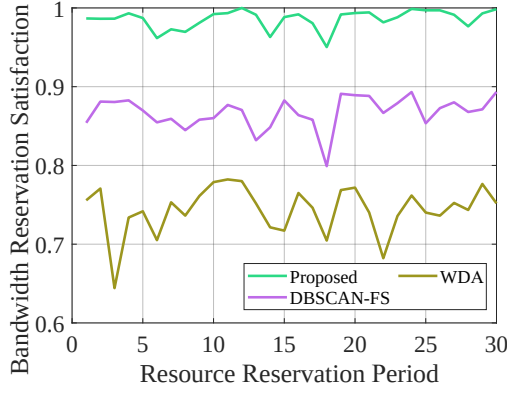
### 3.5.3 System Utility Performance Evaluation

As illustrated in Fig. 3.9, we first present the performance comparison of various system utility components under different schemes. With respect to bandwidth and VM reservation satisfaction, the proposed scheme can achieve a relatively high satisfaction level. Especially, bandwidth reservation satisfaction exhibits less fluctuation compared with VM reservation satisfaction. This is because the number of reserved VMs directly affects the speed of video transcoding, leading to variations in video quality and buffer length. Consequently, when the number of reserved VMs is insufficient, user satisfaction tends to fluctuate more significantly. In comparison to other schemes, WDA has the lowest satisfaction level. This is attributed to the lack of user DAs, which disables the network controller from swiftly and accurately analyzing users' similarities from their historical data to precisely construct multicast groups. As a result, bandwidth and VM instance reservation cannot meet the actual resource demands of each multicast group, thereby leading to lower user satisfaction.

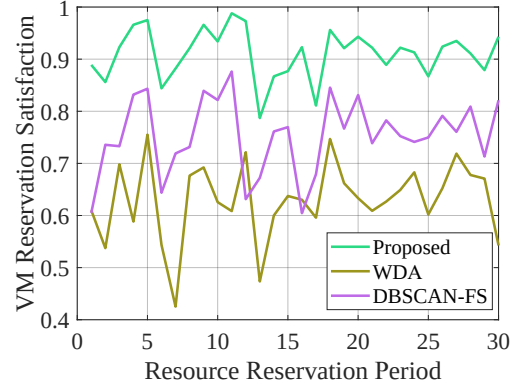
In terms of bandwidth and VM operation cost, WDA and DBSCAN-FS schemes can achieve a lower resource operation cost compared with the proposed DA-assisted resource reservation scheme, as shown in Fig. 3.9(c)(d). This is because these two schemes mainly rely on video traffic distribution to predict each multicast group's resource demand without incorporating the impact of users' swipe behaviors. In each resource reservation window, resources are reserved as close as possible to the predicted resource demands, which can effectively reduce resource operation cost. However, user swipe behaviors are dynamic and differentiated, which can cause frequent resource demand fluctuations. Without considering the impact of users' swipe behaviors, the resource demand prediction can be not accurate enough, which can cause frequent resource reconfiguration and rebuffering events. Therefore, although these two schemes can achieve a lower resource operation cost, they also sacrifice more resource reconfiguration costs and degrade user QoE.

## 3.6 Summary

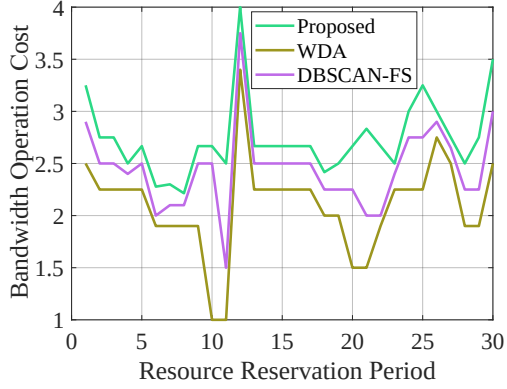
In this chapter, we have studied a novel resource management issue to enhance the multicast short video streaming performance. We have proposed a DA-assisted resource reservation



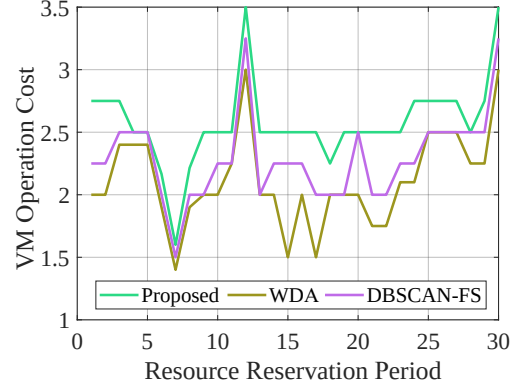
(a) Bandwidth Reservation Satisfaction



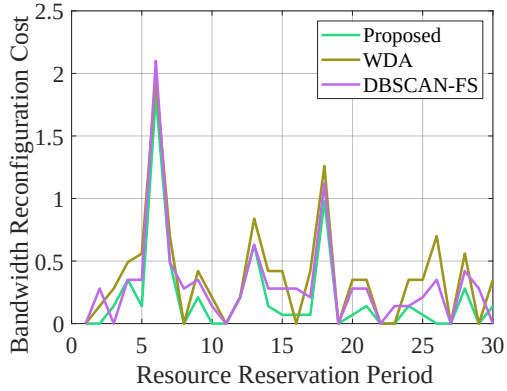
(b) VM Reservation Satisfaction



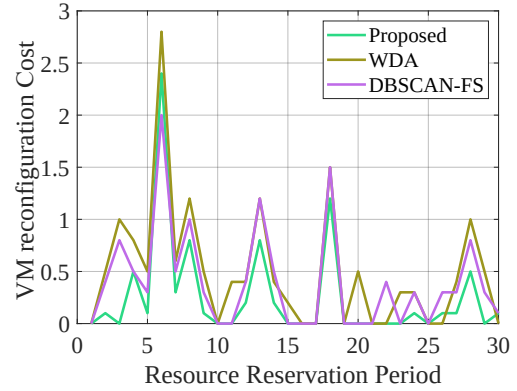
(c) Bandwidth Operation Cost



(d) VM Operation Cost



(e) Bandwidth Reconfiguration Cost



(f) VM Reconfiguration Cost

Figure 3.9: The performance comparison of each component in the system utility.

scheme to abstract the swipe probability distribution and recommended video list for the bandwidth and computing resource demand prediction. Furthermore, we have proposed a user satisfaction model by taking the user's personalized preference and service sensitivity into account. A low-complexity resource scheduling algorithm has been designed to determine the joint bandwidth and computing resource reservation.

## Chapter 4

# Digital Agent-Based Resource Allocation for QoE-Driven Multicast Short Video Streaming

In this chapter, we propose a DA-based network management scheme to enhance QoE. Firstly, user status emulated by the DA is utilized to estimate the transmission capabilities and watching probability distributions of sub-multicast groups for adaptive segment buffering. The sub-multicast groups' buffers are aligned to the unique virtual buffers managed by the DA for a fine-grained buffer update. Then, a multicast QoE model consisting of re-buffering time, video quality, and quality variation is developed, by considering the mutual influence of segment buffering among sub-multicast groups. Finally, a joint optimization problem of segment version selection and slot division is formulated to maximize QoE. To efficiently solve the problem, a data-model-driven algorithm is proposed by integrating a convex optimization method and a deep reinforcement learning algorithm. Simulation results based on the real-world dataset demonstrate that the proposed DA-based network management scheme outperforms benchmark schemes in terms of QoE improvement.

### 4.1 Background and Motivations

Short video platforms such as TikTok, Instagram Reels, and YouTube Shorts have experienced a dramatic surge in user scale, with TikTok's monthly active users reaching 1.7 billion in 2023. Seamless video requests bring a huge traffic and computing burden to communication networks, which cause frequent playback lags and video quality fluctuation, especially

in areas with high user density, thereby deteriorating users' watching experience [99]. Multicast transmission, as an essential technology in wireless networks, can enable a single data stream to be disseminated to numerous users in a group simultaneously. By multicasting short videos to the users with similar characteristics and geographical locations, the redundancy in video transcoding and transmission can be effectively reduced, thereby alleviating the network traffic and computing burden [40].

For multicast short video streaming services, one of the key metrics to evaluate the performance is QoE. It is typically a multi-dimensional metric used to quantify users' subjective and objective watching experience [100]. For instance, video quality (resolution, bitrate), latency, as well as rebuffering events are the most important factors that dominate the QoE level [101]. To maintain users' QoE at a high level, efficient network management is essential, such as buffer management and resource scheduling [102, 103]. Specifically, due to the limited bandwidth and computing resources, it needs to be determined which versions and quantities of segments (sequential components of a video sequence) should be transmitted to users' buffers, and at what priority levels. Furthermore, due to asynchronous swipe behaviors in one multicast group, one multicast group can be further divided into multiple sub-multicast groups. The sub-multicast group with a leading video playback is defined as the leading sub-multicast group, otherwise, the lagging sub-multicast group. The bandwidth and computing resources need to be flexibly and accurately assigned to each sub-multicast group to reduce service delay, which consists of multicast transmission delay and transcoding delay. Nevertheless, existing network management schemes are mainly based on the sequential buffering principle and general QoE model while neglecting the impact of users' swipe behaviors and the mutual influence of sub-multicast groups' segment buffering. As a result, users may suffer from frequent playback lags and low video quality, thus leading to a low QoE.

DA is a promising technique to optimize network management for better QoE. As an essential component embedded in the next-generation communication networks, DA is comprised of a data pool as well as several data processing and decision-making modules, which can efficiently emulate user behaviors and network conditions, abstract distilled features, and make network management decisions [104, 105]. Owing to its powerful emulation, analysis, and decision-making capabilities, communication networks can more intelligently perceive sub-multicast groups' behavior patterns, finely control sub-multicast groups' buffer update, and provide customized network management strategies to enhance user QoE. In this work, the precise role of constructed DA is to emulate users' future network conditions and swipe behaviors, abstract the watching probability distribution of segments, and make tailored network management decisions based on the emulated user status and abstracted information.

The motivation of the DA-based network management for multicast short video streaming includes three aspects. Firstly, due to users' stochastic swipe behaviors, users' viewing sequences are usually non-sequential. The existing sequential buffering principle can cause the segments to be swiped to not being buffered in time, resulting in playback lags. Moreover, users' buffer lengths are usually overestimated due to multicast segment buffering, which can bring inaccurate rebuffering time estimation. Therefore, it is essential to develop an efficient segment buffering scheme that adapts well to users' swipe behaviors. Secondly, the QoE model is typically composed of multiple factors. Due to the impact of multicast transmission and segment buffering priority, different QoE factors among sub-multicast groups influence each other, collectively impacting the multicast group's QoE. Therefore, it is paramount to establish a QoE model specifically tailored for multicast short video streaming. Thirdly, since network management is usually a multi-variable decision-making problem, such as the segment version selection and resource scheduling, the interplay among the variables makes the optimization problem complex and difficult to solve. Therefore, how to design an efficient algorithm to solve it for high user QoE is important.

Designing an efficient DA-based network management scheme needs to address the following challenges: 1) incorporating the impact of users' swipe behaviors in multicast segment buffering; 2) establishing an accurate multicast QoE model; 3) designing an efficient algorithm to solve the complex multi-variable decision-making problem. Specifically, users' swipe behaviors are stochastic and spatiotemporally varied, which are difficult to accurately predict in real time. Therefore, how to conduct effective data abstraction to obtain the distilled swipe feature and utilize it to facilitate accurate multicast segment buffering is challenging. Furthermore, since the lagging sub-multicast group can still receive segments from other leading sub-multicast groups in its scheduling slot, this interactivity results in complex QoE estimation. Therefore, how to characterize the impact of multicast segment buffering among sub-multicast groups on the QoE model is challenging. Finally, since the network management problem is usually a mixed-integer non-convex problem, directly using model-based or data-driven algorithms can lead to a loss in the system performance. Therefore, how to ingeniously combine the advantages of model-based and data-driven algorithms to efficiently solve the formulated problem is challenging.

In this paper, we propose a DA-based network management scheme, which can effectively enhance user QoE. The main contributions are summarized as follows:

- Firstly, we propose a novel DA-assisted buffer management scheme to incorporate the impact of swipe behaviors. Specifically, users' historical status, including locations, channel conditions, preferences, and swipe timestamps, is stored in the DA for status emulation. The emulated status is used to abstract the sub-multicast groups' trans-

mission capabilities and the watching probability distribution of segments. Based on the abstracted information, DA can make an adaptive segment buffering decision. Furthermore, DA is utilized to construct and manage virtual buffers for each sub-multicast group for a fine-grained buffer update.

- Secondly, we establish a multicast QoE model to quantify the impact of multicast segment buffering among sub-multicast groups. Specifically, the multicast QoE model is built as a weighted sum of rebuffering time, video quality, and quality variation, where the weighting factors are the integration of buffering order and users' sensitivity degrees. The rebuffering time estimation relies on the multicast transmission delay as well as the parallel transcoding process. The video quality and quality variation depend on the relationship between the segment version and Structural Similarity Index Measure (SSIM). Based on these elements, an accurate multicast QoE model is established.
- Thirdly, we formulate a joint optimization problem of segment version selection and slot division to maximize QoE. Since the formulated problem is a mixed-integer non-linear programming problem, it is hard to directly use a model-based or data-driven algorithm to solve it. Therefore, we propose a data-model-driven algorithm. Specifically, a convex optimization method is embedded in a DRL algorithm to decouple the joint optimization problem and reduce the action dimension, which can efficiently solve the formulated problem. The extensive simulation results on real-world short video streaming datasets show that the proposed DA-based network management scheme can effectively enhance QoE as compared with state-of-the-art network management schemes.

The remainder of this chapter is organized as follows. The system model, including a DA-based resource allocation framework, DA construction, DA-based segment buffering and update, and multicast QoE model, is presented in Section 4.2. The problem formulation and the proposed scheduling algorithm are shown in Sections 4.3 and 4.4, followed by simulation results in Section 4.5. Finally, Section 4.6 summarizes this chapter.

## 4.2 System Model

In this section, we first propose a DA-based resource allocation framework. Then, we present how to construct DAs and utilize them to realize accurate segment buffering and buffer updates. Finally, a multicast QoE model is established.

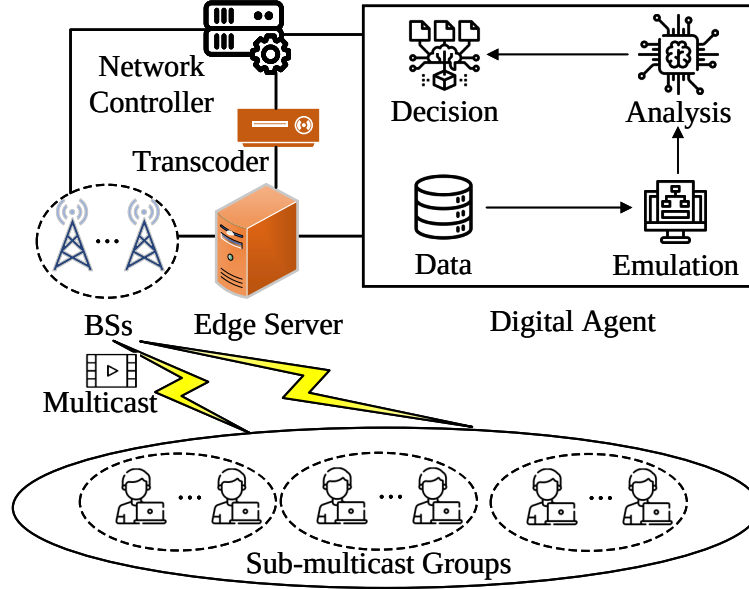


Figure 4.1: DA-based resource allocation framework for multicast short video streaming.

#### 4.2.1 Digital Agent-Based Resource Allocation Framework

As shown in Fig. 4.1, we consider a DA-assisted MSVS scenario, which consists of an Access Point (AP), multiple multicast groups, and one DA.

- **Access Point:** The AP owns communication, computing, and caching capabilities. Based on users' requests, cached video sequences will be transcoded to appropriate bitrates and then transmitted to each multicast group. In addition, it is responsible for collecting users' network-related and behavior-related information to update DA data.
- **Multicast Group:** Each one consists of multiple users using short video services. The same video sequences will be transmitted from the AP to one multicast group over shared wireless channels. The multicast group's construction and update mainly depend on users' similarities in swipe behaviors, channel conditions, locations, and preferences. Due to asynchronous swipe behaviors, users' devices in the same multicast group still have different playback stamps and buffer lengths. Therefore, one multicast group is further divided into multiple sub-multicast groups, denoted by  $\mathcal{G} = \{1, \dots, G\}$ . The number of total users in the system is denoted by  $K$ . The set of users in sub-multicast group  $g$  is denoted by  $\{\mathcal{K}_g\}_{g \in \mathcal{G}}$ .

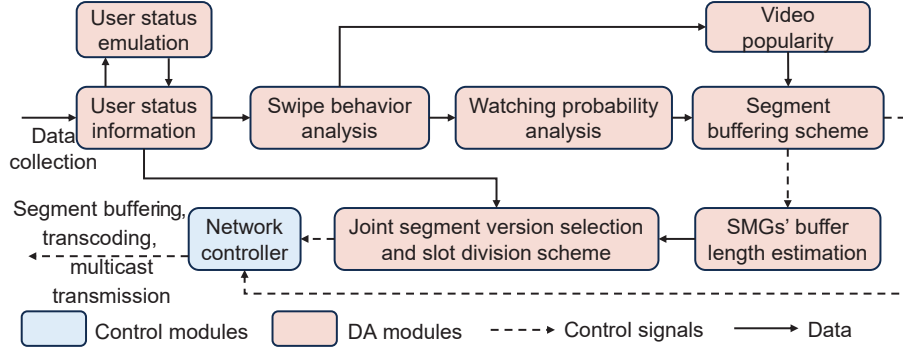


Figure 4.2: Proposed DA-assisted multicast short video streaming workflow.

- **Digital Agent:** It consists of a database storing users' status information, a status emulation module, as well as multiple data analysis and decision-making modules. DA data comes from two aspects, i.e., data measurement and emulation module. The former is used to obtain the label data to detect whether the emulated data is accurate enough and supplement DA data when emulated data has a large deviation. The latter uses machine learning-based algorithms to predict users' future status. The data analysis module is responsible for abstracting distilled data features, such as swipe probability distribution, request density, user satisfaction, etc, which can facilitate tailored network management in the decision-making module.

As shown in Fig. 4.2, we present the proposed DA-assisted multicast short video streaming workflow. Specifically, the user status emulation module can generate user status information. When the generated data has a significant deviation from the actual user status data, new actual data is collected to correct user status information. The user status information is first utilized to abstract the user's swipe feature, such as swipe probability distribution, which can update the video popularity and watching probability distribution of segments. The updated information is then used to make the tailored segment buffering and sub-multicast groups' buffer update decisions. Next, sub-multicast groups' buffer lengths are estimated, which are integrated with user status information to make a joint segment version selection and slot division decision. Finally, all network management decisions are transferred to the network controller to guide the physical entities to implement.

The main notations and their definitions are listed in Table 4.1.

Table 4.1: Notations and definitions

| Notation     | Definition                                                                       |
|--------------|----------------------------------------------------------------------------------|
| $v_{i,j}$    | The recommended segment $j$ in video $i$                                         |
| $w_{i,j}$    | The watching probability of segment $i$ in video $i$                             |
| $p_{i,j}$    | The swipe probability of segment $i$ in video $i$                                |
| $\Lambda_i$  | The number of segments in video $i$                                              |
| $T_s$        | The scheduling slot length                                                       |
| $\tau$       | The time duration of one segment                                                 |
| $r_{g,k}$    | The transmission capability of user $k$ in sub-multicast group $g$               |
| $q_g^f$      | The length of buffer $f$ in sub-multicast group $g$                              |
| $D_g$        | The transmission delay of sub-multicast group $g$                                |
| $S_g$        | The service delay of sub-multicast group $g$                                     |
| $Q_g$        | The video quality of newly buffered segments in sub-multicast group $g$          |
| $V_g$        | The video quality variation between adjacent segments in sub-multicast group $g$ |
| $\Upsilon_g$ | The multicast QoE of sub-multicast group $g$                                     |

### 4.2.2 Digital Agent Construction

DA consists of multiple modules, which are summarized as status emulation, data analysis, and decision-making, as shown in Fig. 4.2.

Firstly, since network conditions and user behaviors are essential to reflect users' characteristics and requirements, we utilize the AP to collect users' data from two aspects, i.e., networking-related data and behavior-related data. The networking-related data include users' channel conditions and locations, which are utilized to estimate the transmission capabilities. The behavior-related data consists of users' swipe timestamps and preferences. The AP accumulates network-related data primarily through channel measurements, while behavior-related data is collected relying on users' periodical upload. To reduce data collection costs while maintaining data freshness, a higher data collection frequency is implemented on networking-related data compared with behavior-related data. Furthermore, the LSTM network is utilized to mine data correlation, which can effectively emulate users' future networking-related data and behavior-related data. This operation can effectively

reduce frequent data interaction costs between DA and users. By integrating these two kinds of data, DA can accurately emulate users' real-time status for network management.

Secondly, two kinds of data analysis modules are embedded in the DA, i.e., the swipe behavior analysis module and the watching probability analysis module. The former has been investigated in Chapter 3 for resource demand prediction. The latter is our focus in this Chapter, which aims at assisting the buffer management. Specifically, due to the impact of sequential playback and swipe behaviors, each segment's watching probability has a dependable relationship. From the aspects of the first and subsequent segments' swipe probabilities, the watching probability distribution of segments is derived in DA. Based on the derived information, DA can realize an accurate buffer management.

Thirdly, based on the emulated status and analyzed data, DA can realize tailored decision-making to further enhance user QoE. Three kinds of decision-making modules are designed to make the segment buffering scheme, the sub-multicast groups' buffer update scheme, and the joint segment version selection and slot division scheme, respectively. These modules are intricately coupled, where the output of one module seamlessly transitions into the input of the following module, collectively influencing QoE. A data-model-driven algorithm is proposed to decouple the joint optimization problem and reduce the action dimension for efficient network management.

DA is considered as a secure third-party platform between AP and video service providers, which can collect both networking-related data and behavior-related data. The collected behavior-related data by DA is used to analyze the statistical information of multicast groups instead of direct utilization of individual behaviors. The collected network-related information is used to estimate multicast groups' transmission capabilities and update multicast groups, which is not shared with short video streaming applications. The fundamental role of DA is to emulate users' future status, abstract the watching probability distribution of segments, and make tailored network management decisions.

### 4.2.3 Digital Agent-Based Segment Buffering

Due to users' diversified swipe behaviors, the segments to be watched do not follow a sequential order, which can cause different watching probabilities. To reduce rebuffering events, we need to analyze which segments should be prioritized for buffering. Therefore, we design a DA-assisted segment buffering scheme. The specific analysis is as follows.

As shown in Fig. 4.3, DA recommends a video list for the multicast group in each large timescale. Each video sequence consists of multiple segments with the same time length. The recommended video list is denoted by  $\{v_{i,j}\}_{i \in \mathcal{N}, j \in \{1, \dots, \Lambda_i\}}$ , where  $\mathcal{N}$  is the recommended

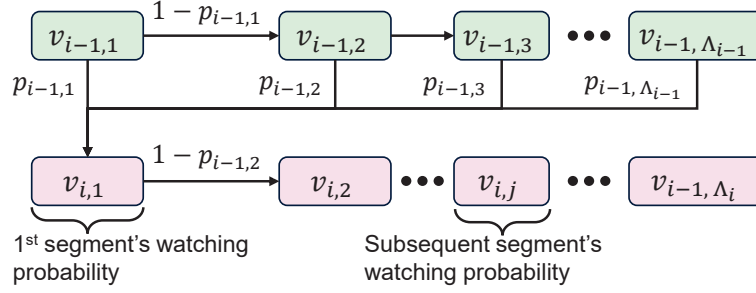


Figure 4.3: Watching probability analysis.

video index list, and  $\Lambda_i$  is the number of segments in video  $i$ . Assuming that there are no quit behaviors during video watching. Since the probability of watching next segment  $j+1$  only depends on the current state [84], i.e., the watching probability of current segment  $j$  and the swipe probability from  $j$  to  $j+1$ , we estimate the watching probability distribution of multicast group by modeling the Markovian transition on users' swipe behaviors. Let  $w_{i,j}$  represent the watching probability of segment  $v_{i,j}$ . For any two successive segments  $j$  and  $j+1$  in video  $i$ , the watching probability of segment  $j+1$  is derived as

$$w_{i,j+1} = w_{i,j}(1 - p_{i,j}), \forall i \in \mathcal{N}, j \in \{1, \dots, \Lambda_i - 1\}, \quad (4.1)$$

where  $p_{i,j}$  is the swipe probability of segment  $v_{i,j}$ . Based on Eq. (4.1), we further analyze the watching probability distribution from the perspectives of 1<sup>st</sup> and subsequent segments.

For the 1<sup>st</sup> segment of video  $i$ , its watching probability is related to the swipe probability of all segments in previous video  $i-1$ . Therefore, the corresponding watching probability,  $w_{i,1}$ , is expressed as

$$w_{i,1} = w_{i-1,1} \left( p_{i-1,1} + \sum_{j=1}^{\Lambda_{i-1}-1} p_{i-1,j+1} \prod_{k=1}^j (1 - p_{i-1,k}) \right). \quad (4.2)$$

For the subsequent segment  $j$  of video  $i$ , its watching probability is only related to the swipe probability of previous segment  $j-1$ . Therefore, the corresponding watching probability,  $w_{i,j}$ , is given by

$$w_{i,j} = w_{i,1} \prod_{k=1}^{j-1} (1 - p_{i,k}), \forall j \in \{2, \dots, \Lambda_i\}. \quad (4.3)$$

Based on the watching probability distribution, we can determine the segment buffering order from high to low. Furthermore, we analyze how many segments should be added to

the multicast group's buffer in each scheduling slot, which needs to satisfy two requirements, i.e., 1) buffer requirement: buffer minimum segments to avoid buffer length empty; 2) resource requirement: assign all bandwidth and computing resources to each sub-multicast group to observe how many segments can be buffered.

For the buffer requirement, we need to guarantee the buffer length is larger than the scheduling slot length for each sub-multicast group. Therefore, we can have

$$n_{\text{buffer}} = \sum_{g \in \mathcal{G}} \left[ \frac{T_s - q_g}{\tau} \right]^+, \quad (4.4)$$

where  $T_s$  and  $\tau$  represent the scheduling slot length and the segment length, respectively. Here,  $q_g$  represents sub-multicast group  $g$ 's buffer length for current watching video sequence and function  $[x]^+ = \max\{x, 0\}$ , respectively.

For the resource requirement, we analyze the multicast transmission mechanism among sub-multicast groups, as shown in Fig. 4.4. Each video sequence consists of multiple segments, represented by different colors. The index orders of sub-multicast groups are consistent with the video viewing progress, sorted from low to high. Sub-multicast groups can be classified into lagging sub-multicast groups and leading sub-multicast groups based on their video viewing progress. The lagging sub-multicast groups can receive the video segments from leading sub-multicast groups in their scheduling slot through multicast transmission in a multicast group, while the leading sub-multicast groups do not receive the video segments from the lagging sub-multicast groups in their scheduling slot. Through this way, the video segments can be buffered in advance for lagging sub-multicast groups, which can avoid repeated segment buffering in the future scheduling slot. Therefore, the transmission capability of sub-multicast group  $g + 1$  needs to consider users' channel conditions both from itself and sub-multicast group  $g$ . Based on the above analysis, the maximum buffered segments within all reserved bandwidth and computing resources need to satisfy the following requirements, i.e.,

$$\begin{cases} \sum_{m=1}^{\tilde{n}_g^B} \sum_{l=1}^{\bar{l}} z_{g,m}^l \leq \min_{k \in \bigcup_{d=1}^g \mathcal{K}_d} T_s r_{g,k} \leq \sum_{m=1}^{\tilde{n}_g^B+1} \sum_{l=1}^{\bar{l}} z_{g,m}^l, \forall g \in \mathcal{G}, \\ \mu \sum_{m=1}^{\tilde{n}_g^C} \sum_{l=2}^{\bar{l}} z_{g,m}^l \leq T_s C \leq \mu \sum_{m=1}^{\tilde{n}_g^C+1} \sum_{l=2}^{\bar{l}} z_{g,m}^l, \forall g \in \mathcal{G}, \end{cases} \quad (4.5)$$

where  $z_{g,m}^l$  is the file size of segment  $m$  of version  $l$  for sub-multicast group  $g$ , which adopts the SVC principle [106]. Here,  $\tilde{n}_g^B$  and  $\tilde{n}_g^C$  represent the segment buffering numbers by allocating reserved bandwidth resources and computing resources, respectively. Here,  $\bar{l}$

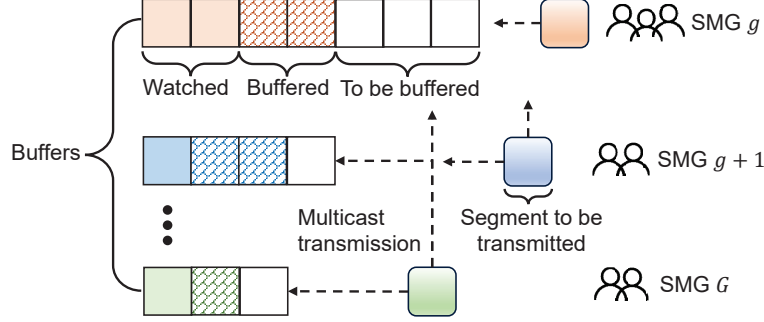


Figure 4.4: Multicast transmission mechanism among sub-multicast groups.

is the average segment version and parameter  $\mu$  is the computing density for segment transcoding. Here,  $C$  is the computing capacity of AP. Furthermore,  $r_{g,k}$  is the data rate for user  $k$  in sub-multicast group  $g$ , which is given by

$$r_{g,k} = B \log_2 \left( 1 + |h_{g,k}|^2 P_D / N_0 \right), \quad (4.6)$$

where  $h_{g,k}$  is the channel gain of user  $k$  in sub-multicast group  $g$  and  $N_0$  is the noise power. Here,  $P_D$  and  $B$  represent the downlink transmission power and the reserved bandwidth resources for the multicast group, respectively. Considering users' diversity in swipe behaviors, video preferences, and locations, the number of users in each sub-multicast group is very limited to guarantee transmission efficiency, usually 3 to 5 users with certain distances. Therefore, the signal interference between users within sub-multicast group  $g$  is relatively small, which is ignored in Eq. (4.6) for calculation convenience. To ensure that all users can receive the data, the transmission capability supported by the worst user channel condition in the sub-multicast group is selected [107, 108].

Since Eq. (4.5) consists of monotonically increasing functions, the maximum segment buffering numbers  $\tilde{n}_g^B$  and  $\tilde{n}_g^C$  can be uniquely determined by increasing their own values. To satisfy both bandwidth and computing resource requirements, we select the smaller one between  $\tilde{n}_g^B$  and  $\tilde{n}_g^C$  as the segment buffering number, i.e.,  $\tilde{n}_g = \min \{ \tilde{n}_g^B, \tilde{n}_g^C \}$ . For one multicast group, the segment buffering number in the resource requirement is the maximum value of all sub-multicast groups' segment buffering numbers, i.e.,  $n_{\text{resource}} = \max_{g \in \mathcal{G}} \tilde{n}_g$ .

According to the buffer and resource requirements, we can determine the segment buffering number, i.e.,  $n = \lfloor \max \{ n_{\text{buffer}}, n_{\text{resource}} \} \rfloor$ , where  $\lfloor \cdot \rfloor$  is the floor operation. Then, the buffered segment sequence of sub-multicast group  $g$  can be obtained by integrating segment buffering number and order, denoted by  $\Omega_g$ .

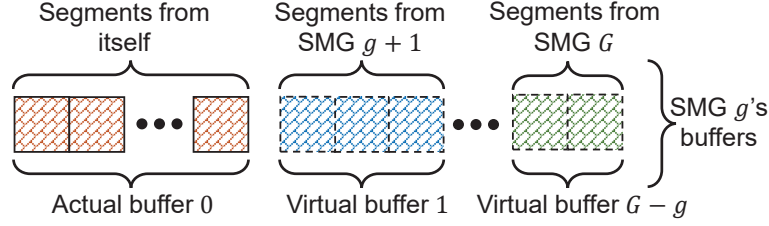


Figure 4.5: DA-assisted sub-multicast group's buffer management.

#### 4.2.4 Digital Agent-Based Buffer Update

As shown in Fig. 4.5, we utilize the virtualization technology of DA to construct multiple virtual buffers for sub-multicast group  $g$ , where each virtual buffer corresponds to the divided and subsequent sub-multicast group's buffer. The buffer index of sub-multicast group  $g$  is denoted by  $f$ , ranging from 0 to  $G - g$ . Since the original buffer update function usually characterizes the relationship between the current buffer size, downloaded video length, and scheduling slot length, without considering multicast transmission among sub-multicast groups [85, 102], we modify the buffer update function, as follows:

$$q_{g,t+1}^f = \begin{cases} \left[ q_{g,t}^f - T_s + \tau |\Omega_g| \right]^+, & f = 0, \\ q_{g,t}^f + \tau |\Omega_{g+f}|, & \forall f \in \{1, \dots, G - g\}, \end{cases} \quad (4.7)$$

where  $|\Omega_g|$  is the segment buffering number of sub-multicast group  $g$ .

When sub-multicast group  $g$  starts to watch the next video, the current buffers 0 to  $G - 1$  will be replaced by new buffers 1 to  $G$ , and a new empty virtual buffer will be added as buffer  $G$ . Assuming that within each scheduling slot, the number of swipe behaviors for each sub-multicast group is at most once. An indicator function  $\mathbb{S}_{g,t}$  is introduced, where  $\mathbb{S}_{g,t} = 1$  denotes a swipe behavior and  $\mathbb{S}_{g,t} = 0$  denotes no swipe behavior. By integrating the impact of swipe behaviors, sub-multicast group  $g$ 's buffer update can be further modified as

$$\tilde{q}_{g,t+1}^f = \begin{cases} q_{g,t+1}^f, & \mathbb{S}_{g,t} = 0, \\ q_{g,t+1}^{f+1}, & \mathbb{S}_{g,t} = 1 \& f \in [0, G - g - 1], \\ 0, & \mathbb{S}_{g,t} = 1 \& f = G - g. \end{cases} \quad (4.8)$$

### 4.2.5 Multicast QoE Model

To evaluate the system performance, we construct the multicast QoE model, consisting of rebuffering time, video quality, and quality variation, which considers the mutual influence of multicast segment buffering. The specific analysis is presented as follows.

In each time slot, since each sub-multicast group can occupy the total bandwidth and computing resources in its scheduling slot, we further divide a scheduling slot into multiple mini-slots, where each sub-multicast group can occupy the total bandwidth and computing resources in its mini-slot. We define the variable  $\beta_{g,t}$  as the division ratio for sub-multicast group  $g$  in scheduling slot  $t$ , which need to satisfy the following constraint, i.e.,

$$\sum_{g=1}^G \beta_{g,t} \leq 1, \forall t \in \mathcal{T}. \quad (4.9)$$

where  $\mathcal{T}$  is the scheduling slot set. For the simplification of expression, we omit  $t$  in the following section. Since the lagging sub-multicast group can receive segments from the leading sub-multicast group in its mini-slot, the transmission delay of leading sub-multicast group needs to consider users' channel conditions of itself and its previous sub-multicast groups. Correspondingly, the multicast transmission delay,  $D_g$ , is derived by

$$D_g = \frac{\sum_{l=1}^L \sum_{m \in \Omega_g} a_{g,m}^l \sum_{j=1}^l z_{g,m}^j}{\min_{k \in \bigcup_{d=1}^g \mathcal{K}_d} \beta_g r_{g,k}}. \quad (4.10)$$

Since each segment can have multiple layers that correspond to different versions, we define a binary version selection variable  $a_{g,m}^l$ , where  $a_{g,m}^l = 1$  indicates segment layer  $l$  of video  $m$  is selected for buffering in sub-multicast group  $g$ , otherwise,  $a_{g,m}^l = 0$ .

To avoid repeated video transmission, we assume that only one segment version can be selected for buffering in sub-multicast group  $g$  in each time slot, which can be expressed as

$$\sum_{l=1}^L a_{g,m}^l = 1, \forall m \in \Omega_g. \quad (4.11)$$

Since the transmission process and transcoding process can be conducted in parallel, the service delay of sub-multicast group  $g$  equals to the larger one between the multicast transmission delay and video transcoding delay, i.e.,

$$S_g = \max \left\{ D_g, \frac{\mu \sum_{l=1}^L \sum_{m \in \Omega_g} a_{g,m}^l \sum_{j=1}^l z_{g,m}^j}{\beta_g C} \right\}. \quad (4.12)$$

Based on sub-multicast group  $g$ 's current actual buffer 0 and service delay, we refer to [102] to derive the rebuffering time, i.e.,

$$R_g = [S_g - \tilde{q}_g^0]^+. \quad (4.13)$$

SSIM is a common metric used in video quality assessment. The relationship between video bitrate,  $b$ , and SSIM,  $Q$ , can be depicted as  $Q = 1 - \frac{1}{2b+1}$  [85], where  $Q$  ranges from 0 to 1 and a higher  $Q$  means a higher video quality. Based on this mathematical relationship, the video quality of new buffered segments in sub-multicast group  $g$  can be expressed as

$$Q_g = \sum_{m \in \Omega_g} 1 - \frac{1}{2 \sum_{l=1}^L a_{g,m}^l \sum_{j=1}^l z_{g,m}^j / \tau + 1}. \quad (4.14)$$

Based on video quality, we can analyze the video quality variation,  $V_g$ , between adjacent segments in sub-multicast group  $g$ , depicted by

$$V_g = \frac{1}{|\Omega_g|} \sum_{m=1}^{|\Omega_g|} |Q_{g,m} - Q_{g,m-1}|. \quad (4.15)$$

where  $Q_{g,0}$  is the video quality of the last segment in sub-multicast group  $g$ 's buffer before new segment buffering.

To reflect the user's satisfaction with network management, three factors including rebuffering time, video quality, and quality variation, can be integrated into the multicast QoE model, as referred in [109], which can be expressed as

$$\Upsilon_g(\mathbf{a}_g, \beta_g) = Q_g - \lambda_{g,1} R_g - \lambda_{g,2} V_g, \quad (4.16)$$

where  $\mathbf{a}_g$  is a vector whose unit element is  $a_{g,m}^l$ . Here,  $\lambda_{g,1}$  and  $\lambda_{g,2}$  represent users' sensitivity degrees of rebuffering time and quality variation in sub-multicast group  $g$ , respectively.

Considering each sub-multicast group has multiple segments to be buffered with different buffering orders, we need to incorporate the impact of buffering order into resource allocation. Therefore, we transform the buffering order into weighting factors, which can be expressed as

$$\omega_g = \frac{\sum_{m=1}^{\Omega_g} \phi_{g,m}}{\sum_{g=1}^G \sum_{m=1}^{\Omega_g} \phi_{g,m}}, \quad (4.17)$$

where  $\phi_{g,m}$  is the buffering order of segment  $m$  for sub-multicast group  $g$ , and a larger value corresponds to a higher buffering priority. Based on weighting factors, we can refine the established QoE model, i.e.,

$$\tilde{\Upsilon}_g = \omega_g \Upsilon_g. \quad (4.18)$$

Here, users' network-related and behavior-related information is utilized to update sub-multicast groups and recommended video list, as well as estimate transmission capabilities and watching probability distribution, which are important elements to constitute the multicast QoE,  $\tilde{\Upsilon}_g$ .

### 4.3 Problem Formulation

Since each sub-multicast group's bandwidth and computing resource demands are dynamic due to users' dynamic swipe behaviors and video requests, the reserved bandwidth and computing resources need to be flexibly and accurately allocated to each sub-multicast group at each scheduling slot to reduce playback lags. Furthermore, the versions of segments to be buffered need to be as high and stable as possible to ensure high video playback quality. By achieving low playback lags and high video playback quality, users can obtain better QoE. Based on the above analysis, our objective is to maximize long-term user QoE by optimizing segment version selection and slot division at each scheduling slot. The formulated problem **P1** is given by

$$\mathbf{P1} : \max_{\{\mathbf{a}_{g,t}, \beta_{g,t}\}_{t \in \mathcal{T}}} \frac{1}{T} \sum_{t=1}^T \sum_{g \in \mathcal{G}} \tilde{\Upsilon}_g(\mathbf{a}_{g,t}, \beta_{g,t}) \quad (4.19a)$$

$$\text{s.t. (4.9) and (4.11),}$$

$$a_{g,m,t}^l \in \{0, 1\}, \forall g \in \mathcal{G}, m \in \Omega_g, l \in \mathcal{L}, t \in \mathcal{T}, \quad (4.19b)$$

$$\beta_{g,t} \in [0, 1], \forall g \in \mathcal{G}, t \in \mathcal{T}. \quad (4.19c)$$

Constraint (4.9) is the resource capacity constraint, which guarantees that the total scheduled bandwidths and computing resources cannot exceed the system capacity. Constraint (4.11) is the video transmission constraint, which avoids the repeated video transmission for each sub-multicast group.

### 4.4 Proposed Solution

We omit  $t$  in this subsection for the simplification of expression. The formulated problem is a mixed-integer nonlinear programming problem with the objective of maximizing long-term user QoE. The dimension of  $\mathbf{a}$  is  $2^L \sum_{g \in \mathcal{G}} g|\Omega_g|$ , which is very huge when the numbers of sub-multicast groups, recommended segments, and segment versions are high. The variable vector  $\beta$  are continuous values ranging from 0 to 1. Since these variables are

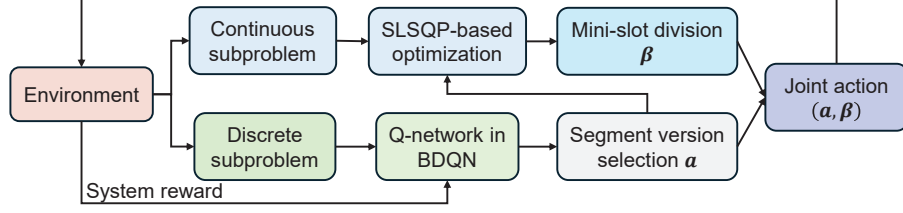


Figure 4.6: The proposed data-model-driven algorithm structure.

coupled with each other, it is hard to directly use a data-driven or model-based method to solve it. Therefore, we consider a data-model-driven algorithm to solve it, as shown in Fig. 4.6. On the one hand, convex optimization excels in handling optimization problems involving continuous variables, especially when the problem structure is provably convex. On the other hand, the non-convex nature induced by discrete variables makes traditional optimization techniques less effective. Here, DRL can play a crucial role as it learns optimization policies through interactions with the environment, adapting and learning even amidst uncertainty and complexity in problem dynamics and feedback. The state from the environment is first input to the Q-network in BDQN to obtain the segment version selection decision  $\mathbf{a}$ . Then, the Sequential Least Squares Quadratic Programming (SLSQP)-based convex optimization algorithm [98] utilizes environment information and determined decision  $\mathbf{a}$  to solve the continuous optimization subproblem for obtaining mini-slot division  $\beta$ . Next, the joint action  $(\mathbf{a}, \beta)$  is fed back to the environment to implement the network management decision. Finally, the reward from the environment is used to update the Q-network in BDQN. The detailed procedure is presented as follows.

#### 4.4.1 Deep Reinforcement Learning-Based Segment Version Selection

To solve the segment version selection optimization subproblem, we employ the Branching Dueling Q-Network (BDQN) algorithm, whose architecture is shown in Fig. 4.7.

As shown in Fig. 4.7, sub-multicast groups' state consisting of buffer lengths, channel gains, and video quality, i.e.,  $\mathbf{s} = \left\{ \left\{ \tilde{q}_g^0 \right\}_{g \in \mathcal{G}}, \left\{ h_{g,k} \right\}_{k \in \mathcal{K}_g, g \in \mathcal{G}}, \left\{ Q_{g,0} \right\}_{g \in \mathcal{G}} \right\}$ , are concatenated and input into the fully connected neural networks. The BDQN architecture is an amalgamation of Dueling Q-Networks [110] and a branching action mechanism, which comprises two key components, i.e., value function and advantage function. The value function  $V(\mathbf{s})$  estimates the value of being in a particular state, independent of any specific action. Advantage function  $A_g$  exists in each action branch  $g$ , which estimates the additional value of

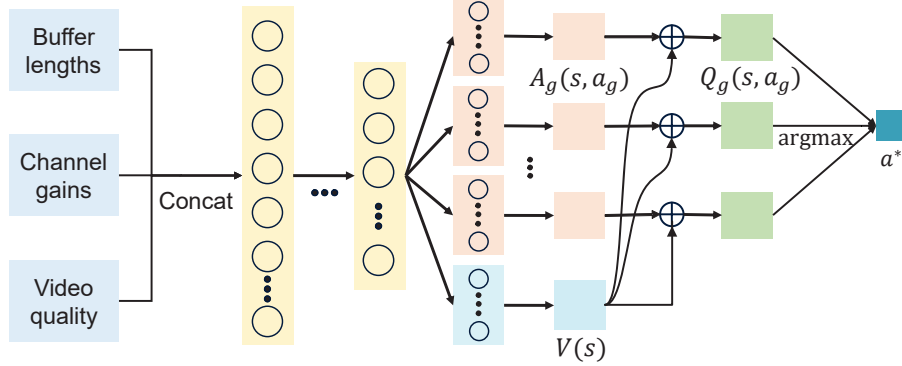


Figure 4.7: The constructed BDQN architecture.

taking a particular action in a given state, relative to other actions. This design allows the network to scale more effectively with the number of decision variables compared to traditional methods that might treat the action space as a single, flat structure. Furthermore, by decoupling the estimation of state values and the advantages of actions, BDQN can more accurately assess the relative value of each action in a given state. The segment version selection decision is split into  $G$  sub-actions, where the sub-action for sub-multicast group  $g$  is denoted by  $\mathbf{a}_g = \{a_{g,m}^l\}_{m \in \Omega_g, l \in \mathcal{L}}$ . Each action branch corresponds to a sub-multicast group. The action dimension of each branching Q-network consists of  $M$  rows and  $L$  columns, where  $M$  and  $L$  represent the maximum number of buffering segments and segment versions in one sub-multicast group, respectively. To guarantee only one segment version is selected for each segment, the position of the maximum value in each row is selected as the segment version selection value. The reward function is sub-multicast groups' QoE, i.e.,  $\sum_{g \in \mathcal{G}} \tilde{\Upsilon}_g(\mathbf{a}_g, \beta_g)$ . The advantage function of each sub-action, i.e.,  $A_g(\mathbf{s}, \mathbf{a}_g)$  is trained with the common state value  $V(\mathbf{s})$  by experience replay. The Q-value of each sub-action is updated based on the average advantage functions, which can be expressed as

$$\mathbb{Q}_g(\mathbf{s}, \mathbf{a}_g) = V(\mathbf{s}) + \left( A_g(\mathbf{s}, a_g) - \frac{1}{\rho_g} \sum_{\mathbf{a}'_g \in A_g} A_g(\mathbf{s}, \mathbf{a}'_g) \right), \quad (4.20)$$

where  $\mathbf{a}'_g$  is the next-step sub-action for sub-multicast group  $g$  and  $\rho_g$  is the sub-action dimension, i.e.,  $2 \times \|\Omega_g\| \times L$ . In each step, BDQN has a probability, i.e.,  $(1 - \varsigma)$ , to select the action that can obtain the highest Q-value, which can be expressed as

$$\mathbf{a} = \left\{ \arg \max_{\mathbf{a}'_1} \mathbb{Q}_1(\mathbf{s}, \mathbf{a}'_1; \theta_1), \dots, \arg \max_{\mathbf{a}'_G} \mathbb{Q}_G(\mathbf{s}, \mathbf{a}'_G; \theta_G) \right\}, \quad (4.21)$$

where  $\theta_g$  is the network weights of Q-network  $g$ .

To enable the agent to efficiently train, we employ the temporal-difference (TD) target after every step, which is an estimate of the expected return (future cumulative reward) for a given state-action pair, expressed by

$$y = r + \gamma \frac{1}{G} \sum_g \hat{\mathbb{Q}}_g \left( \mathbf{s}', \arg \max_{\mathbf{a}'_g} \mathbb{Q}_g (\mathbf{s}', \mathbf{a}'_g) \right). \quad (4.22)$$

To quantify the difference between the predicted and target Q-values, a loss function is essential, which can guide the optimization of neural network parameters to improve learning accuracy. The loss is the expected value of mean square error across the branches, i.e.,

$$L(\theta) = \mathbb{E}_{(\mathbf{s}, \mathbf{a}, r, \mathbf{s}') \sim \mathcal{D}} \left[ \frac{1}{G} \sum_g (y - \mathbb{Q}_g(\mathbf{s}, \mathbf{a}_g))^2 \right]. \quad (4.23)$$

Furthermore, the prioritization error is crucial for efficiently focusing the learning process on the most informative experiences, by prioritizing those with higher TD errors in the training process. Here, the prioritization error is defined by summing across a transition's absolute, i.e.,

$$\delta_{\mathcal{D}}(\mathbf{s}, \mathbf{a}, r, \mathbf{s}') = \sum_g |y_g - \mathbb{Q}_g(\mathbf{s}, \mathbf{a}_g)|. \quad (4.24)$$

Based on the above analysis, the detailed algorithm procedure to determine segment version selection variable  $\mathbf{a}$  is shown in Algorithm 2.

#### 4.4.2 Optimization-Based Slot Division

We first transform the maximization problem into the minimization problem. When  $\mathbf{a}$  is determined, denoted by  $\mathbf{a}^*$ , the original objective function in the maximization problem can be transformed to the opposite objective function in the minimization problem, i.e.,  $\sum_{g \in \mathcal{G}} \hat{\Upsilon}_g(\mathbf{a}_g^*, \beta_g) = - \sum_{g \in \mathcal{G}} \tilde{\Upsilon}_g(\mathbf{a}_g^*, \beta_g)$ .

**Theorem 1:** The transformed objective function  $\sum_{g \in \mathcal{G}} \hat{\Upsilon}_g(\mathbf{a}_g^*, \beta_g)$  is convex about  $\beta$ .

**Proof:** See Appendix A.

For the transformed convex problem, we need to find an effective algorithm to solve it. SLSQP is adept at solving the formulated linear constrained convex optimization problem

---

**Algorithm 2:** BDQN-based segment version selection

---

```

1 Input: Sub-multicast groups' buffer lengths  $\tilde{\mathbf{q}}$ , channel gains  $\mathbf{h}$ , video quality  $\mathbf{Q}$ ,
   and network update threshold  $\Gamma$ ;
2 Output: Segment version selection  $\mathbf{a}$ ;
3 Initialize: Replay memory  $\mathcal{D}$ , action-value function  $\mathbb{Q}$  with random weights  $\theta$ ,
   target action-value function  $\hat{\mathbb{Q}}$  with weights  $\theta'$ ;
4 for each episode do
5   Reset initial state  $s_1$ ;
6   for each step  $t \in \{1, \dots, T\}$  do
7     With probability  $\varsigma$  select a random action  $\mathbf{a}_t$ , otherwise, the action is
       selected based on Eq. (4.21);
8     Implement the Algorithm 3 to obtain slot division  $\beta_t$ ;
9     Execute joint action  $(\mathbf{a}_t, \beta_t)$  in the environment;
10    Observe reward  $r_t$  and new state  $s_{t+1}$ ;
11    Store transition  $(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$  in  $\mathcal{D}$ ;
12    Prioritize replay based on Eq. (4.24) to obtain a transition  $(\mathbf{s}_j, \mathbf{a}_j, r_j, \mathbf{s}_{j+1})$ 
       from  $\mathcal{D}$ ;
13    Calculate the TD target based on Eq. (4.22);
14    Perform a gradient descent step based on the loss in Eq. (4.23);
15    Every  $\Gamma$  steps, reset  $\hat{\mathbb{Q}} = \mathbb{Q}$ ;
16  end
17 end

```

---

by iteratively approximating them into quadratic subproblems [111]. The algorithm can converge to a global minimum for a convex problem due to the following reasons. First, by leveraging the gradient information and incorporating the constraints into Lagrange multipliers, SLSQP ensures that the solution not only optimizes the objective function but also strictly adheres to the problem's constraints. Second, given that our decision variables are continuous and bounded, SLSQP's design inherently aligns with the problem's structure, making it reliable to find the optimal solution. SLSQP needs the gradient information, but the optimization problem is not derivative at every point. Therefore, we employ the sub-gradient method [112] to analyze its sub-gradient.

In iteration  $i$ , the sub-gradient of  $\hat{\Upsilon}_g(\mathbf{a}_g^*, \beta_g^{(i)})$  can be determined based on the value of  $(x)^+$  function. Let denote  $\hat{\Upsilon}_g(\mathbf{a}_g^*, \beta_g^{(i)}) = \frac{1}{\beta_g^{(i)}}\varphi_{g,1} - \varphi_{g,2}$ , where functions  $\varphi_{g,1}$  and  $\varphi_{g,2}$  can

be expressed as  $\varphi_{g,1} = \omega_g \lambda_{g,1} \max \{\Xi_1(\beta_g), \Xi_2(\beta_g)\}$  and  $\varphi_{g,2} = \tilde{q}_g^0(\mathbf{a}_g^*)$ . Functions  $\varphi_{g,1}$  and  $\varphi_{g,2}$  are constants related with  $\mathbf{a}_g^*$ . In addition, functions  $\Xi_1(\beta_g)$  and  $\Xi_2(\beta_g)$  are defined in Appendix A.

Then, we can have the sub-gradient of variable for slot division variable  $\beta_g^{(i)}$ :

$$\nabla \hat{\Upsilon}(\beta_g^{(i)}) = \begin{cases} -\frac{\varphi_{g,1}}{(\beta_g^{(i)})^2}, & \beta_g^{(i)} < \frac{\varphi_{g,1}}{\varphi_{g,2}}, \\ -\sigma \frac{\varphi_{g,1}}{(\beta_g^{(i)})^2}, & \beta_g^{(i)} = \frac{\varphi_{g,1}}{\varphi_{g,2}}, \\ 0, & \text{otherwise,} \end{cases} \quad (4.25)$$

where  $\sigma$  is a positive value with the range of  $(0, 1]$ .

Based on the sub-gradient information, the Hessian matrix,  $\mathbf{H}$ , is expressed as

$$\text{diag} \left( \nabla^2 \hat{\Upsilon}(\beta_1^{(i)}), \dots, \nabla^2 \hat{\Upsilon}(\beta_G^{(i)}) \right).$$

Since the quadratic subproblem is a fundamental step in the SLSQP algorithm, we formulate the quadratic subproblem as

$$\mathbf{P2:} \min_{\beta^{(i)}} \hat{\Upsilon}(\mathbf{a}^*, \beta^{(i)}) + \nabla \hat{\Upsilon}(\mathbf{a}^*, \beta^{(i)})^T \mathbf{d} + \frac{1}{2} \mathbf{d}^T \mathbf{H} \mathbf{d} \quad (4.26a)$$

$$\text{s.t. } c(\beta^{(i)}) + \nabla c(\beta^{(i)})^T \mathbf{d} \leq 0, \quad (4.26b)$$

$$\beta^{(i)} + \mathbf{d} \in [0, 1], \quad (4.26c)$$

where the objective function is the quadratic approximation around  $(\mathbf{a}^*, \beta^{(i)})$ . Here,  $c$  and  $\mathbf{d}$  represent the function of Eq. (4.9) and the direction of change, respectively.

To solve the formulated quadratic subproblem, we can utilize the QP solver in the CVXOPT<sup>1</sup>. The specific algorithm is shown in Algorithm 3.

### 4.4.3 Computational Complexity Analysis

The proposed data-model-driven algorithm is a convex optimization embedded BDQN algorithm, as shown in Algorithm 1, where Algorithm 2 is embedded. The computing complexity is related to the algorithm structure. Specifically, the number of training episodes and steps are set to  $\mathcal{E}$  and  $T$ . In each step, there exist several important computing processes,

<sup>1</sup>CVXOPT: <https://cvxopt.org/examples/tutorial/qp.html>

---

**Algorithm 3:** SLSQP-based slot division

---

```
1 Input: Segment version selection  $\mathbf{a}$ , video bitrate sequence  $\mathbf{z}$ , bandwidth  $B$ ,  
   computing capacity  $C$ , downlink transmission power  $P^{\text{DL}}$ , noise power  $N_0$ , and  
   buffer length  $\tilde{\mathbf{q}}$ ;  
2 Output: Optimal slot division  $\beta^*$ ;  
3 Initialize:  $i = 0$ ,  $\beta^{(i)} = \beta^{(0)}$ , converge = False;  
4 while converge == False do  
5   Calculate the gradient  $\nabla \hat{\mathbf{Y}}(\beta^{(i)})$ ;  
6   Approximate the objective function and constraints at  $(\mathbf{a}^*, \beta^{(i)})$  by quadratic  
   functions;  
7   Formulate the quadratic subproblem P2;  
8   Solve the quadratic subproblem to get direction  $\mathbf{d}^{(i)}$ ;  
9   Update the solution using a line search:  $\beta^{(i+1)} = \beta^{(i)} + \alpha \cdot \mathbf{d}^{(i)}$ ;  
10  if  $\|\beta^{(i+1)} - \beta^{(i)}\| < \varepsilon$  then  
11    converge = True;  
12  end  
13  Increase the iteration number,  $i \leftarrow i + 1$ ;  
14 end  
15 Return  $\beta^* = \beta^{(i)}$ ;
```

---

i.e., action selection, SLSQP-based slot division, prioritized replay, TD target calculation, and gradient descent. The computing complexity of action selection mainly depends on the forward propagation, i.e.,  $O(\mathcal{P})$ , where  $\mathcal{P}$  is related to model parameters. Since the SLSQP-based slot division mainly includes gradient calculation, quadratic problem solving, and line search, its computing complexity can be expressed as  $O(M + n^3 + v \times M)$ , where  $v$ ,  $M$ , and  $n$  represent the evaluation number, the number of terms and variables in the objective function, respectively. The computing complexity of prioritized replay mainly depends on the replay buffer size, i.e.,  $O(\log \mathcal{D})$ . The computing complexity of TD target calculation and gradient descent both depend on the model parameters, i.e.,  $O(2\mathcal{P})$ . Therefore, the computing complexity of the designed data-model-driven algorithm is  $O(\mathcal{E} \times T \times (3\mathcal{P} + M + n^3 + v \times M + \log \mathcal{D}))$ .

Table 4.2: Simulation parameters

| Parameter | Value              | Parameter   | Value         |
|-----------|--------------------|-------------|---------------|
| $B$       | [6, 14] MHz        | $K$         | [10, 26]      |
| $C$       | [8, 12] G cycles/s | $\mu$       | 4 G cycles/Mb |
| $T_s$     | 5 sec              | $\lambda_1$ | [0.2, 0.4]    |
| $\tau$    | 2 sec              | $\lambda_2$ | [0.5, 0.7]    |
| $P_D$     | 27 dBm             | $N_0$       | -174 dBm      |

## 4.5 Simulation Results

We conduct extensive simulations on the real-world dataset to evaluate the performance of the proposed DA-based resource allocation scheme.

### 4.5.1 Simulation Setup

We adopt the short video streaming dataset<sup>2</sup> to obtain users' swipe behaviors. We sample 1000 short videos from the YouTube 8M dataset<sup>3</sup>, which includes 8 video types, i.e., Entertainment, Games, Food, Sports, Science, Dance, Travel, and News. Each video sequence is encoded into four versions, i.e.,  $L = 4$ . The time duration of each video sequence is 30 sec. We consider the scenario where two BSs are deployed at the University of Waterloo (UW) campus and sixty users' initial positions are randomly and uniformly generated around two BSs, as shown in Fig. 4.8. Each user moves along a prescribed path within the UW campus at a speed of  $2 \sim 5$  km/h, and the corresponding channel path loss is obtained by the propagationModel at Matlab. The main simulation parameters are presented in Table 4.2.

Our BDQN architecture consists of four fully connected layers, transitioning from an initial state-size input to a layer with 128 nodes. This architecture is split into two streams: a value stream with a single output and multiple advantage streams, which can produce a matrix of action advantage values that represents the combination of actions and their respective choices. To refine the learning process, we integrate a prioritized replay buffer, which emphasizes learning from experiences with higher predicted errors. The parameter setting and model structure are presented in Table 4.3.

<sup>2</sup>ACM MM Grand Challenges: <https://github.com/AItransCompetition/Short-Video-Streaming-Challenge/tree/main/data>

<sup>3</sup>YouTube 8M dataset: <https://research.google.com/youtube8m/index.html>

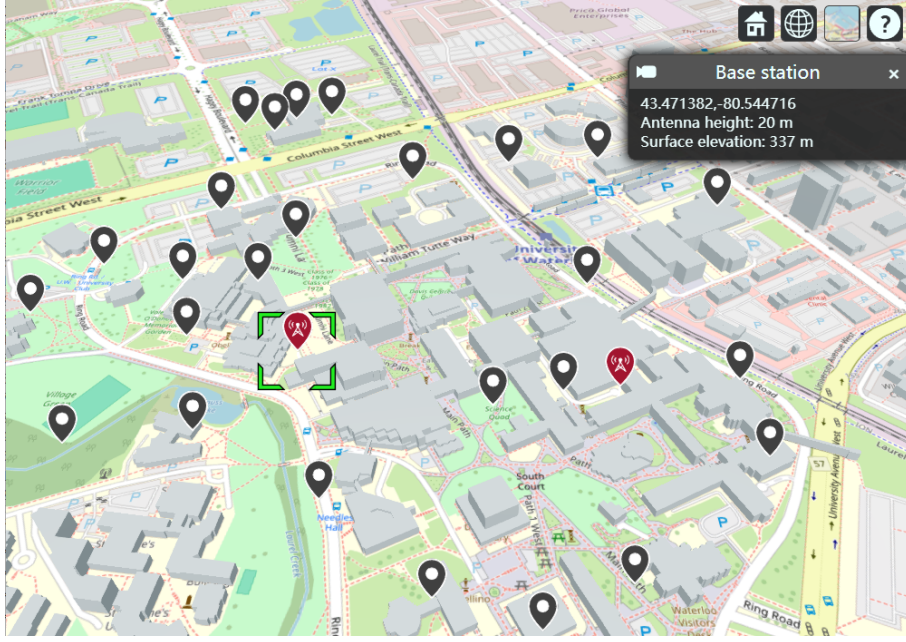


Figure 4.8: The simulation scene, where BSs and users are represented by red and gray icons, respectively.

Table 4.3: BDQN parameters

| Parameter                  | Value                                  | Parameter              | Value          |
|----------------------------|----------------------------------------|------------------------|----------------|
| Memory size                | 5000                                   | Episode length         | 75             |
| Initial epsilon            | 1                                      | Discount factor        | 0.9            |
| Epsilon decay              | 0.99                                   | Learning rate          | 0.001          |
| Final epsilon              | 0.1                                    | Batch size             | 64             |
| Number of episodes         | 500                                    | NN layer connection    | FC             |
| Hidden layer structure     | $512 \times 256 \times 256 \times 128$ | Activation function    | ReLU           |
| Advantage stream structure | $128 \times 36$                        | Value stream structure | $128 \times 1$ |

The proposed DA consists of three modules, i.e., status emulation module, data feature abstraction module, and network decision-making module, implemented in the Python 3.9

environment. In the DA status emulation module, users' trajectories are emulated with the Levy flight model that refers to a random Markovian walk. Based on users' real-time trajectories, the real-time channel conditions are emulated by analyzing the channel fading between users and base stations. Users' swipe behaviors and preferences are sampled from the real-world video swipe dataset. In the DA data feature abstraction module, an improved user clustering method proposed in Chapter 3 is used to abstract the watching probability distribution of segments. In the DA network decision-making module, the multicast segment buffering decisions are determined based on abstracted features, and the proposed data-model-driven method is used to determine the segment version selection and mini-slot division.

We compare the proposed DA-based network management scheme with the following benchmark schemes:

- **Without DA (WDA) scheme:** The segment buffering is based on the sequential principle. The rebuffering time estimation is based on the service delay and the sub-multicast groups' currently total buffered segments. The determination of segment version selection and slot division employs the same data-model-driven method proposed in the DA-based network management scheme.
- **DA-BB scheme** [101]: The segment buffering and buffer update employ the same principle proposed in the DA-based network management scheme. The scheduling slot is discretized into 10 mini-slots. Each mini-slot is first provisionally allocated to each sub-multicast group, and then the corresponding segment version selection is determined by the branch and bound algorithm. Finally, the mini-slot is ultimately allocated to the SMG that can obtain the maximum QoE.
- **WDA-DRL scheme:** The segment buffering is based on the sequential principle. The rebuffering time estimation is based on the service delay and the sub-multicast groups' currently total buffered segments. The joint optimization of segment version selection and slot division is determined by a novel hierarchical reward-based DDPG algorithm [113], where the range of segment version selection action is divided into four parts, i.e.,  $[0, 0.25)$ ,  $[0.25, 0.5)$ ,  $[0.5, 0.75)$ ,  $[0.75, 1]$ , corresponding to the segment version from low to high. Furthermore, a hierarchical reward is used to accelerate the algorithm's training speed. If the constraints can be satisfied, a reward can be given based on multicast group's QoE, otherwise, a penalty will be given based on the violation degrees of constraints.

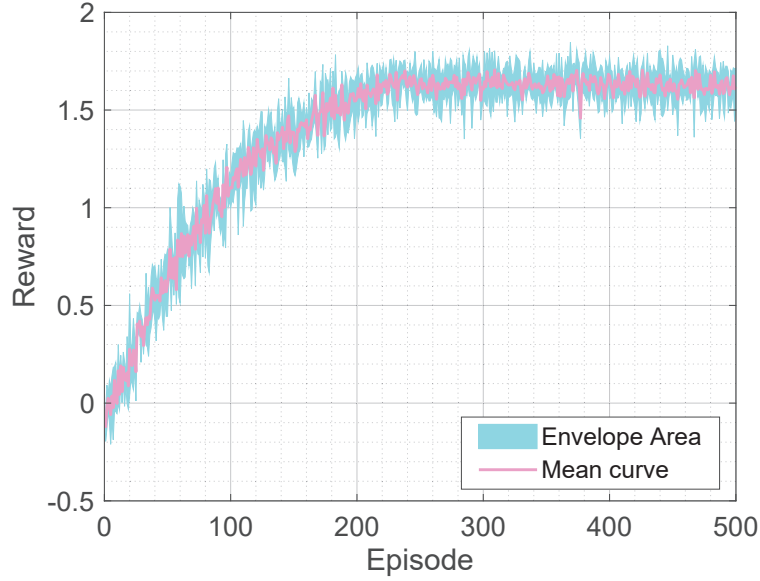


Figure 4.9: Convergence performance of proposed DA-based network management scheme.

### 4.5.2 Convergence Analysis

We analyze the convergence performance of the proposed scheme within 500 training episodes. As shown in Fig. 4.9, we present the convergence curve of the DA-based network management scheme. We conducted four training trials to draw the corresponding envelope area and mean curve, where each training trail corresponds to a unique seed for action exploration and experience replay. Each episode consists of 75 steps, and the corresponding reward is the average reward for all steps within an episode. It can be observed that as the number of episodes increases, the reward gradually grows larger. When the number of episodes approaches nearly 230, the reward converges to a stable state, indicating that the DA-based network management scheme can achieve a high and stable QoE for users.

### 4.5.3 Performance Evaluation of QoE Under Different Settings

We present QoE comparison under different users, bandwidths, and computing capacities among different schemes in Fig. 4.10. Overall, the proposed scheme can achieve superior performance under different simulation settings.

Fig. 4.10(a) describes the correlation between the number of users and QoE. Based on the observation, we find that QoE initially increases and then decreases as the number of

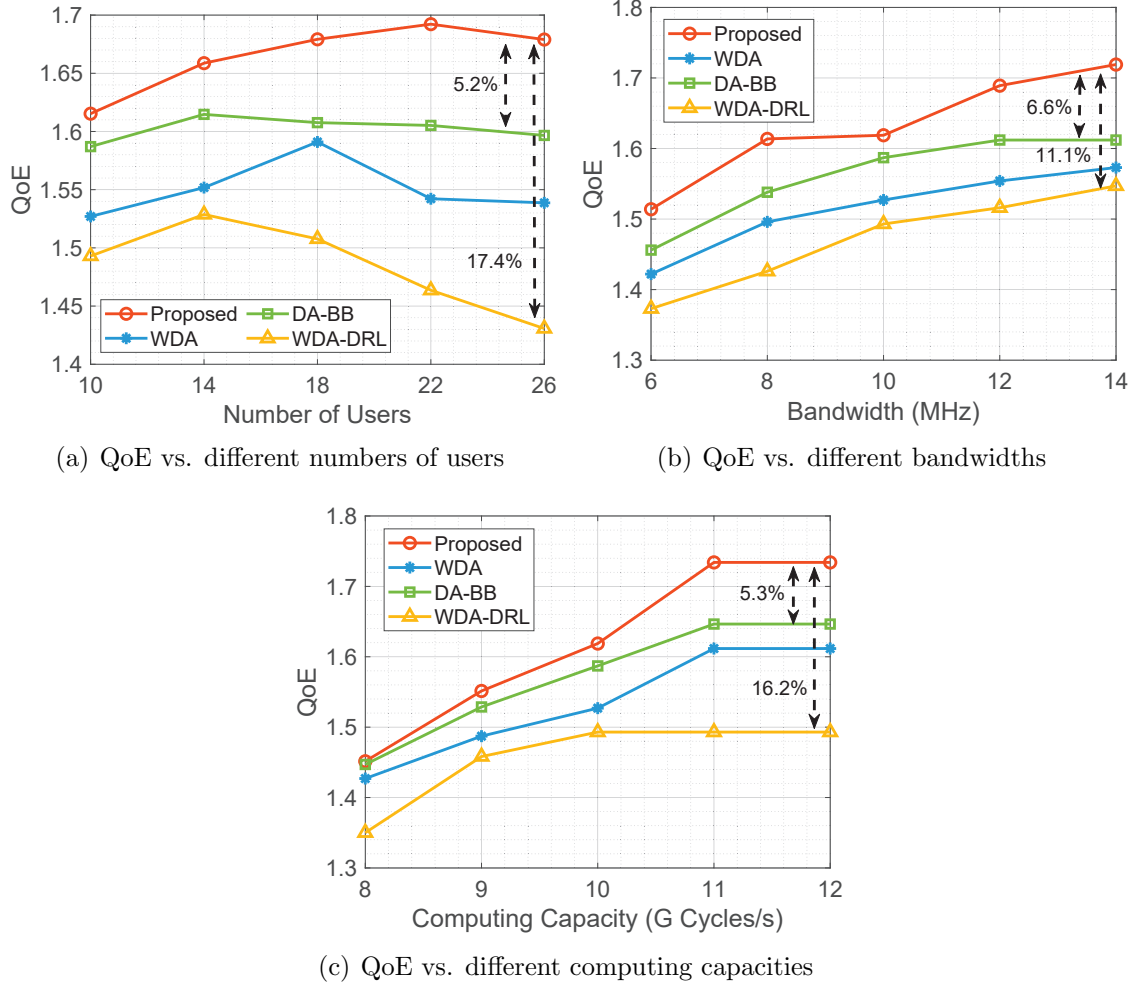


Figure 4.10: QoE comparison under different users, bandwidths, and computing capacities.

users increases due to the mutual influence of multicast segment buffering. Specifically, the increased users can lead to more users being clustered into one sub-multicast group and enrich users' diversity. Initially, the same multicast segments watched by the original users may gain a better QoE for newly clustered users who have low sensitivity degrees. However, with the increased number of users, the multicast segments cannot always satisfy users' differentiated watching requirements, which finally leads to low QoE. The proposed scheme exhibits a QoE of approximately 1.68 when catering to 26 users, a 5.2% and 17.4% increase compared to the DA-BB and WDA-DRL schemes, which demonstrates the effectiveness

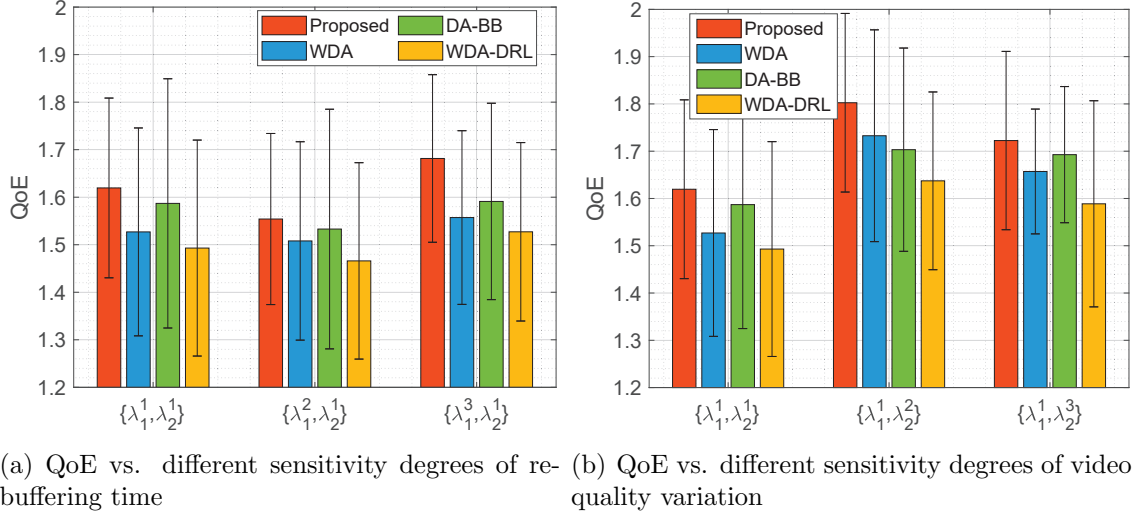


Figure 4.11: QoE comparison under different sensitivity degrees.

of the proposed scheme. The spikes and non-smoothness in Fig. 4.10(a) are most obvious compared to other subfigures mainly due to three reasons. First, with the increasing number of users, the limited bandwidth and computing resources cannot always guarantee user QoE is maintained at a high level. Second, since multicast transmission can tolerate a certain user diversity, it can temporarily increase user QoE by multicasting the same short video sequences to more users, but eventually decrease user QoE due to increasing user diversities in sub-multicast groups. Third, since the intervals of the horizontal axis are relatively small, local fluctuations in QoE may occur; however, these will be less noticeable if larger intervals are used.

Fig. 4.10(b) presents the QoE variation with the increased bandwidth. The proposed scheme's upward tendency in QoE with increased bandwidth implies that it can well adapt to varying network capacities, which is essential for ensuring service quality during peak usage times or in bandwidth-constrained environments. When the bandwidth reaches 14 MHz, the proposed scheme achieves a QoE increment of 11.1% compared to the WDA-DRL scheme, suggesting that the proposed scheme can efficiently utilize available bandwidth to enhance the user experience.

Fig. 4.10(c) correlates the computing capacity with QoE. The computing capacity directly influences the transcoding version and the speed for segments to be delivered to sub-multicast groups. The proposed scheme's pronounced improvement in QoE with the increasing computing capacity reflects its ability to leverage additional computational re-

sources effectively for video transcoding management. Here, the proposed scheme reached a QoE of around 1.75 with a computing capacity of 12 Gcycles/s, which is a significant 16.2% improvement over the WDA-DRL method. This suggests that the proposed scheme can effectively harness computing power to enhance video quality for QoE improvement.

Furthermore, we present the QoE comparison under different sensitivity degrees among different schemes in Fig. 4.11. As shown in Fig. 4.11(a), we adjust the sub-multicast group's sensitivity degrees of video rebuffering time with the fixed sensitivity degrees of video quality variation. The parameter  $\lambda_1^1$  includes three elements, i.e., (0.4, 0.3, 0.2), which represents three sub-multicast groups' sensitivity degrees of rebuffering time, respectively. The parameter  $\lambda_2^1$  also includes three elements, i.e., (0.7, 0.6, 0.5). While the parameters  $\lambda_1^2$  and  $\lambda_1^3$  are set to (0.3, 0.3, 0.3) and (0.2, 0.3, 0.4), respectively. Different settings of sensitivity degrees of rebuffering time aim at validating the effectiveness of the proposed scheme in the diversified sub-multicast groups. It can be observed that our proposed scheme can always achieve the highest QoE with a comparatively tighter range of variance under different parameters  $\lambda_1$ . As shown in Fig. 4.11(b), we change the sub-multicast groups' sensitivity degrees of video quality variation with the fixed sensitivity degrees of rebuffering time. The parameters  $\lambda_2^2$  and  $\lambda_2^3$  are set to (0.6, 0.6, 0.6) and (0.5, 0.6, 0.7), respectively. Compared with the first error bar, the last two error bars obtain higher QoE values with similar variance. This is because the last two sub-multicast groups are much more sensitive to video quality variation and they need to be allocated more bandwidth and computing resources to guarantee their smooth and high-quality video playback. Therefore, with a higher weighting factor, sub-multicast groups' QoE can be effectively enhanced.

## 4.6 Summary

In this chapter, we have proposed a novel DA-based network management scheme to enhance QoE in multicast short video streaming. Furthermore, we have established a multicast QoE model to quantify the impact of multicast segment buffering among sub-multicast groups. A convex optimization embedded DRL algorithm has been designed to determine the joint segment version selection and slot division.

## Chapter 5

# Adaptive Digital Agent Management for QoE-Driven Multicast Short Video Streaming

In this chapter, we propose an adaptive DA-assisted resource management scheme to enhance QoE in multicast short video streaming. Particularly, DAs consist of user status data and data-based models, which can update multicast groups and abstract users' swipe features. An adaptive DA management mechanism is developed to adapt to users' swipe behavior dynamics. Then, a fine-grained QoE model is established by considering the impact of resource constraints and DA model accuracy, leading to accurate buffer control. Finally, a joint optimization problem of resource management is formulated to maximize long-term QoE. To efficiently solve this problem, a diffusion-based DRL algorithm is proposed, which utilizes the denoising technique to improve the action exploration capabilities of DRL. Simulation results based on a real-world dataset demonstrate that the proposed DA-assisted resource management scheme outperforms benchmark schemes in terms of QoE, with improvements of 18.4% and 20.5% under low and high user dynamics, respectively.

### 5.1 Background and Motivations

In the digital media landscape, short video streaming has become a dominant form of content consumption, significantly altering global information sharing and consumption. Short

video platforms like TikTok, Instagram Reels, and Snapchat have revolutionized the media sector by providing platforms that are tailored for quick, engaging, and often ephemeral video content. Measuring users' satisfaction with provided short video streaming services necessitates an efficient performance metric. QoE, as an essential subjective performance metric, assesses the overall satisfaction level of a service perceived by end-users, encompassing factors such as rebuffer time, video quality, and quality variation [114]. Different users usually have different sensitive degrees of QoE factors, and these sensitivities change over time. The well-constructed QoE model can guide efficient communication, computing, and buffer control management.

Efficient resource management management is crucial for short video streaming services to enhance user QoE. The tremendous surge in video service requests (global monthly active users of platforms like TikTok and Instagram have reached 2.4 billion in 2024<sup>1</sup>) has posed significant challenges to the limited communication and computing resources in networks, leading to frequent playback interruptions and fluctuations in video quality, especially in densely populated areas [115, 116]. To cope with these challenges, multicast transmission, as a crucial technology in wireless networks, is widely used to allow a single data stream to be simultaneously distributed to a group of users. By employing multicast transmission to deliver short videos to users who share similar preferences and locations, redundancy in video processing and transmission can be effectively reduced, thereby alleviating overload on network traffic and computation [78]. However, optimizing multicast short video streaming with proper bandwidth and power allocation, along with adaptive user clustering, involves complex resource management problems.

Generative AI (GAI) can help address this by learning and making near-optimal strategies in real time. For instance, it can model network dynamics and user behaviors to assist efficient resource allocation schemes and user clustering, enhancing transmission throughput [117]. In computing resource management, GAI can intelligently distribute video processing tasks to cloud and edge servers, improving processing speed [118]. For buffer control, it can adjust pre-loaded video data based on network conditions and playback status to minimize interruptions [119]. By using GAI to solve complex resource management problems, user QoE can be effectively enhanced.

To further enhance user QoE, efficient resource management management requires the real-time perception of user dynamics and the abstraction of user behavior patterns. Therefore, DA is incorporated into mobile communication networks. DA consists of multiple user DAs and each user DA corresponds to a physical user. User DAs can record the user status through data collection and prediction, and analyze the user's statistical information, such

---

<sup>1</sup><https://www.searchlogistics.com/learn/statistics/tiktok-user-statistics/>

as preference and mobility pattern [120]. User DAs can help improve resource management management performance in MSVS from three aspects, i.e., personalized QoE model construction, users' future state prediction, and users' behavior pattern abstraction [104]. First, personalized QoE models can reflect users' sensitive degrees on different QoE factors that can guide tailored resource management management [90]. Second, users' future state prediction of playback status and swipe behaviors can help proactive video segment buffering, thus mitigating potential playback lags [121]. Third, abstracting users' behavior patterns can help update recommended video lists and reserve accurate network resources, thus alleviating network resource wastage [122].

Designing an efficient DA-assisted resource management management scheme for multicast short video streaming will face the following challenges.

- Firstly, due to the stochastic nature and spatiotemporal differences in users' swipe behaviors, designing an efficient DA management mechanism to characterize and process the highly dynamic and multi-dimensional user behavior data is challenging.
- Secondly, using a general QoE model that lacks accurate buffer control and rebuffer time estimation can cause frequent playback lags. It is challenging to construct a fine-grained QoE model by considering the effect of resource constraints on buffer update and DA model accuracy on multicast transmission delay estimation.
- Thirdly, since the resource management management problem is usually complex mixed-integer nonlinear programming, traditional DRL algorithms usually struggle with highly dynamic environments due to spatiotemporal variation in user behaviors, leading to suboptimal performance [123, 124]. Therefore, an improved DRL algorithm should be developed to solve the complex resource management management problem.

In this paper, we propose a novel adaptive DA-assisted resource management management scheme for MSVS, which can effectively enhance QoE. The main contributions are summarized as follows:

- Firstly, we propose an adaptive DA management mechanism to adapt to user behavior dynamics. Specifically, a precise measurement model of DA Data Processing (DAP) is constructed to characterize the relationship among DA model size, user dynamics, and user clustering accuracy. The proposed DA management mechanism can adaptively trigger based on the error and entropy of predicted user status and user clustering result, respectively. An improved incremental learning method is further

proposed to improve DA Model Update (DAU) efficiency by neural network growth and pruning.

- Secondly, we construct a fine-grained QoE model to incorporate the impact of resource constraints and DA model accuracy. Specifically, a two-layer segment buffering principle based on resource constraints is designed to realize accurate buffer control. The multicast rebuffer time estimation incorporates the impact of DA model accuracy on multicast transmission. Multicast rebuffer time, video quality, and video quality variation are determined by resource management management, together constituting the QoE model.
- Thirdly, we propose a diffusion-based DRL algorithm to solve the joint resource management optimization problem of bandwidth and computing resource allocation as well as video segment version selection. Diffusion model adopts the denoising technique to improve the action exploration capabilities of DRL algorithm. The extensive simulation results on real-world short video streaming datasets show that the proposed diffusion-based DRL algorithm can effectively improve QoE compared with benchmark schemes.

The remainder of this chapter is organized as follows. The system model comprising adaptive DA management model and fine-grained QoE model are presented in Section 5.2. Then, the problem formulation and proposed diffusion-based DRL algorithm are shown in Sections 5.3 and 5.4, followed by simulation results in Section 5.5. Finally, Section 5.6 concludes this chapter.

## 5.2 System Model

In this section, we first propose a DA-based resource management framework. Then, we present how to construct DAs and adaptively manage DAs. Finally, we propose a multicast buffer control method and construct a fine-grained QoE model.

### 5.2.1 Adaptive Digital Agent-Based Resource Management Framework

As shown in Fig. 5.1, we consider an adaptive DA-assisted multicast short video streaming scenario, which mainly consists of multiple BSs, users with multicast groups, one edge

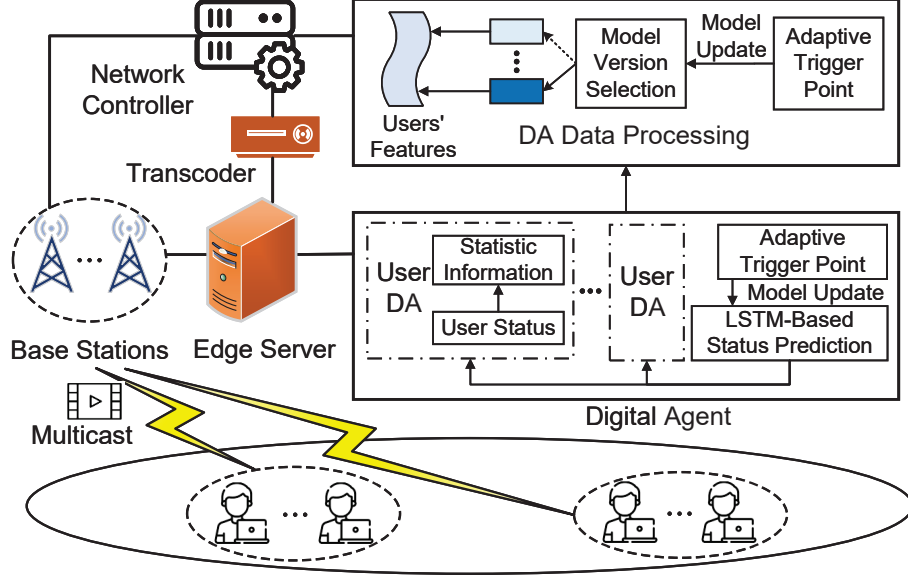


Figure 5.1: Adaptive DA-based resource management framework.

server equipped with a transcoder for video transcoding, multiple user DAs, and one network controller. The network operates in a time-slotted manner. In each real-time scheduling slot, the network controller needs to determine resource management. Communication and computing resources are periodically reserved for multicast groups, with the interval between two resource reservation instances known as a resource reservation window. This window encompasses several scheduling slots. The indexes of scheduling slots and resource reservation windows are denoted by  $t \in \{1, \dots, T\}$  and  $N \in \{1, \dots, N\}$ , respectively.

- **Multicast group:** Users within one BS are clustered into multiple multicast groups based on the similarity of user status, including locations, channel conditions, swipe timestamps, and video preferences. The users within one multicast group can receive the same video data through multicast transmission. Video transmission to one multicast group is on the same frequency band. The user set of multicast group  $g$  is denoted by  $\mathcal{U}_g$ , and the number of multicast groups is  $G$ . In each resource reservation window, multicast group  $g$  will be recommended a video list  $\Delta_g$ .
- **Base station:** BSs are responsible for transmitting short video sequences to users through multicast transmission, and collecting user status to update user DA data.

- Edge server: The edge server owns limited caching capability, which usually stores popular short video sequences in fast response to users' requests. The edge server is equipped with a transcoder, which connects to multiple BSs to transcode short video sequences into different versions, i.e., in different bitrates.
- Digital agent: A DA consists of multiple user DAs, deployed at the edge server, which stores users' status data. The data origins from two aspects, i.e., collected from physical networks and predicted through a LSTM model. Based on the stored data, a DAP module in the network controller adopts a user clustering model to analyze user similarity for accurate multicast group update. The user clustering model is an autoencoder-based DDQN model proposed in Chapter 1. Running such a user clustering model may consume plenty of computing resources even if the user status is relatively static. Therefore, we deploy multiple user clustering models of various sizes and adaptively select an appropriate one based on user dynamics<sup>2</sup>. Furthermore, since the accuracy of LSTM model and user clustering model can gradually decrease with the evolving user dynamics, we utilize an improved incremental learning method to assist model update.
- Video transcoding and delivery: Video transcoding uses the SVC technology to transcode video sequences into multiple versions to adapt to users' buffer dynamics. Then, transcoded video sequences are delivered to each multicast group through multicast transmission.
- Network controller: It is deployed at the edge server. On one hand, it decides the update trigger point and training batches for DA model<sup>3</sup> update. On the other hand, it is responsible for real-time resource management for multicast short video streaming.

The adaptive DA-assisted multicast short video streaming operates as follows. In the network operation stage, an appropriate user clustering model is first activated to implement multicast group update. Then, video transcoding and multicast transmission are started based on real-time resource management from the network controller. A cumulative error detector is used to monitor the accuracy of predicted user status in user DAs and clustering result in the DAP module. If the cumulative error exceeds a prescribed value, the network enters the DAU stage. In this stage, DAU is implemented in parallel

---

<sup>2</sup>To reduce the computational load of DAs, AI-based modules can be simplified and replaced by model-based modules.

<sup>3</sup>We consider three DA models, i.e., LSTM model, autoencoder model, and DDQN model, in this work.

Table 5.1: Notations and definitions

| Notation                                                                    | Definition                                                                         |
|-----------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| $\psi_N$                                                                    | The user dynamics in resource reservation window $N$                               |
| $m_i$                                                                       | The model size of DA model version $i$                                             |
| $f(m_i, \psi_N)$                                                            | The clustering accuracy with model size $m_i$ and user dynamics $\psi_N$           |
| $\Psi_t^k$                                                                  | The accuracy of DA model $k$ at time $t$                                           |
| $\Theta_t^k$                                                                | The average error of DA model $k$ at time $t$                                      |
| $\tau_k$                                                                    | The per batch training time for DA model $k$ update                                |
| $q_k$                                                                       | The gap between the current accuracy of model $k$ and specified accuracy           |
| $\varphi$                                                                   | The time duration of overall DAU                                                   |
| $Q_{t,g}$                                                                   | Users' average buffer length in multicast group $g$ at time $t$                    |
| $\mathcal{C}_t^{\text{DTP}}, \mathcal{C}_t^{\text{DTU}}, \mathcal{C}_{t,g}$ | The allocated computing resources for DAP, DAU and multicast group $g$ at time $t$ |
| $B_{t,g}$                                                                   | The allocated bandwidths for MG $g$ at time $t$                                    |
| $e_i$                                                                       | The selected DA model version $i$                                                  |
| $v_{t,g}^l$                                                                 | The selected segment version $l$ for multicast group $g$ at time $t$               |

with video transcoding and multicast transmission. The network controller determines real-time resource management for video transcoding, multicast transmission and DAU. When the DAU is completed, the network re-enters the network operation stage.

The main notations and their definitions are listed in Table 5.1.

### 5.2.2 Digital Agent Construction

The proposed DA is a virtual representation of physical users, containing multiple user DAs. There exist two kinds of DA data, i.e., networking-related data and behavior-related data. The former includes users' channel conditions, which can be utilized to estimate users' transmission capabilities. The latter includes users' locations, swipe timestamps, and preferences, which can be used to analyze users' mobility trajectories, swipe probability distributions, and video popularity. DA data originates from two aspects, i.e., realistic

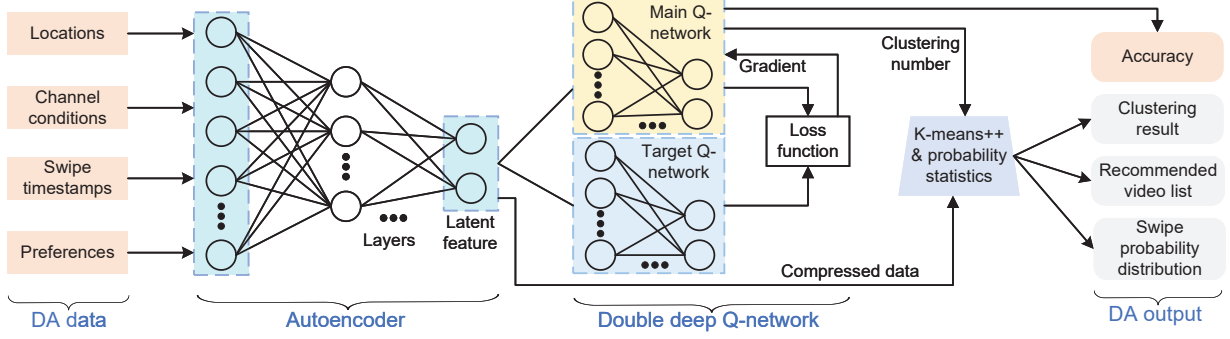


Figure 5.2: DA data processing procedure.

user-status data collected from the physical mobile communication network<sup>4</sup> and predicted user-status data from the LSTM model. DA data in resource reservation window  $N$  consists of four user-status vectors, i.e., locations,  $\mathbf{Y}_N$ , channel conditions,  $\mathbf{H}_N$ , swipe timestamps,  $\mathbf{W}_N$ , and preferences,  $\mathbf{E}_N$ . Each user-status vector is collected or predicted with different time frequencies in user DAs during each resource reservation window.

The proposed DA includes two function modules, i.e., status prediction and feature abstraction. Specifically, based on collected data from the physical mobile communication network, DA can conduct status prediction by using the LSTM model, which can effectively reduce data collection overhead. The output of status prediction module is the predicted user status for efficient network management. Additionally, DA can abstract accurate users' behavior patterns based on the data-driven methods.

### 5.2.3 Digital Agent Data Processing

As shown in Fig. 5.2, we present a detailed DAP procedure, which uses the same neural network structure in Chapter 1. The DAP module mainly consists of three parts, i.e., autoencoder, DDQN, and K-means++ with probability statistics. Specifically, the autoencoder is responsible for compressing time-series DA data into low-dimensional latent features by learning to encode significant data features. The DDQN analyzes the latent features to determine the appropriate clustering number in the user clustering model and output model accuracy by using two neural networks to minimize the overestimation. Here, the accuracy refers to the probability of the action output in DDQN. Based on the compressed DA data and clustering number, the K-means++ method can realize fast user

<sup>4</sup>To handle DA data noise and incompleteness, the Kalman Filter and linear interpolation methods can be used.

clustering while the probability statistics module can analyze the swipe probability distribution and recommended video list.

In this procedure, the autoencoder and DDQN occupy the majority of computing workload due to the hidden neural network layers. Considering the fluctuation of user dynamics, the light-weight autoencoder and DDQN can be used to process DA data when the user status is relatively static, which can effectively reduce DAP consumption.

To identify the relationship between DAP model size, user dynamics, and clustering accuracy, we conduct numerical experiments to fit the mathematical relationship. The user dynamics, denoted by  $\psi_N$ , adopts data variation to represent its value, given by:

$$\psi_N = \delta_1 \text{var}(\mathbf{H}_N) + \delta_2 \text{var}(\mathbf{Y}_N) + \delta_3 \text{var}(\mathbf{W}_N) + \delta_4 \text{var}(\mathbf{E}_N), \quad (5.1)$$

where  $\delta_1, \delta_2, \delta_3, \delta_4$  represent the weighting factors of variations in different user-status vectors for normalization.

To guarantee the flexibility of DAP model, we set three kinds of model versions by adjusting the number of hidden neural network layers<sup>5</sup>. The model selection variable is a binary variable, denoted by  $e_i$ . If model version  $i$  is activated,  $e_i = 1$ ; otherwise,  $e_i = 0$ . The model size of model version  $i$  is denoted by  $m_i$  in megabytes (MB), which is mainly determined by the weights and biases in the autoencoder and DDQN as well as the clustering number and data dimension, denoted by  $K$  and  $d$ , in the K-means++, as follows:

$$m_i = S_p \left( \sum_{j=1}^J (w_{i,j}^C + b_{i,j}^C) + \sum_{j=1}^{J'} (w_{i,j}^D + b_{i,j}^D) \right) + K \times d \times S_d, \quad (5.2)$$

where  $w_{i,j}^C$ ,  $w_{i,j}^D$ ,  $b_{i,j}^C$ , and  $b_{i,j}^D$  are the number of weights and biases in a **Conv2D** and a **Dense** layer, respectively. Parameter  $J$  and  $J'$  are the numbers of **Conv2D** and **Dense** layers, respectively. Here,  $S_p$  and  $S_d$  represent the size of each parameter and data feature, respectively.

We conduct extensive simulation measurements of user dynamics and clustering accuracy under different DA model sizes, as shown in Fig. 5.3. It can be observed that with the increasing model size and decreasing user dynamics, the clustering accuracy gradually rises and approaches 1. The quadratic polynomial in the Curve Fitting Toolbox<sup>6</sup> is utilized

---

<sup>5</sup>The number of model versions can be various for different scenarios. We set it as 3 for an example to demonstrate our idea.

<sup>6</sup><https://www.mathworks.com/help/curvefit/>

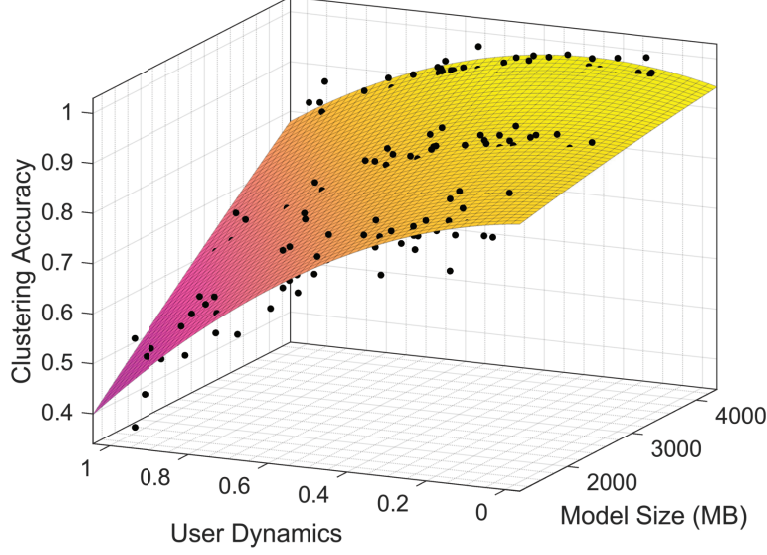


Figure 5.3: Data fitting on DA data processing accuracy.

to fit the mathematical relationship<sup>7</sup> among DA model size, user dynamics, and clustering accuracy, as follows:

$$f(m_i, \psi_N) = P_1 + P_2 m_i + P_3 \psi_N + P_4 m_i^2 + P_5 m_i \psi_N + P_6 \psi_N^2, \quad (5.3)$$

where  $f(m_i, \psi_N)$  is the clustering accuracy with model size  $m_i$  and user dynamics  $\psi_N$ .

#### 5.2.4 Digital Agent Model Update

Without adaptive parameter adjustment, the LSTM model and autoencoder-based DDQN model in DA may gradually become inaccurate due to the evolving user dynamics. Therefore, we use an improved incremental learning method to update DA models in an on-line manner. Specifically, the incremental learning for the LSTM model update adopts elastic weight consolidation on newly collected training batches to prevent knowledge forgetting [125]. In the autoencoder-based DDQN model, the autoencoder integrates new training data, which is fine-tuned at low learning rates to preserve learned features. The

<sup>7</sup>In our considered scenario,  $P_1 = 0.8246$ ,  $P_2 = 3.793 \times 10^{-5}$ ,  $P_3 = -0.2262$ ,  $P_4 = -2.044 \times 10^{-9}$ ,  $P_5 = 3.931 \times 10^{-5}$ ,  $P_6 = -0.3294$ . The coefficient of determination and Root Mean Square Error (RMSE) are 0.9334 and 0.0326, which indicate that the fitting function has high fitting accuracy.

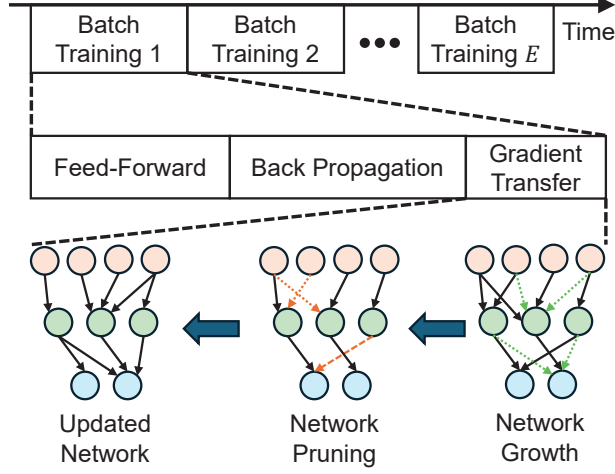


Figure 5.4: The procedure of DA model update process.

DDQN model uses experience replay to maintain a buffer containing both new and old experiences for training.

The incremental learning is utilized to update DA models by using the mini-batch stochastic gradient descent [126], as follows:

$$\theta_{t+b} \leftarrow \theta_t - \frac{\eta}{b} \nabla_{\theta_t} \sum_{z \sim b} \mathcal{L}(F(s_{t+z}; \theta_t), y_{t+z}), \quad (5.4)$$

where  $\theta_t$  and  $s_t$  are the neural network parameters and input at time  $t$ , respectively. Here, functions  $\mathcal{L}$  and  $F$  represent the loss function and strategic function, respectively. Here,  $\eta$  and  $b$  represent the learning rate and batch size, respectively. In the DAU stage, multiple batches are sampled from experience replay to fine-tune DA model. The number of batches for LSTM, autoencoder, and DDQN model update are set to  $n_1$ ,  $n_2$ , and  $n_3$ , respectively. Assuming that the relationship between model accuracy and training batches is logarithmic [127, 128], we construct a model accuracy update method for LSTM, autoencoder, and DDQN, as follows:

$$\Psi_{t+\tau_k n_k}^k = \Psi_t^k + \lambda_{k,1} \ln(b^k n_k + \lambda_{k,2}), \quad (5.5)$$

where index  $k$ , ranging from 1 to 3, represents different DA models of LSTM, autoencoder, and DDQN, respectively. Parameters  $\lambda_{k,1}$  and  $\lambda_{k,2}$  represent the logarithmic parameters for DA model  $k$ , which reflect the rate of change in model accuracy as training batches increase.

As shown in Fig. 5.4, DAU consists of multiple batch training. Each batch training consists of three parts, i.e., feed-forward, back-propagation, and gradient transfer. In the

gradient transfer, there exist three stages, i.e., network growth for activating connections with large gradients, network pruning for removing connections with small gradients, and network update for updating connections. Accordingly, the computation load can be expressed as

$$C_{\text{operation}} = \mathcal{F}_{\text{operation}} + \mathcal{B}_{\text{operation}} + \mathcal{G}_{\text{operation}}, \quad (5.6)$$

where  $\mathcal{F}_{\text{operation}}$ ,  $\mathcal{B}_{\text{operation}}$ , and  $\mathcal{G}_{\text{operation}}$  represent the computation load of feed-forward, back-propagation, and gradient transfer operations, respectively. The computation load of feed-forward operation,  $\mathcal{F}_{\text{operation}}$ , equals to that of the back-propagation operation,  $\mathcal{B}_{\text{operation}}$ , which is related to neural network structures, calculated as:

$$\begin{cases} \mathcal{F}_{\text{operation}}^{\text{LSTM}} = 4\mathbf{b}^{\text{LSTM}}_{\wp} (d \cdot H + H^2 + H), \\ \mathcal{F}_{\text{operation}}^{\text{Auto}} = \mathbf{b}^{\text{Auto}} (W_{\text{enc}} + W_{\text{dec}}), \\ \mathcal{F}_{\text{operation}}^{\text{DDQN}} = \mathbf{b}^{\text{DDQN}} \sum_i W_i, \end{cases} \quad (5.7)$$

where  $\wp$  and  $H$  represent the sequence length and the number of neurons in the hidden layers of LSTM model, respectively. Here,  $W_{\text{enc}}$ ,  $W_{\text{dec}}$ , and  $W_i$  represent the weight operations in the encoder, decoder, and DDQN layer  $i$ , respectively.

Next, we derive the computation load of gradient transfer operations in the proposed network growth and pruning mechanism. Specifically, *in the network growth stage*, the connections between neurons with large gradients are activated. For connection  $w$ , a mask value is used and set to be 1 when its gradient exceeds the  $\mathbf{a}$  percentile of the gradient of overall weight vector  $W_{\text{grad}}$ . We utilize an indicator function to define this operation on the mask value, i.e.,  $\mathbb{I}(|w_{\text{grad}}| \geq \text{Pr}_{\mathbf{a}}(W_{\text{grad}}))$ , where  $w_{\text{grad}}$  is the gradient of connection  $w$ . To guarantee that the newly activated connections can be quickly adjusted in the direction of reducing loss, the gradient needs to integrate with the current learning rate, i.e.,  $w_{\text{grad}} \leftarrow \eta \cdot w_{\text{grad}}$ . *In the network pruning stage*, connections of neurons with small gradients are removed. For connection  $w$ , a mask value is used and set to be 0 when its gradient is below the  $\mathbf{c}$  percentile of the gradients of overall weight vector, which can be expressed as

$$w_{\text{grad}} \leftarrow w_{\text{grad}} \cdot \mathbb{I}(|w_{\text{grad}}| \geq \text{Pr}_{\mathbf{c}}(W_{\text{grad}})). \quad (5.8)$$

After network growth and pruning, the network connections are updated, and a gradient transfer process is completed. During each network pruning and growth stage, assuming that each weight undergoes addition and multiplication with the same probability, the computational load, represented by  $\mathcal{G}_{\text{operation}}$ , equals twice the number of network parameters.

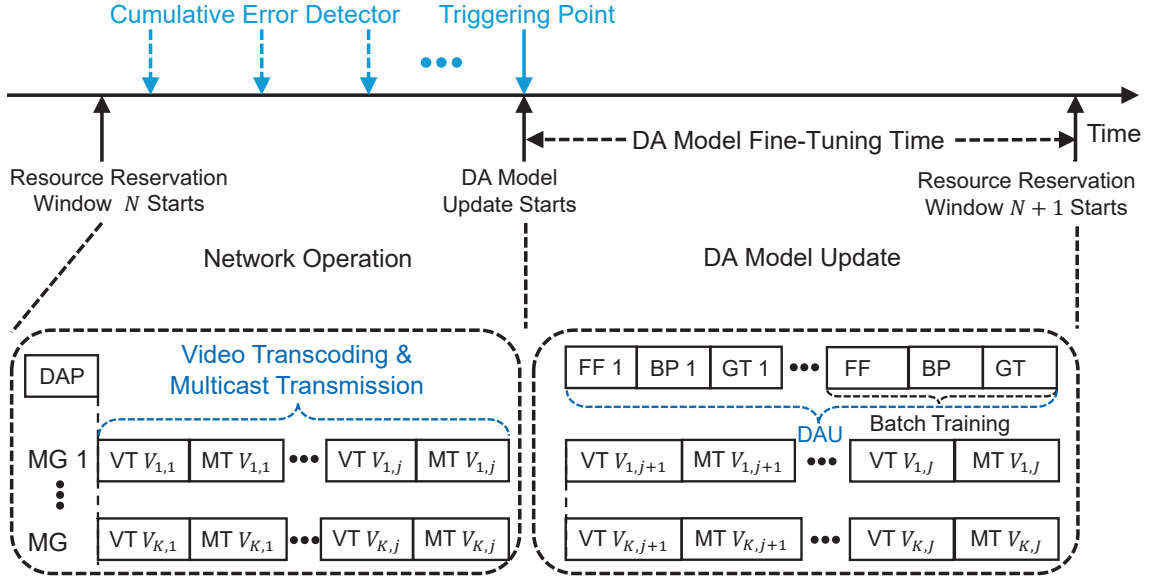


Figure 5.5: The procedure of adaptive DTP and DTU.

### 5.2.5 Adaptive Digital Agent Management

As shown in Fig. 5.5, we propose an adaptive DAU and DAP mechanism. In the network operation stage, an appropriate DAP model is first activated to implement multicast group update. Then, video transcoding (VT) and multicast transmission (MT) are started based on real-time resource management from the network controller. A cumulative error detector is used to monitor the accuracy of predicted user status and clustering result. If the cumulative error exceeds a prescribed value, the network enters the DAU stage. In this stage, DAU is implemented in parallel with video transcoding and multicast transmission. The network controller determines real-time resource management for video transcoding, multicast transmission and DAU. When the DAU is completed, a new network operation stage is started.

We first need to determine when to trigger DAU. Since the output of the LSTM model is predicted user status, we utilize the average error between predicted user status and collected one to trigger DAU. The average error is given by

$$\Theta_t^{\text{LSTM}} = \frac{1}{\|t - t_0\|} \left( \left| \mathbf{H}_{t_0 \sim t} - \tilde{\mathbf{H}}_{t_0 \sim t} \right| + \left| \mathbf{Y}_{t_0 \sim t} - \tilde{\mathbf{Y}}_{t_0 \sim t} \right| + \left| \mathbf{W}_{t_0 \sim t} - \tilde{\mathbf{W}}_{t_0 \sim t} \right| + \left| \mathbf{E}_{t_0 \sim t} - \tilde{\mathbf{E}}_{t_0 \sim t} \right| \right), \quad (5.9)$$

where  $t_0$  and  $t$  represent the beginning and current time in the current resource reservation window. Vectors  $\widetilde{\mathbf{H}}_{t_0 \sim t}$ ,  $\widetilde{\mathbf{Y}}_{t_0 \sim t}$ ,  $\widetilde{\mathbf{W}}_{t_0 \sim t}$ , and  $\widetilde{\mathbf{E}}_{t_0 \sim t}$  are collected users' channel conditions, locations, swipe timestamps, and preferences from time  $t_0$  to  $t$ , respectively. If the average error exceeds the threshold  $\Gamma_1$ , the LSTM model is updated using incremental learning.

Similarly, the autoencoder model in the DAP module uses the average error between the input user status and the reconstructed one to trigger DAU. Finding an adaptive trigger point for DDQN model update can be challenging since it is hard to obtain a realistic and optimal user clustering result for error estimation. Since the action probability distribution of DDQN can reflect the accuracy of decision-making, we utilize its entropy to trigger DAU. The entropy is given by:

$$\Theta_t^{\text{DDQN}} = -\frac{1}{\|t - t_0\|} \sum_{t=t_0}^t \sum_{j=1}^{\mathcal{D}} P(y_{t,j}) \log P(y_{t,j}), \quad (5.10)$$

where  $P(y_{t,j})$  is the output probability of action  $j$  in time  $t$ , and  $\mathcal{D}$  is the action dimension. The update thresholds of autoencoder and DDQN are denoted by  $\Gamma_2$  and  $\Gamma_3$ .

After determining the adaptive trigger point for DAU, we need to analyze how many training batches should be used for different DA models. The time duration of DAU is up to the number of training batches and allocated computing resources for batch training. The time duration of per batch training for DA model  $k$  update, denoted by  $\tau_k$ , can be expressed as

$$\tau_k = \frac{\mu_1 C_{\text{operation}}^k \mathbb{I}(\Theta_k \geq \Gamma_k)}{\mathcal{C}_k^{\text{DTU}}}, \quad (5.11)$$

where  $\Gamma_k$  and  $\mu_1$  represent the model update threshold of model  $k$  and the computing density for model update. Here,  $\mathcal{C}_k^{\text{DTU}}$  is the allocated computing resources for model  $k$  update. Based on Eq. (5.5), we can analyze how many training batches are needed for DA model  $k$  to increase accuracy  $q_k$ , i.e.,

$$q_k = \lambda_{k,1} \ln(b^k \tilde{n}_k + \lambda_{k,2}), \forall k \in \{1, 2, 3\}, \quad (5.12)$$

where  $q_k$  is the gap between the current accuracy of model  $k$  and specified accuracy of model  $k$ . Here,  $k$  ranging from 1  $\sim$  3 corresponds to the LSTM model, autoencoder model, and DDQN model, respectively, in the constructed DA. By solving this equation, the number of training batches,  $\tilde{n}_k$ , for DA model  $k$  to obtain  $q_k$  accuracy increase is

$$\tilde{n}_k = \lceil (\exp(q_k / \lambda_{k,1}) - \lambda_{k,2}) / b^k \rceil, \quad (5.13)$$

where  $\lceil \cdot \rceil$  is the ceiling function.

By combining the number of batches,  $n_k$ , for DA model  $k$  update, the time duration,  $\varphi$ , of overall DAU is given by

$$\varphi = \sum_{k=1}^3 \tau_k \tilde{n}_k. \quad (5.14)$$

To select an appropriate DA model version to process user status data, DAP activates in the initial network operation stage in Fig. 5.5. The DA model version selection  $e_i$  is based on future user dynamics. The range of user dynamics  $\psi_N$  is split into three parts, i.e.,  $[0, 0.33\kappa)$ ,  $[0.33\kappa, 0.66\kappa)$ , and  $[0.66\kappa, \kappa]$ , which corresponds to DA model version  $i$  from 1 to 3. Parameter  $\kappa$  represents the threshold of user dynamics.

### 5.2.6 Multicast Buffer Control

To guarantee smooth and high-quality video playback, accurate buffer control is necessary, which includes the number and version of segments to be buffered in scheduling slot  $t$ . The buffer control should consider two perspectives, i.e., 1) buffer perspective: buffer minimum segments to avoid buffer length empty; 2) resource perspective: buffer maximum segments based on available communication and computing resources.

From the buffer perspective, the number of buffered segments,  $\Lambda_{t,g}^1$ , should guarantee multicast group  $g$ 's buffer length is larger than scheduling slot length,  $T_s$ , to avoid the playback lags. Therefore, variable  $\Lambda_{t,g}^1$  is determined by

$$\Lambda_{t,g}^1 = [(T_s - Q_{t,g}) / \chi]^+, \quad (5.15)$$

where  $Q_{t,g}$  and  $\chi$  represent users' average buffer length in multicast group  $g$  at scheduling slot  $t$  and the time duration of one segment, respectively.

From the resource perspective, the number of buffered segments,  $\Lambda_{t,g}^2$ , should guarantee video transcoding and multicast transmission can be completed in each scheduling slot. Variable  $\Lambda_{t,g}^2$  is determined by

$$\begin{cases} \sum_{x=1}^{X_{t,g}} \sum_{l=1}^{\bar{l}} z_{t,g,x}^l \leq oT_s \min_{u \in \mathcal{U}_g} \hat{r}_{t,u} \leq \sum_{x=1}^{X_{t,g}+1} \sum_{l=1}^{\bar{l}} z_{t,g,x}^l, \forall g \in \mathcal{G}, \\ \sum_{x=1}^{\tilde{X}_{t,g}} \sum_{l=2}^{\bar{l}} z_{t,g,x}^l \leq \frac{(1-o)T_s \hat{C}_{t,g}}{\mu_3} \leq \sum_{x=1}^{\tilde{X}_{t,g}+1} \sum_{l=2}^{\bar{l}} z_{t,g,x}^l, \forall g \in \mathcal{G}, \end{cases} \quad (5.16)$$

where  $z_{t,g,x}^l$  is the file size of segment  $x$  of version  $l$  for multicast group  $g$  at scheduling slot  $t$ , which adopts the SVC principle [106]. Here,  $\hat{r}_{t,u}$  and  $\hat{\mathcal{C}}_{t,g}$  are the estimated transmission and computing capabilities of user  $u$  and multicast group  $g$  based on the recent statistical information. Parameter  $\mu_3$  is the computing density of video transcoding. Here,  $l$  and  $\tilde{l}$  represent the segment version index and average segment version. Since video transcoding is related to the enhancement layers, the segment version index ranges from 2 to  $\tilde{l}$ . Here,  $o$  is the ratio of time occupation between multicast transmission and video transcoding processes, which is set to 0.5, because these two processes usually consume almost equal time. Since Eq. (5.16) consists of monotonically increasing functions, the maximum segment buffering numbers, denoted by  $X_{t,g}$  and  $\tilde{X}_{t,g}$ , can be uniquely determined by increasing their respective values. To satisfy network resource constraints, we select the smaller one between  $X_{t,g}$  and  $\tilde{X}_{t,g}$  as the number of buffered segments, i.e.,  $\Lambda_{t,g}^2 = \min\{X_{t,g}, \tilde{X}_{t,g}\}$ .

Based on the buffer and resource perspectives, we can determine the segment buffering number, i.e.,  $\Lambda_{t,g} = \max\{\Lambda_{t,g}^1, \Lambda_{t,g}^2\}$ . The above analysis of segment buffering number depends on the average segment version. To guarantee users' high-quality video playback, it is important to determine the segment version selection in each time scheduling slot, which will be analyzed in the following subsections.

### 5.2.7 Fine-Grained QoE Model

The fine-grained QoE model consists of multicast rebuffer time, video quality, and quality variation. The multicast rebuffer time is related to the DAP latency, as well as real-time video transcoding and multicast transmission latency, and buffer length. The average multicast rebuffer time of all multicast groups is expressed in Eq. (5.17)

$$R_t(\mathcal{C}_t, \mathbf{B}_t, \mathbf{v}_t) = \frac{\mu_2 \sum_i e_i m_i}{\mathcal{C}} + \frac{1}{G} \sum_{g=1}^G \left[ \left( \frac{\mu_3 \sum_l \sum_x v_{t,g}^l z_{t,g,x}^l}{\mathcal{C}_{t,g}} + \frac{\sum_l \sum_x v_{t,g}^l z_{t,g,x}^l}{\sum_i f(e_i m_i, \psi_{t,g}) \min_{u \in \mathcal{U}_g} B_{t,g} \log \left( 1 + \frac{|h_{t,u}|^2 P_D}{N_0 + N_I} \right)} - Q_{t,g} \right) \right]^+, \quad (5.17)$$

where  $\mathcal{C}_{t,g}$  and  $B_{t,g}$  represent the allocated computing and bandwidth resources for multicast group  $g$  in scheduling slot  $t$ , respectively. To ensure that all users in multicast group  $g$  can receive segments, the transmission capability supported by the worst user channel condition in multicast group  $g$  is selected [108]. In Eq. (5.17),  $h_{t,u}$ ,  $P_D$ ,  $N_0$ , and  $N_I$  refer to user  $u$ ' channel gain, transmission power, noise power, and interference power, respectively.

Parameter  $\mu_2$  represents the computing density of DAP. Additionally,  $v_{t,g}^l$  is a binary variable. If segment version  $l$  is selected for multicast group  $g$  in scheduling slot  $t$ ,  $v_{t,g}^l = 1$ ; otherwise,  $v_{t,g}^l = 0$ . It needs to satisfy

$$\sum_l v_{t,g}^l = 1, \forall t \in [t_N, t_{N+1}], g \in \mathcal{G}, \quad (5.18)$$

which can guarantee only one segment version is selected for multicast group  $g$  to avoid repeated video transcoding and multicast transmission. Here,  $t_{N+1}$  denotes the beginning time of resource reservation window  $N + 1$ .

Furthermore,  $\mathcal{C}_t^{\text{DTP}}$ ,  $\mathcal{C}_t^{\text{DTU}}$ , and  $\mathcal{C}_{t,g}$  are the allocated computing resources for DAP, DAU and multicast group  $g$  in scheduling slot  $t$ , which have the following constraints:

$$\begin{cases} \mathcal{C}_t^{\text{DTP}} = \mathcal{C}, & \forall t \in [t_N, t_N + \varsigma_N), \\ \sum_g \mathcal{C}_{t,g} = \mathcal{C}, & \forall t \in [t_N + \varsigma_N, t_{N+1} - \varphi_N), \\ \mathcal{C}_t^{\text{DTU}} + \sum_g \mathcal{C}_{t,g} = \mathcal{C}, & \forall t \in [t_{N+1} - \varphi_N, t_{N+1}], \end{cases} \quad (5.19)$$

where  $\varsigma_N$  is the DAP time in resource reservation window  $N$ , expressed by the first term of the right side in Eq. (5.17). If there exists no DAU, the time duration of resource reservation window  $N$  is set to a fixed value,  $\mathcal{W}$ , otherwise, we can have  $t_{N+1} = t_N^S + \varphi_N$ , where  $t_N^S$  is the start time of DAU in resource reservation window  $N$ . Additionally,  $Q_{t,g}$  is the buffer length of multicast group  $g$ , which is updated based on the time length of downloaded segment, as follows:

$$Q_{t+1,g} = \left[ Q_{t,g} - T_s + \min \left\{ 1, \frac{T_s}{T_{t,g}^V + T_{t,g}^M} \right\} \Lambda_{t,g} \chi \right]^+, \quad (5.20)$$

where  $T_{t,g}^V$  and  $T_{t,g}^M$  represent the video transcoding and multicast transmission delay of multicast group  $g$  in scheduling slot  $t$ , which are expressed in the first two terms in  $[\cdot]^+$  function in Eq. (5.17), respectively. Here,  $[x]^+$  equals to  $\max\{x, 0\}$ . In addition, the minimization function is used to analyze the ratio of transmitted segments.

SSIM is a common metric to assess video quality by comparing luminance, contrast, and structural characteristics. The relationship between video bitrate,  $z$ , and SSIM,  $O$ , can be depicted as  $O = 1 - \frac{1}{2z+1}$  [85], where a higher  $O$  means a higher video quality. Based on this mathematical relationship, the average video quality of new buffered segments in all multicast groups in scheduling slot  $t$  is expressed as

$$O_t(\mathbf{v}_t) = \frac{1}{G} \sum_{g=1}^G \sum_{x=1}^{\Lambda_{t,g}} 1 - \frac{1}{2 \sum_l v_{t,g}^l z_{t,g,x}^l + 1}. \quad (5.21)$$

Based on the video quality in Eq. (5.21), we can further analyze the video quality variation between adjacent segments in all multicast groups. Since video quality decrease can have a negative effect on user QoE, we utilize the video quality decrease between adjacent segments to measure the video quality variation. Specifically, if the video quality at time  $t$  is lower than that at time  $t - 1$ , video quality variation, defined by  $J_t$ , equals to  $O_{t-1} - O_t$ ; otherwise,  $J_t = 0$ . Therefore, video quality variation can be expressed as

$$J_t(\mathbf{v}_t) = [O_{t-1}(\mathbf{v}_{t-1}) - O_t(\mathbf{v}_t)]^+. \quad (5.22)$$

The proposed fine-grained QoE model consists of three parts, i.e., multicast rebuffer time, video quality, and video quality variation. We adopt a linear weighted function, which is widely used to integrate these factors [109], to reflect users' satisfaction with network management. The constructed QoE model is expressed as

$$QoE_t(\mathcal{C}_t, \mathbf{B}_t, \mathbf{v}_t) = \varpi_1 O_t(\mathbf{v}_t) - \varpi_2 R_t(\mathcal{C}_t, \mathbf{B}_t, \mathbf{v}_t) - \varpi_3 J_t(\mathbf{v}_t), \quad (5.23)$$

where  $\varpi_1$ ,  $\varpi_2$  and  $\varpi_3$  represent multicast groups' sensitivity degrees of video quality, rebuffer time and quality variation, respectively.

### 5.3 Problem Formulation

Due to multicast groups' dynamic video requests and swipe behaviors, bandwidth and computing resources need to be flexibly and accurately allocated to DA operation as well as video transcoding and multicast transmission, to reduce playback rebuffer time. Furthermore, the versions of transmitted segments should be as high and consistent as possible to improve video quality and reduce video quality variation. Therefore, the objective is to maximize multicast groups' long-term QoE by jointly optimizing bandwidth and computing resource allocation, as well as segment version selection at each scheduling slot. The formulated problem **P1** is given by

$$\mathbf{P1}: \max_{\{\mathcal{C}_t, \mathbf{B}_t, \mathbf{v}_t\}_{t \in \mathcal{T}}} \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T QoE_t(\mathcal{C}_t, \mathbf{B}_t, \mathbf{v}_t) \quad (5.24a)$$

s.t. (5.18), (5.19),

$$\sum_{g=1}^G B_{t,g} \leq B, \forall t \in \mathcal{T}, \quad (5.24b)$$

$$\mathcal{C}_{t,g}, \mathcal{C}_t^{\text{DTU}} \in [0, \mathcal{C}], \forall t \in \mathcal{T}, g \in \mathcal{G}, \quad (5.24c)$$

$$B_{t,g} \in [0, B], \forall t \in \mathcal{T}, g \in \mathcal{G}, \quad (5.24d)$$

$$v_{t,g}^l \in \{0, 1\}, \forall t \in \mathcal{T}, g \in \mathcal{G}, l \in \mathcal{L}. \quad (5.24e)$$

Constraint (5.24b) guarantees that the allocated bandwidth resources to multicast groups do not exceed the network bandwidth limit.

## 5.4 Proposed Solution

In this section, we first transform the optimization problem into a MDP, and then develop a diffusion-based DRL algorithm to solve it.

### 5.4.1 Problem Transformation

Since the formulated problem is mixed-integer nonlinear programming with a complex objective function, it is hard to directly use a model-based method to solve it. Considering that the transition of users' playback status satisfies the Markov properties and the objective is to maximize the long-term QoE, the optimization problem **P1** can be modeled as a MDP. MDP is a discrete-time stochastic control process, including four essential elements, i.e., state, action, state transition probability, and reward. At step (scheduling slot)  $t$ , the agent, i.e., network controller, makes action  $a_t$  only based on current state  $s_t$ . Then, state  $s_t$  can evolve into  $s_{t+1}$  according to state transition probability  $\Pr(s_{t+1}|s_t, a_t)$ , which can obtain reward  $r_t$ .

The state includes six elements, i.e., user dynamics  $\psi_t$ , multicast groups' worst channel gains  $\tilde{\mathbf{h}}_t$ , DA model size  $m_t$ , DA model accuracy  $\Psi_t$ , DAU time duration  $\varphi_t$ , and multicast groups' buffer lengths  $\mathbf{Q}_t$ , which can be expressed as

$$s_t = \left\{ \psi_t, \left\{ \tilde{h}_{t,g} \right\}_{g \in \mathcal{G}}, m_t, \Psi_t, \varphi_t, \{Q_{t,g}\}_{g \in \mathcal{G}} \right\}. \quad (5.25)$$

The action consists of optimization variables, i.e., computing resource allocation variable  $\mathcal{C}_t$ , bandwidth resource allocation variable  $\mathbf{B}_t$ , and segment version selection variable  $\mathbf{v}_t$ . Since discrete variable  $\mathbf{v}_t$  hardly directly uses a gradient descend algorithm for neural network training, we relax it to a continuous vector,  $\tilde{\mathbf{v}}_t$ , ranging from 0 to 1. When any element in vector  $\tilde{\mathbf{v}}_t$  belongs to  $[0, 0.5)$ , the element will be rounded to 0; otherwise, it will be rounded to 1. Based on the above analysis, the action can be expressed as

$$a_t = \{\mathcal{C}_t, \mathbf{B}_t, \tilde{\mathbf{v}}_t\}. \quad (5.26)$$

Furthermore, to guarantee that vectors  $\mathbf{C}_t$  and  $\mathbf{B}_t$  can satisfy the constraints, the output action of each element in the vectors needs to be normalized.

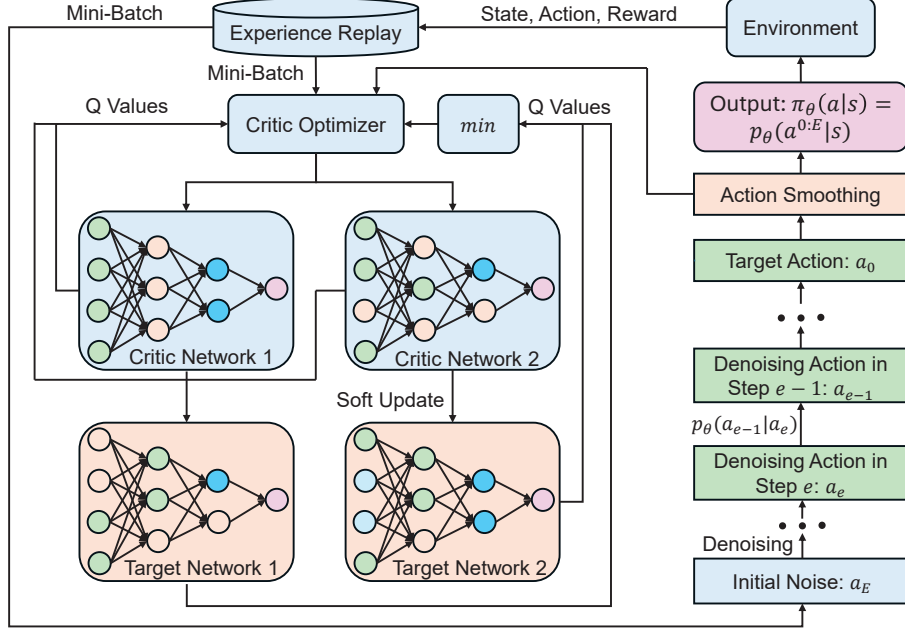


Figure 5.6: The proposed diffusion-based TD3 algorithm framework.

The state transition probability is given by

$$\begin{aligned} \Pr(s_{t+1}|s_t, a_t) &= \Pr(\psi_{t+1}|\psi_t) \prod_{g \in \mathcal{G}} \Pr(\tilde{h}_{t+1,g}|\tilde{h}_{t,g}) \Pr(m_{t+1}|m_t) \Pr(\Psi_{t+1}|\Psi_t, \mathcal{C}_t) \\ &\quad \cdot \Pr(\varphi_{t+1}|\varphi_t, \mathcal{C}_t) \prod_{g \in \mathcal{G}} \Pr(Q_{t+1,g}|Q_{t,g}, \mathcal{C}_t, \mathbf{B}_t, \mathbf{v}_t). \end{aligned} \quad (5.27)$$

The equation holds due to the independence of different state components. The first two components are evolved based on user and channel dynamics. The third component is evolved based on the adaptive DA mechanism, while the fourth and fifth components are evolved based on the computing resource allocation. The last component is the evolution of multicast groups' buffers based on the resource management decision.

The reward is designed to maximize multicast groups' QoE in Eq. (5.24a) in step  $t$ , which is defined as  $r_t = QoE_t(\mathcal{C}_t, \mathbf{B}_t, \mathbf{v}_t)$ .

#### 5.4.2 Diffusion-Based Deep Reinforcement Learning Algorithm

As shown in Fig. 5.6, we have proposed a diffusion-based Twin Delayed Deep Deterministic

istic Policy Gradient (TD3) scheduling algorithm [123] that enhances the actor-network for resource management. This approach utilizes a diffusion model, which is a method for gradually introducing variability into the decision-making process. Unlike traditional techniques that add a fixed amount of noise, the diffusion model adjusts noise levels dynamically during action generation. This unique feature enables our algorithm to explore a wider range of potential actions, potentially discovering superior strategies that standard methods might miss. Furthermore, the diffusion model’s ability to produce a variety of actions from the same starting condition enhances the robustness of our scheme.

At each step, the network controller needs to make network decisions based on the current state  $s$ , i.e.,  $\pi_\theta(a|s)$ . The reverse process of conditional diffusion models is utilized to modify the actor network. To distinguish the timestamps used in the TD3 training and the reverse process, we utilize subscripts  $e \in \{1, \dots, E\}$  to represent denoising steps<sup>8</sup>.

The reverse process is also known as the sampling process, which can infer target action  $a_0$  from a noise sample  $a_E \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ . The transition from  $a_e$  to  $a_{e-1}$  in state  $s$  with network parameter  $\theta$  is defined as  $p_\theta(a_{e-1}|a_e, s)$ , which follows a Gaussian distribution, as follows:

$$p_\theta(a_{e-1}|a_e, s) = \mathcal{N}\left(a_{e-1}; \mu_\theta(a_e, e, s), \sum_\theta(a_e, e, s)\right), \quad (5.28)$$

where the covariance matrix  $\sum_\theta(a_e, e, s)$  is determined by  $\beta_e \mathbf{I}$ . We can parameterize the noise prediction model as described in [129], where the mean value is expressed as:

$$\mu_\theta(a_e, s, e) = \frac{1}{\sqrt{\alpha_e}} \left( a_e - \frac{\beta_e}{\sqrt{1 - \bar{\alpha}_e}} \varepsilon_\theta(a_e, s, e) \right), \quad (5.29)$$

where  $\varepsilon$  is an approximator function. We have  $\alpha_e = 1 - \beta_e$  and  $\bar{\alpha}_e = \prod_{k=1}^e \alpha_k$ . The forward process variance  $\beta_e$  can be learned by re-parameterization [130] or held as a constant. To sample  $a_e$  from  $p_\theta(a_{e-1}|a_e, s)$ , we can have the following relationship:

$$a_{e-1} = \frac{1}{\sqrt{\alpha_e}} \left( a_e - \frac{\beta_e}{\sqrt{1 - \bar{\alpha}_e}} \varepsilon_\theta(a_e, s, e) \right) + \sqrt{\beta_e} \mathcal{A}, \quad (5.30)$$

where  $\mathcal{A}$  satisfies the Gaussian distribution, i.e.,  $\mathcal{A} \sim (\mathbf{0}, \mathbf{I})$ . When  $e$  equals to 1,  $\varepsilon$  is set to 0 to enhance sample quality. The simplified training loss in [129] is used to train our conditional  $\varepsilon$ -model, as follows:

$$\mathcal{L}_D(\theta) = \mathbb{E}_{e, \varepsilon, (s, a)} \left[ \|\varepsilon - \varepsilon_\theta(\sqrt{\bar{\alpha}_e} a_0 + \sqrt{1 - \bar{\alpha}_e} \varepsilon, s, e)\|^2 \right]. \quad (5.31)$$

---

<sup>8</sup>To decrease the computational load required for diffusion-based TD3 model inference, we can reduce the number of denoising steps.

The training process involves two key components: the actor and the critic networks. Two critic networks and their corresponding target networks are utilized to mitigate the overestimation bias and implement a delayed policy update mechanism. Specifically, the role of critic networks is to evaluate the policy by estimating action-value function  $\mathcal{Q}(s, a)$ . With two critic networks,  $\mathcal{Q}_1$  and  $\mathcal{Q}_2$ , we can compute the minimum value between these two critic networks to avoid overestimation. The target networks are updated at a slower rate using a soft update rule. In contrast, the critic networks are updated by minimizing the loss calculated from the Mean Square Error (MSE) between the predicted Q-values and the target Q-values. The target Q-value is computed by using the formula:

$$y_t = r_{t+1} + \gamma \min_{j=1,2} \mathcal{Q}(s_{t+1}, \tilde{a}_{t+1}; \mathbf{w}_t^j), \quad (5.32)$$

where  $\gamma$  is the discount factor. Here,  $\tilde{a}_{t+1}$  is the smooth policy that adds the clipped noise, i.e.,

$$\tilde{a}_{t+1} = \text{clip}(a_{t+1} + \tilde{\varepsilon}, [a_{\min}, a_{\max}]). \quad (5.33)$$

Parameter  $\tilde{\varepsilon}$  equals to  $\text{clip}(\mathcal{N}(0, \tilde{\sigma}^2), [-c, c])$ , which denotes  $\mathcal{N}(0, \tilde{\sigma}^2)$  is truncated between  $-c$  and  $c$ . Additionally,  $\mathbf{w}_t^j$  is the network weight of  $\mathcal{Q}$  network  $j$  at step  $t$ . The weights of two critic networks are updated based on MSE, as follows:

$$\mathbf{w}_{t+1}^j = \arg \min_{\mathbf{w}_t^j} \frac{1}{|\mathbf{B}_t|} \sum_{k=1}^{|\mathbf{B}_t|} (y_t - \mathcal{Q}(s_k, a_k; \mathbf{w}_t^j))^2, \quad (5.34)$$

where  $\mathbf{B}_t$  is the training batch at step  $t$ .

In the policy improvement stage for the diffusion-based actor  $\pi_\theta$ , the function designated for actor updates integrates two main components: the policy regularization  $\mathcal{L}_D(\theta)$  and the policy improvement  $\mathcal{L}_q(\theta)$ . The Q-value-based policy improvement  $\mathcal{L}_q(\theta)$  can guide the diffusion model to prefer the action that is expected to yield a higher Q value. Therefore, our policy learning objective for actor updates is given by

$$\pi = \arg \min_{\pi_\theta} \mathcal{L}_D(\theta) - \xi \cdot \mathbb{E}_{s \sim \mathbf{B}, a_0 \sim \pi_\theta} [\min_j \mathcal{Q}(s, a_0, \mathbf{w}^j)], \quad (5.35)$$

where  $a_0$  is re-parameterized by Eq. (5.30), which allows the gradient of the Q value with respect to the action to propagate backward through the entire diffusion sequence. Parameter  $\xi$  is the weighting factor to balance the influence between the diffusion process and the actor-critic process. The detailed algorithm procedure of proposed Diffusion-Based TD3 (DFFTD3) for resource management is presented in Algorithm 4.

---

**Algorithm 4:** Diffusion-Based TD3 (DFFTD3).

---

```
1 Input: Bandwidth and computing resource capacities  $B$  and  $\mathcal{C}$ , video bitrate  $\mathbf{z}$ ,  
   computing density vector  $\mu$ , DA model size  $\mathbf{m}$ , users' channel gains  $\mathbf{H}$ , and the  
   number of training batches for DAU  $\tilde{\mathbf{n}}$ ;  
2 Output: Network management decision  $a$ , including bandwidth and computing  
   resource allocation  $\mathbf{B}$  and  $\mathcal{C}$ , and segment version selection  $\mathbf{v}$ ;  
3 Initialize: Critic networks  $\mathcal{Q}_1, \mathcal{Q}_2$  and policy  $\pi_\theta$  with random parameters  $\mathbf{w}_1, \mathbf{w}_2$ ,  
    $\theta$ ; Target networks  $\mathbf{w}'_1 \leftarrow \mathbf{w}_1, \mathbf{w}'_2 \leftarrow \mathbf{w}_2, \theta' \leftarrow \theta$ ; Replay buffer  $\mathcal{R}$ ;  
4 for each episode do  
5   Reset initial state  $s_1$ ;  
6   for each step  $t \in \{1, \dots, T\}$  do  
7     Observe current state  $s$  and initialize a random normal distribution  
        $a_E \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ;  
8     for denoising step  $e = E : 1$  do  
9       Calculate the reverse transition distribution based on Eq. (5.28);  
10      Calculate mean value  $\mu_\theta$  based on Eq. (5.29);  
11      Calculate the action distribution based on Eq. (5.30);  
12    end  
13    Select action  $a$  based on Eq. (5.33);  
14    Execute action  $a$  and observe reward  $r$ ;  
15    Store transition tuple  $(s, a, r, s')$  in  $\mathcal{R}$ ;  
16    if  $t \bmod \mathcal{F}_1 == 0$  (time to update) then  
17      Compute target Q values based on Eq. (5.32);  
18      Update Q-functions based on Eq. (5.34);  
19    end  
20    if  $t \bmod \mathcal{F}_2 == 0$  (policy delay) then  
21      Update the policy based on Eq. (5.35);  
22      Update target network parameters:  $\mathbf{w}'_j \leftarrow \zeta \mathbf{w}_j + (1 - \zeta) \mathbf{w}'_j$ , for  $j = 1, 2$ ,  
        $\theta' \leftarrow \zeta \theta + (1 - \zeta) \theta'$ ;  
23    end  
24  end  
25 end
```

---

In Algorithm 4, critical steps include initializing the actor and critic networks at lines 3-4, followed by a loop where the current state is observed and noise is introduced for diffusion at lines 6-7. The core of the algorithm involves a diffusion process detailed from lines 8-12, where the action is refined through several denoising steps, calculating the

reverse transition and mean action values to generate a new action distribution. Actions are then selected based on the updated policy and executed to observe rewards at lines 13-14. Critically, the network undergoes periodic updates, where Q-functions are updated every  $F_1$  steps and the policy network every  $F_2$  steps as specified in lines 16-21.

### 5.4.3 Time Complexity Analysis

In the training phase, the training time complexity of the diffusion-based TD3 algorithm is mainly determined by the operations in both critic networks and the actor network with the diffusion process. Specifically, for a fully connected neural network, the time complexity of one forward pass is proportional to the number of layers  $L_c$  and the number of neurons per layer  $n_c$ . Since there are two critic networks and each network processes a batch of size  $B$ , the total complexity for updating both critic networks in one iteration is  $O(2 \cdot B \cdot L_c \cdot n_c^2)$ . The actor network involves multiple forward passes to denoise the initial action. Each forward pass through the actor network has a complexity of  $O(L_a \cdot n_a^2)$ , where  $L_a$  is the number of layers and  $n_a$  is the number of neurons in the actor network. Since the diffusion process involves  $e$  denoising steps, the total complexity for generating actions in the actor network is  $O(e \cdot L_a \cdot n_a^2)$ . Overall, the total time complexity for one training iteration is  $O(2 \cdot B \cdot L_c \cdot n_c^2) + O(e \cdot L_a \cdot n_a^2)$ .

In the inference phase, the inference time complexity is primarily driven by the actor network with the diffusion process, as the critic networks are not used during inference. For each forward pass through the actor network, the complexity is  $O(L_a \cdot n_a^2)$ . The diffusion process involves  $e$  denoising steps, meaning  $e$  forward passes are required. Therefore, the total inference complexity is  $O(e \cdot L_a \cdot n_a^2)$ .

## 5.5 Simulation Results

We conduct extensive simulations on the real-world dataset to evaluate the performance of the proposed adaptive DA-assisted resource management scheme.

### 5.5.1 Simulation Setup

We consider a scenario where two BSs are deployed at the University of Waterloo campus to provide multicast short video streaming services for 60 users. Users' initial distribution

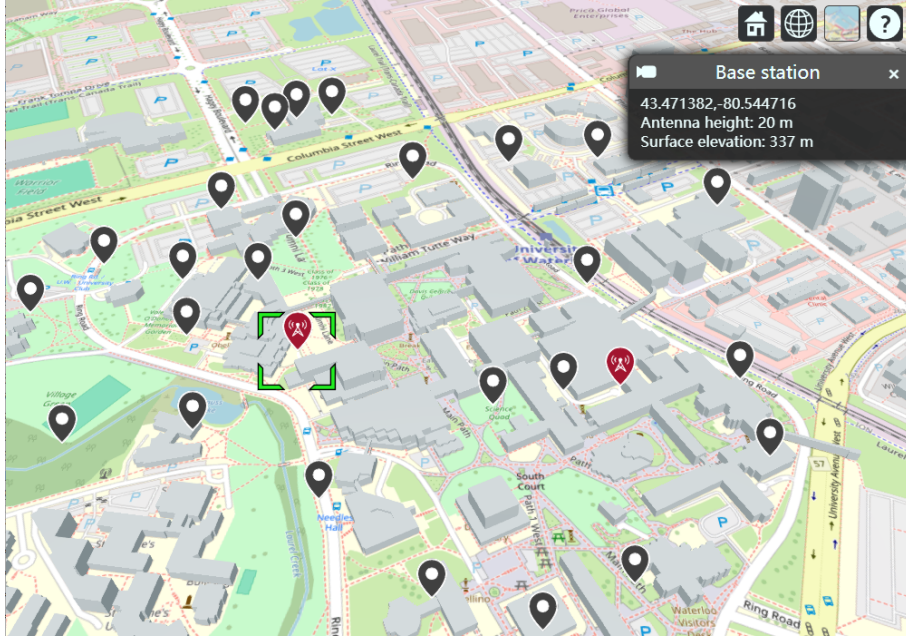


Figure 5.7: The simulation scene, where BSs and users are represented by red and gray icons, respectively.

is shown in Fig. 5.7, moving at speeds of  $2 \sim 5 \text{ km/h}$ . Their channel path losses are calculated using `propagationModel` at Matlab. Users' swipe behaviors are generated based on the short video streaming dataset<sup>9</sup>. We use the YouTube 8M dataset<sup>10</sup> to sample 1000 short videos across 8 categories, i.e., Entertainment, Games, Food, Sports, Science, Dance, Travel, and News. Each video sequence of  $30 \text{ sec}$  time duration is encoded into 15 segments. The network parameters are presented in Table 5.2.

The proposed DFFTD3 neural network integrates a diffusion-based actor-network to generate actions and uses target networks to stabilize the learning process. The critic networks consist of two separate neural networks designed to estimate the Q-values in parallel. In the action selection, the trade-off between exploration and exploitation is considered, with noise added at a given timestep to simulate the reverse diffusion process. The neural network parameter setting is shown in Table 5.3.

<sup>9</sup>ACM MM Grand Challenges: <https://github.com/AItransCompetition/Short-Video-Streaming-Challenge/tree/main/data>

<sup>10</sup>YouTube 8M dataset: <https://research.google.com/youtube8m/index.html>

Table 5.2: Communication network parameters

| Parameter    | Value                     | Parameter                      | Value           |
|--------------|---------------------------|--------------------------------|-----------------|
| $B$          | [1.2, 2.2] MHz            | $\eta$                         | 0.001           |
| $C$          | [7, 12] G cycles/s        | $\mathbf{b}$                   | 64              |
| $T_s$        | 4 sec                     | $\lambda_{1,1}, \lambda_{1,2}$ | {0.015, 100}    |
| $\delta_1$   | $3 \times 10^8$           | $\lambda_{2,1}, \lambda_{2,2}$ | {0.013, 200}    |
| $\delta_2$   | 0.4                       | $\lambda_{3,1}, \lambda_{3,2}$ | {0.011, 1200}   |
| $\delta_3$   | 0.03                      | $\varpi$                       | {1.5, 0.4, 1.4} |
| $\delta_4$   | 0.01                      | $N_I$                          | -100 dBm        |
| $\mu_1$      | $2 \times 10^7$           | $N_0$                          | -174 dBm        |
| $\mu_2$      | $1 \times 10^7$ Cycles/MB | $P_D$                          | 27 dBm          |
| $\mu_3$      | $2 \times 10^8$ Cycles/MB | $\kappa$                       | 12              |
| $\mathbf{a}$ | 30                        | $\mathbf{c}$                   | 5               |

The constructed DA comprises two modules: 1) status prediction and 2) data feature abstraction, developed in Python 3.9. The status prediction module uses the Levy flight model to predict users' trajectories and analyzes channel fading to predict real-time channel conditions. It also samples users' swipe behaviors and preferences from a real-world dataset. A two-layer LSTM with the activation functions 'tanh' for the LSTM gates and 'sigmoid' for the recurrent step is used to make data predictions. At the beginning of each resource reservation window, the LSTM returns the full sequence of DA status. The learning rate is set at 0.001, utilizing the Adam optimizer in handling sparse gradients. The loss function is based on the MSE of DA status data. To combat overfitting, a dropout rate of 20% is applied to the inputs and the connections within the LSTM layers. The data feature abstraction module uses a two-stage probability statistics method to abstract the watching probability distribution.

We compare the proposed adaptive DA-assisted resource management scheme with the following benchmark schemes:

- Fixed DA Operation-Based (FDA) scheme: The DAU and DAP use the fixed update frequency and model version. The former is updated every 5 *min*, while the latter utilizes the DA model version 2. In each scheduling slot, the resource management decision is solved by the proposed DFFTD3 algorithm.

Table 5.3: DFFTD3 algorithm parameters

| Parameter              | Value                                  | Parameter                       | Value             |
|------------------------|----------------------------------------|---------------------------------|-------------------|
| Memory size            | 5000                                   | Episode length                  | 180               |
| Initial epsilon        | 1                                      | Foward process variance         | $[10^{-3}, 0.02]$ |
| Epsilon decay          | 0.99                                   | Noise clip                      | 0.1               |
| Final epsilon          | 0.05                                   | Denoising steps                 | 10                |
| Number of episodes     | 500                                    | Neural network layer connection | Dense             |
| Hidden layer structure | $512 \times 512 \times 256 \times 256$ | Activation function             | ReLu              |
| Batch size             | 128                                    | Policy frequency                | 2                 |
| Actor learning rate    | $10^{-3}$                              | Critic learning rate            | $10^{-4}$         |

- Myopic Optimization-based (MOP) scheme [101]: The DAU and DAP use the same strategies as the FDA scheme. Bandwidth and computing resources are divided into 10 equal units, respectively. During each scheduling slot, bandwidth units are sequentially allocated to each multicast group across four segment versions, and computing units are allocated to the DAU and multicast groups. The allocation iteration yielding the highest QoE determines the combination of bandwidth and computing resources for a specific segment version.
- Model-based Branch DQN (MBDQN) [131] scheme: The DAU and DAP use the same mechanism in the proposed adaptive DA scheme. In each scheduling slot, BDQN is responsible for determining the segment version. The bandwidth and computing resources are split into 10 equal units, which are allocated to DAU and multicast groups based on the branch and bound method [132]. The joint resource management decision is fed back to the MSVS environment to update the BDQN network.
- Hierarchical Reward-based TD3 (HRTD3) scheme [133]: The DAU and DAP use the same mechanism in the proposed adaptive DA scheme. The action discretization uses the same principle in the proposed scheme. The reward function is a two-layer representation, i.e., basic reward and enhanced reward. The former depends on the violation degrees of constraints, while the latter is based on the objective function.

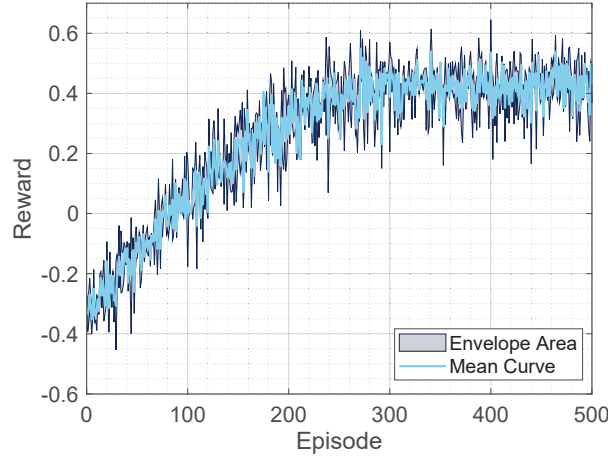


Figure 5.8: Convergence performance.

### 5.5.2 Convergence Analysis

As shown in Fig. 5.8, we present the convergence performance of proposed adaptive DA-based resource management scheme. Each episode consists of 180 steps, where the time duration of each step is 4 *sec*. Two training trials with random seeds 10 and 30 are conducted to draw the corresponding envelope area and mean curve, where the random seeds are used for action exploration and experience replay. The reward of each episode is the average QoE of all steps within its episode. It can be observed that as the number of episodes increases, the reward gradually grows. When the number of episodes approaches nearly 260, the reward converges to a stable state, indicating that the proposed adaptive DA-based resource management scheme can achieve a high QoE consistently.

### 5.5.3 The Influence of DA Data Processing and Model Update on QoE

To evaluate the influence of DAP and DAU on QoE performance, we present the real-time QoE of each step in one episode in Fig. 5.9. Initially, the QoE value remains low as the DAP conducts multicast grouping and updates the swipe probability distribution for video list recommendations, which pauses video transcoding and transmission until DAP completion. Around step 20, the QoE value decreases a lot because the current DA model cannot capture new swipe behavior patterns. Then, QoE slightly improves but is disrupted again around step 90 due to changes in swipe behaviors, necessitating another

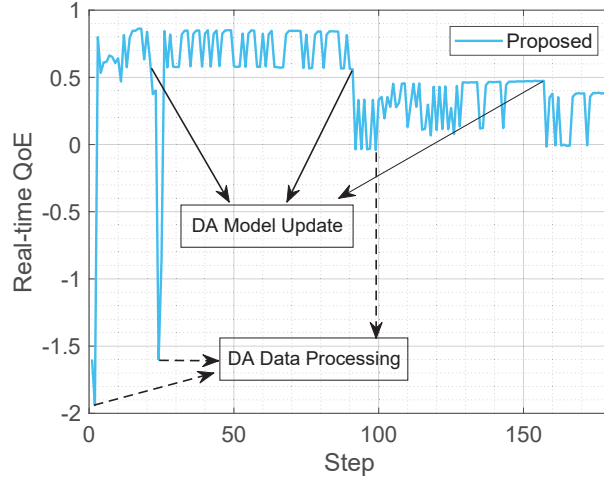


Figure 5.9: Real-time QoE in each step.

DAU for precise adjustments. However, QoE is still lower than earlier levels because increased multicast groups due to diverse user locations and behaviors strain the limited network resources, thus affecting video quality. Overall, the adaptive DA-assisted resource management approach can effectively adapt to users' swipe behaviors to maintain high QoE.

#### 5.5.4 Performance Evaluation of QoE Components

In this subsection, we evaluate the QoE performance from buffer length, video quality, and video quality variation.

The Cumulative Distribution Function (CDF) of buffer length for various schemes is shown in Fig. 5.10(a). The proposed scheme's buffer lengths mainly range from 4 to 5 *sec*, with probabilities of approximately from 10% to 90%, while other algorithms extend up to 6 *sec* but typically stay between 2 to 4 *sec*. The proposed algorithm adapts well to dynamic swipe behaviors and service delay, unlike other schemes which lead to more frequent buffering. The MOP scheme is least effective, primarily because it uses a fixed strategy for DAP and model updates that cannot match the dynamic swipe behaviors, resulting in poor user grouping and video recommendations. Moreover, MOP's focus on immediate performance gains causes the uneven segment buffering, resulting in the buffer length not concentrating at a higher stable level.

As shown in Fig. 5.10(b), we present the CDF of video quality across different schemes.

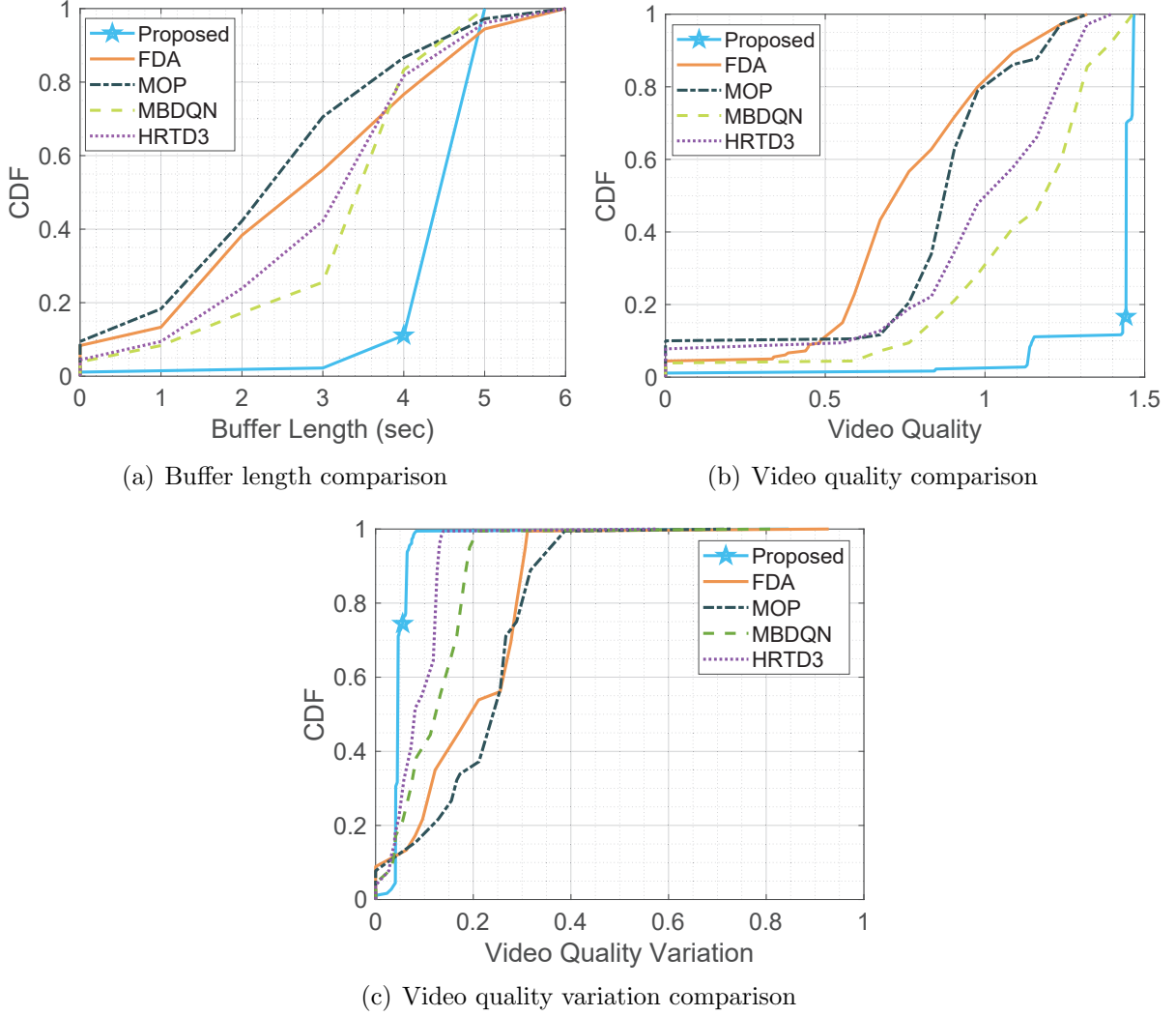


Figure 5.10: The performance comparison of different QoE components.

The proposed algorithm consistently delivers video quality between 1.1 and 1.4, surpassing other methods. The FDA algorithm underperforms, generally yielding video quality between 0.6 to 1.1. This is due to its fixed DAP and model update strategies, which do not match user preferences, causing frequent swipes and necessitating quick buffering of lower-quality videos to decrease rebuffering times. In contrast, the proposed resource management algorithm optimizes QoE holistically, ensuring longer buffer lengths and consistent video quality, thereby enhancing overall QoE.

Table 5.4: QoE performance under different user velocities

| QoE                             | Proposed | FDA    | MOP    | MBDQN  | HRTD3  |
|---------------------------------|----------|--------|--------|--------|--------|
| Low velocities<br>(2~5 Km/h)    | 0.4649   | 0.3947 | 0.3918 | 0.4382 | 0.4081 |
| High velocities<br>(20~40 Km/h) | 0.4317   | 0.3631 | 0.3583 | 0.4022 | 0.3714 |

As shown in Fig. 5.10(c), we present the CDF of video quality variation for different schemes. The proposed algorithm shows significantly lower fluctuations, mainly between 0.02 and 0.04, indicating a more consistent video quality for users. The HRTD3 and MBDQN schemes follow closely, using a hierarchical TD3 and a model-based BDQN algorithm respectively, to manage resources more effectively than the traditional MOP algorithm. The outstanding performance of our proposed scheme is due to enhanced action exploration in DRL, which adapts better to network and user dynamics.

### 5.5.5 QoE Evaluation Under Different Network Settings

As shown in Fig. 5.11, we evaluate the QoE under different numbers of users, bandwidths, and computing resources. Fig. 5.11(a) illustrates the QoE performance with the increasing number of users and fixed network bandwidth at 1.2 MHz and computing resources at 12 Gcycles/s. As the number of users increases from 40 to 60, all schemes exhibit a decreasing trend in QoE, indicating that higher user densities generally degrade service quality. However, the proposed solution shows a more gradual decline compared to the other schemes, indicating its relative robustness in handling user density increases. Fig. 5.11(b) analyzes the effect of increasing bandwidth on QoE. The proposed scheme capitalizes on the additional bandwidth more effectively than other schemes. At the maximum bandwidth of 2.2 MHz, the proposed scheme exhibits a QoE that is 12.2% higher than that the MOP scheme. This is because the MOP scheme uses a myopic strategy that only allocates as many bandwidth resources to the user group that can obtain the best QoE in the current time slot. Fig. 5.11(c) explores how varying computing resources affect QoE. At the highest computing resource allocation of 12 Gcycles/s, the proposed scheme achieves a QoE that is 44.7% superior to that of the worst scheme. This dramatic improvement highlights the proposed solution's efficiency in leveraging additional computational power to enhance data processing and ultimately improve the overall user experience.

Table 5.4 presents a comparative analysis of QoE under low and high user velocities

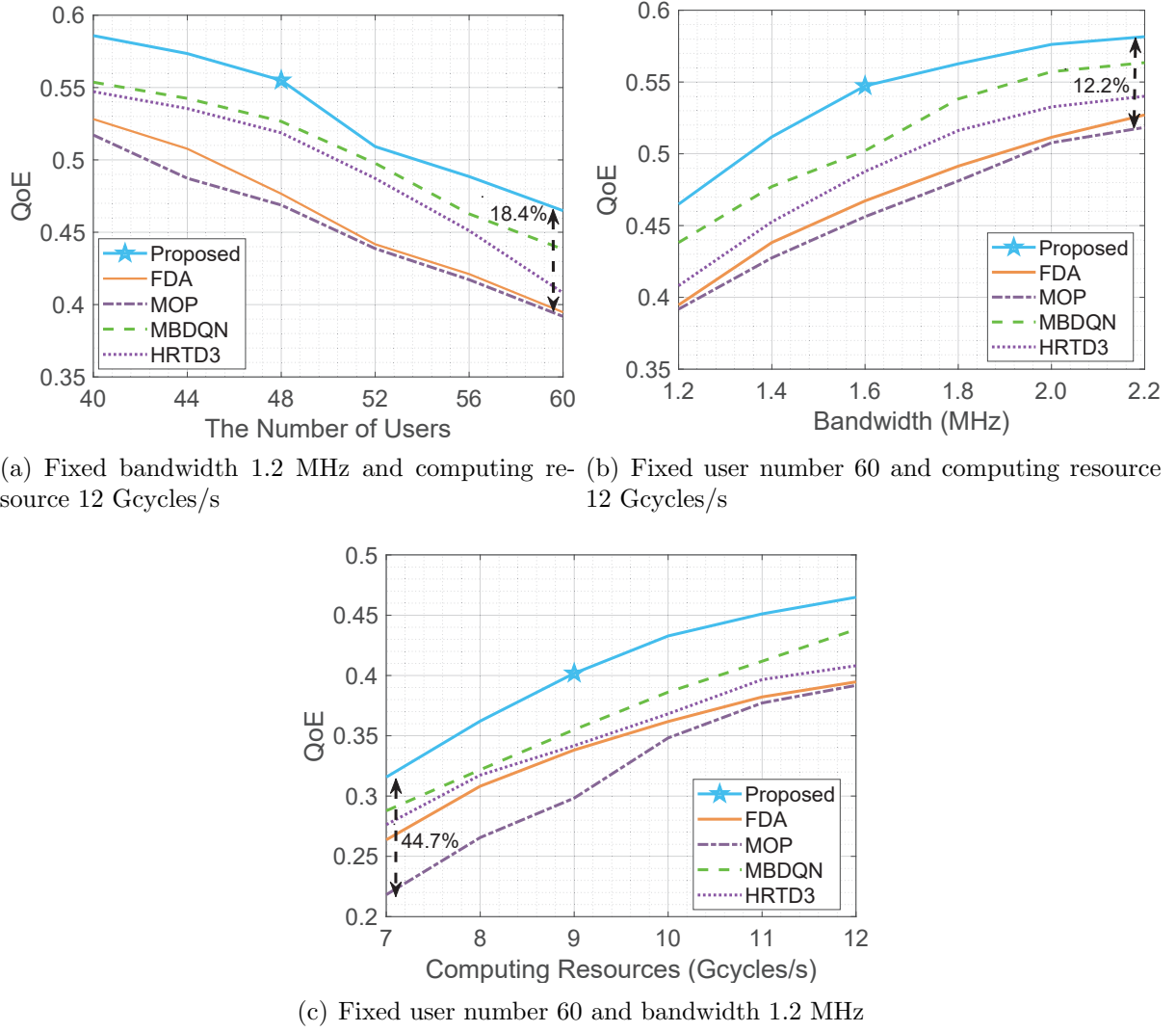


Figure 5.11: The performance of QoE under different settings.

(40 Km/h speed limit in the campus area). In the low user dynamics scenario, the proposed scheme exhibits superior performance with a QoE of 0.4649, outperforming other schemes. Notably, the MBDQN scheme is the closest competitor with a QoE of 0.4382. In high user dynamics scenarios, the proposed scheme maintains its lead with a QoE of 0.4217, indicating its robustness and effectiveness even under more dynamic conditions. From the low to high user velocities, although the proposed scheme experiences performance

degradation, its QoE performance is still higher than the baseline scheme (MOP) with 18.4% and 20.5%, because the proposed DA data processing and model update mechanism can well adapt to user dynamics.

## 5.6 Summary

In this chapter, we have proposed an adaptive DA-assisted resource management scheme to maximize long-term QoE in multicast short video streaming. Specifically, we have designed an adaptive DAP and DAU mechanism to adapt to user behavior dynamics. Furthermore, we have established a fine-grained QoE model that considers the impact of resource constraints and DA model accuracy. A diffusion-based DRL algorithm has been proposed to determine the joint resource management decision.

# Chapter 6

## Conclusions and Future Works

In this chapter, we summarize the main results and contributions of this thesis and present our future research directions.

### 6.1 Main Research Contributions

In this thesis, we have investigated DA-based resource management schemes to enhance user QoE in multicast networks for short video streaming services. First, in the large timescale, we developed a DA-based resource reservation scheme by incorporating the impact of user swipe behaviors and QoE models to make tailored resource reservations. Then, in the small timescale, we developed a DA-based resource allocation scheme that can emulate user buffer dynamics for accurate segment buffering to enhance user QoE. Last, an adaptive DA management scheme is proposed to adapt to user behavior dynamics to improve resource utilization for user QoE enhancement. The main contributions of this thesis are summarized as follows.

- A DA-based resource reservation scheme has been proposed to enhance user QoE as well as reduce resource consumption. Firstly, we design a user DA-assisted resource reservation framework. Specifically, user DAs are constructed for individual users, which store users' historical data for updating multicast groups and abstracting useful information. The swipe probability distributions and recommended video lists are abstracted from user DAs to predict bandwidth and computing resource demands. Parameterized sigmoid functions are leveraged to characterize multicast groups' user

satisfaction. Secondly, we formulate a joint non-convex bandwidth and computing resource reservation problem which is transformed into a convex piecewise problem by utilizing a tangent function to approximately substitute the concave part. A low-complexity scheduling algorithm is then developed to find the optimal resource reservation decisions. The work can be applied to analyze the user behavior pattern and integrate its impact on resource management in interactive media scenarios.

- A DA-based resource allocation scheme has been developed to enhance user QoE. Firstly, user status emulated by the DA is utilized to estimate the transmission capabilities and watching probability distributions of sub-multicast groups for adaptive segment buffering. The sub-multicast groups' buffers are aligned to the unique virtual buffers managed by the DA for a fine-grained buffer update. Then, a multicast QoE model consisting of rebuffering time, video quality, and quality variation is developed, by considering the mutual influence of segment buffering among sub-multicast groups. Finally, a joint optimization problem of segment version selection and slot division is formulated to maximize QoE. To efficiently solve the problem, a data-model-driven algorithm is proposed by integrating a convex optimization method and a deep reinforcement learning algorithm. The work can contribute to the physical entity buffer dynamics analysis on the network side for tailored content caching.
- An adaptive DA-assisted resource management scheme has been proposed to enhance user QoE. Particularly, DAs consist of user status data and data-based models, which can update multicast groups and abstract users' swipe features. An adaptive DA management mechanism is developed to adapt to user swipe behavior dynamics. Then, a fine-grained QoE model is established by considering the impact of resource constraints and DA model accuracy, leading to accurate buffer control. Finally, a joint optimization problem of resource management is formulated to maximize long-term QoE. To efficiently solve this problem, a diffusion-based DRL algorithm is proposed, which utilizes the denoising technique to improve the action exploration capabilities of DRL. The work can address the challenges of customized resource management in scenarios with high user dynamics.

In summary, this thesis has investigated resource management in multicast networks for short video streaming in three aspects: swipe behavior-aware resource reservation, buffer dynamics-based resource allocation, and behavior dynamics-driven DA management. All three research issues in this thesis have explored the nature of network dynamics, and we have proposed tailored resource management strategies to enhance user QoE. The proposed approaches and theoretical analysis can provide valuable guidelines for implementing multicast short video streaming in 6G networks.

## 6.2 Future Research Directions

Although the proposed framework can effectively facilitate experience-centric network orchestration and slicing to improve user QoE, there still exist many open research issues. For the future research, there are some interesting and promising research directions listed as follows:

- **Efficient Multi-modal DA Data Collection and Generation:** To construct and maintain level-one and level-two DAs, it is imperative to collect multi-modal data such as user behaviors, network performance metrics, and environmental contextual information. However, these data are inherently heterogeneous, encompassing various formats and update intervals, leading to high communication overhead in traditional network settings. To alleviate this issue, semantic communication can be employed to reduce the amount of data transmitted. By extracting and transmitting only the semantic essence, the communication load is significantly decreased while preserving the information necessary for DA maintenance. Furthermore, generating these multi-modal data in real-time introduces considerable computational overhead, even though DAs are deployed on edge and cloud servers. To address this challenge, lightweight data generation models such as knowledge distillation can be utilized to reduce computational costs.
- **Scalable Two-level DA Deployment and Migration:** Level-one DAs are typically deployed on edge nodes close to users to facilitate low-latency and real-time interactions, while level-two DAs are usually hosted on central nodes to leverage powerful computational resources for extensive data processing. However, due to the diversity of terminal services and varying user demands, selecting appropriate nodes for deploying these two-level DAs is crucial for enhancing processing speed and reducing latency. An effective deployment solution involves a dynamic node selection mechanism that considers factors such as service requirements, network conditions, and node capabilities. By analyzing these factors, the system can assign level-one and level-two DAs to optimal nodes within a hierarchical edge-cloud architecture, thereby optimizing resource utilization and performance.
- **Efficient Resource Management for DA Operation:** Although DAs can efficiently process networking data to improve network management performance, the maintenance of DAs is equally important, which consumes substantial communication, caching, and computing resources. This is attributed to voluminous networking and user data storage, as well as resource-intensive DA models for network status emulation, feature abstraction, and network decision-making. Since the limited network resources

need to cater to both users' service requests and DA operation, it is imperative to accurately quantify the impact of DA operation on QoS and QoE. Based on the quantified result, the network resource scheduling for both DA operation and users' service requests can be more tailored to further improve QoS and QoE. By integrating real-time user feedback loops and AI-based predictive analytics, DAs can dynamically optimize network resource management to meet diverse user requirements.

In summary, while the proposed framework provides a solid foundation for experience-centric resource management, future research should focus on enhancing data collection efficiency, optimizing DA deployment strategies, and improving resource management mechanisms. By addressing these challenges, the 6G networks can become more adaptive, scalable, and efficient, thereby further improving user QoE in increasingly complex and dynamic network environments.

# References

- [1] 3GPP. Study on management aspect of network digital twin. Technical Report TS 28.915, 3GPP, Sept. 2024. [Accessed: 15-Mar-2025].
- [2] Michael Grieves. Digital twin: manufacturing excellence through virtual factory replication. *White paper*, 1(2014):1–7, 2014.
- [3] Latif U Khan, Walid Saad, Dusit Niyato, Zhu Han, and Choong Seon Hong. Digital-twin-enabled 6G: Vision, architectural trends, and future directions. *IEEE Commun. Mag.*, 60(1):74–80, 2022.
- [4] Xuemin Shen, Xinyu Huang, Jianzhe Xue, Conghao Zhou, Xiufang Shi, and Weihua Zhuang. Revolutionizing QoE-driven network management with digital agents in 6G. *IEEE Commun. Mag.*, pages 1–1, 2025.
- [5] Davide Villa, Miead Tehrani-Moayyed, Clifton Paul Robinson, Leonardo Bonati, Pedram Johari, Michele Polese, and Tommaso Melodia. Colosseum as a digital twin: Bridging real-world experimentation and wireless network emulation. *IEEE Trans. Mobile Comput.*, 23(10):9150–9166, 2024.
- [6] Lumin Xing, Wenjian Liu, Xiaoliang Liu, and Xin Li. An enhanced vision transformer model in digital twins powered internet of medical things for pneumonia diagnosis. *IEEE J. Sel. Areas Commun.*, 41(11):3677–3689, 2023.
- [7] Eduardo Garro, Manuel Fuentes, Jose Luis Carcel, Hongzhi Chen, De Mi, Fasil Tesema, Jordi Joan Gimenez, and David Gomez-Barquero. 5G mixed mode: NR multicast-broadcast services. *IEEE Trans. Broadcast.*, 66(2):390–403, 2020.
- [8] Guanghui Zhang, Jie Zhang, Ke Liu, Jing Guo, Jack YB Lee, Haibo Hu, and Vaneet Aggarwal. Duasvs: A mobile data saving strategy in short-form video streaming. *IEEE Trans. Serv. Comput.*, 16(2):1066–1078, 2022.

- [9] Jian Yang, Enzhong Yang, Yongyi Ran, Yifeng Bi, and Jun Wang. Controllable multicast for adaptive scalable video streaming in software-defined networks. *IEEE Trans. Multimedia*, 20(5):1260–1274, 2017.
- [10] Mingfu Li and Yung-Hsiang Wu. Performance analysis of adaptive multicast streaming services in wireless cellular networks. *IEEE Trans. Mobile Comput.*, 18(11):2616–2630, 2018.
- [11] Nadezhda Chukhno, Olga Chukhno, Sara Pizzi, Antonella Molinaro, Antonio Iera, and Giuseppe Araniti. Efficient management of multicast traffic in directional mmwave networks. *IEEE Trans. Broadcast.*, 67(3):593–605, 2021.
- [12] Xuying Zhou, Wei Wang, Chau Yuen, and Dusit Niyato. Bandwidth-cache pricing-based network slicing for partially cached video streaming delivery. *IEEE Trans. Commun.*, 72(1):334–347, 2023.
- [13] Yuming Zhang, Yan Liu, Lingfeng Guo, and Jack YB Lee. Measurement of a large-scale short-video service over mobile and wireless networks. *IEEE Trans. Mobile Comput.*, 22(6):3472–3488, 2022.
- [14] Jean-Marie Vella and Saviour Zammit. A survey of multicasting over wireless access networks. *IEEE Commun. Surv. Tut.*, 15(2):718–753, 2013.
- [15] Wentao Huang, Tracey Ho, Michael Langberg, and Joerg Kliewer. Single-unicast secure network coding and network error correction are as hard as multiple-unicast network coding. *IEEE Trans. Inf. Theory*, 64(6):4496–4512, 2018.
- [16] Shuangwu Chen, Bowen Yang, Jian Yang, and Lajos Hanzo. Dynamic resource allocation for scalable video multirate multicast over wireless networks. *IEEE Trans. Veh. Technol.*, 69(9):10227–10241, 2020.
- [17] Betty Searcy, Zurh Farus, Bronny Bush, Kevin Muhammad, and Zubair Clinton. Efficient resource management in multicast short video streaming systems. *arXiv preprint arXiv:2407.00552*, 2024.
- [18] Sadaf ul Zuhra, Karl-Ludwig Besser, Prasanna Chaporkar, Abhay Karandikar, and H Vincent Poor. Optimal resource allocation for loss-tolerant multicast video streaming. *Entropy*, 25(7):1045, 2023.
- [19] Doyle Kwon and Duk Kyung Kim. Optimal resource allocation for a single-cell multicast transmission scheme with a supplementary multicast channel. *Electronics*, 8(6):704, 2019.

- [20] Ran Dubin, Amit Dvir, Ofer Hadar, Nissim Harel, and Refael Barkan. Multicast adaptive logic for dynamic adaptive streaming over HTTP network. In *IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, pages 269–274. IEEE, 2015.
- [21] Jeong Yun Kim, Gyu Myoung Lee, and Jun Kyun Choi. Efficient multicast schemes using in-network caching for optimal content delivery. *IEEE Commun. Lett.*, 17(5):1048–1051, 2013.
- [22] Fangxin Wang, Feng Wang, Jiangchuan Liu, Ryan Shea, and Lifeng Sun. Intelligent video caching at network edge: A multi-agent deep reinforcement learning approach. In *IEEE Conf. Comput. Commun. (INFOCOM)*, pages 2499–2508. IEEE, 2020.
- [23] Ruohan Shi, Qilin Fan, Shu Fu, Xu Zhang, Xiuhua Li, and Meng Chen. COCAM: a cooperative video edge caching and multicasting approach based on multi-agent deep reinforcement learning in multi-clouds environment. *J. Cloud Comput.*, 12(1):123, 2023.
- [24] Zhilong Zhang, Luzhi Xu, Xiaomeng Gao, and Danpu Liu. Joint video streaming and hybrid beamforming in multi-connectivity enabled mmwave networks. *IEEE Trans. Veh. Technol.*, 73(8):11289–11302, 2024.
- [25] Sadaf ul Zuhra, Prasanna Chaporkar, Abhay Karandikar, and H Vincent Poor. Multi-connectivity for multicast video streaming in cellular networks. *Netw.*, 4(2):175–195, 2024.
- [26] Siyu Xie, Qi Jiang, Victor C. M. Leung, Zhuo Ma, and Kofi Kwarteng Abrokwa. Joint bitrate adaptation and resource allocation by leveraging SVC setting for wireless scalable video multicast. *IEEE Wirel. Commun. Lett.*, 14(4):1099–1103, 2025.
- [27] Peilong Li, Honghai Zhang, Baohua Zhao, and Sampath Rangarajan. Fast optimal resource allocation for scalable video multicast in broadband wireless networks. In *IEEE Int. Conf. Commun. (ICC)*, pages 1–6. IEEE, 2010.
- [28] Hardik Soni, Walid Dabbous, Thierry Turletti, and Hitoshi Asaeda. NFV-based scalable guaranteed-bandwidth multicast service for software defined ISP networks. *IEEE Trans. Netw. Ser. Manage.*, 14(4):1157–1170, 2017.
- [29] Shisheng Hu, Jie Gao, Xinyu Huang, Conghao Zhou, Mingcheng He, and Xuemin Sherman Shen. Model drift-adaptive resource reservation in ISAC networks: A digital twin-based approach. In *IEEE/CIC Int. Conf. Commun. China (ICCC)*, pages 2143–2148. IEEE, 2024.

- [30] Lun Tang, Qiang Hou, Wen Wen, Dongxu Fang, and Qianbin Chen. Digital-twin-assisted VNF migration through resource prediction in SDN/NVF-enabled IoT networks. *IEEE Internet Things J.*, 11(21):35445–35464, 2024.
- [31] Conghao Zhou, Jie Gao, Mushu Li, Xuemin Sherman Shen, and Weihua Zhuang. Digital twin-empowered network planning for multi-tier computing. *J. Commun. Inform. Netw.*, 7(3):221–238, 2022.
- [32] Jing Li, Song Guo, Weifa Liang, Quan Chen, Zichuan Xu, Wenzheng Xu, and Albert Y Zomaya. Digital twin-assisted, SFC-enabled service provisioning in mobile edge computing. *IEEE Trans. Mobile Comput.*, 23(1):393–408, 2022.
- [33] Sekione Reward Jeremiah, Laurence Tianruo Yang, and Jong Hyuk Park. Digital twin-assisted resource allocation framework based on edge collaboration for vehicular edge computing. *Fut. Gen. Comput. Syst.*, 150:243–254, 2024.
- [34] Yuyin Ma, Fei Song, Giovanni Pau, Ilsun You, and Hongke Zhang. Adaptive service provisioning for dynamic resource allocation in network digital twin. *IEEE Netw.*, 38(1):61–68, 2023.
- [35] Yixue Hao, Jiayi Wang, Dongkun Huo, Nadra Guizani, Long Hu, and Min Chen. Digital twin-assisted URLLC-enabled task offloading in mobile edge network via robust combinatorial optimization. *IEEE J. Sel. Areas Commun.*, 41(10):3022–3033, 2023.
- [36] Sekione Reward Jeremiah, David Camacho, and Jong Hyuk Park. Maximizing throughput in NOMA-enabled industrial IoT networks using digital twin and reinforcement learning. *J. Advanced Research*, 66:59–70, 2024.
- [37] Muhammad Awais, Haris Pervaiz, Qiang Ni, and Wenjuan Yu. Task dependency aware optimal resource allocation for URLLC edge network: A digital twin approach using finite blocklength. *IEEE Trans. Green Commun. Network.*, 9(1):177–190, 2025.
- [38] Dang Van Huynh, Van-Dinh Nguyen, Saeed R Khosravirad, Vishal Sharma, Octavia A Dobre, Hyundong Shin, and Trung Q Duong. URLLC edge networks with joint optimal user association, task offloading and resource allocation: A digital twin approach. *IEEE Trans. Commun.*, 70(11):7669–7682, 2022.
- [39] Muhammet Hevesli, Abegaz Mohammed Seid, Aiman Erbad, and Mohamed Abdallah. Task offloading optimization in digital twin assisted MEC-enabled air-ground IIoT 6G networks. *IEEE Trans. Veh. Technol.*, 73(11):17527–17542, 2024.

- [40] Xinyu Huang, Wen Wu, Shisheng Hu, Mushu Li, Conghao Zhou, and Xuemin Shen. Digital twin based user-centric resource management for multicast short video streaming. *IEEE J. Sel. Topics Signal Process.*, 18(1):50–65, 2024.
- [41] Bo Zhao, Qian Dang, Qin Zhang, Chunhui Du, and Chunliang Li. Intelligent monitoring and scheduling of real-time dynamic data based on digital twin. In *Int. Conf. Front. Electron., Informa. & Comput. Technol. (ICFEICT)*, pages 536–541. IEEE, 2023.
- [42] Deepak Kumar Jain, Sumarga Kumar Sah Tyagi, Subramani Neelakandan, Mohan Prakash, and Lakshmaiya Natrayan. Metaheuristic optimization-based resource allocation technique for cybertwin-driven 6G on IoE environment. *IEEE Trans. Ind. Informa.*, 18(7):4884–4892, 2021.
- [43] Mahnoor Yaqoob, Ramona Trestian, Mallik Tatipamula, and Huan X Nguyen. Digital-twin-driven end-to-end network slicing toward 6G. *IEEE Internet Comput.*, 28(2):47–55, 2023.
- [44] Jing Zhao, Yuanmou Chen, and Yi Huang. QoE-driven wireless communication resource allocation based on digital twin edge network. *IEEE Radio Freq. Identif.*, 8:277–281, 2023.
- [45] Shoufeng Wang, Hua-Min Chen, Jianchao Guo, Ye Ouyang, Fan Li, Xuan Chen, Shuling Wang, Sen Bian, Limeng Ma, and Zhanwu Li. Trinity equilibrium algorithm for computing network digital twin. In *Proc. IEEE Int. Conf. Digital Twins & Parallel Intell. (DTPI)*, pages 405–410. IEEE, 2024.
- [46] Zhengming Zhang, Yongming Huang, Cheng Zhang, Qingbi Zheng, Luxi Yang, and Xiaohu You. Digital twin-enhanced deep reinforcement learning for resource management in networks slicing. *IEEE Trans. Commun.*, 72(10):6209–6224, 2024.
- [47] Haijun Zhang, Xu Ma, Xiangnan Liu, Linpei Li, and Kai Sun. GNN-based power allocation and user association in digital twin network for the terahertz band. *IEEE J. Sel. Areas Commun.*, 41(10):3111–3121, 2023.
- [48] Fangyi Mou, Jiong Lou, Zhiqing Tang, Yuan Wu, Weijia Jia, Yan Zhang, and Wei Zhao. Adaptive digital twin migration in vehicular edge computing and networks. *IEEE Trans. Veh. Technol.*, 74(3):4839–4854, 2025.

- [49] Wenshuai Liu, Yaru Fu, Yongna Guo, Fu Lee Wang, Wen Sun, and Yan Zhang. Two-timescale synchronization and migration for digital twin networks: a multi-agent deep reinforcement learning approach. *IEEE Trans. Wirel. Commun.*, 2024.
- [50] Yejun He, Mengna Yang, Zhou He, and Mohsen Guizani. Resource allocation based on digital twin-enabled federated learning framework in heterogeneous cellular network. *IEEE Trans. Veh. Technol.*, 72(1):1149–1158, 2022.
- [51] Andrzej Sikora, Adam Zielonka, Muhammad Fazal Ijaz, and Marcin Woźniak. Digital twin heuristic positioning of insulation in multimodal electric systems. *IEEE Trans. Consum. Electron.*, 70(1):3436–3445, 2024.
- [52] Bishmita Hazarika, Keshav Singh, Chih-Peng Li, Anke Schmeink, and Kim Fung Tsang. RADiT: Resource allocation in digital twin-driven UAV-aided internet of vehicle networks. *IEEE J. Sel. Areas Commun.*, 41(11):3369–3385, 2023.
- [53] Zhihan Lv, Liang Qiao, and Robert Nowak. Energy-efficient resource allocation of wireless energy transfer for the internet of everything in digital twins. *IEEE Commun. Mag.*, 60(8):68–73, 2022.
- [54] Lun Tang, Yanzhou Yi, Songlin Li, Qiong Huang, and Qianbin Chen. Digital twin construction and resource allocation on internet of vehicles. *IEEE Internet Things J.*, 2024.
- [55] Xinyu Huang, Xue Qin, Mushu Li, Cheng Huang, and Xuemin Shen. Adaptive digital twin-assisted 3C management for QoE-driven MSVS: A GAI-based DRL approach. *IEEE Trans. Cogn. Commun. Network.*, 11(2):858–872, 2025.
- [56] Hong Chen, Terence D Todd, Dongmei Zhao, and George Karakostas. Digital twin model selection for feature accuracy. *IEEE Internet Things J.*, 11(7):11415–11426, 2023.
- [57] Haoye Chai, Huandong Wang, Tong Li, and Zhaocheng Wang. Generative AI-driven digital twin for mobile networks. *IEEE Netw.*, 38(5):84–92, 2024.
- [58] Dianlei Xu, Xiang Su, Sasu Tarkoma, and Pan Hui. Toward sustainable 6g leveraging digital twin and artificial intelligence: Framework and case study. *IEEE Commun. Mag.*, pages 1–7, 2025.
- [59] Zhihan Lv, Dongliang Chen, Hailin Feng, Hu Zhu, and Haibin Lv. Digital twins in unmanned aerial vehicles for rapid medical resource delivery in epidemics. *IEEE Trans. Intell. Transport. Syst.*, 23(12):25106–25114, 2021.

- [60] Xinyu Huang, Haojun Yang, Conghao Zhou, Mingcheng He, Xuemin Shen, and Weihua Zhuang. When digital twin meets generative AI: Intelligent closed-loop network management. *IEEE Netw.*, pages 1–1, 2024.
- [61] Wen Wu, Xinyu Huang, and Tom H Luan. AI-native network digital twin for intelligent network management in 6G. *arXiv preprint arXiv:2410.01584*, 2024.
- [62] Mohammad Sajid Shahriar, Suresh Subramaniam, Motoharu Matsuura, Hiroshi Hasegawa, and Shih-Chun Lin. Digital twin enabled data-driven approach for traffic efficiency and software-defined vehicular network optimization. In *Proc. IEEE Veh. Technol. Conf. (VTC-Fall)*, pages 1–7. IEEE, 2024.
- [63] Qiushuo Hou, Matteo Zecchin, Sangwoo Park, Yunlong Cai, Guanding Yu, Kaushik Chowdhury, and Osvaldo Simeone. Automatic AI model selection for wireless systems: Online learning via digital twinning. *IEEE Trans. Wirel. Commun.*, pages 1–1, 2025.
- [64] Aurélien Chambon, Abderrahim Sahli, Abderrezak Rachedi, and Ahmed Mebarki. Analyzing the integration of 3D building digital twins in optimal iot networks deployment. In *Proc. Int. Wirel. Commun. Mobile Comput. (IWCMC)*, pages 616–621. IEEE, 2024.
- [65] Sébastien Faye, Miguel Camelo, Jean-Sébastien Sottet, Christoph Sommer, Mario Franke, Julien Baudouin, German Castellanos, Régis Decorme, Maria Pia Fanti, Ramin Fuladi, et al. Integrating network digital twinning into future ai-based 6G systems: The 6G-twin vision. In *Proc. Joint European Conf. Networks Commun. & 6G Summit (EuCNC/6G Summit)*, pages 883–888. IEEE, 2024.
- [66] Paolo Bellavista, Carlo Giannelli, Marco Mamei, Matteo Mendula, and Marco Picone. Application-driven network-aware digital twin management in industrial edge environments. *IEEE Trans. Ind. Inform.*, 17(11):7791–7801, 2021.
- [67] Yagmur Yigit, Hamed Ahmadi, Gokhan Yurdakul, Berk Canberk, Trang Hoang, and Trung Q Duong. Digi-infrastructure: Digital twin-enabled traffic shaping with low-latency for 6G smart cities. *IEEE Commun. Stand. Mag.*, 8(3):28–34, 2024.
- [68] Jing Zhao, Xiaoming Xiong, and Yuanmou Chen. Design and application of a network planning system based on digital twin network. *IEEE J. Radio Freq. Identif.*, 6:900–904, 2022.

- [69] Dawei Chen, Dan Wang, Yifei Zhu, and Zhu Han. Digital twin for federated analytics using a bayesian approach. *IEEE Internet Things J.*, 8(22):16301–16312, 2021.
- [70] Abubakar Isah, Hyeju Shin, Seungmin Oh, Sangwon Oh, Ibrahim Aliyu, Tai-won Um, and Jinsul Kim. Digital twins temporal dependencies-based on time series using multivariate long short-term memory. *Electronics*, 12(19):4187, 2023.
- [71] Wen Sun, Peng Wang, Ning Xu, Gaozu Wang, and Yan Zhang. Dynamic digital twin and distributed incentives for resource allocation in aerial-assisted internet of vehicles. *IEEE Internet Things J.*, 9(8):5839–5852, 2021.
- [72] Yinlin Ren, Shaoyong Guo, Bin Cao, and Xuesong Qiu. End-to-end network SLA quality assurance for C-RAN: A closed-loop management method based on digital twin network. *IEEE Trans. Mobile Comput.*, 23(5):4405–4422, 2023.
- [73] Joe David, Andrei Lobov, and Minna Lanz. Attaining learning objectives by ontological reasoning using digital twins. *Procedia Manufactur.*, 31:349–355, 2019.
- [74] Hui Yuan, Shiyun Zhao, Junhui Hou, Xuekai Wei, and Sam Kwong. Spatial and temporal consistency-aware dynamic adaptive streaming for 360-degree videos. *IEEE J. Sel. Top. Signal Process.*, 14(1):177–193, 2019.
- [75] Kunlun Wang, Jiong Jin, Yang Yang, Tao Zhang, Arumugam Nallanathan, Chintha Tellambura, and Bijan Jabbari. Task offloading with multi-tier computing resources in next generation wireless networks. *IEEE J. Sel. Areas Commun.*, 41(2):306–319, 2022.
- [76] Wenxiao Zhang, Feng Qian, Bo Han, and Pan Hui. Deepvista: 16K panoramic cinema on your mobile device. In *Proc. Int. World Wide Web (WWW)*, pages 2232–2244, Ljubljana, Slovenia, 2021.
- [77] Jinxi Li, Yutong Xu, Yang Cao, Jiabin Zhu, and Desheng Wang. Utility-driven joint caching and bitrate allocation for real-time immersive videos. *IEEE J. Sel. Top. Signal Process.*, 2023.
- [78] Kamran Zahoor, Kashif Bilal, Aiman Erbad, and Amr Mohamed. Service-less video multicast in 5G: Enablers and challenges. *IEEE Netw.*, 34(3):270–276, 2020.
- [79] Kunlun Wang, Wen Chen, Jun Li, Yang Yang, and Lajos Hanzo. Joint task offloading and caching for massive MIMO-aided multi-tier computing networks. *IEEE Trans. Commun.*, 70(3):1820–1833, 2022.

- [80] Navid Reyhanian and Zhi-Quan Luo. Data-driven adaptive network slicing for multi-tenant networks. *IEEE J. Sel. Top. Signal Process.*, 16(1):113–128, 2021.
- [81] Xinyu Huang, Mushu Li, Wen Wu, Conghao Zhou, and Xuemin Shen. Digital twin-assisted collaborative transcoding for better user satisfaction in live streaming. In *Proc. IEEE Int. Conf. Commun. (ICC)*, pages 4051–4056, Rome, Italy, 2023.
- [82] Xiaobin Tan, Lei Xu, Quan Zheng, Simin Li, and Bei Liu. QoE-driven DASH multicast scheme for 5G mobile edge network. *J. Commun. Inf. Netw.*, 6(2):153–165, 2021.
- [83] Shangyue Zhu, Theo Karagioules, Emir Halepovic, Alamin Mohammed, and Aaron D Striegel. Swipe along: a measurement study of short video services. In *Proc. ACM Multimedia Syst. Conf.*, pages 123–135, Athlone, Ireland, 2022.
- [84] Zhuqi Li, Yaxiong Xie, Ravi Netravali, and Kyle Jamieson. Dashlet: Taming swipe uncertainty for robust short video streaming. In *Proc. USENIX Symp. Netw. Syst. Des. Implement. (NSDI)*, pages 1583–1599, Boston, MA, USA, 2023.
- [85] Guanglun Huang, Wei Gong, Baoxian Zhang, Chunxi Li, and Cheng Li. An on-line buffer-aware resource allocation algorithm for multiuser mobile video streaming. *IEEE Trans. Veh. Technol.*, 69(3):3357–3369, 2020.
- [86] Xudong Gong, Qinlin Feng, Yuan Zhang, Jiangling Qin, Weijie Ding, Biao Li, Peng Jiang, and Kun Gai. Real-time short video recommendation on mobile devices. In *Proc. ACM Int. Conf. Infor. Knowl. Management*, pages 3103–3112, Atlanta, GA, USA, 2022.
- [87] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.
- [88] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proc. AAAI Conf. Artif. Intell.*, volume 30, Phoenix, Arizona, USA, 2016.
- [89] Andrew Ng. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011.
- [90] Xinyu Huang, Conghao Zhou, Wen Wu, Mushu Li, Huaqing Wu, and Xuemin Shen. Personalized QoE enhancement for adaptive video streaming: A digital twin-assisted scheme. In *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, pages 4001–4006, 2022.

- [91] Xinfeng Zhang, Chao Yang, Haiqiang Wang, Wei Xu, and C-C Jay Kuo. Satisfied-user-ratio modeling for compressed video. *IEEE Trans. Image Process.*, 29:3777–3789, 2020.
- [92] Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli. A control-theoretic approach for dynamic adaptive video streaming over HTTP. In *Proc. ACM SIGCOMM*, pages 325–338, 2015.
- [93] Wen Wu, Nan Chen, Conghao Zhou, Mushu Li, Xuemin Shen, Weihua Zhuang, and Xu Li. Dynamic RAN slicing for service-oriented vehicular networks via constrained learning. *IEEE J. Sel. Areas Commun.*, 39(7):2076–2089, 2021.
- [94] Dimitri Bertsekas. *Network optimization: Continuous and discrete models*, volume 8. Athena Scientific, 1998.
- [95] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [96] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Trans. Database Syst.*, 42(3):1–21, 2017.
- [97] Vladimir I Norkin, Georg Ch Pflug, and Andrzej Ruszczyński. A branch and bound method for stochastic global optimization. *Math. Program.*, 83:425–450, 1998.
- [98] Arash Tavakoli, Fabio Pardo, and Petar Kormushev. Action branching architectures for deep reinforcement learning. In *Proc. AAAI Conf. Artif. Intell.*, volume 32, 2018.
- [99] Xiantao Jiang, F Richard Yu, Tian Song, and Victor CM Leung. A survey on multi-access edge computing applied to video streaming: Some research issues and challenges. *IEEE Commun. Surv. Tut.*, 23(2):871–903, 2021.
- [100] James Nightingale, Pablo Salva-Garcia, Alcaraz Calero, and Qi Wang. 5G-QoE: QoE modelling for ultra-HD video streaming in 5G networks. *IEEE Trans. Broadcast.*, 64(2):621–634, 2018.
- [101] Xinyu Huang, Lijun He, Liejun Wang, and Fan Li. Towards 5G: Joint optimization of video segment caching, transcoding and resource allocation for adaptive video streaming in a multi-access edge computing network. *IEEE Trans. Veh. Technol.*, 70(10):10909–10924, 2021.

- [102] Weiwei Huang, Yipeng Zhou, Xueyan Xie, Di Wu, Min Chen, and Edith Ngai. Buffer state is enough: Simplifying the design of QoE-aware HTTP adaptive video streaming. *IEEE Trans. Broadcast.*, 64(2):590–601, 2018.
- [103] Shuoyao Wang, Suzhi Bi, and Yingjun Zhang. Adaptive wireless video streaming: Joint transcoding and transmission resource allocation. *IEEE Trans. Wirel. Commun.*, 21(5):3208–3221, 2022.
- [104] Xinyu Huang, Haojun Yang, Shisheng Hu, and Xuemin Shen. Digital twin-driven network architecture for video streaming. *IEEE Netw.*, 38(6):334–341, 2024.
- [105] Qi Guo, Fengxiao Tang, Tiago Rodrigues, and Nei Kato. Five disruptive technologies in 6G to support digital twin networks. *IEEE Wirel. Commun.*, pages 1–8, 2023, Early Access.
- [106] Mathias Wien, Heiko Schwarz, and Tobias Oelbaum. Performance analysis of SVC. *IEEE Trans. Circuits Syst. Video Technol.*, 17(9):1194–1203, 2007.
- [107] Yijie Mao, Bruno Clerckx, and Victor OK Li. Rate-splitting for multi-antenna non-orthogonal unicast and multicast transmission: Spectral and energy efficiency analysis. *IEEE Trans. Commun.*, 67(12):8754–8770, Dec. 2019.
- [108] Ahmet Zahid Yalcin, Melda Yuksel, and Bruno Clerckx. Rate splitting for multi-group multicasting with a common message. *IEEE Trans. Veh. Technol.*, 69(10):12281–12285, Jul. 2020.
- [109] Chao Zhou, Yixuan Ban, Yangchao Zhao, Liang Guo, and Bing Yu. PDAS: Probability-driven adaptive streaming for short video. In *Proc. ACM Int. Conf. Multimedia (ACM MM)*, pages 7021–7025, Lisboa, Portugal, 2022.
- [110] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pages 1995–2003, New York, NY, USA, 2016.
- [111] Dieter Kraft. A software package for sequential quadratic programming. *Forschungsbericht- DFVLR*, 1988.
- [112] Stephen Boyd, Lin Xiao, and Almir Mutapcic. Subgradient methods. *lecture notes of EE392o, Stanford University, Autumn Quarter*, 2004:2004–2005, 2003.

- [113] Tantan Zhao, Fan Li, and Lijun He. DRL-based joint resource allocation and device orchestration for hierarchical federated learning in NOMA-enabled industrial IoT. *IEEE Trans. Ind. Informat.*, 19(6):7468–7479, 2023.
- [114] Yun Gao, Xin Wei, and Liang Zhou. Personalized QoE improvement for networking video service. *IEEE J. Sel. Areas Commun.*, 38(10):2311–2323, 2020.
- [115] Zhaoyu Zhang, Xiaoxue Wang, Chen Sun, and Yan Zhang. Cooperative transmission for multimedia video distribution: Models, algorithms and implementation. *IEEE Netw.*, 39(1):174–180, 2025.
- [116] Yousuf Abo Rahama, Mohamed S. Hassan, and Mahmoud H. Ismail. A stochastic-based rate control approach for video streaming over cognitive radio networks. *IEEE Trans. Cogn. Commun. Netw.*, 5(1):181–192, 2019.
- [117] Zhenyu Tao, Wei Xu, Yongming Huang, Xiaoyun Wang, and Xiaohu You. Wireless network digital twin for 6G: Generative AI as a key enabler. *IEEE Wirel. Commun.*, 31(4):24–31, 2024.
- [118] Yun-Cheng Wang, Jintang Xue, Chengwei Wei, and C-C Jay Kuo. An overview on generative AI at scale with edge-cloud computing. *IEEE Open J. Commun. Soc.*, 2023.
- [119] Emanuele Artioli. Generative AI for HTTP adaptive streaming. In *Proc. ACM Multimedia Syst. Conf.*, pages 516–519, Bari, Italy, 2024.
- [120] Xuemin Shen, Jie Gao, Wen Wu, Mushu Li, Conghao Zhou, and Weihua Zhuang. Holistic network virtualization and pervasive network intelligence for 6G. *IEEE Commun. Surveys Tuts.*, 24(1):1–30, 2022.
- [121] Emanuele Artioli, Farzad Tashtarian, and Christian Timmerer. DIGITWISE: Digital twin-based modeling of adaptive video streaming engagement. In *Proc. ACM Multimedia Syst. Conf.*, pages 78–88, 2024.
- [122] Lianyong Qi, Xiaolong Xu, Xiaotong Wu, Qiang Ni, Yuan Yuan, and Xuyun Zhang. Digital-twin-enabled 6G mobile network video streaming using mobile crowdsourcing. *IEEE J. Sel. Areas Commun.*, 2023.
- [123] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pages 1587–1596, Stockholmsmässan, Stockholm, Sweden, Jul. 2018.

- [124] Geng Sun, Wenwen Xie, Dusit Niyato, Fang Mei, Jiawen Kang, Hongyang Du, and Shiwen Mao. Generative ai for deep reinforcement learning: Framework, analysis, and use cases. *IEEE Wirel. Commun.*, pages 1–10, 2025.
- [125] Xialei Liu, Marc Masana, Luis Herranz, Joost Van de Weijer, Antonio M Lopez, and Andrew D Bagdanov. Rotate your networks: Better weight consolidation and less catastrophic forgetting. In *Proc. Int. Conf. Pattern Recognit. (ICPR)*, pages 2262–2268, Beijing, China, Aug. 2018.
- [126] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pages 233–248, Munich, Germany, Sept. 2018.
- [127] Jiangpeng He, Runyu Mao, Zeman Shao, and Fengqing Zhu. Incremental learning in online scenario. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recogn. (CVPR)*, pages 13926–13935, Virtual, Jun. 2020.
- [128] Pavlo M Radiuk. Impact of training set batch size on the performance of convolutional neural networks for diverse datasets. *Infor. Technol. Manage. Sci.*, 20(1):20–24, Dec. 2017.
- [129] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 33, pages 6840–6851, Virtual, Dec. 2020.
- [130] Hongyang Du, Zonghang Li, Dusit Niyato, Jiawen Kang, Zehui Xiong, Huawei Huang, and Shiwen Mao. Diffusion-based reinforcement learning for edge-enabled AI-generated content services. *IEEE Trans. Mob. Comput.*, pages 1–16, Early Access, 2024.
- [131] Xinyu Huang, Shisheng Hu, Haojun Yang, Xinghan Wang, Yingying Pei, and Xuemin Shen. Digital twin-based network management for better QoE in multicast short video streaming. *IEEE Trans. Wirel. Commun.*, 23(11):16187–16202, 2024.
- [132] Stephen Boyd and Jacob Mattingley. Branch and bound methods. *Notes for EE364b, Stanford University*, 2006:07, 2007.
- [133] Tantan Zhao, Fan Li, and Lijun He. Secure video offloading in MEC-enabled IIoT networks: A multicell federated deep reinforcement learning approach. *IEEE Trans. Ind. Informat.*, 20(2):1618–1629, 2024.

# Appendix A

When the variable  $\mathbf{a}$  is determined, the opposite objective function in the minimization problem can be expressed in Eq. (1).

$$\begin{aligned} \hat{\Upsilon}_g(a_g^*, \beta_g) = & \omega_g \left( \sum_{m \in \Omega_g} -1 + \frac{1}{2a_{g,m}^{l^*} \sum_{j=1}^{l^*} z_{g,m}^j / \tau + 1} + \frac{\lambda_{g,2}}{|\Omega_g|} \sum_{m \in \Omega_g} \left| Q_{g,m}^{l^*}(a_{g,m}^{l^*}) - Q_{g,m-1}^{l^*}(a_{g,m-1}^{l^*}) \right| \right) \\ & + \omega_g \lambda_{g,1} \left[ \max \left\{ \frac{\sum_{m \in \Omega_g} a_{g,m}^{l^*} \sum_{j=1}^{l^*} z_{g,m}^j}{\min_{k \in \bigcup_{d=1}^g \mathcal{K}_d} \beta_g B \log_2 \left( 1 + \frac{|h_{g,k}|^2 P_D}{N_0} \right)}, \frac{\mu \sum_{m \in \Omega_g} a_{g,m}^{l^*} \sum_{j=1}^{l^*} z_{g,m}^j}{\beta_g C} \right\} - \tilde{q}_g^0 \right]^+. \end{aligned} \quad (1)$$

Let denote

$$\Xi_1(\beta_g) = \left\{ \frac{\sum_{m \in \Omega_g} a_{g,m}^{l^*} \sum_{j=1}^{l^*} z_{g,m}^j}{\min_{k \in \bigcup_{d=1}^g \mathcal{K}_d} \beta_g B \log_2 \left( 1 + \frac{|h_{g,k}|^2 P_D}{N_0} \right)} \right\}$$

and

$$\Xi_2(\beta_g) = \left\{ \frac{\mu \sum_{m \in \Omega_g} a_{g,m}^{l^*} \sum_{j=1}^{l^*} z_{g,m}^j}{\beta_g C} \right\},$$

then the last term of Eq. (1) can be transformed into

$$\omega_g \lambda_{g,1} \left( \max \left\{ [\Xi_1(\beta_g) - \tilde{q}_g^0]^+, [\Xi_2(\beta_g) - \tilde{q}_g^0]^+ \right\} \right).$$

Since the second derivative of functions  $\Xi_1(\beta_g)$  and  $\Xi_2(\beta_g)$  are positive values, i.e.,  $\frac{\partial^2 \Xi_1(\beta_g)}{\partial^2 \beta_g} \geq 0$ ,  $\frac{\partial^2 \Xi_2(\beta_g)}{\partial^2 \beta_g} \geq 0$ , they are convex functions. Then, we need to prove the convexity of function  $\psi_1(\beta_g) = [\Xi_1(\beta_g) - \tilde{q}_g^0]^+$  and  $\psi_2(\beta_g) = [\Xi_2(\beta_g) - \tilde{q}_g^0]^+$ .

Consider an arbitrary value  $\theta \in (0, 1)$ , and arbitrary values  $\beta_{g,1}$ , and  $\beta_{g,2}$ , we have

$$\begin{aligned}
& \psi_1(\theta\beta_{g,1} + (1 - \theta)\beta_{g,2}) \\
&= \max\{\Xi_1(\theta\beta_{g,1} + (1 - \theta)\beta_{g,2}) - \tilde{q}_g^0, 0\} \\
&\leq \max\{\theta\Xi_1(\beta_{g,1}) + (1 - \theta)\Xi_1(\beta_{g,2}) - \tilde{q}_g^0, 0\} \\
&\leq \theta \max\{\Xi_1(\beta_{g,1}) - \tilde{q}_g^0, 0\} \\
&\quad + (1 - \theta) \max\{\Xi_1(\beta_{g,2}) - \tilde{q}_g^0, 0\} \\
&= \theta\psi_1(\beta_{g,1}) + (1 - \theta)\psi_1(\beta_{g,2}). \tag{2}
\end{aligned}$$

Therefore, the function  $\psi_1(\beta_g)$  is convex. The same validation method can be applied to  $\psi_2(\beta_g)$  and  $\max\{\psi_1(\beta_g), \psi_2(\beta_g)\}$ . Therefore, the component  $\omega_g\lambda_{g,1}(\max\{\psi_1(\beta_g), \psi_2(\beta_g)\})$  is convex. Since the first two terms of the first line of Eq. (1) are constant, they do not affect the convexity of the transformed objective function. Therefore, function  $\hat{\Upsilon}_g(a_g^*, \beta_g)$  is convex. For functions  $\hat{\Upsilon}_g(a_g^*, \beta_g)$  with different  $g$ , they have an identical domain and are mutually independent with each other, so their summation is also a convex function. Based on the above analysis, the transformed objective function is convex.