Digital Twin-Based Network Management for Better QoE in Multicast Short Video Streaming

Xinyu Huang, *Graduate Student Member, IEEE*, Shisheng Hu, *Graduate Student Member, IEEE*, Haojun Yang, *Member, IEEE*, Xinghan Wang, Yingying Pei, *Graduate Student Member, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*— Multicast short video streaming can enhance bandwidth utilization by enabling simultaneous video transmission to multiple users over shared wireless channels. The existing network management schemes mainly rely on the sequential buffering principle and general quality of experience (QoE) model, which may deteriorate QoE when users' swipe behaviors exhibit distinct spatiotemporal variation. In this paper, we propose a digital twin (DT)-based network management scheme to enhance QoE. Firstly, user status emulated by the DT is utilized to estimate the transmission capabilities and watching probability distributions of sub-multicast groups (SMGs) for an adaptive segment buffering. The SMGs' buffers are aligned to the unique virtual buffers managed by the DT 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 SMGs. 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 DT-based network management scheme outperforms benchmark schemes in terms of QoE improvement.**

***Index Terms*— Digital twin, network management, multicast transmission, short video, QoE.**

I. Introduction

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 [1]. 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 [2]. 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 [3].

For multicast short video streaming (MSVS) services, one of the key metrics to evaluate the performance is quality of experience (QoE). It is typically a multi-dimensional metric used to quantify users' subjective and objective watching experience [4]. For instance, video quality (resolution, bitrate), latency, as well as rebuffering events are the most important factors that dominate the QoE level [5]. To maintain users' QoE at a high level, efficient network management is essential, such as buffer management and resource scheduling [6], [7]. 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 (MG), one MG can be further divided into multiple sub-MGs (SMGs). The SMG with a leading video playback is defined as the leading SMG, otherwise, the lagging SMG. The bandwidth and computing resources need to be flexibly and accurately assigned to each SMG to reduce service delay, which consists of multicast transmission delay and trancoding 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 SMGs' segment buffering. As a result, users may suffer from frequent playback lags and low video quality, thus leading to a low QoE.

Digital twin (DT) is a promising technique to optimize network management for better QoE. DT is defined as a full digital representation of a physical object, and real-time synchronization between the physical object and its corresponding digital replica [8]. As an essential component embedded in the next-generation communication networks, DT is comprised of a data pool as well as several data processing and decision-making modules, which can efficiently emulate users' behaviors and network conditions, abstract distilled features,

Manuscript received 19 January 2024; revised 27 May 2024; accepted 29 July 2024. Date of publication 16 August 2024; date of current version 13 November 2024. The associate editor coordinating the review of this article and approving it for publication was S. R. Khosravirad. *(Corresponding author: Xinyu Huang.)*

The authors are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: x357huan@uwaterloo.ca; s97hu@uwaterloo.ca; h88yang@uwaterloo.ca; x243wang@uwaterloo.ca; y32pei@uwaterloo.ca; sshen@uwaterloo.ca).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TWC.2024.3438159.

Digital Object Identifier 10.1109/TWC.2024.3438159

and make network management decisions [9], [10]. Owing to its powerful emulation, analysis, and decision-making capabilities, communication networks can more intelligently perceive SMGs' behavior patterns, finely control SMGs' buffer update, and provide customized network management strategies to enhance users' QoE. In this work, the precise role of constructed DT 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 DT-based network management for MSVS 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 SMGs influence each other, collectively impacting the MG's QoE. Therefore, it is paramount to establish a QoE model specifically tailored for MSVS. 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 users' high QoE is important.

Designing an efficient DT-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 SMG can still receive segments from other leading SMGs in its scheduling slot, this interactivity results in complex QoE estimation. Therefore, how to characterize the impact of multicast segment buffering among SMGs 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 DT-based network management scheme, which can effectively enhance users' QoE. The main contributions are summarized as follows:

- Firstly, we propose a novel DT-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 DT for status emulation. The emulated status is used to abstract the SMGs' transmission capabilities and the watching probability distribution of segments. Based on the abstracted information, DT can make an adaptive segment buffering decision. Furthermore, DT is utilized to construct and manage virtual buffers for each SMG for a fine-grained buffer update.
- Secondly, we establish a multicast QoE model to quantify the impact of multicast segment buffering among SMGs. 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 nonlinear 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 deep reinforcement learning (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 DT-based network management scheme can effectively enhance QoE as compared with the state-of-the-art network management schemes.

The remainder of this paper is organized as follows. Related works are introduced in Section II. The system model is first built in Section III. Then, the DT-assisted buffer management and the multicast QoE model are presented in Sections IV and V, respectively. Next, the problem formulation and the proposed scheduling algorithm are shown in Section VI, followed by simulation results in Section VII. Finally, Section VIII concludes this paper.

II. Related Work

In this section, we introduce the existing related work from three aspects, i.e., network management for multicast streaming, DT-assisted network management, and main differences compared with our work.

A. Network Management for Multicast Streaming

To facilitate the efficient MSVS within radio access networks (RANs), extensive works are devoted to optimizing network management performance from different directions, such as adaptive video bitrate, multi-connectivity management, and transmission and transcoding scheduling. Specifically,

Zhang et al. conducted a large-scale measurement of short video services over wireless networks from three aspects, i.e., video content characteristics, network analytics, and video streaming analytics, which revealed the statistical relationship between user behaviors and short video traffic [11]. Considering the impact of users' diversified viewing behaviors on bandwidth wastage, the authors in [12], [13], and [14] studied the advanced video prefetching/buffering schemes to reduce bandwidth wastage. To reduce video quality fluctuation, Guo et al. employed deep reinforcement learning to automatically generate a quality-driven bitrate decision model to dynamically determine the bitrate for each short video segment [15]. However, the advanced short video buffering schemes still need to be closely coordinated with intelligent and customized network management to provide better short video streaming services. In addition, Taha et al. studied the impact of the characteristics of videos, wireless channel capabilities, and users' profiles on the MG's QoE, and designed an efficient machine learning algorithm to adaptively adjust the video bitrate [16]. Lie et al. integrated the scalable video coding (SVC) technology with the transmission delay constraint, and exploited the hard deadline constrained prioritized data structure and user feedback to make an optimal adaptive encoding and scheduling strategy, which can enhance the average network throughput [17]. To achieve high-quality and cost-efficient multicast video services, Zhong et al. proposed a novel buffer-nadir-based multicast mechanism, formulated the multicast-aware task offloading problem, and devised a joint optimization algorithm for data scheduling and task offloading, respectively [18]. Additionally, Zuhra et al. proposed the procedures of establishing multi-connectivity and a greedy approximation algorithm to solve the associated resource allocation problem, which can effectively increase the number of served users [19]. Daher et al. proposed a dynamic clustering algorithm based on the minimization of a submodular function that integrated the traffic in each cell and the MG's average signal to interference plus noise ratio (SINR), which can maintain an acceptable transmission failure probability and enhance the MG's average SINR [20]. To improve the robustness of multicast transmission, Zhang et al. proposed a cooperative multicast framework, where users can recover videos with quality proportional to their channel conditions. A joint power allocation and segment scheduling problem was formulated to minimize the overall distortion and solved by a provably convergent optimal algorithm [21]. Considering the characteristic of stochastic channels, Zhang et al. further analyzed the video-layer recovery failure probability and estimated the MG's average QoE. Based on the information, an optimal scheduling algorithm was developed based on the hidden monotonicity of the problem to maximize the MG's QoE [22]. However, these schemes usually require real-time data collection and efficient data processing on user and network status information, which poses a high requirement for efficient network management.

B. Digital Twin-Assisted Network Management

As an essential virtualization technology, DT was first introduced to monitor and mitigate anomalous events for flying vehicles [23]. By introducing DT into RANs, we can realize holistic network virtualization for efficient network management. We refer readers to recent comprehensive surveys and tutorials on DT to become familiar with this topic [24], [25], [26]. There also exist some technical papers aiming at utilizing DT to improve network management performance. Specifically, Bellavista et al. proposed an application-driven DT networking middleware to simplify the interaction with heterogeneous distributed industrial devices and flexibly manage network resources, which can effectively reduce communication overhead [27]. Qi et al. leveraged the DT as a centric controller to encourage edge devices to share their idle resources and get a reward from other devices with poor network performance, which can efficiently reduce the service delay [28]. Considering that the real networks are not static, Dong et al. constructed the DT of the wireless networks to generate labeled training samples, where the network topology, channel and queueing models, and fundamental rules were adopted in the DT to mirror the real networks [29]. A DT-assisted resource demand prediction scheme was proposed to enhance prediction accuracy for MSVS, including the DT construction, the accurate and fast MG construction, and the MG's swipe probability distribution abstraction [30]. To further enhance users' QoE in dynamic and heterogeneous environments, an intelligent resource allocation strategy with low communication overhead was proposed, where DT was utilized to monitor the current network operation status and enable intelligent decision-making [31]. Furthermore, Jeremiah et al. proposed to construct a central DT to simulate dynamic and heterogeneous networks, which can enhance the efficiency of edge collaboration and real-time resource information availability [32]. These pioneering works can effectively improve communication overhead, service delay, and QoE, but lack the effective network pattern information abstraction mechanism, which may cause the inefficiency of network management.

C. Main Differences Compared With Our Work

In our previous work [3], we first designed an efficient data abstraction mechanism for the digital twin to abstract MGs' swipe probability distribution. Then, a novel user satisfaction model was established to quantify the impact of bandwidth and computing resource reservation on user satisfaction. Finally, A low-complexity scheduling algorithm was developed to find the optimal resource reservation decisions. Compared with existing short video streaming frameworks [12], [13], [14], [15], [33], this work focuses on filling the research gap of the multicast transmission mechanism applied to short video streaming services. Firstly, the existing adaptive video buffering schemes cannot be well applied to multicast short video streaming due to the mutual influence between users. Secondly, the existing short video streaming frameworks mainly used the common QoE model that cannot accurately quantify the impact of multicast segment buffering on user QoE. Therefore, we have further proposed a novel DT-assisted buffer management scheme to incorporate the impact of swipe behaviors. Furthermore, we have established a multicast QoE

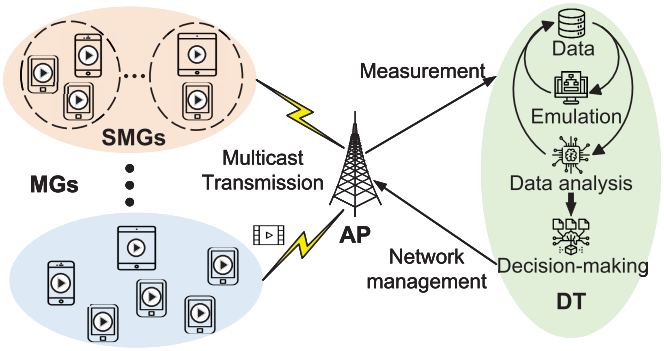


Fig. 1. DT-assisted MSVS framework.

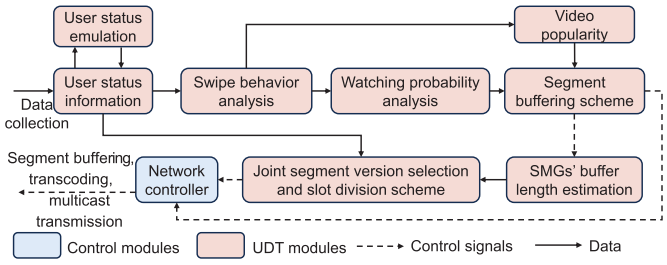


Fig. 2. Proposed DT-assisted MSVS workflow.

model to quantify the impact of multicast segment buffering among SMGs. Finally, a data-model-driven algorithm has been developed to efficiently solve the formulated mixed-integer nonlinear programming problem for better QoE.

III. System Model

As shown in Fig. 1, we consider a DT-assisted MSVS scenario, which consists of an access point (AP), multiple MGs, and one DT.

- 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 MG. In addition, it is responsible for collecting users' network-related and behavior-related information to update DT data.
- Multicast group: Each MG consists of multiple users using short video services. The same video sequences will be transmitted from the AP to one MG over shared wireless channels. The MG's construction and update mainly depend on users' similarities in swipe behaviors, channel conditions, locations, and preferences, as discussed in [3]. Due to asynchronous swipe behaviors, users' devices in the same MG still have different playback stamps and buffer lengths. Therefore, one MG is further divided into multiple SMGs, denoted by $\mathcal{G} = \{1, \cdots, G\}$. The number of total users in the system is denoted by K. The set of users in SMG g is denoted by $\{\mathcal{K}_g\}_{g\in\mathcal{G}}$.
- Digital twin: It consists of a database storing users' status information, a status emulation module, as well as multiple data analysis and decision-making modules. DT 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 DT 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. 2, we present the proposed DT-assisted MSVS 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 SMGs' buffer update decisions. Next, SMGs' 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.

TABLE I
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
Λ_i	The number of segments in video i
T_s	The scheduling slot length
τ	The time duration of one segment
$r_{g,k}$	The transmission capability of user k in SMG g
q_g^f	The length of buffer f in SMG g
D_g	The transmission delay of SMG g
S_g	The service delay of SMG g
Q_g	The video quality of newly buffered segments in SMG g
V_g	The video quality variation between adjacent segments in SMG g
Υ_g	The multicast QoE of SMG g

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

IV. DT-Assisted Buffer Management Scheme

In this section, we first propose an adaptive segment buffering scheme based on the DT-analyzed watching probability distribution, and then design a fine-grained buffer update scheme based on the DT-assisted virtual buffer management.

A. DT Construction

DT consists of multiple modules, which are summarized as status emulation, data analysis, and decision-making, as shown in Fig. 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 long short-term memory (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 DT and users. By integrating these two kinds of data, DT can accurately emulate users' real-time status for network management.

Secondly, two kinds of data analysis modules are embedded in the DT, i.e., the swipe behavior analysis module and the watching probability analysis module. The former has been investigated in [3] for resource demand prediction. The latter is our focus in this paper, 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 DT. Based on the derived information, DT can realize an accurate buffer management.

Thirdly, based on the emulated status and analyzed data, DT can realize tailored decision-making to further enhance users' QoE. Three kinds of decision-making modules are designed to make the segment buffering scheme, the SMGs' 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.

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

B. DT-Assisted Segment Buffering Scheme

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 DT-assisted segment buffering scheme. The specific analysis is as follows.

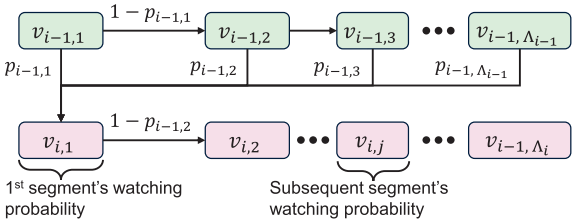


Fig. 3. Watching probability analysis.

As shown in Fig. 3, DT recommends a video list for the MG in each large timescale, referred in [3]. Each video sequence consists of multiple segments with the same time length. The recommended video list is denoted by $\{v_{i,j}\}_{i\in\mathcal{N},j\in\{1,\cdots,\Lambda_i\}}$, where $\mathcal{N}$ is the recommended video index list, and Λ_i is the number of segments in video i. Since the probability of watching next segment $j+1$ only depends on the current state [34], 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 MG 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

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

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

For the 1st 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

$$w_{i,1} = w_{i-1,1}\left(p_{i-1,1} + \sum_{j=1}^{\Lambda_{i-1}-1} p_{i-1,j+1}\prod_{k=1}^{j}(1-p_{i-1,k})\right). \quad (2)$$

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

$$w_{i,j} = w_{i,1}\prod_{k=1}^{j-1}(1-p_{i,k}), \forall j \in \{2,\ldots,\Lambda_i\}. \quad (3)$$

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 MG'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 SMG to observe how many segments can be buffered.

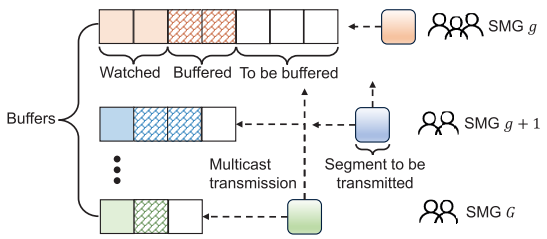


Fig. 4. Multicast transmission mechanism among SMG groups.

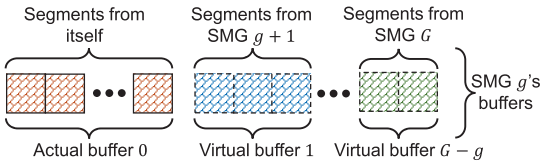


Fig. 5. DT-assisted SMG's buffer management.

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

$$n_{\text{buffer}} = \sum_{g\in\mathcal{G}} \left[\frac{T_{\text{s}} - q_g}{\tau}\right]^+, \tag{4}$$

where T_{s} and τ represent the scheduling slot length and the segment length, respectively. Here, q_g represents SMG 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 SMG groups, as shown in Fig. 4. Each video sequence consists of multiple segments, represented by different colors. The index orders of SMGs are consistent with the video viewing progress, sorted from low to high. SMGs can be classified into lagging SMGs and leading SMGs based on their video viewing progress. The lagging SMGs can receive the video segments from leading SMGs in their scheduling slot through multicast transmission in a MG, while the leading SMGs do not receive the video segments from the lagging SMGs in their scheduling slot. Through this way, the video segments can be buffered in advance for lagging SMGs, which can avoid repeated segment buffering in the future scheduling slot. Therefore, the transmission capability of SMG $g+1$ needs to consider users' channel conditions both from itself and SMG 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.,

$$\begin{cases} \sum_{m=1}^{\tilde{n}_g^{\text{B}}} \sum_{l=1}^{\bar{l}} z_{g,m}^l \le \min_{k\in\bigcup_{d=1}^{g}\mathcal{K}_d} T_{\text{s}} r_{g,k} \le \sum_{m=1}^{\tilde{n}_g^{\text{B}}+1} \sum_{l=1}^{\bar{l}} z_{g,m}^l, \forall g \in \mathcal{G}, \\ \mu \sum_{m=1}^{\tilde{n}_g^{\text{C}}} \sum_{l=2}^{\bar{l}} z_{g,m}^l \le T_{\text{s}} C \le \mu \sum_{m=1}^{\tilde{n}_g^{\text{C}}+1} \sum_{l=2}^{\bar{l}} z_{g,m}^l, \forall g \in \mathcal{G}, \end{cases} \tag{5}$$

where $z_{g,m}^l$ is the file size of segment m of version l for SMG g, which adopts the SVC principle [35]. Here, $\tilde{n}_g^{\text{B}}$ and $\tilde{n}_g^{\text{C}}$ represent the segment buffering numbers by allocating reserved bandwidth resources and computing resources, respectively. Here, $\bar{l}$ is the average segment version and parameter μ 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 SMG g, which is given by

$$r_{g,k} = B\log_2\left(1 + \left|h_{g,k}\right|^2 P_{\text{D}}/N_0\right), \tag{6}$$

where $h_{g,k}$ is the channel gain of user k in SMG 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 MG, respectively. Considering users' diversity in swipe behaviors, video preferences, and locations, the number of users in each SMG is very limited to guarantee transmission efficiency, usually 3 to 5 users with certain distances. Therefore, the signal interference between users within SMG g is relatively small, which is ignored in Eq. (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 SMG is selected [36], [37].

Since Eq. (5) consists of monotonically increasing functions, the maximum segment buffering numbers $\tilde{n}_g^{\text{B}}$ and $\tilde{n}_g^{\text{C}}$ can be uniquely determined by increasing their own values. To satisfy both bandwidth and computing resource requirements, we select the smaller one between $\tilde{n}_g^{\text{B}}$ and $\tilde{n}_g^{\text{C}}$ as the segment buffering number, i.e., $\tilde{n}_g = \min\left\{\tilde{n}_g^{\text{B}}, \tilde{n}_g^{\text{C}}\right\}$. For one MG, the segment buffering number in the resource requirement is the maximum value of all SMGs' 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 SMG g can be obtained by integrating segment buffering number and order, denoted by Ω_g.

C. DT-Assisted Buffer Update Scheme

As shown in Fig. 5, we utilize the virtualization technology of DT to construct multiple virtual buffers for SMG g, where each virtual buffer corresponds to the divided and subsequent SMG's buffer. The buffer index of SMG 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 SMGs [38], [6], we modify the buffer update function, as follows:

$$q_{g,t+1}^f = \begin{cases} \left[q_{g,t}^f - T_{\text{s}} + \tau\left|\Omega_g\right|\right]^+, & f = 0, \\ q_{g,t}^f + \tau\left|\Omega_{g+f}\right|, & \forall f \in \{1, \cdots G-g\}, \end{cases} \tag{7}$$

where $|\Omega_g|$ is the segment buffering number of SMG g.

When SMG 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 SMG 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, SMG g's buffer update can be further modified as

$$\widetilde{q}^{f}_{g,t+1} = \begin{cases} q^{f}_{g,t+1}, & \mathbb{S}_{g,t} = 0, \\ q^{f+1}_{g,t+1}, & \mathbb{S}_{g,t} = 1 \ \& \ f \in [0, G-g-1], \\ 0, & \mathbb{S}_{g,t} = 1 \ \& \ f = G-g. \end{cases} \tag{8}$$

V. Multicast QoE Model Establishment

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

$$\sum_{g=1}^{G} \beta_{g,t} \leq 1, \quad \forall t \in \mathcal{T}. \tag{9}$$

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

$$D_g = \frac{\sum_{l=1}^{L} \sum_{m \in \Omega_g} a^{l}_{g,m} \sum_{j=1}^{l} z^{j}_{g,m}}{\min\limits_{k \in \bigcup_{d=1}^{g} \mathcal{K}_d} \beta_g r_{g,k}}. \tag{10}$$

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

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

$$\sum\nolimits_{l=1}^{L} a^{l}_{g,m} = 1, \quad \forall m \in \Omega_g. \tag{11}$$

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

$$S_g = \max \left\{ D_g, \frac{\mu \sum_{l=1}^{L} \sum_{m \in \Omega_g} a^{l}_{g,m} \sum_{j=1}^{l} z^{j}_{g,m}}{\beta_g C} \right\}. \tag{12}$$

Based on SMG g's current actual buffer 0 and service delay, we refer to [6] to derive the rebuffering time, i.e.,

$$R_g = \left[S_g - \widetilde{q}^{0}_{g} \right]^{+}. \tag{13}$$

SSIM is a common metric used in video quality assessment. The relationship between video bitrate, b, and SSIM, Q, can be depicted as $Q = 1 - \frac{1}{2b+1}$ [38], 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 SMG g can be expressed as

$$Q_g = \sum_{m \in \Omega_g} 1 - \frac{1}{2 \sum_{l=1}^{L} a^{l}_{g,m} \sum_{j=1}^{l} z^{j}_{g,m} / \tau + 1}. \tag{14}$$

Based on video quality, we can analyze the video quality variation, V_g, between adjacent segments in SMG g, depicted by

$$V_g = \frac{1}{|\Omega_g|} \sum_{m=1}^{|\Omega_g|} |Q_{g,m} - Q_{g,m-1}|. \tag{15}$$

where $Q_{g,0}$ is the video quality of the last segment in SMG 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 [33], which can be expressed as

$$\Upsilon_g \left(\boldsymbol{a}_g, \beta_g \right) = Q_g - \lambda_{g,1} R_g - \lambda_{g,2} V_g, \tag{16}$$

where $\boldsymbol{a}_g$ is a vector whose unit element is $a^{l}_{g,m}$. Here, $\lambda_{g,1}$ and $\lambda_{g,2}$ represent users' sensitivity degrees of rebuffering time and quality variation in SMG g, respectively.

Considering each SMG 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

$$\omega_g = \frac{\sum_{m=1}^{\Omega_g} \phi_{g,m}}{\sum_{g=1}^{G} \sum_{m=1}^{\Omega_g} \phi_{g,m}}, \tag{17}$$

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

$$\widetilde{\Upsilon}_g = \omega_g \Upsilon_g. \tag{18}$$

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

VI. Problem Formulation and Solution

A. Problem Formulation

Since each SMG'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 SMG 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

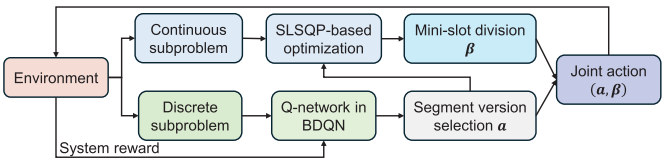


Fig. 6. The proposed data-model-driven algorithm structure.

video playback quality, users can obtain better QoE. Based on the above analysis, our objective is to maximize users' long-term QoE by optimizing segment version selection and slot division at each scheduling slot. The formulated problem **P1** is given by

$$\mathbf{P1}: \max_{\{\boldsymbol{a}_{g,t},\beta_{g,t}\}_{t\in\mathcal{T}}} \frac{1}{T}\sum_{t=1}^{T}\sum_{g\in\mathcal{G}}\widetilde{\Upsilon}_g\left(\boldsymbol{a}_{g,t},\beta_{g,t}\right) \tag{19}$$

$$\text{s.t.} \quad (9) \text{ and } (11),$$

$$a^l_{g,m,t}\in\{0,1\}, \forall g\in\mathcal{G}, m\in\Omega_g, l\in\mathcal{L}, t\in\mathcal{T}, \tag{19a}$$

$$\beta_{g,t}\in[0,1], \forall g\in\mathcal{G}, t\in\mathcal{T}. \tag{19b}$$

Constraint (9) is the resource capacity constraint, which guarantees that the total scheduled bandwidths and computing resources cannot exceed the system capacity. Constraint (11) is the video transmission constraint, which avoids the repeated video transmission for each SMG.

B. 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 users' long-term QoE. The dimension of $\boldsymbol{a}$ is $2^{L\sum_{g\in\mathcal{G}} g|\Omega_g|}$, which is very huge when the numbers of SMGs, recommended segments, and segment versions are high. The variable vector $\boldsymbol{\beta}$ are continuous values ranging from 0 to 1. Since these variables are coupled with each other, it is hard to directly use a data-driven or model-based method to solve it [39], [40], [41]. Therefore, we consider a data-model-driven algorithm to solve it, as shown in Fig. 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 $\boldsymbol{a}$. Then, the sequential least squares quadratic programming (SLSQP)-based convex optimization algorithm [42] utilizes environment information and determined decision $\boldsymbol{a}$ to solve the continuous optimization subproblem for obtaining mini-slot division $\boldsymbol{\beta}$. Next, the joint action $(\boldsymbol{a}, \boldsymbol{\beta})$ 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.

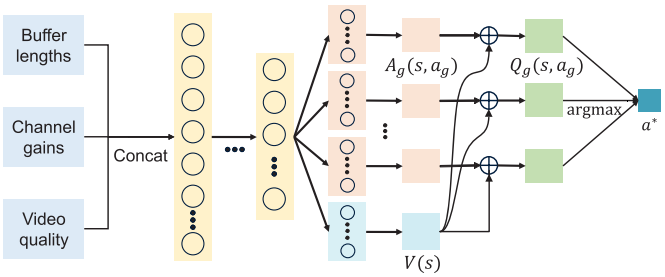


Fig. 7. The constructed BDQN architecture.

1) DRL-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. 7.

As shown in Fig. 7, SMGs' state consisting of buffer lengths, channel gains, and video quality, i.e., $s = \left\{\left\{\tilde{q}^0_g\right\}_{g\in\mathcal{G}}, \{h_{g,k}\}_{k\in\mathcal{K}_g, g\in\mathcal{G}}, \{Q_{g,0}\}_{g\in\mathcal{G}}\right\}$, are concatenated and input into the fully connected neural networks. The BDQN architecture is an amalgamation of Dueling Q-Networks [43] and a branching action mechanism, which comprises two key components, i.e., value function and advantage function. The value function $V(s)$ 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 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 SMG g is denoted by $\boldsymbol{a}_g = \left\{a^l_{g,m}\right\}_{m\in\Omega_g, l\in\mathcal{L}}$. Each action branch corresponds to a SMG. 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 SMG, 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 SMGs' QoE, i.e., $\sum_{g\in\mathcal{G}}\widetilde{\Upsilon}_g\left(\boldsymbol{a}_g,\beta_g\right)$. The advantage function of each sub-action, i.e., $A_g(s,\boldsymbol{a}_g)$ is trained with the common state value $V(s)$ by experience replay. The Q-value of each sub-action is updated based on the average advantage functions, which can be expressed as

$$\mathbb{Q}_g\left(s,\boldsymbol{a}_g\right) = V\left(s\right) + \left(A_g\left(s,a_g\right) - \frac{1}{\rho_g}\sum_{\boldsymbol{a}'_g\in A_g} A_g\left(s,\boldsymbol{a}'_g\right)\right), \tag{20}$$

where $\boldsymbol{a}'_g$ is the next-step sub-action for SMG g and ρ_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

$$\boldsymbol{a}=\left\{\arg\max_{\boldsymbol{a}_1'}\mathbb{Q}_1\left(s,\boldsymbol{a}_1';\boldsymbol{\theta}_1\right),\cdots,\arg\max_{\boldsymbol{a}_G'}\mathbb{Q}_G\left(s,\boldsymbol{a}_G';\boldsymbol{\theta}_G\right)\right\}, \tag{21}$$

where $\boldsymbol{\theta}_g$ 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

$$y=r+\vartheta\frac{1}{G}\sum_g\widehat{\mathbb{Q}}_g\left(s',\arg\max_{\boldsymbol{a}_g'}\mathbb{Q}_g\left(s',\boldsymbol{a}_g'\right)\right). \tag{22}$$

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.,

$$L\left(\boldsymbol{\theta}\right)=\mathrm{E}_{(s,\boldsymbol{a},r,s')\sim\mathcal{D}}\left[\frac{1}{G}\sum_g\left(y-\mathbb{Q}_g\left(s,\boldsymbol{a}_g\right)\right)^2\right]. \tag{23}$$

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.,

$$\delta_{\mathcal{D}}\left(s,\boldsymbol{a},r,s'\right)=\sum_g\left|y_g-\mathbb{Q}_g\left(s,\boldsymbol{a}_g\right)\right|. \tag{24}$$

Based on the above analysis, the detailed algorithm procedure to determine segment version selection variable $\boldsymbol{a}$ is shown in Algorithm 1.

2) Optimization-Based Slot Division: We first transform the maximization problem into the minimization problem. When $\boldsymbol{a}$ is determined, denoted by $\boldsymbol{a}^*$, the original objective function in the maximization problem can be transformed to the opposite objective function in the minimization problem, i.e., $\sum_{g\in\mathcal{G}}\widehat{\Upsilon}_g\left({\boldsymbol{a}_g}^*,\beta_g\right)=-\sum_{g\in\mathcal{G}}\widetilde{\Upsilon}_g\left({\boldsymbol{a}_g}^*,\beta_g\right)$.

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

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 by iteratively approximating them into quadratic subproblems [44]. 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 [45] to analyze its sub-gradient.

Algorithm 1 BDQN-Based Segment Version Selection

Input: SMGs' buffer lengths $\widetilde{\boldsymbol{q}}$, channel gains $\boldsymbol{h}$, video quality $\boldsymbol{Q}$, and network update threshold Γ;
Output: Segment version selection $\boldsymbol{a}$;
Initialize: Replay memory $\mathcal{D}$, action-value function $\mathbb{Q}$ with random weights $\boldsymbol{\theta}$, target action-value function $\widehat{\mathbb{Q}}$ with weights $\boldsymbol{\theta}'$;
for *each episode* **do**
  Reset initial state s_1;
  for *each step* $t\in\{1,\cdots,T\}$ **do**
    With probability ς select a random action $\boldsymbol{a}_t$, otherwise, the action is selected based on Eq. (21);
    Implement the Algorithm 2 to obtain slot division $\boldsymbol{\beta}_t$;
    Execute joint action $(\boldsymbol{a}_t,\boldsymbol{\beta}_t)$ in the environment;
    Observe reward r_t and new state s_{t+1};
    Store transition $(s_t,\boldsymbol{a}_t,r_t,s_{t+1})$ in $\mathcal{D}$;
    Prioritize replay based on Eq. (24) to obtain a transition (s_j,a_j,r_j,s_{j+1}) from $\mathcal{D}$;
    Calculate the TD target based on Eq. (22);
    Perform a gradient descent step based on the loss in Eq. (23);
    Every Γ steps, reset $\widehat{\mathbb{Q}}=\mathbb{Q}$;
  end
end

In iteration i, the sub-gradient of $\widehat{\Upsilon}_g\left({\boldsymbol{a}_g}^*,\beta_g^{(i)}\right)$ can be determined based on the value of $(x)^+$ function. Let denote $\widehat{\Upsilon}_g\left({\boldsymbol{a}_g}^*,\beta_g^{(i)}\right)=\frac{1}{\beta_g^{(i)}}\varphi_{g,1}-\varphi_{g,2}$, 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 $\varphi_{g,2}=\widetilde{q}_g^0(\boldsymbol{a}_g^*)$. Functions $\varphi_{g,1}$ and $\varphi_{g,2}$ are constants related with ${\boldsymbol{a}_g}^*$. 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 $\beta_g^{(i)}$:

$$\nabla\widehat{\Upsilon}(\beta_g^{(i)})=\begin{cases}-\dfrac{\varphi_{g,1}}{\left(\beta_g^{(i)}\right)^2}, & \beta_g^{(i)}<\dfrac{\varphi_{g,1}}{\varphi_{g,2}},\\ -\sigma\dfrac{\varphi_{g,1}}{\left(\beta_g^{(i)}\right)^2}, & \beta_g^{(i)}=\dfrac{\varphi_{g,1}}{\varphi_{g,2}},\\ 0, & \text{otherwise},\end{cases} \tag{25}$$

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

Based on the sub-gradient information, the Hessian matrix, $\mathbf{H}$, is expressed as $\mathtt{diag}\left(\nabla^2\widehat{\Upsilon}\left(\beta_1^{(i)}\right),\ldots,\nabla^2\widehat{\Upsilon}\left(\beta_G^{(i)}\right)\right)$. Since the quadratic subproblem is a fundamental step in the SLSQP algorithm, we formulate the quadratic subproblem as

$$\mathbf{P2}:\ \min_{\boldsymbol{\beta}^{(i)}}\widehat{\Upsilon}\left(\boldsymbol{a}^*,\boldsymbol{\beta}^{(i)}\right)+\nabla\widehat{\Upsilon}\left(\boldsymbol{a}^*,\boldsymbol{\beta}^{(i)}\right)^T\mathbf{d}+\frac{1}{2}\mathbf{d}^T\mathbf{H}\mathbf{d} \tag{26}$$

$$\text{s.t.}\quad c\left(\boldsymbol{\beta}^{(i)}\right)+\nabla c\left(\boldsymbol{\beta}^{(i)}\right)^T\mathbf{d}\le 0, \tag{26a}$$

$$\boldsymbol{\beta}^{(i)}+\mathbf{d}\in[0,1], \tag{26b}$$

Algorithm 2 SLSQP-Based Slot Division

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

where the objective function is the quadratic approximation around $(\boldsymbol{a}^*, \boldsymbol{\beta}^{(i)})$. Here, c and $\mathbf{d}$ represent the function of Eq. (9) and the direction of change, respectively.

To solve the formulated quadratic subproblem, we can utilize the QP solver in the CVXOPT.[1] The specific algorithm is shown in Algorithm 2.

C. Computing 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, 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 $O(\mathcal{E} \times T \times (3\mathcal{P} + M + n^3 + v \times M + \log \mathcal{D}))$.

[1]CVXOPT: https://cvxopt.org/examples/tutorial/qp.html

TABLE II
SIMULATION PARAMETERS

Parameter	Value	Parameter	Value
B	$[6, 14]$ MHz	K	$[10, 26]$
C	$[8, 12]$ Gcycles/s	μ	4 Gcycles/Mb
T_{s}	5 sec	λ_1	$[0.2, 0.4]$
τ	2 sec	λ_2	$[0.5, 0.7]$
P_{D}	27 dBm	N_0	-174 dBm

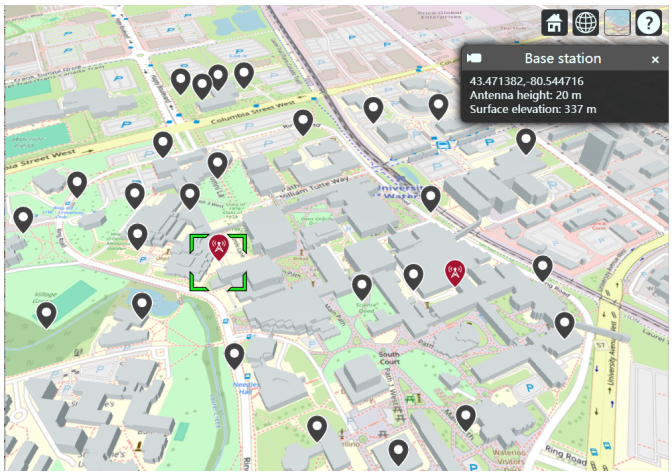


Fig. 8. The simulation scene, where BSs and users are represented by red and gray icons, respectively.

VII. Simulation Results

We conduct extensive simulations on the real-world dataset to evaluate the performance of the proposed DT-based network management scheme.

A. Simulation Setup

We adopt the short video streaming dataset[2] to obtain users' swipe behaviors. We sample 1000 short videos from the YouTube 8M dataset,[3] 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 users' initial positions are randomly and uniformly generated around two BSs, as shown in Fig. 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 II.

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

[2]ACM MM Grand Challenges: https://github.com/AItransCompetition/Short-Video-Streaming-Challenge/tree/main/data
[3]YouTube 8M dataset: https://research.google.com/youtube8m/index.html

TABLE III
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	$512\times256\times256\times128$	Activation function	ReLu
Advantage stream structure	128×36	Value stream structure	128×1

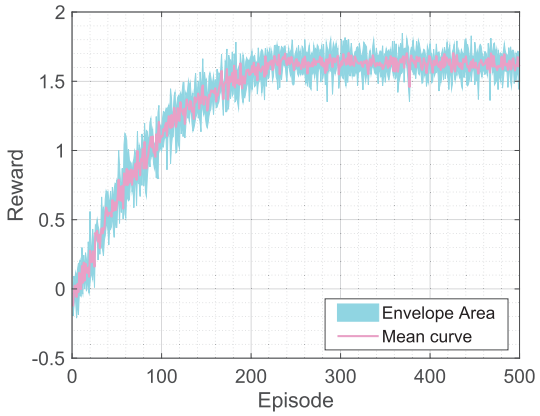


Fig. 9. Convergence performance of proposed DT-based network management scheme.

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 III.

The proposed DT 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 DT 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 DT data feature abstraction module, an improved user clustering method proposed in [3] is used to abstract the watching probability distribution of segments. In the DT 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 DT-based network management scheme with the following benchmark schemes:

- **Without DT (WDT) scheme**: The segment buffering is based on the sequential principle. The rebuffering time estimation is based on the service delay and the SMGs' currently total buffered segments. The determination of segment version selection and slot division employs the same data-model-driven method proposed in the DT-based network management scheme.
- **DT-BB scheme** [5]: The segment buffering and buffer update employ the same principle proposed in the DT-based network management scheme. The scheduling slot is discretized into 10 mini-slots. Each mini-slot is first provisionally allocated to each SMG, 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.
- **WDT-DRL scheme**: The segment buffering is based on the sequential principle. The rebuffering time estimation is based on the service delay and the SMGs' currently total buffered segments. The joint optimization of segment version selection and slot division is determined by a novel hierarchical reward-based DDPG algorithm [46], 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 MG's QoE, otherwise, a penalty will be given based on the violation degrees of constraints.

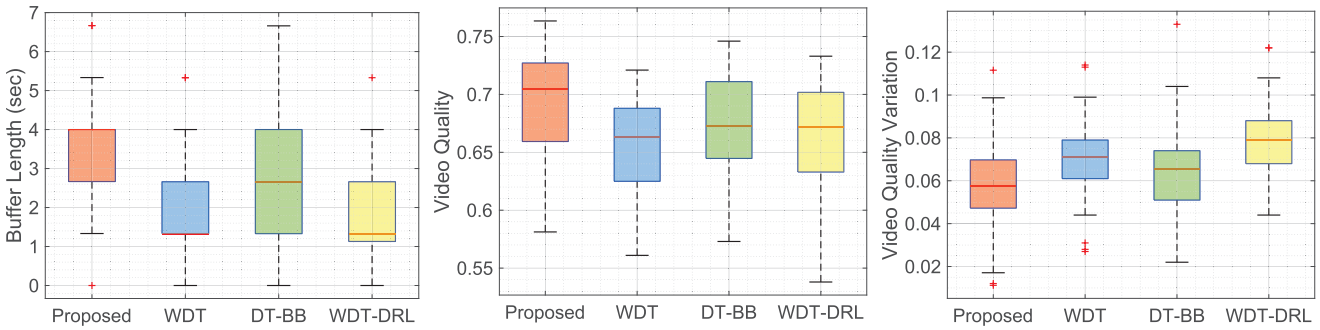
B. Convergence Analysis

In this subsection, we analyze the convergence performance of the proposed scheme within 500 training episodes. As shown in Fig. 9, we present the convergence curve of the DT-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 DT-based network management scheme can achieve a high and stable QoE for users.

C. Performance Evaluation of QoE Components

In this subsection, we evaluate the performance of the MG's QoE components. The bandwidth, computing capacity, and user number are set to 10 MHz, 10 Gcycles/s, and 10 respectively. We present the box plot comparison of buffer length, video quality, and video quality variation across four different schemes in Fig. 10. Each box plot delineates the interquartile range, median, and outliers.

As shown in Fig. 10(a), the proposed scheme demonstrates a higher median buffer length and a more compact interquartile range relative to the other schemes, which indicates users'



(a) Buffer length comparison (b) Video quality comparison (c) Video quality variation comparison

Fig. 10. The performance comparison of different QoE components.

TABLE IV
WATCHING PROBABILITY COMPARISON

Entertainment	1st Seg.	2nd Seg.	3th Seg.	4th Seg.	5th Seg.	6th Seg.	7th Seg.	8th Seg.	9th Seg.	10th Seg.	11st Seg.	12nd Seg.	13th Seg.	14th Seg.	15th Seg.	Accuracy
Actual Prob.	0.93	0.88	0.86	0.83	0.77	0.71	0.62	0.53	0.43	0.35	0.29	0.21	0.15	0.11	0.11	89.8%
Estimated Prob.	0.89	0.83	0.81	0.78	0.74	0.71	0.65	0.58	0.50	0.41	0.33	0.25	0.18	0.13	0.12	
Travel	1st Seg.	2nd Seg.	3th Seg.	4th Seg.	5th Seg.	6th Seg.	7th Seg.	8th Seg.	9th Seg.	10th Seg.	11st Seg.	12nd Seg.	13th Seg.	14th Seg.	15th Seg.	Accuracy
Actual Prob.	0.77	0.74	0.71	0.66	0.60	0.52	0.48	0.43	0.41	0.37	0.34	0.31	0.28	0.26	0.24	92.2%
Estimated Prob.	0.82	0.77	0.73	0.69	0.65	0.59	0.54	0.48	0.43	0.40	0.38	0.34	0.30	0.27	0.22	

equipment can buffer more segments to reduce the rebuffering probability. The compactness of the proposed scheme suggests lower variability in buffer length, which could bring a more stable user experience during video playback. The reason is attributed to the proposed DT-assisted segment buffering scheme, which can effectively abstract the watching probability distribution for priority-based buffering and maintain virtual buffers for accurate buffer length updates. Furthermore, since the DT-BB scheme employs the DT-based segment buffering scheme that can well adapt to the dynamics of users' swipe behaviors and network conditions to make the appropriate segment buffering number, it can have a better buffer length than the WDT-DRL scheme.

Fig. 10(b), presents a comparative analysis of video quality performance among different schemes with the range from 0.53 to 0.77. The proposed scheme reveals the highest median video quality, as well as a relatively narrower interquartile range compared to the other schemes. This suggests that the proposed scheme not only delivers a relatively higher video quality but also ensures smoother playback in the quality of streaming content. The reduced spread of data points and fewer outliers underscores its ability to provide a reliably high-quality watching experience, due to its advanced segment version selection algorithm that can achieve a good trade-off between buffering and streaming quality.

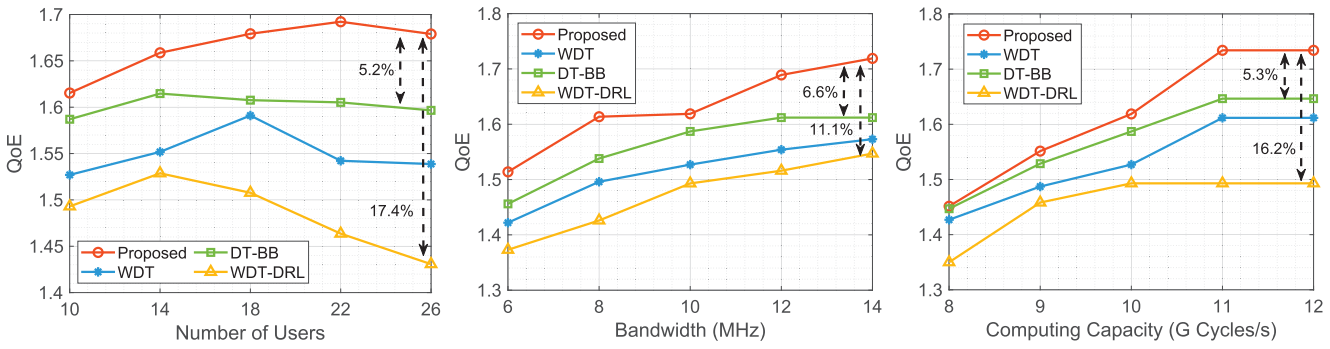
As shown in Fig. 10(c), we present the comparison of video quality variation. Based on the observation, the proposed scheme shows the lowest median value that suggests a central tendency towards lower variation in video quality. Despite the interquartile range of the proposed scheme being very close to the other schemes, the concentration of data around a lower median value and the reduced number of extreme outliers reflect its effectiveness in ensuring stable video quality. Furthermore, since the branch and bound-based scheduling scheme can obtain the near-optimal segment version selection and mini-slot division while deep deterministic policy gradient (DDPG)-based resource scheduling scheme scarifies a part of network performance by discretizing output actions, the DT-BB scheme can have a better video quality than the WDT-DRL scheme.

We also select two kinds of video sequences, i.e., Entertainment and Travel, to illustrate the gap between the actual and estimated watching probability distribution. As shown in Table IV, each video sequence has fifteen segments with a decreasing watching probability distribution. The absolute value error is used to measure the accuracy of estimated watching probability distributions. The accuracy of two kinds of video sequences can achieve up to 89.8% and 92.2%, respectively, which demonstrates the constructed Markovian-based watching probability model can well reflect users' realistic watching process.

D. Performance Evaluation of QoE Under Different Settings

We present QoE comparison under different users, bandwidths, and computing capacities among different schemes in Fig. 11. Overall, the proposed scheme can achieve superior performance under different simulation settings.

Fig. 11(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 users increases due to the mutual influence of multicast segment buffering. Specifically, the increased users can lead to more users being clustered into one SMG and enrich users' diversity. Initially, the same multicast segments watched by the original users may gain a better QoE for newly clustered users who



(a) QoE vs. different numbers of users (b) QoE vs. different bandwidths (c) QoE vs. different computing capacities

Fig. 11. QoE comparison under different users, bandwidths, and computing capacities.

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 DT-BB and WDT-DRL schemes, which demonstrates the effectiveness of the proposed scheme. The spikes and non-smoothness in Fig. 11(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 MGs. 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.

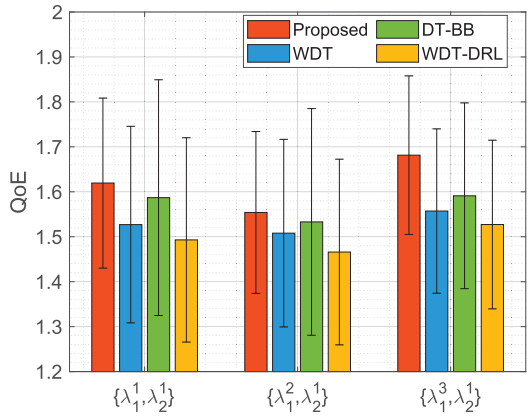
Fig. 11(b) presents the QoE variation with the increased bandwidth. The proposed scheme's upward tendency in QoE with increased bandwidth implies that it can well adapt to varying network capacities, which is essential for ensuring service quality during peak usage times or in bandwidth-constrained environments. When the bandwidth reaches 14 MHz, the proposed scheme achieves a QoE increment of 11.1% compared to the WDT-DRL scheme, suggesting that the proposed scheme can efficiently utilize available bandwidth to enhance the user experience.

Fig. 11(c) correlates the computing capacity with QoE. In MSVS, the computing capacity directly influences the transcoding version and the speed for segments to be delivered to SMGs. The proposed scheme's pronounced improvement in QoE with the increasing computing capacity reflects its ability to leverage additional computational resources 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 WDT-DRL method. This suggests that the proposed scheme can effectively harness computing power to enhance video quality for QoE improvement.

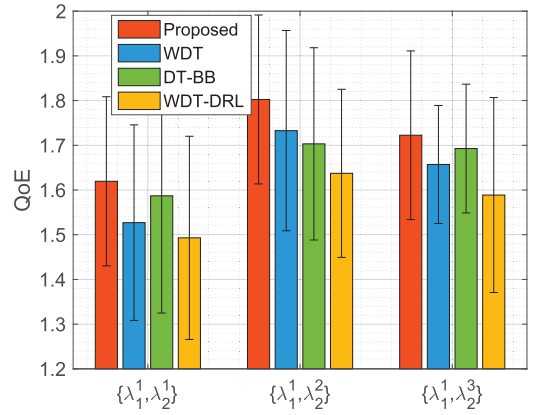
The spikes and non-smoothness in Fig. 11(b)(c) are mainly due to the constructed nonlinear multicast QoE model consisting of rebuffer time, video quality, and video quality variation. Specifically, these factors are related to segment version selection and mini-slot division, which exhibits a nonlinear relationship. Therefore, with the increasing bandwidth and computing resources, the multicast QoE will gradually increase but not with a smooth trend.

The reasons why the proposed method can obtain the best QoE performance compared to other baseline methods mainly consist of two aspects. Firstly, the proposed DT-assisted segment buffering scheme can adapt to users' swipe behaviors and be selectively buffered to users' buffers, thus effectively reducing the users' rebuffering time. Secondly, the proposed data-model-driven scheduling algorithm can utilize the respective advantages of reinforcement learning and convex optimization to efficiently handle the complex and dynamic joint optimization problem of segment version selection and mini-slot division. Therefore, the proposed method can achieve the best QoE performance under different users, bandwidths, and computing capacities.

Furthermore, we present the QoE comparison under different sensitivity degrees among different schemes in Fig. 12. As shown in Fig. 12(a), we adjust the SMGs's sensitivity degrees of video rebuffering time with the fixed sensitivity degrees of video quality variation. The parameter λ_1^1 includes three elements, i.e., $(0.4, 0.3, 0.2)$, which represents three SMGs' sensitivity degrees of rebuffering time, respectively. The parameter λ_2^1 also includes three elements, i.e., $(0.7, 0.6, 0.5)$. While the parameters λ_1^2 and λ_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 SMGs. It can be observed that our proposed scheme can always achieve the highest QoE with a comparatively tighter range of variance under different parameters λ_1. As shown in Fig. 12(b), we change the SMGs' sensitivity degrees of video quality variation with the fixed sensitivity degrees of rebuffering time. The parameters λ_2^2 and λ_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 SMGs 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



(a) QoE vs. different sensitivity degrees of rebuffering time



(b) QoE vs. different sensitivity degrees of video quality variation

Fig. 12. QoE comparison under different sensitivity degrees.

playback. Therefore, with a higher weighting factor, SMGs' QoE can be effectively enhanced.

VIII. Conclusion

In this paper, we have proposed a novel DT-based network management scheme to enhance QoE in MSVS. Furthermore, we have established a multicast QoE model to quantify the impact of multicast segment buffering among SMGs. A convex optimization embedded DRL algorithm has been designed to determine the joint segment version selection and slot division. The proposed DT-based network management scheme can efficiently multicast segment buffering in MSVS. For the future work, we will investigate the adaptive granularity of DT data collection and abstraction to reduce the network overhead of DT.

Appendix A
Proof of Theorem 1

When the variable $\boldsymbol{a}$ is determined, the opposite objective function in the minimization problem can be expressed in Eq. (27), shown at the bottom of the page.

Let denote

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

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. (27) can be transformed into

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

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 $\psi_1(\beta_g) = \left[\Xi_1(\beta_g) - \widetilde{q}_g^0\right]^+$ and $\psi_2(\beta_g) = \left[\Xi_2(\beta_g) - \widetilde{q}_g^0\right]^+$.

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

$$\begin{aligned}
&\psi_1(\theta\beta_{g,1} + (1-\theta)\beta_{g,2}) \\
&\quad = \max\{\Xi_1(\theta\beta_{g,1} + (1-\theta)\beta_{g,2}) - \widetilde{q}_g^0, 0\} \\
&\quad \leq \max\{\theta\Xi_1(\beta_{g,1}) + (1-\theta)\Xi_1(\beta_{g,2}) - \widetilde{q}_g^0, 0\} \\
&\quad \leq \theta \max\{\Xi_1(\beta_{g,1}) - \widetilde{q}_g^0, 0\} \\
&\qquad + (1-\theta)\max\{\Xi_1(\beta_{g,2}) - \widetilde{q}_g^0, 0\} \\
&\quad = \theta\psi_1(\beta_{g,1}) + (1-\theta)\psi_1(\beta_{g,2}).
\end{aligned} \tag{28}$$

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. (27) are constant, they do not affect the convexity of the transformed objective function. Therefore, function $\widehat{\Upsilon}_g(a_g{}^*, \beta_g)$ is convex. For functions $\widehat{\Upsilon}_g(a_g{}^*, \beta_g)$ with different g, they have an identical domain

$$\begin{aligned}
\widehat{\Upsilon}_g(a_g{}^*, \beta_g) = \omega_g &\left(\sum_{m\in\Omega_g} -1 + \frac{1}{2a_{g,m}^{l^*}\sum_{j=1}^{l^*} z_{g,m}^j/\tau + 1} + \frac{\lambda_{g,2}}{|\Omega_g|} \sum_{m\in\Omega_g} \left| Q_{g,m}^{l^*}\left(a_{g,m}^{l^*}\right) - Q_{g,m-1}^{l^*}\left(a_{g,m-1}^{l^*}\right) \right| \right) \\
&+ \omega_g\lambda_{g,1} \left[\max\left\{ \frac{\sum_{m\in\Omega_g} a_{g,m}^{l^*}\sum_{j=1}^{l^*} z_{g,m}^j}{\min\limits_{k\in\bigcup_{d=1}^{g}\mathcal{K}_d} \beta_g B\log_2\left(1+\frac{|h_{g,k}|^2 P_D}{N_0}\right)}, \frac{\mu\sum_{m\in\Omega_g} a_{g,m}^{l^*}\sum_{j=1}^{l^*} z_{g,m}^j}{\beta_g C} \right\} - \widetilde{q}_g^0 \right]^+ .
\end{aligned} \tag{27}$$

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.

REFERENCES

[1] Outloud Group. (2023). *Top Tiktok Statistics (2023): Key Facts, Figures, & Data*. [Online]. Available: https://www.outloudgroup.com/post/top-tik-tok-statistics-demographics

[2] X. Jiang, F. R. Yu, T. Song, and V. C. M. Leung, "A survey on multi-access edge computing applied to video streaming: Some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 871–903, 2nd Quart., 2021.

[3] X. Huang, W. Wu, S. Hu, M. Li, C. Zhou, and X. Shen, "Digital twin based user-centric resource management for multicast short video streaming," *IEEE J. Sel. Topics Signal Process.*, vol. 18, no. 1, pp. 50–65, Jan. 2024.

[4] J. Nightingale, P. Salva-Garcia, J. M. A. Calero, and Q. Wang, "5G-QoE: QoE modelling for ultra-HD video streaming in 5G networks," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 621–634, Jun. 2018.

[5] X. Huang, L. He, L. Wang, and F. 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.*, vol. 70, no. 10, pp. 10909–10924, Oct. 2021.

[6] W. Huang, Y. Zhou, X. Xie, D. Wu, M. Chen, and E. Ngai, "Buffer state is enough: Simplifying the design of QoE-aware HTTP adaptive video streaming," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 590–601, Jun. 2018.

[7] S. Wang, S. Bi, and Y. A. Zhang, "Adaptive wireless video streaming: Joint transcoding and transmission resource allocation," *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 3208–3221, May 2022.

[8] M. Grieves, "Digital twin: Manufacturing excellence through virtual factory replication," *White paper*, vol. 1, pp. 1–7, Jun. 2014.

[9] X. Huang, H. Yang, S. Hu, and X. Shen, "Digital twin-driven network architecture for video streaming," *IEEE Netw.*, early access, Apr. 8, 2024, doi: 10.1109/MNET.2024.3386030.

[10] Q. Guo, F. Tang, T. K. Rodrigues, and N. Kato, "Five disruptive technologies in 6G to support digital twin networks," *IEEE Wireless Commun.*, early access, Jan. 9, 2023, doi: 10.1109/MWC.013.2200296.

[11] Y. Zhang, Y. Liu, L. Guo, and J. Y. B. Lee, "Measurement of a large-scale short-video service over mobile and wireless networks," *IEEE Trans. Mobile Comput.*, vol. 22, no. 6, pp. 3472–3488, Jun. 2023.

[12] G. Zhang, K. Liu, H. Hu, and J. Guo, "Short video streaming with data wastage awareness," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2021, pp. 1–6.

[13] G. Zhang et al., "DUASVS: A mobile data saving strategy in short-form video streaming," *IEEE Trans. Services Comput.*, vol. 16, no. 2, pp. 1066–1078, Mar. 2023.

[14] X. Wu, L. Zhang, and L. Cui, "QoE-aware download control and bitrate adaptation for short video streaming," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 7115–7119.

[15] J. Guo and G. Zhang, "A video-quality driven strategy in short video streaming," in *Proc. 24th Int. ACM Conf. Model., Anal. Simul. Wireless Mobile Syst.*, Nov. 2021, pp. 221–228.

[16] M. Taha, A. Canovas, J. Lloret, and A. Ali, "A QoE adaptive management system for high definition video streaming over wireless networks," *Telecommun. Syst.*, vol. 77, no. 1, pp. 63–81, May 2021.

[17] H. Li, B. Li, T. T. Tran, and D. C. Sicker, "Transmission schemes for multicasting hard deadline constrained prioritized data in wireless multimedia streaming," *IEEE Trans. Wireless Commun.*, vol. 15, no. 3, pp. 1631–1641, Mar. 2016.

[18] L. Zhong et al., "A multi-user cost-efficient crowd-assisted VR content delivery solution in 5G-and-beyond heterogeneous networks," *IEEE Trans. Mobile Comput.*, vol. 22, no. 8, pp. 4405–4421, Sep. 2022.

[19] S. ul Zuhra, P. Chaporkar, A. Karandikar, and P. Jha, "Leveraging multi-connectivity for multicast video streaming," 2022, *arXiv:2202.05053*.

[20] A. Daher, M. Coupechoux, P. Godlewski, P. Ngouat, and P. Minot, "A dynamic clustering algorithm for multi-point transmissions in mission-critical communications," *IEEE Trans. Wireless Commun.*, vol. 19, no. 7, pp. 4934–4946, Jul. 2020.

[21] M. Zhang, H. Lu, and P. Hong, "Cooperative robust video multicast in integrated terrestrial-satellite networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 8276–8291, Oct. 2022.

[22] M. Zhang, H. Lu, F. Wu, and C. W. Chen, "NOMA-based scalable video multicast in mobile networks with statistical channels," *IEEE Trans. Mobile Comput.*, vol. 20, no. 6, pp. 2238–2253, Jun. 2021.

[23] E. Glaessgen and D. Stargel, "The digital twin paradigm for future NASA and U.S. Air force vehicles," in *Proc. 53rd AIAA/ASME/ASCE/AHS/ASC Struct., Structural Dyn. Mater. Conf. 20th AIAA/ASME/AHS Adapt. Struct. Conf.,14th AIAA*, Apr. 2012, p. 1818.

[24] X. Shen, J. Gao, W. Wu, M. Li, C. Zhou, and W. Zhuang, "Holistic network virtualization and pervasive network intelligence for 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 1–30, 1st Quart., 2022.

[25] T. Q. Duong, D. Van Huynh, S. R. Khosravirad, V. Sharma, O. A. Dobre, and H. Shin, "From digital twin to metaverse: The role of 6G ultra-reliable and low-latency communications with multi-tier computing," *IEEE Wireless Commun.*, vol. 30, no. 3, pp. 140–146, Jun. 2023.

[26] A. Masaracchia, V. Sharma, B. Canberk, O. A. Dobre, and T. Q. Duong, "Digital twin for 6G: Taxonomy, research challenges, and the road ahead," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 2137–2150, 2022.

[27] P. Bellavista, C. Giannelli, M. Mamei, M. Mendula, and M. Picone, "Application-driven network-aware digital twin management in industrial edge environments," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7791–7801, Nov. 2021.

[28] L. Qi, X. Xu, X. Wu, Q. Ni, Y. Yuan, and X. Zhang, "Digital-Twin-Enabled 6G mobile network video streaming using mobile crowdsourcing," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 10, pp. 3161–3174, Oct. 2023.

[29] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, "Deep learning for hybrid 5G services in mobile edge computing systems: Learn from a digital twin," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4692–4707, Oct. 2019.

[30] X. Huang, W. Wu, and X. S. Shen, "Digital twin-assisted resource demand prediction for multicast short video streaming," in *Proc. IEEE 43rd Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2023, pp. 967–968.

[31] Q. Guo, F. Tang, and N. Kato, "Resource allocation for aerial assisted digital twin edge mobile network," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 10, pp. 3070–3079, Oct. 2023.

[32] S. R. Jeremiah, L. T. Yang, and J. H. Park, "Digital twin-assisted resource allocation framework based on edge collaboration for vehicular edge computing," *Future Gener. Comput. Syst.*, vol. 150, pp. 243–254, Jan. 2024.

[33] C. Zhou, Y. Ban, Y. Zhao, L. Guo, and B. Yu, "PDAS: Probability-driven adaptive streaming for short video," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 7021–7025.

[34] Z. Li, Y. Xie, R. Netravali, and K. Jamieson, "Dashlet: Taming swipe uncertainty for robust short video streaming," in *Proc. USENIX Symp. Networked Syst. Design Implement. (NSDI)*, 2023, pp. 1583–1599.

[35] M. Wien, H. Schwarz, and T. Oelbaum, "Performance analysis of SVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1194–1203, Sep. 2007.

[36] Y. Mao, B. Clerckx, and V. O. K. Li, "Rate-splitting for multi-antenna non-orthogonal unicast and multicast transmission: Spectral and energy efficiency analysis," *IEEE Trans. Commun.*, vol. 67, no. 12, pp. 8754–8770, Dec. 2019.

[37] A. Z. Yalcin, M. Yuksel, and B. Clerckx, "Rate splitting for multi-group multicasting with a common message," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12281–12285, Oct. 2020.

[38] G. Huang, W. Gong, B. Zhang, C. Li, and C. Li, "An online buffer-aware resource allocation algorithm for multiuser mobile video streaming," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3357–3369, Mar. 2020.

[39] B. He et al., "ShuttleBus: Dense packet assembling with QUIC stream multiplexing for massive IoT," *IEEE Trans. Mobile Comput.*, early access, Dec. 25, 2024, doi: 10.1109/TMC.2023.3345898.

[40] W. Jiang, B. Ai, M. Li, W. Wu, and X. Shen, "Average age of information minimization in aerial IRS-assisted data delivery," *IEEE Internet Things J.*, vol. 10, no. 17, pp. 15133–15146, Oct. 2023.

[41] J. Xue, K. Yu, T. Zhang, H. Zhou, L. Zhao, and X. Shen, "Cooperative deep reinforcement learning enabled power allocation for packet duplication URLLC in multi-connectivity vehicular networks," *IEEE Trans. Mobile Comput.*, early access, Jan. 3, 2024, doi: 10.1109/TMC.2023.3347580.

[42] A. Tavakoli, F. Pardo, and P. Kormushev, "Action branching architectures for deep reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, 2018, pp. 4131–4138.

[43] Z. Wang, N. D. Freitas, and M. Lanctot, "Dueling network architectures for deep reinforcement learning," in *Proc. Int. Conf. Int. Conf. Mach. Learn.*, vol. 48, Jun. 2016, pp. 1995–2003.
[44] D. Kraft. (1988). *A Software Package for Sequential Quadratic Programming*. [Online]. Available: http://degenerateconic.com/uploads/2018/03/DFVLR_FB_88_28.pdf
[45] S. Boyd, L. Xiao, and A. Mutapcic, *Subgradient Methods*. Stanford, CA, USA: Stanford Univ. Press, 2003.
[46] T. Zhao, F. Li, and L. He, "DRL-based joint resource allocation and device orchestration for hierarchical federated learning in NOMA-enabled industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 19, no. 6, pp. 7468–7479, Nov. 2022.

Xinyu Huang (Graduate Student Member, IEEE) received the B.E. degree in Qian Xuesen experimental class from Xidian University, Xi'an, China, and the M.S. degree in information and communications engineering from Xi'an Jiaotong University, Xi'an, in 2018 and 2021, respectively. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Waterloo, Waterloo, ON, Canada. His research interests include digital twins, multimedia communication, and network resource management.

Shisheng Hu (Graduate Student Member, IEEE) received the B.Eng. and M.A.Sc. degrees from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2018 and 2021, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research interests include AI for wireless networks and networking for AI.

Haojun Yang (Member, IEEE) received the B.S. degree in communication engineering and the Ph.D. degree in information and communication engineering from Beijing University of Posts and Telecommunications (BUPT), Beijing, China. He is currently a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada. His research interests include ultra-reliable and low-latency communications, radio resource management, and vehicular networks.

Xinghan Wang received the Ph.D. degree from the School of Cyber Science and Engineering, Southeast University, Nanjing, China. He is currently a Post-Doctoral Fellow with the Peng Cheng Laboratory, Shenzhen, China. His research interests include network protocol design and analysis, reinforcement learning, and edge computing.

Yingying Pei (Graduate Student Member, IEEE) received the B.E. degree from Huazhong University of Science and Technology, Wuhan, China, in 2014, and the M.Sc. degree from the University of Macau, Macau, China, in 2019. She is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. Her current research interests include network resource management for extended reality services.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990.

He is currently an University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on network resource management, wireless network security, the Internet of Things, AI for networks, and vehicular networks. He is a Registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Member, an International Fellow of the Engineering Academy of Japan, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society. He received "West Lake Friendship Award" from Zhejiang Province in 2023, President's Excellence in Research from the University of Waterloo in 2022, Canadian Award for Telecommunications Research from Canadian Society of Information Theory (CSIT) in 2021, the R. A. Fessenden Award in 2019 from IEEE, Canada, Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) in 2019, James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society (ComSoc), and Technical Recognition Award from Wireless Communications Technical Committee (2019) and AHSN Technical Committee (2013). He has also received the Excellent Graduate Supervision Award in 2006 from the University of Waterloo and the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada. He is the Past President of the IEEE Communications Society. He was the Vice President of Technical & Educational Activities, the Vice President of Publications, a Member-at-Large on the Board of Governors, the Chair of the Distinguished Lecturer Selection Committee, and a member of IEEE Fellow Selection Committee of the ComSoc. He served as the Editor-in-Chief for the IEEE INTERNET OF THINGS JOURNAL, *IEEE Network*, and *IET Communications*.