QoE-Aware Volumetric Video Caching and Rendering for Mobile Extended Reality Services

Yingying Pei, *Graduate Student Member, IEEE*, Mushu Li, *Member, IEEE*, Xinyu Huang, *Graduate Student Member, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—In this article, we propose a novel volumetric video caching and rendering approach for an edge-assisted extended reality (XR) system to enhance user Quality of Experience (QoE). Particularly, user QoE consists of visual quality and quality variation. Different quality of volumetric videos are required to be cached, rendered, and delivered to XR devices for different viewing distances within a time latency. Given the limited caching, computing, and communication resources on the edge server, we formulate a long-term user QoE maximization problem to jointly optimize video caching and rendering by considering user locations and viewing distances. To solve this problem, we first design an online optimization algorithm in which caching decisions are obtained using a regularization technique. We then develop a low-complexity binary search algorithm to determine optimal rendering quality. Extensive simulations are conducted to demonstrate that our proposed approach outperforms benchmark schemes by an average 46% improvement in terms of long-term user QoE.**

***Index Terms*—Binary search, extended reality (XR), Quality of Experience (QoE), regularization technique, volumetric video streaming.**

I. INTRODUCTION

EXTENDED reality (XR) serves as the technological foundation for immersive communications, enabling the creation of high-fidelity and interactive environments for mobile users [1], [2]. The immersive XR experience is achieved through the playback of volumetric videos on XR devices. When watching volumetric videos, users can freely explore the video content in three-dimensional (3-D) space, including both rotational movement (yaw, pitch, and roll) and translational movement (X, Y, and Z) [3]. Volumetric videos are composed of point clouds. A point cloud is a set of data points in 3-D space, and each point contains spatial coordinates and color attributes.

Depending on the number of points that compose an object, point clouds representing the same object can have multiple versions with different densities. When an XR user changes its viewpoint, point clouds are rendered to generate rendered videos, which are then displayed on the user's device. Rendering dense point clouds can provide high-quality videos, but this requires considerable computational resources for rendering and a large buffer size for storing dense point clouds [4]. Standalone XR devices, such as augmented reality (AR) glasses, are usually lightweight and equipped with limited storage and computing resources. Therefore, achieving high-quality and low-latency local rendering on XR devices is challenging, which potentially hinders XR users from obtaining a satisfactory Quality of Experience (QoE).

To enhance XR user QoE, mobile-edge computing (MEC) technologies can be adapted to efficiently utilize storage and computing resources at access points (APs) near XR users [5]. Edge servers located in these APs can store high-density point clouds and perform high-quality rendering with reduced latency, thus improving XR user QoE. However, QoE models designed for conventional 2-D videos are ineffective in estimating XR user QoE. In conventional video delivery, user QoE is determined by factors such as resolution, startup delay, and rebuffering time [6], [7]. However, when viewing volumetric videos, user QoE is influenced not only by the above conventional factors but also by the distance between the users' viewpoint and objects in the videos, referred to as viewing distance [8]. With a longer viewing distance, objects appear smaller in the displayed view, making XR users less sensitive to video quality [9]. Therefore, even when viewing the same video at the same quality, XR users with different viewing distances may have different QoE.

Due to the dependency between QoE and user viewing dynamics, the following challenges should be addressed. *First*, user QoE needs to be accurately estimated. However, measuring the impact of users' viewing distances on QoE is technically challenging. *Second*, since different users have diverse and dynamic viewing distances, the density of cached point clouds and the rendering quality from the edge server side should be dynamically adjusted to enhance the overall QoE for all users. In addition, decisions made about caching and rendering at one time instant may affect the QoE performance at subsequent time instants, while users' dynamic viewing status (including location and viewing distance) cannot be accurately obtained in advance. *Third*, caching and rendering decisions are highly coupled. Only precached point clouds can start rendering instantaneously. This coupling presents a challenge, which necessitates a joint decision-making strategy for caching and rendering.

Received 2 December 2024; revised 19 February 2025; accepted 4 March 2025. Date of publication 14 March 2025; date of current version 9 June 2025. This article was presented in part at the IEEE/CIC International Conference on Communications in China, Hangzhou, China, 2024 [4] [DOI: 10.1109/ICCC62479.2024.10681793]. *(Corresponding author: Xinyu Huang.)*

Yingying Pei, Xinyu Huang, and Xuemin Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: y32pei@uwaterloo.ca; x357huan@uwaterloo.ca; sshen@uwaterloo.ca).

Mushu Li is with the Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA 18015 USA (e-mail: mul224@lehigh.edu).

Digital Object Identifier 10.1109/JIOT.2025.3551237

2327-4662 © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Waterloo. Downloaded on June 16,2025 at 15:26:53 UTC from IEEE Xplore. Restrictions apply.

TABLE I
COMPARISON OF RELATED STUDIES

Reference	Video Type	Research Focus	QoE-Driven	Viewing Distance Awareness	Edge Resource Optimization
[11]	2D Video	ABR-assisted video caching and delivery	×	×	✓
[6]	360-degree Video	Tile-based edge caching and delivery	×	×	✓
[20]	Volumetric Video	User behavior-aware tile compression and streaming	✓	×	×
[9]	Volumetric Video	Visibility-aware tile compression and streaming	×	✓	×
[8]	Volumetric Video	SR-enhanced video streaming	✓	✓	×
Our work	Volumetric Video	QoE-driven edge caching and rendering	✓	✓	✓

In this article, we propose a viewing distance-aware caching and rendering approach for edge-assisted volumetric video streaming in XR. We first adopt a viewing distance-aware QoE model to accurately estimate user QoE. Based on users' dynamic viewing status, the edge server dynamically selects point cloud versions for caching to reduce the overall video delivery latency. In addition, to fulfill real-time rendering requests from each user, the edge server selects an appropriate cached version for rendering. Our objective is to find the optimal caching and rendering decisions that maximize the long-term QoE of all users while satisfying communication, storage, and computing resource constraints. Our main contributions are summarized as follows.

1) A novel edge caching and rendering approach for volumetric video-based XR services is proposed. By incorporating user viewing distances, users' video quality requirements are accurately estimated, thus enabling enhanced resource utilization and improved user QoE.
2) A joint volumetric video caching and rendering problem is studied to maximize the long-term QoE of all XR users. Given the uncertainty of future network conditions and users' viewing dynamics, an online optimization algorithm is designed to make efficient caching decisions.
3) Given caching decisions, closed-form derivations are performed based on Karush–Kuhn–Tucker (KKT) conditions to determine the optimal rendering decisions. In particular, the optimal rendering decision is obtained through a low-complexity binary search-based algorithm.

The remainder of this article is organized as follows. Section II provides an overview of the related work. Section III describes the system model and problem formulation. An online optimization scheme is discussed in Section IV. Simulation results are presented in Section V to demonstrate the performance of the proposed scheme. Section VI concludes the research work. Table I provides a comparison between our work and the most related studies. Table II provides a list of important notations.

TABLE II
SUMMARY OF IMPORTANT NOTATIONS

Notation	Definition
$\mathcal{M}, \mathcal{I}, \mathcal{Q}, \mathcal{T}$	Sets of volumetric videos, XR users, density versions, and time slots, respectively.
$x_{m,q,t}$	Caching decision for version q of video m at slot t.
$y_{i,m,q,t}$	Rendering decision for user i at slot t.
$L_{i,t}$	Rendering quality for user i at slot t.
$V_{i,t}(\Delta V_{i,t})$	Visual quality (quality variation) for user i at slot t.
$d_{i,t}$	Viewing distance of user i at slot t.
$\alpha(\beta)$	Weighting parameter for visual quality (quality variation) in QoE.
$q_{i,t}$	Instantaneous QoE of user i at slot t.
$\tau^{\mathrm{C}}(\tau^{\mathrm{B}})$	Latency requirement for volumetric video rendering (rendered video transmission).
$S_{\max}(C_{\max}, B_{\max})$	Maximum storage (computing, bandwidth) resources available at the edge server.
$R^{\mathrm{C}}_{i,t}$ $(R^{\mathrm{B}}_{i,t})$	Computing (bandwidth) resources consumed for user i at slot t.
σ_q	Storage resources required to store a video of version q.
ρ	Computation intensity for rendering tasks.
χ_q	Data size of the video segment generated by rending a video of version q.
$r_{i,t}$	Physical distance between user i and the AP at slot t.
$E_{i,t}$	Spectrum efficiency for user i's device at slot t.

II. RELATED WORK

In this section, we first review existing streaming techniques for conventional 2-D videos and 360-degree videos, followed by state-of-the-art works in volumetric video streaming.

A. Conventional Video Streaming

To adapt to highly dynamic network environments, the main 2-D video delivery technique is adaptive bitrate (ABR) streaming [10], [11], [12]. ABR splits videos into multiple chunks, each of which is then encoded at a certain quality level to adapt to fluctuating network bandwidth or user demands. For 2-D videos, the user QoE mainly depends on the video quality, with key impact factors including startup delay, rebuffering delay, video resolution, and quality variations. Most existing ABR algorithms maximize QoE by balancing these factors according to user preferences or network conditions.

Unlike 2-D videos, which offer a fixed perspective, 360-degree videos capture every direction simultaneously using an omnidirectional camera or a collection of cameras, providing a panoramic view of the entire surrounding area [13]. This format supports user movements within three degrees of freedom (3-DoF), i.e., yaw, pitch, and roll, and allows users to turn their heads in any direction within the video sphere. Given dynamic network conditions, the transmission of users' Field of View (FoV) is prioritized, which is supported by tiling. Specifically, the entire video is divided into tiles both spatially and temporally, and different tiles can be encoded at different quality levels [14]. Video tiling enables selective streaming of

Authorized licensed use limited to: University of Waterloo. Downloaded on June 16,2025 at 15:26:53 UTC from IEEE Xplore. Restrictions apply.

high-quality tiles within the user's FoV and low-quality tiles (or no streaming) outside the FoV, depending on the user's viewing perspective and bandwidth limitations.

B. Volumetric Video Streaming

Compared to 360-degree videos, volumetric videos provide more immersive viewing experiences by supporting three extra DoF movements (up/down, left/right, and forward/backward), i.e., 6-DoF movements. Currently, the most popular data format for volumetric video is the point cloud. The complex structure of point clouds and their large data sizes require substantial computation and storage resources for effective caching and rendering. Moreover, users' constantly changing viewing perspectives require continuous rendering and the transmission of rendered videos, which further poses challenges. The main approaches to addressing these challenges include point cloud compression to reduce data sizes, point cloud tiling to manage content delivery, and 6-DoF viewpoint prediction to enhance viewer interaction.

Point cloud compression can significantly reduce the data volume in volumetric videos through high compression ratio levels. Depending on available bandwidth and the user's viewpoint, point cloud segments of appropriate quality (bit rate or density) can be dynamically compressed and transmitted to the user's device [15], [16]. Recently, artificial intelligence (AI) empowered techniques, such as super-resolution (SR), can work as a complement to video compression and transmission [17], [18]. SR techniques utilize powerful models, such as pretrained deep neural network models, to generate high-resolution video streams from compressed low-resolution ones. For example, the work [18] integrates SR with tile-based streaming, where video tile selection and tile upgradation are jointly optimized to maximize user QoE, while considering dynamic bandwidth and limited computing capabilities.

Similar to 360-degree video tiling, 3-D tiling refers to dividing the point cloud into small tiles in 3-D space [19]. To reduce bandwidth consumption, only tiles within the user's FoV are encoded at different quality levels and then transmitted to users. For example, Li et al. [20] proposed a 3-D visual saliency-based tiling scheme that matches streamed content with regions of high visual interest for users, with these regions identified using saliency mapping and user motion estimation. To further reduce redundant transmissions of volumetric videos, proactive video streaming strategies, such as viewpoint prediction-based methods, were proposed [21], [22]. These methods predicted users' future viewing areas to improve streaming efficiency. To address prediction uncertainties, Liu et al. [22] proposed a multiview approach that generates multiple candidate views of potential future viewpoints using 6-DoF prediction models. These candidate views are generated and delivered to the XR device in real time. The XR device then selects the best candidate view that maximizes the QoE for video display.

In summary, the above works all focused on efficient tiling, streaming, and rendering of volumetric videos while considering the 3-D features of point clouds. However, from users' point of view, their QoE is influenced by both video

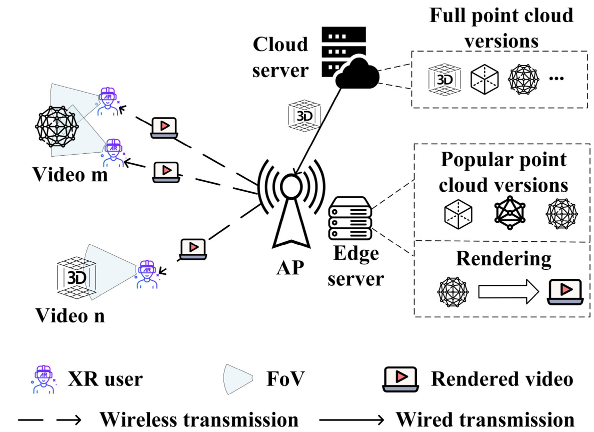

Fig. 1. MEC-assisted XR system.

quality and viewing distances [8], [9]. This feature motivates us to reconsider a QoE-driven and viewing distance-aware approach for volumetric video caching and rendering, aiming to maximize users' overall QoE given limited edge resources.

III. System Model and Problem Formulation

A. Network Scenario

As shown in Fig. 1, we consider an MEC-assisted XR system consisting of one AP associated with an edge server, multiple XR users, and multiple virtual objects. The XR service is deployed within the coverage area of the AP. Within the service area, virtual objects are presented to XR users in the form of volumetric videos, with indexes represented by set $\mathcal{M} = \{1, 2, \ldots, M\}$. Each frame of a volumetric video is represented by point clouds and is available in multiple-density versions. The set of versions for each video is denoted by $\mathcal{Q} = \{1, 2, \ldots, Q\}$. The cloud server stores all versions of all volumetric videos. XR users, denoted by set $\mathcal{I} = \{1, 2, \ldots, I\}$, move freely within the service area and view videos through XR devices (e.g., AR glasses). When a user changes its pose (including location and orientation), the scene displayed on the XR device is updated accordingly by rendering the corresponding point cloud from the user's current viewing perspective.

The system operates in a time-slotted manner with each slot duration δ. Let $\mathcal{T} = \{1, 2, \ldots, T\}$ denote the set of time slots. During each slot, each XR device continuously detects the user's pose by using live video analytic schemes [23], [24]. If a pose change is detected, a rendering task is generated and sent to the edge server. The edge server then identifies which virtual objects are within the user's current FoV based on the user's pose and the spatial distribution of all virtual objects. To ensure low-latency rendering, the edge server downloads and caches selected point cloud versions from the cloud server.[1] For each rendering task, the edge server selects a cached version to perform rendering and generates a video segment. Finally, the

[1]The wired backbone network (e.g., fiber-optic links) is assumed to provide sufficient and stable bandwidth to guarantee seamless point cloud downloading [25]. Therefore, the wired transmission latency is negligible compared to edge computing and wireless downlink latency.

Authorized licensed use limited to: University of Waterloo. Downloaded on June 16,2025 at 15:26:53 UTC from IEEE Xplore. Restrictions apply.

rendered video segment is transmitted back to the XR device for display.

Let the binary variable $x_{m,q,t}$ indicate whether version q of video m at slot t is cached at the edge server or not. Here, $x_{m,q,t} = 1$ indicates that the video version is cached; otherwise, we set $x_{m,q,t} = 0$. The total storage resource consumption at the edge server during slot t, denoted by S_t, is calculated as

$$S_t = \sum_{m \in \mathcal{M}} \sum_{q \in \mathcal{Q}} x_{m,q,t} \sigma_q \quad \forall t \in \mathcal{T} \tag{1}$$

where σ_q represents the storage resource consumption, measured in bits, required to store a video with a time duration of δ at version q. For simplicity, we assume that all videos of the same density have identical data sizes. The total storage resource consumption at any time slot cannot exceed the maximum storage capacity, which is expressed as

$$S_t \leq S_{\max} \quad \forall t \in \mathcal{T}. \tag{2}$$

Let the binary variable $y_{i,m,q,t}$ represent the rendering decision for user i at slot t. If version q of video m is rendered for user i at slot t, we set $y_{i,m,q,t} = 1$; otherwise, we set $y_{i,m,q,t} = 0$.

For a feasible rendering decision, two constraints must be satisfied. First, the video version selected for rendering for any user i must be cached at the current time slot, which can be expressed as

$$y_{i,m,q,t} \leq x_{m,q,t} \quad \forall m \in \mathcal{M} \ \forall q \in \mathcal{Q} \ \forall i \in \mathcal{I} \ \forall t \in \mathcal{T}. \tag{3}$$

Second, for any user i, at most one version of any video can be selected for rendering, which is expressed as

$$\sum_{m \in \mathcal{M}} \sum_{q \in \mathcal{Q}} y_{i,m,q,t} \leq 1 \quad \forall i \in \mathcal{I} \ \forall t \in \mathcal{T}. \tag{4}$$

B. QoE Model

We adopt a viewing distance-aware QoE model to estimate the QoE for volumetric video streaming.[2] For the QoE impact factors, we consider visual quality and quality variations.

1) Visual Quality: The visual quality of a volumetric video perceived by a user depends on both the quality displayed on the device (i.e., rendering quality) and the user's viewing distance. The rendering quality for user i at slot t, denoted by $L_{i,t}$, can be expressed as

$$L_{i,t} = \sum_{q \in \mathcal{Q}} q \sum_{m \in \mathcal{M}} y_{i,m,q,t}. \tag{5}$$

For user i and video m, the viewing distance at slot t is given by the following equation:

$$d_{i,t} = \sqrt{|a_{i,t} - a_m|^2 + |b_{i,t} - b_m|^2} \tag{6}$$

where $(a_{i,t}, b_{i,t})$ represents the coordinates of user i at slot t, and (a_m, b_m) are the coordinates of video m. At slot t, if user i is viewing video $m^\star (m^\star \in \mathcal{M})$, then for any $m' (m' \in \mathcal{M}/\{m^\star\})$, we have $y_{i,m',q,t} = 0$.

Let $V_{i,t}$ denote the visual quality of user i at slot t, defined by

$$V_{i,t} = \alpha\big(d_{i,t}\big) \cdot L_{i,t}. \tag{7}$$

In (7), the function $\alpha(d_{i,t})$ quantifies users' sensitivity to visual quality based on viewing distance, which can be evaluated as a weight parameterized by $d_{i,t}$. The function $\alpha(d_{i,t})$ is a decreasing function with respect to $d_{i,t}$, ranging from [0, 1]. This is because, for a longer viewing distance, changes in video quality lead to less degradation or improvement in QoE. Conversely, if a user is closer to the requested video, changes in video quality can have a greater impact on user QoE.

2) Quality Variation: Quality variation refers to the fluctuation in visual quality between video segments at successive time slots. A user experiences quality degradation when the visual quality decreases from the previous slot to the next. Let $\Delta V_{i,t}$ denote the quality variation at slot t, given by

$$\Delta V_{i,t} = \big[V_{i,t-1} - V_{i,t}\big]^+. \tag{8}$$

In (8), $[V_{i,t-1} - V_{i,t}]^+ = \max\{V_{i,t-1} - V_{i,t}, 0\}$.

3) Overall QoE: For modeling the overall QoE in volumetric video delivery, we use a linear combination of visual quality $V_{i,t}$ and quality variation $\Delta V_{i,t}$, a form widely used in existing research [8], [27]. For user i at slot t, the instantaneous QoE is given by

$$q_{i,t} = V_{i,t-1} - \beta\big(d_{i,t}\big) \cdot \Delta V_{i,t}. \tag{9}$$

Here, $\beta(d_{i,t})$ is the weight parameterized by the viewing distance $d_{i,t}$. Adjusting this weight allows for tuning the influence of quality variation $\Delta V_{i,t}$ on a user's overall QoE. Denote $\mathbf{x} = \{x_{m,q,t} | m \in \mathcal{M}, q \in \mathcal{Q}, t \in \mathcal{T}\}$ and $\mathbf{y} = \{y_{i,m,q,t} | i \in \mathcal{I}, m \in \mathcal{M}, q \in \mathcal{Q}, t \in \mathcal{T}\}$ as the sets of caching and rendering decisions, respectively. The accumulated QoE of all users over T slots is given by

$$\mathrm{QoE}(\mathbf{x}, \mathbf{y}) = \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{I}} q_{i,t}. \tag{10}$$

C. Resource Consumption Model

For XR services, the round-trip latency, also known as pose-to-render-to-photon latency, should be less than 50 ms [28]. As defined in [28], this latency consists of two components: user interaction delay and age of content. The former is defined as the time duration from the moment a user's pose changes until this change is recognized by the edge server. The latter, i.e., the age of content, is defined as the time duration from when a point cloud starts being rendered until the corresponding rendered video segment is displayed on the XR device. For simplicity, we assume that the user interaction delay is consistently controlled to be within a certain time, e.g., 20 ms [29]. We only evaluate the age of content, which includes the point cloud rendering latency and the rendered video transmission latency.

Delivering volumetric videos of different qualities usually requires different amounts of radio resources. In addition, network channel conditions can vary significantly based on user locations. To ensure timely delivery of rendered video segments, the edge server must allocate sufficient computing resources and radio spectrum to each XR device for efficient rendering and downlink transmission.[3] We define ρ as the

[2]This model is a simplified version of a more comprehensive model proposed in [8], [26]. We exclude certain factors (e.g., super-resolution ratio) from the model and treat them as fixed parameters.

[3]Note that we do not optimize the computing and bandwidth resources allocated to individual XR users in this article.

Authorized licensed use limited to: University of Waterloo. Downloaded on June 16,2025 at 15:26:53 UTC from IEEE Xplore. Restrictions apply.

computation intensity required for executing rendering tasks, measured in CPU cycles per bit, and τ^C as the rendering latency requirement. Thus, the required computational resources (in cycles per second) for XR device i at slot t, denoted by $R^C_{i,t}$, can be calculated as follows:

$$R^C_{i,t} = \frac{\rho}{\tau^C} \cdot \sum_{q\in\mathcal{Q}} \sum_{m\in\mathcal{M}} y_{i,m,q,t} \cdot \sigma_q. \tag{11}$$

For the downlink transmission from the edge server to each user, we consider a Line of Sight (LOS) link in Terahertz (THz) communications, known for providing higher bandwidths and faster data rates, which are essential for high-throughput applications. According to [30], the channel coefficient for user i in relation to the edge server is given by

$$L_p(r_{i,t}) = \frac{c^2}{(4\pi f_T)^2} \cdot r_{i,t}^{-\eta} \cdot \exp(-k(f_T) r_{i,t}) \tag{12}$$

where $r_{i,t}$ denotes the physical distance between user i and the AP at slot t, η is the path loss exponent, c represents the speed of light in free space, and f_T denotes the THz operating frequency. The molecular absorption coefficient, $k(f_T)$, which depends on f_T, also plays a crucial role. Using the Shannon capacity formula, the downlink spectrum efficiency of device i at slot t, denoted by $E_{i,t}$, is given by

$$E_{i,t} = \log_2\left(1 + \frac{PG_i L_p(r_{i,t})}{\sigma^2}\right) \tag{13}$$

where P denotes the transmission power, G_i represents the average channel gain between device i and the AP, and σ^2 denotes the average background noise power.

Let $R^B_{i,t}$, measured in Hz, represent the downlink radio spectrum consumed by user i at slot t. $R^B_{i,t}$ is calculated as

$$R^B_{i,t} = \frac{1}{E_{i,t}\tau^B} \cdot \sum_{q\in\mathcal{Q}} \sum_{m\in\mathcal{M}} y_{i,m,q,t} \cdot \chi_q \tag{14}$$

where χ_q is the data size (measured in bits) of a video segment after rendering a video at version q. This term, χ_q, simplifies our model by encapsulating various parameters such as frame rate, time slot durations, and resolution into a single comprehensive measure of data size for volumetric video segments.[4] Additionally, τ^B represents the rendered video transmission latency requirement.

D. Problem Formulation

Our objective is to maximize the accumulated QoE over time while ensuring that all the rendered volumetric video segments can be timely delivered to users given the limited network resources. To this end, we formulate the following optimization problem:

$$\mathbf{P_0}: \max_{\mathbf{x},\mathbf{y}} \ \mathrm{QoE}(\mathbf{x},\mathbf{y}) \tag{15}$$

$$\text{s.t.} \ x_{m,q,t}, y_{i,m,q,t} \in \{0,1\} \quad \forall i\ \forall m\ \forall q\ \forall t \tag{15a}$$

[4]We assume that even though different objects may have varying point cloud data sizes, the final rendered video is mainly dependent on the density.

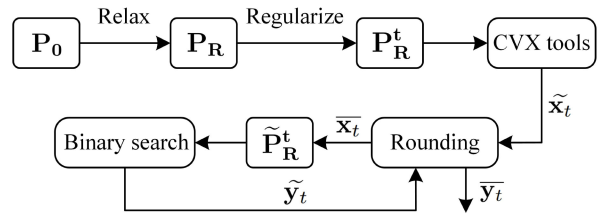

Fig. 2. Problem transformation diagram.

$$(2)-(4)$$

$$\sum_{i\in\mathcal{I}} R^C_{i,t} \le C_{\max}\ \forall t \tag{15b}$$

$$\sum_{i\in\mathcal{I}} R^B_{i,t} \le B_{\max}\ \forall t \tag{15c}$$

where $C_{\max}$ and $B_{\max}$ represent the maximum available computing resources on the edge server and radio resources on the AP, respectively. In the above formulation, (15a) is the binary constraint for point cloud caching and rendering decisions. Constraints (15b) and (15c) ensure that the total consumption of radio and computing resources does not exceed the available limits at the edge server.

Solving optimization problem $\mathbf{P_0}$ is nontrivial due to the following challenges. First, $\mathbf{P_0}$ is a long-term optimization problem that involves time-coupling variables. Solving it requires future information, such as users' future viewing distances and channel conditions. In practical XR systems, such predictions are highly uncertain as these conditions fluctuate over time. This requires online decision-making that can adapt to these uncertainties. Second, the problem is nonconvex and involves multiple integer variables, making it difficult to tackle even if complete information for all time slots is available.

IV. Online Optimization Scheme

In this section, we present a regularization-based online optimization and dependent rounding scheme (ROAD) to address the above challenges. The basic idea is shown in Fig. 2. The original problem, $\mathbf{P_0}$, is first relaxed to a relaxed optimization problem $\mathbf{P_R}$, in which all variables are considered to be continuous. Then, we transform $\mathbf{P_R}$ into a more tractable problem via the regularization technique, i.e., by replacing the time-coupling component with a smooth convex function. In this way, the long-term problem $\mathbf{P_R}$ is decomposed into a series of one-shot relaxed problems $\mathbf{P_R^t}$, each of which is convex and time-independent, thereby can be solved with convex tools to obtain a set of relaxed caching solutions, denoted by $\widetilde{\mathbf{x}}_t = \{\widetilde{x}_{m,q,t} | m \in \mathcal{M}, q \in \mathcal{Q}\}$. Then, we use a rounding algorithm to round elements in $\widetilde{\mathbf{x}}_t$ to obtain a set of integer solutions, i.e., $\overline{\mathbf{x}}_t = \{\overline{x}_{m,q,t} | m \in \mathcal{M}, q \in \mathcal{Q}\}$. Given the feasible caching solution $\overline{\mathbf{x}}_t$, we formulate an optimization problem, $\widetilde{\mathbf{P}}_\mathbf{R}^\mathbf{t}$, and use optimization theory to obtain a set of relaxed rendering solutions, denoted by $\widetilde{\mathbf{y}}_t = \{\widetilde{y}_{i,m,q,t} | i \in \mathcal{I}, m \in \mathcal{M}, q \in \mathcal{Q}\}$. Finally, we round $\widetilde{\mathbf{y}}_t$ to obtain the set of integral rendering solutions $\overline{\mathbf{y}}_t = \{\overline{y}_{i,m,q,t} | i \in \mathcal{I}, m \in \mathcal{M}, q \in \mathcal{Q}\}$.

Authorized licensed use limited to: University of Waterloo. Downloaded on June 16,2025 at 15:26:53 UTC from IEEE Xplore. Restrictions apply.

A. Regularization-Based Online Algorithm

1) Relaxation: We first relax (15a), obtaining the relaxed optimization problem as follows:

$$\mathbf{P_R}: \max_{\mathbf{x},\mathbf{y}} \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{I}}\Big(V_{i,t}-\beta\big(d_{i,t}\big)\cdot\big[V_{i,t-1}-V_{i,t}\big]^+\Big)$$
$$\text{s.t. } x_{m,q,t}\in[0,1], y_{i,m,q,t}\in[0,1] \quad \forall i\ \forall m\ \forall q\ \forall t$$
$$(2)-(4),\ (15b),\ (15c). \tag{16}$$

Solving $\mathbf{P_R}$ requires decoupling it into multiple subproblems for each time slot due to the uncertainty of future information. However, handling the time-coupled terms $[V_{i,t-1}-V_{i,t}]^+$ is a challenge. To overcome this challenge, we first transform these terms and then adopt a regularization technique to substitute each term with a smooth convex function [31].

Lemma 1: Minimizing $[V_{i,t-1} - V_{i,t}]^+$ is equivalent to minimizing the following expression:

$$\sum_{m\in\mathcal{M}}\sum_{q\in\mathcal{Q}}\big[q\cdot\alpha\big(d_{i,t-1}\big)y_{i,m,q,t-1}-q\cdot\alpha\big(d_{i,t}\big)y_{i,m,q,t}\big]^+. \tag{17}$$

Proof: Please refer to Appendix A for the detailed proof. ■

2) Regularization: We transform the objective function in $\mathbf{P_R}$ into a convex and time-decoupled function via regularization. The $[\,\cdot\,]^+$ term in (17) can be approximately transformed into the L_1-distance form, and relative entropy is an efficient alternative to regularize the L_1-distance in online problems [32]. To simplify the mathematical expressions, we replace $y_{i,m,q,t}$ with y_t and $\alpha(d_{i,t})$ with α_t in the following equations. The relative entropy function for $[\alpha_{t-1}y_{t-1}-\alpha_t y_t]^+$ is defined as follows:

$$\begin{aligned}\Delta&\Big(\alpha_{t-1}y_{t-1}\ \Big\|\ \alpha_t y_t\Big)\\ &=\alpha_{t-1}y_{t-1}\cdot\ln\frac{\alpha_{t-1}y_{t-1}}{\alpha_t y_t}+(\alpha_{t-1}y_{t-1}-\alpha_t y_t)\\ &=\alpha_{t-1}y_{t-1}\cdot\left(\ln\frac{\alpha_{t-1}y_{t-1}}{\alpha_t}-\ln y_t+1\right)-\alpha_t y_t. \end{aligned}\tag{18}$$

Some terms in (18) are only related to the previous time slot and thus cannot be optimized further. Therefore, optimizing (18) is equivalent to optimizing $-(\alpha_{t-1}y_{t-1}\ln y_t + \alpha_t y_t)$. Therefore, optimizing $[V_{i,t-1} - V_{i,t}]^+$ is equivalent to optimizing the following expression:

$$-\sum_{m\in\mathcal{M}}\sum_{q\in\mathcal{Q}}q\cdot(\alpha_{t-1}y_{t-1}\ln y_t+\alpha_t y_t). \tag{19}$$

The above equation is a linear combination of the relative entropy term and affine functions of the variables, making the corresponding optimization problem convex. To ensure the validity of the logarithmic function for $y_t = 0$, we introduce a positive constant ϵ and replace $\ln y_t$ with $\ln(y_t+\epsilon)$. Following the approach in [32], we normalize the equation by adding an approximation weight factor ω, where $\omega = \ln(1+\epsilon^{-1})$. The relative entropy function in (19) is then multiplied by ω^{-1}. Let $O_t(\mathbf{x}_t, \mathbf{y}_t)$ denote the one-shot optimization objective, given as follows:

$$O_t(\mathbf{x}_t,\mathbf{y}_t)=\sum_{q\in\mathcal{Q}}q\omega^{-1}\sum_{i\in\mathcal{I}}\beta\big(d_{i,t}\big)\sum_{m\in\mathcal{M}}\Big[\alpha\big(d_{i,t}\big)y_{i,m,q,t}$$
$$+\alpha\big(d_{i,t-1}\big)y_{i,m,q,t-1}\ln\big(y_{i,m,q,t}+\epsilon\big)\Big]+\sum_{i\in\mathcal{I}}V_{i,t}.$$

Algorithm 1 Regularization-Based Online Algorithm

Input: Q, M, I, $S_{\max}$, $C_{\max}$, $B_{\max}$, and other parameters
Output: $\widetilde{\mathbf{x}}_t$, $\widetilde{\mathbf{y}}_t$, $\forall t\in\mathcal{T}$

Set $\widetilde{y}_{i,m,q,0}=0$, $\forall i\in\mathcal{I}$, $m\in\mathcal{M}$, $q\in\mathcal{Q}$.
for each time slot $t\in\mathcal{T}$ **do**
  Observe $\widetilde{y}_{i,m,q,t-1}$, $d_{i,t-1}$, $d_{i,t}$, and $L_p\big(r_{i,t}\big)$;
  Solve $\mathbf{P_R^t}$ using the interior-point method;
  Return the optimal relaxed solutions $\widetilde{\mathbf{x}}_t$ and $\widetilde{\mathbf{y}}_t$;
end for

We can then decouple $\mathbf{P_R}$ into subproblems over time slots, with the subproblem for slot t formulated as follows:

$$\mathbf{P_R^t}: \max_{\mathbf{x}_t,\mathbf{y}_t}\ O_t(\mathbf{x}_t,\mathbf{y}_t)$$
$$\text{s. t. } x_{m,q,t}\in[0,1],\quad y_{i,m,q,t}\in[0,1]\quad\forall i\ \forall m\ \forall q$$
$$(2)-(4),\ (15b),\ (15c).$$

For each subproblem $\mathbf{P_R^t}$ ($t\in\mathcal{T}$), we use the interior-point method to solve it due to its convexity [33]. Algorithm 1 is used to obtain the relaxed solutions for $\mathbf{P_R}$.

B. Dependent Rounding Algorithm

In this subsection, we round the relaxed solutions in $\widetilde{\mathbf{x}}_t$ and $\widetilde{\mathbf{y}}_t$ to integer solutions $\overline{\mathbf{x}}_t$ and $\overline{\mathbf{y}}_t$ for slot t. One straightforward method is the independent randomized rounding algorithm [34]. The basic idea is as follows: for each fractional value $\widetilde{x}\in\mathbb{R}$, round it up to $\overline{x}=\lceil\widetilde{x}\rceil$ with probability $\widetilde{x}-\lfloor\widetilde{x}\rfloor$ (the fractional part of $\widetilde{x}$), or down to $\overline{x}=\lfloor\widetilde{x}\rfloor$ with probability $\lceil\widetilde{x}\rceil-\widetilde{x}$. However, such an independent rounding policy may not always generate feasible rounding solutions. For instance, there is a certain probability that too many variables in $\widetilde{\mathbf{x}}_t$ will be rounded up, potentially violating the storage constraint (2). Similarly, rounding up too many variables in $\widetilde{\mathbf{y}}_t$ could exceed the available computing and radio resources. The second design challenge lies in the covering feature of (3), which requires a carefully designed rounding order to ensure feasibility [35].

To tackle this challenge, we adopt a randomized dependent rounding algorithm to convert the relaxed solutions into integral ones [36]. The basic idea is that each rounded-down variable will be compensated by another rounded-up variable. With such a dependent rounding algorithm, the variables would not be aggressively rounded up or down. Specifically, we first round the outermost covering variable $\widetilde{\mathbf{x}}_t$ to obtain a rounded solution $\overline{\mathbf{x}}_t$.[5] Then, we resolve the one-shot optimization problem to obtain an optimal relaxed solution $\widetilde{\mathbf{y}}_t$ given $\overline{\mathbf{x}}_t$. Finally, we round $\widetilde{\mathbf{y}}_t$ to produce $\overline{\mathbf{y}}_t$.[6] For brevity, we omit the subscript t in the remainder of this section and illustrate the rounding algorithm using $\widetilde{\mathbf{x}}_t$ as an example. We

[5]Rounding $\widetilde{\mathbf{y}}_t$ first might make it impossible to find a feasible set of $\overline{\mathbf{x}}_t$ due to (2) and (3).

[6]Although Algorithm 1 generates relaxed solutions for both $\widetilde{\mathbf{x}}_t$ and $\widetilde{\mathbf{y}}_t$, we do not round them simultaneously, as doing so could violate (3).

Authorized licensed use limited to: University of Waterloo. Downloaded on June 16,2025 at 15:26:53 UTC from IEEE Xplore. Restrictions apply.

leave the detailed rounding procedure for $\widetilde{\mathbf{y}}_t$ in the online technical report [37].

The detailed randomized dependent rounding algorithm is shown in Algorithm 2. We first define two sets: the integer set $\mathcal{X}^+ = \{(m,q)|\widetilde{x}_{m,q} \in \mathbb{Z}\}$, containing all relaxed solutions that are already integers, and the floating set $\mathcal{X}^- = \{(m,q)|\widetilde{x}_{m,q} \in \mathbb{R}^+\}$, containing solutions that require rounding. For each element (m,q) in the floating set, we assign both a probability coefficient $p_{m,q}$ and a weight coefficient $w_{m,q}$ associated with it. Specifically, we set $p_{m,q} = \widetilde{x}_{m,q} - \lfloor \widetilde{x}_{m,q} \rfloor$ and $w_{m,q} = \sigma_q$. To round the elements in $\mathcal{X}^-$, the algorithm executes a series of rounding iterations. In each iteration, it randomly selects two elements (m_1,q_1) and (m_2,q_2) from $\mathcal{X}^-$ and ensures that at least one of them is updated (rounded either up or down).

The rounding probability is decided by φ_1 and φ_2, which are calculated as

$$\begin{cases} \varphi_1 = \min\{1 - p_{m_1,q_1}, \frac{w_{m_2,q_2}}{w_{m_1,q_1}} \cdot p_{m_2,q_2}\} \\ \varphi_2 = \min\{p_{m_1,q_1}, \frac{w_{m_2,q_2}}{w_{m_1,q_1}} \cdot \left(1 - p_{m_2,q_2}\right)\}. \end{cases} \tag{20}$$

Given the rounding probabilities φ_1 and φ_2, the probability coefficients $p_{m,q}$ are updated according to the following rules.

1) With probability $(\varphi_2/[\varphi_1 + \varphi_2])$, update

$$\begin{cases} p_{m_1,q_1} = p_{m_1,q_1} + \varphi_1 \\ p_{m_2,q_2} = p_{m_2,q_2} - \frac{w_{m_1,q_1}}{w_{m_2,q_2}} \cdot \varphi_1. \end{cases} \tag{21}$$

2) With probability $1 - (\varphi_2/[\varphi_1 + \varphi_2])$, update

$$\begin{cases} p_{m_1,q_1} = p_{m_1,q_1} - \varphi_2 \\ p_{m_2,q_2} = p_{m_2,q_2} + \frac{w_{m_1,q_1}}{w_{m_2,q_2}} \cdot \varphi_2. \end{cases} \tag{22}$$

Based on the updated values of p_{m_1,q_1} and p_{m_2,q_2}, at least one of the relaxed solutions $\widetilde{x}_{m_1,q_1}$ and $\widetilde{x}_{m_2,q_2}$ will be rounded up or down, as indicated in lines 10 to 17 of Algorithm 2. For any element $(m,q) \in \mathcal{X}^-$ that is rounded during the current iteration, it will be removed from the set $\mathcal{X}^-$ and added to the set $\mathcal{X}^+$. Finally, when $\mathcal{X}^-$ contains only one element in the last iteration, the algorithm rounds it up and checks whether (2) is satisfied. If the constraint is violated, the element is rounded down instead.

The proposed rounding algorithm is cost-efficient, as it would not aggressively cache point cloud versions or result in severe under-utilization of storage resources. This efficiency is achieved by maintaining the following good properties.

Proposition 1 (Continuously Reducing Property): At least one of the two selected variable pairs, (m_1,q_1) and (m_2,q_2), will be rounded to integers in each iteration.

Proof: Please refer to Appendix B for the detailed proof. ■

Proposition 2 (Weight Conservation Property): After each iteration, the total weighted storage consumption of the selected two elements remains constant, i.e.,

$$\sigma_{q_1}\bar{x}_{m_1,q_1} + \sigma_{q_2}\bar{x}_{m_2,q_2} = \sigma_{q_1}\widetilde{x}_{m_1,q_1} + \sigma_{q_2}\widetilde{x}_{m_2,q_2}. \tag{23}$$

This property ensures that the capacity constraint (2) will not be violated when running Algorithm 2.

Proof: Please refer to Appendix C for the detailed proof. ■

Algorithm 2 Randomized Dependent Rounding Algorithm

Input: $\mathcal{X}^+$, $\mathcal{X}^-$, σ_q
Output: $\bar{\mathbf{x}}_t$
1: **for** each element $(m,q) \in \mathcal{X}^+$ **do**
2: Set $\bar{x}_{m,q} = \widetilde{x}_{m,q}$;
3: **end for**
4: **for** each element $(m,q) \in \mathcal{X}^-$ **do**
5: Let $p_{m,q} = \widetilde{x}_{m,q} - \lfloor \widetilde{x}_{m,q} \rfloor$, $w_{m,q} = \sigma_q$;
6: **end for**
7: **while** $|\mathcal{X}^-| \geq 1$ **do**
8: Randomly select two elements (m_1,q_1) and (m_2,q_2) from the floating set $\mathcal{X}^-$;
9: Update the probability coefficients p_{m_1,q_1} and p_{m_2,q_2} according to Eqs. (20), (21), and (22);
10: **if** $p_{m_1,q_1} \in \{0,1\}$ **then**
11: Set $\bar{x}_{m_1,q_1} = \lfloor \widetilde{x}_{m_1,q_1} \rfloor + p_{m_1,q_1}$;
12: Update the integer set $\mathcal{X}^+ = \mathcal{X}^+ \cup \{(m_1,q_1)\}$ and the floating set $\mathcal{X}^- = \mathcal{X}^- \setminus \{(m_1,q_1)\}$;
13: **end if**
14: **if** $p_{m_2,q_2} \in \{0,1\}$ **then**
15: Set $\bar{x}_{m_2,q_2} = \lfloor \widetilde{x}_{m_2,q_2} \rfloor + p_{m_2,q_2}$;
16: Update the integer set $\mathcal{X}^+ = \mathcal{X}^+ \cup \{(m_2,q_2)\}$ and the floating set $\mathcal{X}^- = \mathcal{X}^- \setminus \{(m_2,q_2)\}$;
17: **end if**
18: **end while**
19: **if** $|\mathcal{X}^-| = 1$ **then**
20: Set $\bar{x}_{m,q} = \lceil \widetilde{x}_{m,q} \rceil$ for $(m,q) \in \mathcal{X}^-$;
21: If constraint (2) is not satisfied, set $\bar{x}_{m,q} = \lfloor \widetilde{x}_{m,q} \rfloor$;
22: **end if**

Proposition 3 (Marginal Distribution Property): The probability that each element $\widetilde{x}_{m,q}$ will be rounded up is given by $\Pr\{\bar{x}_{m,q} = \lceil \widetilde{x}_{m,q} \rceil\} = \widetilde{x}_{m,q} - \lfloor \widetilde{x}_{m,q} \rfloor$, and the probability that it will be rounded down is $\Pr\{\bar{x}_{m,q} = \lfloor \widetilde{x}_{m,q} \rfloor\} = 1 - (\widetilde{x}_{m,q} - \lfloor \widetilde{x}_{m,q} \rfloor)$. The expected value of the rounded solution equals the original relaxed value, $\mathbb{E}[\bar{x}_{m,q}] = \widetilde{x}_{m,q}$.

Proof: Please refer to Appendix D for the detailed proof. ■

C. Binary Search Algorithm

After obtaining the integral caching solution $\bar{\mathbf{x}}_t$, one intuitive way to make real-time rendering decisions is to solve the one-shot optimization problem $\mathbf{P_R^t}$ again by using CVX tools [38]. However, by reformulating $\mathbf{P_R^t}$ into a standard linear program, we can use convex optimization techniques to find the optimal solution with significantly reduced computational complexity compared to using CVX directly. Specifically, the linear optimization problem $\mathbf{P_R^t}$ can be rewritten in its standard form, expressed as follows:

$$\widetilde{\mathbf{P}}_{\mathbf{R}}^{\mathbf{t}} : \min_{\mathbf{y}_t} \; O_t(\bar{\mathbf{x}}_t, \mathbf{y}_t) \tag{24}$$

$$\text{s.t.} \; 0 \leq y_{i,m,q,t} \leq 1 \quad \forall i, m, q \tag{24a}$$

$$y_{i,m,q,t} - \bar{x}_{m,q,t} \leq 0 \quad \forall i, m, q \tag{24b}$$

$$\sum_{m \in \mathcal{M}} \sum_{q \in \mathcal{Q}} y_{i,m,q,t} - 1 \leq 0 \quad \forall i \tag{24c}$$

Authorized licensed use limited to: University of Waterloo. Downloaded on June 16,2025 at 15:26:53 UTC from IEEE Xplore. Restrictions apply.

$$\sum_{i\in\mathcal{I}}\frac{\rho}{\tau^C}\cdot\sum_{q\in\mathcal{Q}}\sigma_q\cdot\sum_{m\in\mathcal{M}}y_{i,m,q,t}-C_{\max}\leq 0 \quad (24d)$$

$$\sum_{i\in\mathcal{I}}\frac{1}{E_{i,t}\tau^B}\cdot\sum_{q\in\mathcal{Q}}\chi_q\cdot\sum_{m\in\mathcal{M}}y_{i,m,q,t}-B_{\max}\leq 0. \quad (24e)$$

For any pair (m, q), if the caching decision $\bar{x}_{m,q,t} = 0$, then the optimal rendering decision, denoted by $y^\star_{i,m,q,t}$, will have $y^\star_{i,m,q,t} = 0$, for all $i \in \mathcal{I}$. Therefore, we focus on cases where $\bar{x}_{m,q,t} = 1$. Under this condition, (24b) is subsumed by (24a) and can be omitted in subsequent analysis. In addition, for any $i \in \mathcal{I}$, if $\sum_{m\in\mathcal{M}}\sum_{q\in\mathcal{Q}} y_{i,m,q,t} \leq 1$, then we will have $y^\star_{i,m,q,t} \leq 1$, for all $m \in \mathcal{M}$ and $q \in \mathcal{Q}$.

We introduce Lagrange multipliers $\{\gamma_i\}_{\forall i}$ for (24c), θ for (24d), and λ for (24e). The Lagrangian L is given as follows:

$$\begin{aligned}L(y,\gamma,\theta,\lambda) = &-\sum_{i\in\mathcal{I}}\sum_{m\in\mathcal{M}}\sum_{q\in\mathcal{Q}}\Big(\delta_i\beta_i\cdot\ln\big(y_{i,m,q,t}+\epsilon\big)\\ &+(1+\beta_i)\cdot\alpha_i y_{i,m,q,t}\Big)\cdot q\\ &+\sum_{i\in\mathcal{I}}\gamma_i\cdot\left(\sum_{m\in\mathcal{M}}\sum_{q\in\mathcal{Q}}y_{i,m,q,t}-1\right)\\ &+\theta\cdot\left(\sum_{i\in\mathcal{I}}\sum_{m\in\mathcal{M}}\sum_{q\in\mathcal{Q}}y_{i,m,q,t}\varsigma_q-C_{\max}\right)\\ &+\lambda\cdot\left(\sum_{i\in\mathcal{I}}\sum_{m\in\mathcal{M}}\sum_{q\in\mathcal{Q}}y_{i,m,q,t}\cdot\frac{\xi_q}{r_{i,t}}-B_{\max}\right).\end{aligned}$$

For brevity, the above equation replaces $\alpha(d_{i,t-1})y_{i,m,q,t-1}/\omega$ with δ_i $(i \in \mathcal{I})$, $\sigma_q\rho_q/\tau^C$ with ς_q $(q \in \mathcal{Q})$, χ_q/τ^B with ξ_q $(q \in \mathcal{Q})$, $\alpha(d_{i,t})$ with α_i $(i \in \mathcal{I})$, and $\beta(d_{i,t})$ with β_i $(i \in \mathcal{I})$.

Next, we use KKT conditions to establish the conditions for optimality in solving $\widetilde{\mathbf{P}}^{\mathbf{t}}_{\mathbf{R}}$. The KKT conditions are given as follows:

$$\begin{cases}\gamma_i\geq 0\quad\forall I\in\mathcal{I};\quad\theta\geq 0,\quad\lambda\geq 0 & (25)\\ \gamma_i\cdot\left(\sum_{m\in\mathcal{M}}\sum_{q\in\mathcal{Q}}y_{i,m,q,t}-1\right)=0\quad\forall i\in\mathcal{I} & (26)\\ \gamma_i+\varsigma_q\theta+\dfrac{\xi_q\lambda}{r_{i,t}} \\ \quad -q\cdot\left(\dfrac{\delta_i\beta_i}{y_{i,m,q,t}+\epsilon}+\alpha_i(1+\beta_i)\right)=0\quad\forall i\in\mathcal{I}. & (27)\end{cases}$$

By substituting (27) into (26), we can obtain the following equation:

$$\underbrace{\left(\frac{q\delta_i\beta_i}{y_{i,m,q,t}+\epsilon}+q\alpha_i(1+\beta_i)-\varsigma_q\theta-\frac{\xi_q\lambda}{r_{i,t}}\right)}_{A_i}\cdot\underbrace{\left(\sum_{m\in\mathcal{M}}\sum_{q\in\mathcal{Q}}y_{i,m,q,t}-1\right)}_{B_i}=0\quad\forall i\in\mathcal{I}. \quad (28)$$

For any given $i \in \mathcal{I}$, we analyze the following two cases.

1) If there exist Lagrange multipliers (θ, λ) satisfying

$$\min_{q\in\mathcal{Q}}\left\{\frac{\varsigma_q\theta+\xi_q r_{i,t}^{-1}\lambda}{q}\right\}>\left(\frac{\delta_i\beta_i}{y_{i,m,q,t}+\epsilon}+\alpha_i(1+\beta_i)\right) \quad (29)$$

we will have $A_i > 0$. To satisfy (28), $B_i = 0$ must hold, i.e., $\sum_{m\in\mathcal{M}}\sum_{q\in\mathcal{Q}} y_{i,m,q,t} - 1 = 0$. The optimal value is derived using the Lagrange multiplier method, which is calculated as follows:

$$y^\star_{i,m,q,t}=\frac{q\delta_i\beta_i}{\gamma^\star_i-q\alpha_i(1+\beta_i)}-\epsilon\quad\forall m\in\mathcal{M}q\in\mathcal{Q} \quad (30)$$

where $\gamma^\star_i$ is obtained by solving the following equation:

$$\sum_{m\in\mathcal{M}}\sum_{q\in\mathcal{Q}}\left(\frac{q\delta_i\beta_i}{\gamma_i}-q\alpha_i(1+\beta_i)-\epsilon\right)=1. \quad (31)$$

2) If $B_i < 0$, i.e., $\sum_{m\in\mathcal{M}}\sum_{q\in\mathcal{Q}} y_{i,m,q,t} < 1$, then we have $\gamma_i = 0$. For all $m \in \mathcal{M}, q \in \mathcal{Q}$, the optimal rendering decision $y^\star_{i,m,q,t}$ is given by

$$y^\star_{i,m,q,t}=\frac{q\delta_i\beta_i}{\varsigma_q\theta+\xi_q\lambda/r_{i,t}-q\alpha_i(1+\beta_i)}-\epsilon. \quad (32)$$

From (30)–(32), we observe that decreasing the values of (θ, λ) results in a nondecreasing optimal value $y^\star_{i,m,q,t}$. As a result, both the computing resource consumption, i.e., $\sum_{i\in\mathcal{I}}\sum_{m\in\mathcal{M}}\sum_{q\in\mathcal{Q}} y_{i,m,q,t}\varsigma_q$ and the radio resource consumption, i.e., $\sum_{i\in\mathcal{I}}\sum_{m\in\mathcal{M}}\sum_{q\in\mathcal{Q}} y_{i,m,q,t}\xi_q/r_{i,t}$ are nonincreasing with respect to (θ, λ). Therefore, we can use a binary search algorithm to efficiently determine the optimal pair of $(\theta^\star, \lambda^\star)$. This approach ensures maximal utilization of either computing or bandwidth resources, satisfying at least one of the constraints in (24d) and (24e).

The searching ranges of θ and λ are determined as follows:

$$\begin{cases}\theta^{\max}=\max_{q\in\mathcal{Q}}\left\{\frac{q}{\varsigma_q}\right\}\max_{i\in\mathcal{I}}\left\{\frac{\delta_i\beta_i}{\epsilon}+\alpha_i(1+\beta_i)\right\}\\ \theta^{\min}=\min_{q\in\mathcal{Q}}\left\{\frac{q}{\varsigma_q}\right\}\min_{i\in\mathcal{I}}\left\{\frac{\delta_i\beta_i}{1+\epsilon}+\alpha_i(1+\beta_i)\right\}\end{cases}$$

$$\begin{cases}\lambda^{\max}=\max_{q\in\mathcal{Q}}\left\{\frac{q}{\xi_q}\right\}\max_{i\in\mathcal{I}}\left\{\left(\frac{\delta_i\beta_i}{\epsilon}+\alpha_i(1+\beta_i)\right)r_{i,t}\right\}\\ \lambda^{\min}=\min_{q\in\mathcal{Q}}\left\{\frac{q}{\xi_q}\right\}\min_{i\in\mathcal{I}}\left\{\left(\frac{\delta_i\beta_i}{1+\epsilon}+\alpha_i(1+\beta_i)\right)r_{i,t}\right\}.\end{cases}$$

The detailed searching procedures are shown in Algorithm 3. At each iteration, the algorithm computes the midpoint values for both parameters θ and λ. Depending on whether (24d) and (24e) are satisfied, the bounds for θ and λ are updated. The algorithm will continue iterating until the difference between the upper and lower bounds falls below a small positive threshold ε.

Authorized licensed use limited to: University of Waterloo. Downloaded on June 16,2025 at 15:26:53 UTC from IEEE Xplore. Restrictions apply.

Algorithm 3 2-D Binary Search Algorithm

Input: $\theta^{\min}$, $\theta^{\max}$, $\lambda^{\min}$, $\lambda^{\max}$, and ε
Output: $\{y^{\star}_{i,m,q,t}\}_{\forall i,m,q}$
repeat
  Update $\theta^{\text{mid}} = \frac{\theta^{\max}+\theta^{\min}}{2}$ and $\lambda^{\text{mid}} = \frac{\lambda^{\max}+\lambda^{\min}}{2}$;
  Compute $y^{\star}_{i,m,q,t}$ using Eqs. (30), (31), and (32);
  Compute the computation and storage resource consumption for the following parameter pairs: $(\theta^{\text{mid}}, \lambda^{\text{mid}})$, $(\theta^{\text{mid}}, \lambda^{\max})$, $(\theta^{\max}, \lambda^{\text{mid}})$, and $(\theta^{\max}, \lambda^{\max})$;
  if the pair $(\theta^{\text{mid}}, \lambda^{\text{mid}})$ satisfies both Eqs. (24d) and (24e) **then**
    Update $\theta^{\max} = \theta^{\text{mid}}$ and $\lambda^{\max} = \lambda^{\text{mid}}$;
  else if both pairs $(\theta^{\max}, \lambda^{\text{mid}})$ and $(\theta^{\text{mid}}, \lambda^{\max})$ violate one of Eqs. (24d) or (24e) **then**
    Update $\theta^{\min} = \theta^{\text{mid}}$ and $\lambda^{\min} = \lambda^{\text{mid}}$;
  else if the pair $(\theta^{\max}, \lambda^{\text{mid}})$ violates one of the Eqs. (24d) or (24e), while the pair $(\theta^{\text{mid}}, \lambda^{\max})$ satisfies both **then**
    Update $\theta^{\max} = \theta^{\text{mid}}$ and $\lambda^{\min} = \lambda^{\text{mid}}$;
  else
    Update $\theta^{\min} = \theta^{\text{mid}}$ and $\lambda^{\max} = \lambda^{\text{mid}}$;
  end if
until $|\lambda^{\max} - \lambda^{\min}| \le \varepsilon$ and $|\theta^{\max} - \theta^{\min}| \le \varepsilon$
Set $\theta^{\star} = \theta^{\text{mid}}$ and $\lambda^{\star} = \lambda^{\text{mid}}$, then obtain $\{y^{\star}_{i,m,q,t}\}_{\forall i,m,q}$;

D. Complexity Analysis

The computational complexity of the proposed scheme can be estimated to be $O((IMQ)^{3.5}\log(\omega^{-1}))$. Specifically, Algorithm 1 uses an interior-point method to solve a convex optimization problem, i.e., $\mathbf{P}^{\mathbf{t}}_{\mathbf{R}}$, for each slot t. As discussed in [33, Sec. 11.5], the number of Newton iterations required to achieve a desired solution accuracy ω is bounded by $O(\log(IMQ/\omega))$, where IMQ represents the number of inequality constraints.[7] For each Newton iteration, the algorithm solves a linear system with a computational complexity of $O((IMQ)^3)$, where IMQ is also the number of variables. Therefore, the overall computational complexity of Algorithm 1 is $O((IMQ)^{3.5}\log(\omega^{-1}))$.

For Algorithm 2, the time complexity is $\mathcal{O}(N)$, where N is the total number of elements in $\mathcal{X}^{-}$ that need to be rounded. Moreover, we have $N \le MQ$. For Algorithm 3, the computational time complexity can be approximated by $O(\log_2(\theta^{\max} - \theta^{\min})\log_2(\lambda^{\max} - \lambda^{\min}))$ [39]. Since the complexity of Algorithm 1 is higher than that of Algorithms 2 and 3, the overall complexity of the proposed scheme is dominated by Algorithm 1, namely, $O((IMQ)^{3.5}\log(\omega^{-1}))$.

V. Performance Evaluation

A. Simulation Setup

The main parameters used in the simulations are listed in Table III. We consider an XR service area of 100 m $\times$ 100 m, within which volumetric videos and XR users are randomly distributed. The edge server is located at the center of this area, with coordinates (50 m, 50 m). The maximum velocity for each user is set not to exceed 1 m/s during any time slot. For repeated simulations, virtual objects' distribution and each user's initial position are fixed.

[7]Note that the number of inequality constraints is dominated by (4).

TABLE III
Simulation Parameters

Parameter	Value	Parameter	Value
M	100	τ^{C}	35 ms
Q	4	τ^{B}	15 ms
I	100	χ_1	24 MB
T	3000	σ_1	28.8 MB
$S_{\max}$	30 GB	f_T	0.8 THz
$C_{\max}$	170 GFLOPS	$k(f_T)$	0.03
$B_{\max}$	320 MHz	P	30 dBm
ρ	0.85 GFLOPS	δ	100 ms

TABLE IV
Weighted Parameters for QoE

Viewing Distance (m)	$(0,1]$	$(1,2]$	$(2,3]$	$(3,4]$	$(4,\infty)$
α	0.5	0.3	0.15	0.05	0
β	1	0.6	0.3	0.1	0

Each volumetric video has four density versions: 1) 25%; 2) 50%; 3) 75%; and 4) 100%. The resource consumption for all volumetric videos is identical and directly proportional to their densities. We also specify that XR users view volumetric videos at a frame rate of 30 frames per second (FPS), with each video segment consisting of three frames. Therefore, the duration of each time slot is 100 ms. In practice, the rendered video segments are encoded at the edge server, transmitted, and then decoded on the XR device. We set a compression ratio of 10 and a total encoding and decoding latency of 10 ms. The parameters used in the QoE model are listed in Table IV. When a user's viewing distance exceeds 4 m, the corresponding volumetric video is not provided to the user.

We introduce the following benchmarks for comparison.

1) *2D-QoE:* This benchmark uses traditional QoE models suitable for 2-D videos, where the user QoE depends only on video quality [7].
2) *Greedy:* This benchmark caches popular video versions by considering the number of users viewing each video, prioritizing versions with the highest user demand. The demand is determined by the number of users within specific distance ranges from the video, where closer distances correspond to higher-quality versions. The rendering policy is the same as that of the proposed ROAD.
3) *IRR:* This benchmark uses the independent randomized rounding algorithm [36]. The key difference between IRR and our proposed ROAD lies in the rounding algorithm.

All algorithms use the same rule for the overall QoE calculation.

B. Varied Resource Capacity

In this section, we evaluate the impact of caching, computing, and radio resource capacity on XR users' accumulated

Authorized licensed use limited to: University of Waterloo. Downloaded on June 16,2025 at 15:26:53 UTC from IEEE Xplore. Restrictions apply.

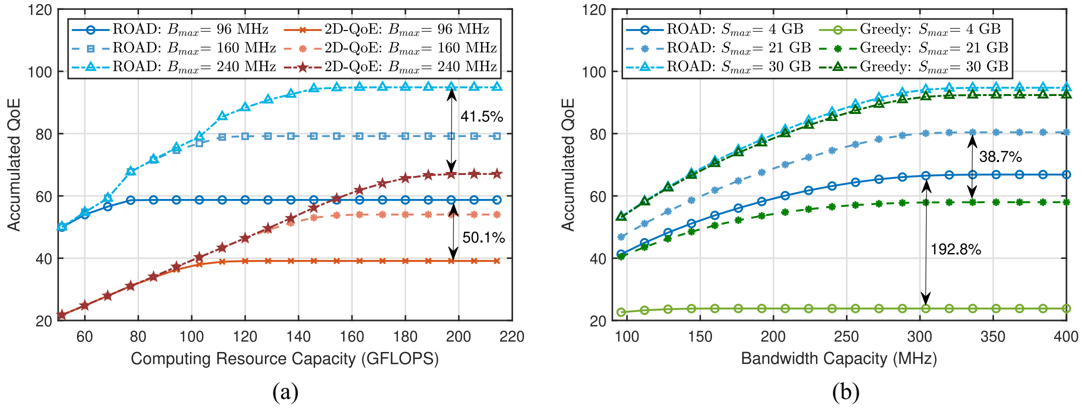

Fig. 3. Achieved QoE versus various resource amounts. (a) Computing resource. (b) Communication resource.

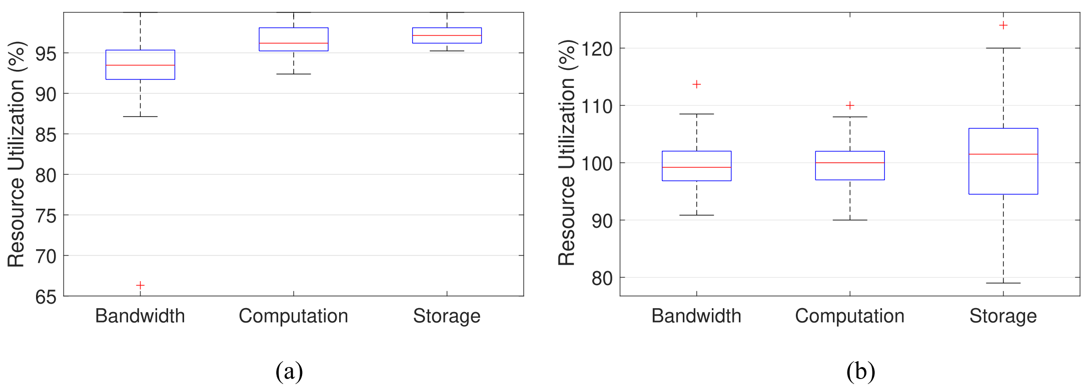

Fig. 4. Resource utilization for (a) ROAD and (b) IRR.

QoE. We first vary the computing resource capacity, as shown in Fig. 3(a). The achieved QoE for all algorithms first increases almost linearly as the capacity increases. This increase occurs because more computing resources allow the selection of high-density point clouds for more rendering tasks, thus enhancing the overall QoE. However, as computing resource capacity further increases, the improvement rate of QoE diminishes. The reason is that the user performance encounters transmission and caching bottlenecks. Although the edge server can support more high-quality rendering, there is insufficient bandwidth to deliver high-quality volumetric videos or inadequate caching space to store high-density point clouds. Fig. 3(b) shows a similar trend in QoE when varying the bandwidth resource capacity.

The above figures also demonstrate the superior performance of the proposed ROAD algorithm compared to the benchmark algorithms. Compared to 2D-QoE, our algorithm achieves a significant improvement in accumulated QoE, with an average increase of 46%. This improvement is due to ROAD's use of a customized QoE model to estimate XR user quality requirements, enabling more resource-efficient caching and rendering decisions. Compared to the greedy caching algorithm, the proposed ROAD algorithm achieves a significantly higher QoE, particularly when caching resources are stringent. This improvement is due to ROAD's QoE-oriented design, which optimally utilizes caching resources to store volumetric videos in their most resource-efficient versions, thereby maximizing users' accumulated QoE.

C. Rounding Performance

Next, we evaluate the rounding performance of ROAD and IRR. For IRR, we consider all rounding results and calculate their resource utilization (measured in %), which is defined as the ratio of the algorithm's decisions to the available resource capacity. As shown in Fig. 4, nearly half of the results for IRR exhibit resource violations in terms of bandwidth, computation, and storage resources. This occurs because IRR has a certain probability of rounding up a large number of caching or rendering decisions. In contrast, ROAD remains within the resource capacity. In addition, IRR exhibits a much larger variance compared to ROAD, which could contribute to its potentially lower QoE performance.

D. Varied User's Sensitivity

We evaluate the impact of users' viewing distance sensitivity on the accumulated QoE of XR users. A discount factor, γ, is introduced and applied to the original parameters α and β to represent user sensitivity. We vary γ from 0.05 to 1 and present the results in Fig. 5. The figure reveals the following information: 1) ROAD consistently outperforms the benchmark algorithms, demonstrating the effectiveness of our approach, which incorporates a QoE model based on viewing distance and 2) as the discount factor γ increases, the performance gap between ROAD and the benchmark algorithms (e.g., 2D-QoE) increases. This trend shows the

Authorized licensed use limited to: University of Waterloo. Downloaded on June 16,2025 at 15:26:53 UTC from IEEE Xplore. Restrictions apply.

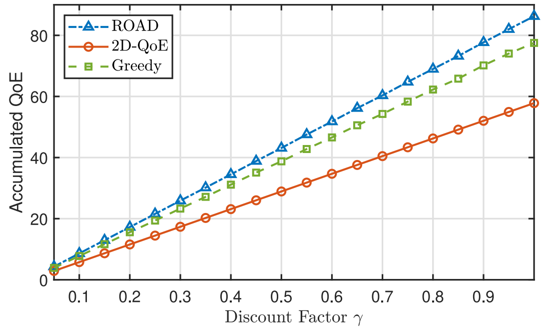

Fig. 5. QoE versus discount factor added on QoE parameters.

significance of our proposed approach in environments where user QoE is highly sensitive to viewing distances.

E. Varied Viewing Preferences

We also evaluate the impact of user viewing preferences on resource consumption. Regarding viewing preferences, we mainly consider users' average viewing durations of volumetric videos. We define two groups of users: 1) Group 1 users have shorter average viewing times for each video, indicating more casual interactions with the virtual environment and 2) in contrast, Group 2 users tend to view videos more carefully, spending more time on each video. We calculate the average downlink bandwidth and computing resource consumption per user over $T = 3000$ time slots. For each scenario, we conduct 60 simulation runs and plot the mean results with 95% confidence intervals.

First, we set the average viewing duration to 50 time slots for Group 1 users and 300 time slots for Group 2 users. The average computation and bandwidth resource consumption per user over T time slots are shown in Fig. 6. From the figure, we find that Group 2 users (green points) consume considerably more computing and bandwidth resources than Group 1 users (orange points).

Next, we vary the average viewing durations for Group 2 users while keeping the viewing duration for Group 1 users unchanged. We calculate the average bandwidth consumption and computational resource consumption for both groups, as shown in Fig. 7. The results show that as the average viewing duration of Group 2 users increases, Group 2 users aggressively occupy more computing and bandwidth resources. In addition, when the viewing duration of Group 2 users is set to 300 time slots, their bandwidth and computing resource consumptions are 149.8% and 132.0% higher, respectively, than that of Group 1 users. This increased consumption by Group 2 is due to their more detailed examination of virtual objects in the videos, which generates longer interaction times and potentially requires higher-quality rendering. These findings reveal that understanding user viewing preferences is crucial for effective resource management.

VI. Conclusion

In this article, we have proposed a novel volumetric video caching and rendering approach for an edge-assisted XR system to enhance user QoE. Our approach has jointly considered the impact of users' viewing behaviors and preferences

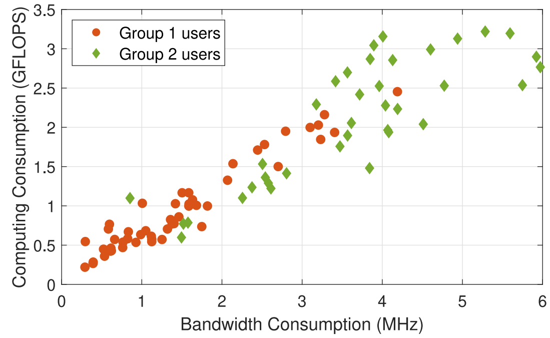

Fig. 6. Time-averaged resource consumption for each user.

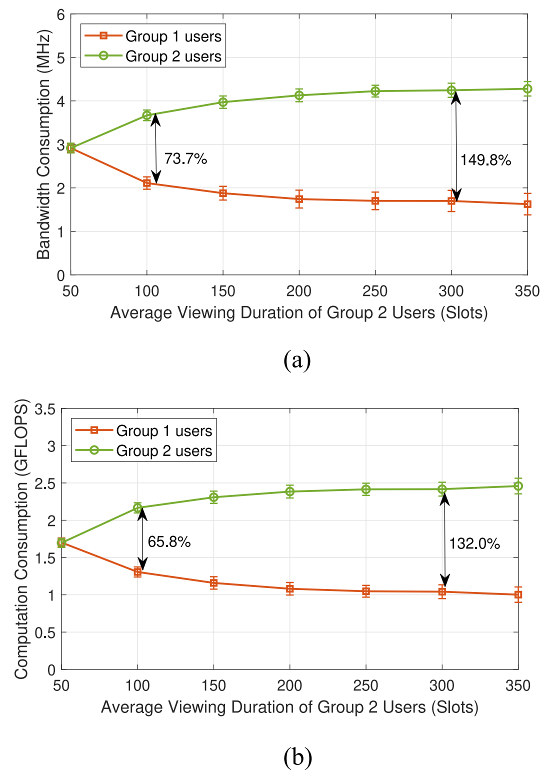

Fig. 7. Average resource consumption versus average viewing duration of Group 2 users. (a) Bandwidth consumption. (b) Computing resource consumption.

on edge resource optimization. By adopting our approach, XR service providers (e.g., AR/VR content platforms) can achieve significant cost savings and increase user satisfaction by delivering the appropriate video quality to each user. In future work, we plan to leverage digital twin technologies to better capture long-term user viewing behavior patterns and enable more efficient resource reservation for XR services.

Appendix A
Proof of Lemma 1

The coupled term across different time slots can be transformed as follows:

$$\begin{aligned}&\left[V_{i,t-1} - V_{i,t}\right]^+ \\ &\quad = \left[\sum_{q\in\mathcal{Q}} q \sum_{m\in\mathcal{M}} \left(\alpha\left(d_{i,t-1}\right) y_{i,m,q,t-1} - \alpha\left(d_{i,t}\right) y_{i,m,q,t}\right)\right]^+\end{aligned}$$

Authorized licensed use limited to: University of Waterloo. Downloaded on June 16,2025 at 15:26:53 UTC from IEEE Xplore. Restrictions apply.

$$\overset{(a)}{=} \Big[\pi_{1,1} + \pi_{1,2} + \cdots + \pi_{M,Q}\Big]^{+}$$
$$\overset{(b)}{\leq} \sum_{m\in\mathcal{M}} \sum_{q\in\mathcal{Q}} [\pi_{m,q}]^{+} \quad \text{(by subadditivity of } [\cdot]^{+})$$
$$= \sum_{q\in\mathcal{Q}} \sum_{m\in\mathcal{M}} q\Big[\alpha\big(d_{i,t-1}\big)\, y_{i,m,q,t-1} - \alpha\big(d_{i,t}\big)\, y_{i,m,q,t}\Big]^{+}.$$

Here, in step (a), we define

$$\pi_{m,q} = q\Big(\alpha\big(d_{i,t-1}\big)\, y_{i,m,q,t-1} - \alpha\big(d_{i,t}\big)\, y_{i,m,q,t}\Big) \quad \forall m, q$$

to decompose the summation into individual terms. Step (b) follows from the subadditivity of the $[\,\cdot\,]^{+}$ operator

$$[a_1 + a_2 + \cdots + a_n]^{+} \leq [a_1]^{+} + [a_2]^{+} + \cdots + [a_n]^{+}$$

which generalizes the triangle inequality for positive parts. Since the inequality in step (b) provides an upper bound, minimizing $\sum_{m,q} [\pi_{m,q}]^{+}$ ensures that $[V_{i,t-1} - V_{i,t}]^{+}$ is also minimized. This completes the proof.

Appendix B
Proof of Proposition 1

In each iteration of Algorithm 2, there are four possible cases for the rounding process, determined by the values of p_{m_i,q_i} and w_{m_i,q_i} for $i \in \{1, 2\}$. For clarity, let $p_i = p_{m_i,q_i}$ and $w_i = w_{m_i,q_i}$ throughout this proof.

1) If $1 - p_1 \geq (w_2/w_1) \cdot p_2$, then $\varphi_1 = (w_1/w_2) \cdot p_2$. According to (21), with probability $(\varphi_2/[\varphi_1 + \varphi_2])$, we have
$$\begin{cases} p_1 = p_1 + \varphi_1 \; = \; p_1 + \frac{w_1}{w_2} \cdot p_2 \\ p_2 = p_2 - \frac{w_2}{w_1} \cdot \varphi_1 \; = \; 0. \end{cases}$$
2) If $1 - p_1 < (w_2/w_1) \cdot p_2$, then $\varphi_1 = 1 - p_1$. Again, from (21), with probability $(\varphi_2/[\varphi_1 + \varphi_2])$, we have
$$\begin{cases} p_1 = p_1 + \varphi_1 \; = \; p_1 + 1 - p_1 = 1 \\ p_2 = p_2 - \frac{w_2}{w_1} \cdot \big(1 - p_1\big). \end{cases}$$
3) If $p_1 \geq (w_2/w_1) \cdot (1 - p_2)$, then $\varphi_2 = (w_2/w_1) \cdot (1 - p_2)$. From (22), with probability $(\varphi_1/[\varphi_1 + \varphi_2])$, we have
$$\begin{cases} p_1 = p_1 - \varphi_2 \; = \; p_1 - \frac{w_2}{w_1} \cdot \big(1 - p_2\big) \\ p_2 = p_2 + \frac{w_1}{w_2} \cdot \varphi_2 \; = \; 1. \end{cases}$$
4) If $p_1 < (w_2/w_1) \cdot (1 - p_2)$, then $\varphi_2 = p_1$. From (22), with probability $(\varphi_1/[\varphi_1 + \varphi_2])$, we have
$$\begin{cases} p_1 = p_1 - \varphi_2 = 0 \\ p_2 = p_2 + \frac{w_1}{w_2} \cdot \varphi_2. \end{cases}$$

Regardless of the values of p_i and w_i, for $i \in \{1, 2\}$, at least one of case 1 and case 2 should happen. At the same time, at least one of case 3 and case 4 should happen. Therefore, after parameter updating according to (21) and (22), at least one of p_1 and p_2 must satisfy $p_i \in \{0, 1\}, i \in \{1, 2\}$. Thus, at least one of lines 10 and 14 in Algorithm 2 will be executed. At least one pair, either (m_1, q_1) or (m_2, q_2) will be rounded, and the number of elements in the set $\mathcal{X}^{-}$ will be reduced by at least 1. This completes the proof.

Appendix C
Proof of Proposition 2

In each iteration of Algorithm 2, two elements are selected. However, there are cases in which only one of those elements can be rounded to an integer. For any element that is not rounded, its probability coefficient $p_{m,q}$ is adjusted during the iteration (see line 9 of Algorithm 2). It is crucial to ensure that the total storage consumption remains constant throughout the algorithm. Denote S^0 be the total storage consumption at the initial (0th) iteration

$$S^0 = \sum_{m,q} \Big(\lfloor \widetilde{x}_{m,q} \rfloor + p^0_{m,q}\Big) \cdot \sigma_q.$$

Let $|K|$ be the total number of iterations in Algorithm 2. Denote S^k be the total storage consumption at iteration k, where $0 \leq k \leq |K| - 1$. We now show that $S^{k+1} = S^k$ under the following two cases.

1) If p_1 and p_2 are updated according to (21), then
$$S^{k+1} = S^k + \sigma_{q_1}\varphi_1 - \sigma_{q_2} \cdot \frac{w_{m_1,q_1}}{w_{m_2,q_2}} \cdot \varphi_1 = S^k.$$
2) If p_1 and p_2 are updated according to (22), then
$$S^{k+1} = S^k + \sigma_{q_1}\varphi_2 - \sigma_{q_2} \cdot \frac{w_{m_1,q_1}}{w_{m_2,q_2}} \cdot \varphi_2 = S^k.$$

In both cases, the total storage consumption remains unchanged after each iteration. This completes the proof of Proposition 2.

Appendix D
Proof of Proposition 3

Let $Y^k_{m,q}$ be the random variable denoting the value of $\widetilde{x}_{m,q}$ at the beginning of iteration k. We will show that

$$\mathbb{E}\big[Y^{k+1}_{m,q}\big] = \mathbb{E}\big[Y^{k}_{m,q}\big] \quad \text{for all } k \geq 1. \tag{33}$$

Then, we will have $\Pr\{\bar{x}_{m,q} = \lceil \widetilde{x}_{m,q} \rceil\} = \mathbb{E}[Y^{|k|}_{m,q}] = \cdots = \mathbb{E}[Y^{1}_{m,q}] = \bar{x}_{m,q} - \lfloor \widetilde{x}_{m,q} \rfloor$. Hence, the marginal distribution property holds.

We now prove (33) for a fixed k. According to line 8 in Algorithm 2, one of the following two cases could occur for the element $\widetilde{x}_{m,q}$ in iteration k.

1) If $\widetilde{x}_{m,q}$ is not selected in line 8, its value remains unchanged. Therefore, for any possible value v of $Y^k_{m,q}$, we have $\mathbb{E}[Y^{k+1}_{m,q} | Y^k_{m,q} = v] \; = v$.
2) If $\widetilde{x}_{m,q}$ is selected in line 8, then line 9 modifies its expected value according to the update rules in this iteration. If line 10 is triggered, the expectation is calculated as $\mathbb{E}[Y^{k+1}_{m,q} | Y^k_{m,q} = v] = (\varphi_2/[\varphi_1 + \varphi_2]) \cdot (v + \varphi_1) + (\varphi_1/[\varphi_1 + \varphi_2]) \cdot (v - \varphi_2) = v$. If line 14 is triggered, the expectation is calculated as $\mathbb{E}[Y^{k+1}_{m,q} | Y^k_{m,q} = v] = (\varphi_2/[\varphi_1 + \varphi_2]) \cdot (v - (w_1/w_2) \cdot \varphi_1) + (\varphi_1/[\varphi_1 + \varphi_2]) \cdot (v + (w_1/w_2) \cdot \varphi_2) = v$.

In both cases, the conditional expectation of $Y^{k+1}_{m,q}$ remains equal to v. Thus, (33) is proved.

Let $\mathcal{V}$ denote the set of all possible values that $Y^k_{m,q}$ can take. Then, we have

Authorized licensed use limited to: University of Waterloo. Downloaded on June 16,2025 at 15:26:53 UTC from IEEE Xplore. Restrictions apply.

$$\mathbb{E}\Big[Y_{m,q}^{k+1}\Big] = \sum_{v\in\mathcal{V}} \mathbb{E}\Big[Y_{m,q}^{k+1} | Y_{m,q}^{k} = v\Big] \cdot \Pr\Big[Y_{m,q}^{k} = v\Big]$$
$$= \sum_{v\in\mathcal{V}} v \cdot \Pr\Big[Y_{m,q}^{k} = v\Big] = \mathbb{E}\Big[Y_{m,q}^{k}\Big].$$

Hence, we complete the proof of Proposition 3.

Acknowledgment

The authors would like to thank Kaige Qu and Wen Wu for many valuable suggestions throughout the work.

References

[1] X. Shen et al., "Toward immersive communications in 6G," *Front. Comput. Sci.*, vol. 4, pp. 1–45, Jan. 2023.

[2] F. Tang, X. Chen, M. Zhao, and N. Kato, "The roadmap of communication and networking in 6G for the Metaverse," *IEEE Wireless Commun.*, vol. 30, no. 4, pp. 72–81, Aug. 2022.

[3] Z. Liu et al., "Point cloud video streaming: Challenges and solutions," *IEEE Netw.*, vol. 35, no. 5, pp. 202–209, Sep./Oct. 2021.

[4] Y. Pei, M. Li, K. Qu, X. Huang, and X. S. Shen, "Viewing distance-aware volumetric video caching and rendering for XR services," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, 2024, pp. 1869–1874.

[5] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proc. IEEE*, vol. 107, no. 8, pp. 1738–1762, Aug. 2019.

[6] M. Li, J. Gao, C. Zhou, X. Shen, and W. Zhuang, "User dynamics-aware edge caching and computing for mobile virtual reality," *IEEE J. Sel. Topics Signal Process.*, vol. 17, no. 5, pp. 1131–1146, Sep. 2023.

[7] X. Huang, S. Hu, H. Yang, X. Wang, Y. Pei, and X. Shen, "Digital twin-based network management for better QoE in multicast short video streaming," *IEEE Trans. Wireless Commun.*, vol. 23, no. 11, pp. 16187–16202, Nov. 2024.

[8] A. Zhang, C. Wang, B. Han, and F. Qian, "YuZu: Neural-enhanced volumetric video streaming," in *Proc. 19th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2022, pp. 137–154.

[9] B. Han, Y. Liu, and F. Qian, "ViVo: Visibility-aware mobile volumetric video streaming," in *Proc. 26th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2020, pp. 1–13.

[10] M. Xing, S. Xiang, and L. Cai, "A real-time adaptive algorithm for video streaming over multiple wireless access networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 4, pp. 795–805, Apr. 2014.

[11] T. X. Tran and D. Pompili, "Adaptive bitrate video caching and processing in mobile-edge computing networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 9, pp. 1965–1978, Sep. 2019.

[12] X. Zuo, J. Yang, M. Wang, and Y. Cui, "Adaptive bitrate with user-level QoE preference for video streaming," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, 2022, pp. 1279–1288.

[13] J. Van Der Hooft et al., "A tutorial on immersive video delivery: From omnidirectional video to holography," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 2, pp. 1336–1375, 2nd Quart., 2023.

[14] Y. Li, C. Dou, Y. Wu, W. Jia, and R. Lu, "NOMA assisted two-tier VR content transmission: A tile-based approach for QoE optimization," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 3769–3784, May 2024.

[15] L. Wang, C. Li, W. Dai, S. Li, J. Zou, and H. Xiong, "QoE-driven adaptive streaming for point clouds," *IEEE Trans. Multimedia*, vol. 25, pp. 2543–2558, Feb. 2022.

[16] J. Van Der Hooft, T. Wauters, F. De Turck, C. Timmerer, and H. Hellwagner, "Towards 6DoF HTTP adaptive streaming through point cloud compression," in *Proc. 27th ACM Int. Conf. Multimedia (MM)*, 2019, pp. 2405–2413.

[17] B. Chai et al., "SDSR: Optimizing metaverse video streaming via saliency-driven dynamic super-resolution," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 4, pp. 978–989, Apr. 2024.

[18] L. Zhang, H. Guo, Y. Dong, F. Wang, L. Cui, and V. C. M. Leung, "Collaborative streaming and super resolution adaptation for mobile immersive videos," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, 2023, pp. 1–10.

[19] J. Park, P. A. Chou, and J.-N. Hwang, "Rate-utility optimized streaming of volumetric media for augmented reality," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 1, pp. 149–162, Mar. 2019.

[20] J. Li, C. Zhang, Z. Liu, R. Hong, and H. Hu, "Optimal volumetric video streaming with hybrid saliency based tiling," *IEEE Trans. Multimedia*, vol. 25, pp. 2939–2953, 2023.

[21] J. Liu et al., "CaV3: Cache-assisted viewport adaptive volumetric video streaming," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces*, 2023, pp. 173–183.

[22] Y. Liu, B. Han, F. Qian, A. Narayanan, and Z.-L. Zhang, "Vues: Practical mobile volumetric video streaming through multiview transcoding," in *Proc. 28th Annu. Int. Conf. Mobile Comput. Netw (MobiCom)*, 2022, pp. 514–527.

[23] Z. Fang et al., "PACP: Priority-aware collaborative perception for connected and autonomous vehicles," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 15003–15018, Dec. 2024.

[24] Z. Fang, S. Hu, J. Wang, Y. Deng, X. Chen, and Y. Fang, "Prioritized information bottleneck theoretic framework with distributed online learning for edge video analytics," *IEEE/ACM Trans. Netw.*, early access, Jan. 20, 2025, doi: 10.1109/TON.2025.3526148.

[25] S. Yuan et al., "Efficient online computing offloading for budget-constrained cloud-edge collaborative video streaming systems," *IEEE Trans. Cloud Comput.*, vol. 13, no. 1, pp. 273–287, Jan.–Mar. 2025.

[26] J. Li et al., "Toward optimal real-time volumetric video streaming: A rolling optimization and deep reinforcement learning based approach," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 12, pp. 7870–7883, Dec. 2023.

[27] Y. Zhu et al., "A semantic-aware transmission with adaptive control scheme for volumetric video service," *IEEE Trans. Multimedia*, vol. 25, pp. 7160–7172, 2023.

[28] "Support of 5G glass-type augmented reality / mixed reality (AR/MR) devices; Version 18.1.0,"3GPP, Sophia Antipolis, France, Rep. TR 26.998, Mar. 2024.

[29] C. Zhou, J. Gao, M. Li, N. Cheng, X. Shen, and W. Zhuang, "Digital-twin-based 3D map management for edge-assisted device pose tracking in mobile AR," *IEEE Internet Things J.*, vol. 11, no. 10, pp. 17812–17826, May 2024.

[30] K. Humadi, I. Trigui, W.-P. Zhu, and W. Ajib, "User-centric cluster design and analysis for hybrid sub-6GHz-mmWave-THz dense networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 7, pp. 7585–7598, Jul. 2022.

[31] Z. Luo and C. Wu, "An online algorithm for VNF service chain scaling in datacenters," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1061–1073, Jun. 2020.

[32] Z. Zhou, Q. Wu, and X. Chen, "Online orchestration of cross-edge service function chaining for cost-efficient edge computing," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 8, pp. 1866–1880, Aug. 2019.

[33] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[34] P. Raghavan and C. D. Tompson, "Randomized rounding: A technique for provably good algorithms and algorithmic proofs," *Combinatorica*, vol. 7, no. 4, pp. 365–374, Dec. 1987.

[35] L. Pu, L. Jiao, X. Chen, L. Wang, Q. Xie, and J. Xu, "Online resource allocation, content placement and request routing for cost-efficient edge caching in cloud radio access networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 8, pp. 1751–1767, Aug. 2018.

[36] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan, "Dependent rounding and its applications to approximation algorithms," *J. ACM*, vol. 53, no. 3, pp. 324–360, May 2006.

[37] "Online technical report." Accessed: Feb. 2025. [Online]. Available: https://www.dropbox.com/scl/fi/d0ss7i30tbpusehcgbsuk/Supporting_file-1.pdf?rlkey=602yxm0m5tk37x7mtba17kt2c&st=qow9mk0y&dl=0

[38] M. Grant and S. Boyd. "CVX: MATLAB software for disciplined convex programming, version 2.1." Mar. 2014. [Online]. Available: https://cvxr.com/cvx

[39] P. Li et al., "Filling the missing: Exploring generative AI for enhanced federated learning over heterogeneous mobile edge devices," *IEEE Trans. Mobile Comput.*, vol. 23, no. 10, pp. 10001–10015, Oct. 2024.

Authorized licensed use limited to: University of Waterloo. Downloaded on June 16,2025 at 15:26:53 UTC from IEEE Xplore. Restrictions apply.

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 research interests include network resource optimization, mobile-edge computing, and immersive communications.

Mushu Li (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2021.

She is an Assistant Professor with the Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA, USA. Prior to this, she was a Postdoctoral Fellow with Toronto Metropolitan University, Toronto, ON, Canada, from 2022 to 2024, and the University of Waterloo from 2021 to 2022. Her research interests include machine learning for communications and networking, cloud and multiaccess edge computing, Intelligent Internet of Things, and 6G wireless networks.

Dr. Li received the Best Land Transportation Paper Award from the IEEE Vehicular Technology Society in 2024, the NSERC Postdoctoral Fellowship in 2022, and the NSERC Canada Graduate Scholarship in 2018.

Xinyu Huang (Graduate Student Member, IEEE) received the B.E. degree from the Qian Xuesen Experimental Class, Xidian University, Xi'an, China, in 2018, and the M.S. degree in information and communications engineering from Xi'an Jiaotong University, Xi'an, in 2021. 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 agents, multimedia communications, and network resource management.

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

He is a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research focuses on network resource management, wireless network security, Internet of Things, AI for networks, and vehicular networks.

Dr. Shen received "West Lake Friendship Award" from Zhejiang Province in 2023, President's Excellence in Research from the University of Waterloo in 2022, the Canadian Award for Telecommunications Research from the Canadian Society of Information Theory 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 in 2019 and AHSN Technical Committee in 2013. He has also received the Excellent Graduate Supervision Award in 2006 from the University of Waterloo and the Premier's Research Excellence Award in 2003 from the Province of Ontario, Canada. He is the Past President of the IEEE Communications Society. He was the Vice President for Technical and Educational Activities and 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*. 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.

Authorized licensed use limited to: University of Waterloo. Downloaded on June 16,2025 at 15:26:53 UTC from IEEE Xplore. Restrictions apply.