Energy-Efficient Multi-User Adaptive 360° Video Streaming: A Two-Step Approach with Device Video Super-Resolution

Yannan Wei, *Graduate Student Member, IEEE*, Qiang (John) Ye, *Senior Member, IEEE*, Weihua Zhuang, *Fellow, IEEE*, and Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—In this paper, we propose a device energy-efficient two-step adaptive scheme for tile-based 360° video streaming to support enhanced multi-user viewing quality-of-experience (QoE) in a time-slotted system. Specifically, each video chunk is first prefetched based on the predictive field-of-view (FoV), and the FoV quality of the chunk is then enhanced at a closer-to-playback time instant based on the updated FoV prediction with improved accuracy. Both transmission-driven and device video super-resolution (VSR)-driven methods are adaptively selected to enable efficient video chunk enhancement. At each time slot, the incremental QoE gain of each user is characterized via a time-difference approach, based on which the best candidate chunk is determined. Then, a single-slot problem is formulated for maximizing the total incremental QoE gain while minimizing the total device energy consumption. A particle swarm optimization (PSO)-based iterative solution is proposed to obtain optimal bandwidth allocation, bitrate level selection, and enhancement method selection for multiple users. Extensive simulation results demonstrate that our proposed solution outperforms benchmark schemes in terms of average viewing QoE, average device energy consumption, and average utility.**

***Index Terms*—Tile-based 360° video streaming, FoV prediction accuracy, video super-resolution, incremental QoE gain, two-step adaptive streaming.**

I. Introduction

In comparison with traditional on-demand video streaming, 360° virtual reality (VR) video streaming enables more immersive and engaging viewing experience, which has attracted significant attention from both academia and industry vertical markets such as remote education and entertainment [1]–[3]. It is forecasted that the number of head-mounted displays (HMDs) and other types of mobile VR devices will reach over 112 million by 2026 [4]. Due to the large area of an omni-directional video content, a 360° video has the data size typically 4-6 times larger than a conventional video at an equivalent quality, which requires an extremely high end-to-end (E2E) transmission rate (e.g., 400 Mbps) for smooth high-resolution video playback and poses significant challenges to current mobile networks [5], [6]. Based on the fact that a user (or video client) watches only part of a 360° video at any time instant due to the limited span of an HMD (e.g., $100^\circ \times 100^\circ$), referred to as field-of-view (FoV), the tile-based adaptive 360° video streaming has been proposed and widely adopted to reduce the high demand on transmission rate [7]. Specifically, at the video server side, a spherical 360° video representation is first transformed into a planer format. Then, the projected 360° video is temporally divided into a sequence of video chunks, each of which is further spatially partitioned into multiple non-overlapping video tiles. Each video tile is encoded into multiple bitrate levels, corresponding to various quality versions, to support adaptive tile-based streaming. At the user side, head movements are tracked by the HMD for FoV prediction [8]. Based on the prediction results and the estimated transmission rate, video tiles are transmitted at different bitrate levels for each video chunk to enhance the viewing experience, where the video tiles covered by the predicted FoV are transmitted with high bitrate levels [9], [10].

The tile-based adaptive 360° video streaming suffers from transmission rate fluctuations and inevitable FoV prediction errors, which affect the user perceived FoV quality. Maintaining a large playback buffer occupancy ensures high video playback smoothness, whereas reducing the FoV prediction accuracy due to a large prediction time gap. In this case, the predictive FoV tiles assigned with high bitrate levels may not be actually watched by the user during the playback of the video chunk due to high FoV prediction errors, which leads to low transmission rate utilization and perceived FoV quality. Conversely, the FoV prediction accuracy becomes high when the playback buffer occupancy is small due to a short prediction time gap. Transmission resources can be efficiently utilized to deliver predictive FoV tiles with high bitrate levels, which can pose high risk of generating stalling events for video playback under transmission rate fluctuations.

There are many existing works addressing this dilemma, focusing on single-user 360° video streaming. For example, the predicted FoV of each video chunk is adaptively enlarged with extension tiles based on the recent FoV prediction accuracy to compensate for possible FoV prediction errors [11], [12]. In addition, various two-tier/layer streaming frameworks have been proposed, which account the impact of FoV prediction accuracy on user perceived FoV quality. Each panoramic video chunk is first prefetched with a basic quality to prioritize the video playback smoothness. Then, the FoV quality of each prefetched chunk is improved based on a more accu-

Yannan Wei, Weihua Zhuang, and Xuemin (Sherman) Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, N2L 3G1 (emails: {y272wei, wzhuang, sshen}@uwaterloo.ca).

Qiang Ye is with the Department of Electrical and Software Engineering, University of Calgary, Calgary, AB, Canada, T2N 1N4 (e-mail: qiang.ye@ucalgary.ca).

This work has been accepted in part at 2025 IEEE Globecom Workshops.

This work was supported by a research grant from the Natural Sciences and Engineering Research Council (NSERC) of Canada.

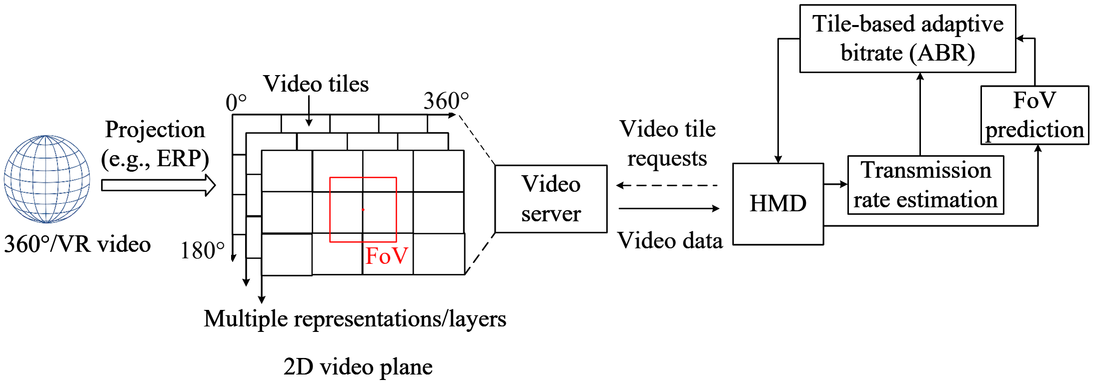


Fig. 1: A general E2E tile-based 360° video streaming system.

rate updated FoV prediction, which is performed when the playback buffer occupancy reaches a certain threshold or is adaptively enabled under transmission rate and playback buffer dynamics [13]–[16]. On the other hand, with increasingly powerful computing capability of an edge server or an end device, the technique of video super-resolution (VSR) has been incorporated into the tile-based adaptive 360° video streaming to further enhance the user viewing experience, which breaks the strong dependency between available transmission rate and user viewing quality-of-experience (QoE) [17], [18]. A remote video server can transmit low-resolution video tiles which are then reconstructed into higher-resolution versions (i.e., higher bitrate levels) by an edge server or a user device using pre-trained deep neural network (DNN)-based VSR models [19]–[21].

Different from the existing studies, in this paper, we consider multi-user 360° video streaming in a time-slotted system and propose a two-step adaptive streaming scheme to achieve device energy-efficient viewing QoE optimization. Each video chunk is first prefetched based on the predictive FoV and then enhanced at a closer-to-playback time instant based on the updated FoV with improved prediction accuracy. Comparing the predictive FoV with the updated FoV of a video chunk, two types of tiles, i.e., *miss-predicted tiles* and *hit tiles*, are respectively enhanced. The bitrate level of predictive FoV tiles in the chunk prefetching step and the enhanced bitrate levels of miss-predicted and hit tiles in the chunk enhancement step are adaptively selected based on the allocated radio resources and the playback buffer dynamics of a user. Transmission-driven and device-VSR-driven methods for chunk enhancement are proposed, which are adaptively enabled for both miss-predicted and hit tiles under device energy consumption constraints. In order to determine when to prefetch a new video chunk or enhance a prefetched chunk, a time-difference approach is applied to characterize the incremental QoE gain of each user at any slot, based on which the best candidate chunk is determined. The best candidate chunk of a user is a to-be-prefetched video chunk for chunk prefetching or a prefetched chunk for chunk enhancement. A single-shot problem is then formulated for maximizing the total incremental QoE gain while minimizing the total device energy consumption. A practical particle swarm optimization (PSO)-based iterative solution is developed to determine the per-slot optimal decisions of bandwidth allocation, bitrate level selection, and enhancement method selection for each user. The main contributions of this paper are summarized as follows.

- A two-step adaptive streaming scheme is proposed to achieve enhanced 360° video viewing QoE for multiple users, which captures the impact of FoV prediction accuracy on user perceived FoV quality. Both transmission-driven and device-VSR-driven methods are proposed and adaptively enabled to achieve device energy-efficient FoV quality enhancement;
- The incremental QoE gain of each user is characterized via a time-difference approach. The best candidate chunk of each user is then determined, which balances the trade-off between video playback smoothness and perceived FoV quality;
- A single-slot problem is formulated for maximizing the total incremental QoE gain while minimizing the total device energy consumption, which is decomposed into multiple independent per-user subproblems given bandwidth allocation decisions. A PSO-based iterative solution is developed to obtain the optimal bandwidth allocation, bitrate level selection, and enhancement method selection decisions for multiple users at each slot;
- Extensive simulation results based on real data traces are presented to validate the improved performance of our proposed solution over benchmark schemes.

The rest of this paper is organized as follows. Section II reviews the background and related work. The system model is provided in Section III. Section IV first presents the user viewing QoE characterization and then the problem formulation. The proposed PSO-based solution is presented in Section V. Simulation results are provided in Section VI. Section VII draws the conclusion. A list of important notations is given in Table I.

II. Background and Related Work

A. Background

Fig. 1 shows a general E2E tile-based 360° video streaming system with emphasis on the end sides. At the server side,

TABLE I: List of important notations

Symbol	Definition
τ, T_c	The slot and video chunk lengths
$\mathcal{I}_u$	The set of video chunks of user u
$\mathcal{J}$	The set of video tiles composing the panoramic video scene
$\mathcal{R}$	The set of encoding bitrate levels for a video tile
$u \in \mathcal{U}_t$	The set of users at slot t
$\mathcal{B}_{t,u}, O_{t,u}$	The playback buffer and the current playback buffer occupancy of user u at slot t
$A_{t,u,i}, e_{t,u,i}, f_{t,u,i}$	The FoV quality, enhancement status, and effective factor for video chunk i of user u at slot t
$r^P_{u,i}, \mathcal{J}^P_{u,i}$	The bitrate level and the set of predictive FoV tiles for video chunk i of user u
$\mathcal{J}^P_{t,u}, \mathcal{J}^M_{t,u,i}, \mathcal{J}^H_{t,u,i}$	The sets of predictive FoV tiles, miss-predicted tiles, and hit tiles when a new video chunk of user u is prefetched or when prefetched chunk i of user u is enhanced at slot t
$r^P_{t,u}, r^M_{t,u,i}, r^H_{t,u,i}$	The bitrate levels of predictive FoV tiles, miss-predicted tiles, and hit tiles when a new video chunk of user u is prefetched or when prefetched chunk i of user u is enhanced at slot t
$\rho^M_{t,u,i}, \rho^H_{t,u,i}$	The enhancement methods for miss-predicted tiles and hit tiles when prefetched chunk i of user u is enhanced at slot t
$w_{t,u}, \kappa_{t,u}$	The fraction of bandwidth allocated to user u at slot t and the corresponding transmission rate
$d^P_{t,u}, d^E_{t,u}$	The total experienced delay when a new video chunk of user u is prefetched or when a prefetched chunk of user u is enhanced at slot t
$\phi_u(r_b, r'_b)$	The time for reconstructing a video tile from bitrate level r_b to a higher bitrate level $r'_b \geq r_b$ by user device u
$Q^1_{t,u}, Q^2_{t,u}, Q^3_{t,u}$	The average perceived FoV quality, video stall time, and temporal quality smoothness of user u at slot t
$G^1_{t,u}, G^2_{t,u}, G^3_{t,u}$	The incremental gains in terms of different QoE components for user u at slot t
(α, β, γ)	The importance coefficients regarding different QoE components
$E_{t,u}$	The incremental average (normalized) device energy consumption of user u at slot t

each (spherical) 360° video is transformed into a planer format using projection techniques such as the equirectangular projection (ERP) and pyramid projection [22]. Each projected 360° video is temporally divided into a sequence of video chunks, each with a playback time of typically 1-10 seconds. Then, each chunk is spatially partitioned into multiple non-overlapping video tiles. Each tile is further independently encoded into multiple representations or interdependent layers, corresponding to different bitrate (or quality) levels, to support adaptive tile-based streaming under transmission rate variations [23], [24].

At the user side, head movements are tracked by the HMD for FoV prediction, which is a critical module in tile-based 360° video streaming. Many FoV prediction approaches have been proposed, which can be classified into three categories: trajectory-based, content-based, and hybrid. Trajectory-based approaches, such as (truncated) linear regression [14], [15], dead-reckoning (DR) [25], and deep learning (DL) models (e.g., recurrent neural network (RNN)) [26], predict a user's future FoV based on the user's historical head movement trajectories. Content-based approaches generate the saliency maps of a 360° video using DNNs to predict which parts of the video attract users the most [27]. Hybrid approaches consider both the saliency patterns of a 360° video and users' head movement traces in FoV prediction to achieve more accurate prediction results [28]–[30]. When a user requests a video chunk, the FoV of the chunk is first predicted. Then, based on the prediction results and the estimated E2E transmission rate, the user selectively requests a set of video tiles with different bitrate levels. Generally, the priority of a video tile being requested depends on its likelihood of being watched by the user. More transmission resources are allocated to deliver the video tiles covered by the predicted FoV than the peripheral tiles outside the predicted FoV. Nevertheless, as the FoV prediction and downloading of a video chunk are conducted before it is played, the predicted FoV may diverge from the real one due to the inevitable prediction errors caused by user viewing behavior dynamics, especially when the prediction time gap becomes larger for a large playback buffer occupancy. A more comprehensive description and discussion of E2E tile-based 360° video streaming, including video pre-processing, distribution, and streaming, can be found in [1].

B. Related Work

Tile-based 360° video streaming suffers from transmission rate fluctuations and inevitable FoV predictions. Specifically, under transmission rate variations, a user needs to adaptively select bitrate levels for requested video tiles to balance between video playback smoothness and video quality [6], [35]–[37]. Moreover, transmission resources are not efficiently utilized when the FoV prediction accuracy is low, as the predictive FoV tiles assigned with high bitrate levels may not be actually watched by a user, leading to low perceived FoV quality. Hence, an adaptive streaming scheme, which considers the interplay among video playback smoothness, perceived FoV quality, and FoV prediction accuracy, is critical for supporting good 360° video viewing experience.

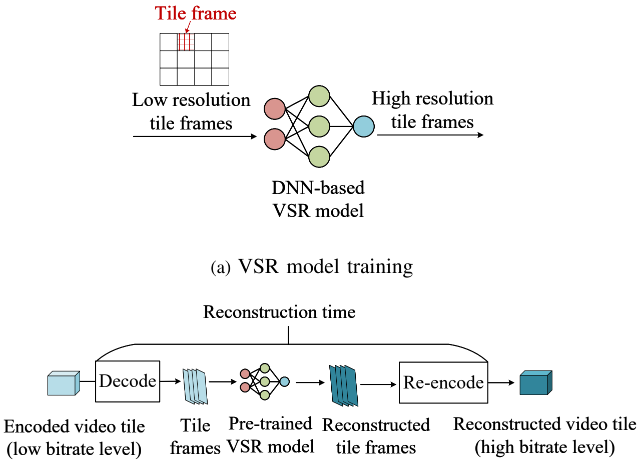
Many works have been done to achieve enhanced user viewing QoE for tile-based 360° video streaming, focusing on single-user case, as shown in Table II. For example, multiple edge-assisted tile-based 360° video streaming frameworks have been proposed [38]. An edge server can cache some (popular) video tiles and deliver them to the intended users with reduced latency [31], [39]. Besides, a cached video tile with a high bitrate level can be transcoded by the edge server into various lower bitrate level versions. In the case when

TABLE II: Comparison among different adaptive streaming schemes for tile-based 360° video streaming

Representative schemes	One/Two-step/tier	VSR assistance	Edge assistance	Main idea(s)
Pano [6]	One-step	×	×	Enhanced quality model; Variable-sized tiling
[31]	One-step	×	√	Edge caching in heterogeneous networks
CUCBSC [32]	One-step	×	√	Edge caching with switching cost considered
DRL-CTCS [33]	One-step	×	√	Collaborative edge caching and transcoding
CSE [34]	One-step	√	×	Device-VSR
SDSR [19]	One-step	√	×	Device-VSR; VSR model selection
ECCSR [21]	One-step	√	√	Edge caching; Collaborative edge- and device-VSR
RAPT360 [12]	One-step	×	×	Probabilistic FoV prediction to accommodate possible FoV predictions
[11]	One-step	×	×	The predicted FoV adaptively appended with extension tiles to reduce FoV prediction errors
[14]	Two-tier	×	×	Always prefetching a new chunk with the lowest bitrate level to prioritize video playback smoothness. Enhancing the FoV quality of a prefetched chunk only when the playback buffer occupancy reaches a preset upper-bound
RAM [16]	Two-tier	×	×	Two-tier streaming with adaptive switching between chunk prefetching with the lowest bitrate level and FoV quality update of a prefetched chunk; Adaptive selection of the prefetched chunk for FoV quality update; Transmission-driven FoV quality update

multiple users request the same 360° video, instead of caching multiple bitrate level versions of a video tile, the video tile with a high bitrate level can be cached at the edge server to save the caching space, at the cost of additional transcoding delay [33]. Therefore, in an edge-assisted tile-based 360° video streaming system, the joint design of adaptive edge caching placement, edge computing resource allocation, and bitrate level selection is crucial [29], [40], [41]. Particularly, the switching cost for caching replacement needs to be considered [32]. For a newly cached video tile, it can be either directly downloaded from a remote server with extra transmission delay or transcoded from a higher bitrate level version of the same tile already cached in the edge server with extra transcoding delay. Besides, for all newly cached video tiles, the edge server can download from the remote server(s) and locally transcode simultaneously to reduce the total switching delay.

Recent advancements in VSR techniques open another promising direction to further enhance video quality, while breaking the strong dependency between user viewing QoE and available transmission rate [17]. Leveraging the computing capability of an edge server and/or a user device, a video chunk or tile with a low bitrate level can be reconstructed into a higher bitrate level version with improved quality. The VSR technique has been exploited and widely investigated in traditional video streaming [42]–[49]. Furthermore, multiple lightweight and scalable DNN-based VSR models, customized for the tile-based 360° video streaming service, have been developed [18], [20], [34], [50]–[53]. As shown in Fig. 2, typically, to train a VSR model offline, a pair of low-resolution and high-resolution tile frames (i.e., a video tile in a frame of a 360° video) of the same video tile is used as the input and output of the VSR model as one training data sample. Each VSR model is trained to learn how to reconstruct a tile frame from a low-resolution one to a high-resolution one (e.g., 240p → 480p). During the inference stage, the reconstruction of an encoded video tile usually goes through the sequential steps of decoding, reconstruction, and re-encoding, which requires certain computations and incurs additional reconstruction time. Many works have investigated VSR-assisted tile-based adaptive 360° video streaming by leveraging the computing capability of an edge server and/or a user device [18]–[21], [34], [54]. A video tile with a low bitrate level can be transmitted by a remote video server first and then reconstructed into a target enhanced bitrate level by the edge server or the user device before being put into the playback buffer. Specifically, when video tile reconstruction is performed at the edge server (*edge-VSR*), the backhaul transmission between the server and the edge server is reduced, whereas enlarging the wireless transmission time due to the larger size of a reconstructed tile. When video tile reconstruction is performed at the user device (*device-VSR*), the E2E transmission time is reduced


Tile frame
Low resolution tile frames
High resolution tile frames
DNN-based VSR model
Reconstruction time
Decode
Re-encode
Encoded video tile (low bitrate level)
Tile frames
Pre-trained VSR model
Reconstructed tile frames
Reconstructed video tile (high bitrate level)

(a) VSR model training

(b) VSR model inference

Fig. 2: DNN-based VSR model training and inference in tile-based 360° video streaming.

at the cost of device energy consumption, which is essential to consider for mobile 360° video delivery due to the limited battery capacity of an HMD. Therefore, in a VSR-assisted tile-based 360° video streaming system, it is critical to adaptively determine the pair of download and reconstructed bitrate levels for each requested video tile, as well as the location for the video tile reconstruction (i.e., edge-VSR or device-VSR), to balance transmission time and reconstruction time while considering device energy consumption.

On the other hand, many previous works have explicitly considered the impact of FoV prediction accuracy on user perceived FoV quality in tile-based adaptive streaming scheme design. For example, probabilistic FoV prediction methods have been proposed to accommodate possible FoV predictions [12], [55]. Extension tiles are adaptively appended to the predicted FoV based on the recent prediction accuracy to form an extended FoV [11]. In addition, some works focus on the playback buffer management for an accurate FoV prediction. The maximum playback buffer length of a video player (i.e., HMD) is set to a small value (e.g., 1-3s) to guarantee high FoV prediction accuracy with a short prediction time gap. To balance between video playback smoothness and perceived FoV quality, multiple two-tier streaming schemes have been proposed [14]–[16], [55]. During the streaming, each panoramic video chunk is first prefetched with the lowest bitrate level (i.e., base tier) to prioritize the video playback smoothness. Then, the updated FoV tiles of a prefetched chunk are downloaded with a higher bitrate level (i.e., enhancement tier) to improve the FoV quality, which is performed when the playback buffer occupancy reaches a preset upper-bound, or is adaptively enabled under transmission rate and playback buffer dynamics.

Different from the existing studies, in this paper, we consider multi-user tile-based 360° video streaming and propose a two-step adaptive streaming scheme for multi-user viewing QoE optimization. Our core idea is that each video chunk is first prefetched based on the predictive FoV and then enhanced at a closer-to-playback time instant based on the updated and more accurate FoV. Specifically, the switching between chunk prefetching and enhancement is adaptively determined for each user to balance between video playback smoothness and perceived FoV quality. Transmission-driven and device-VSR-driven methods are proposed and adaptively enabled to achieve device energy-efficient FoV quality enhancement. The bitrate level of predictive FoV tiles in the chunk prefetching step and the bitrate levels of updated FoV tiles in the chunk enhancement step are adaptively selected for different users under network and playback buffer dynamics to achieve efficient resource utilization for enhanced viewing experience.

III. System Model

A. Network Scenario

We consider a multi-user on-demand 360° video streaming service, as shown in Fig. 3, where time is slotted with constant duration τ. Multiple video users under the coverage of a base station (BS) request 360° videos asynchronously which may have different video lengths. Each user's streaming request is initiated at the beginning of a slot. In each slot, we consider only active users who have an ongoing 360° video streaming session. Let $\mathcal{U}_t$ be the set of users at slot t. For each user $u \in \mathcal{U}_t$, the playback buffer at slot t is denoted by $\mathcal{B}_{t,u}$, which contains the indexes of all prefetched video chunks to be played at slot t, and the playback buffer occupancy at slot t is given by $O_{t,u}$ seconds[1]. The BS is pre-configured with a total radio spectrum bandwidth of W MHz for wireless video data transmission.

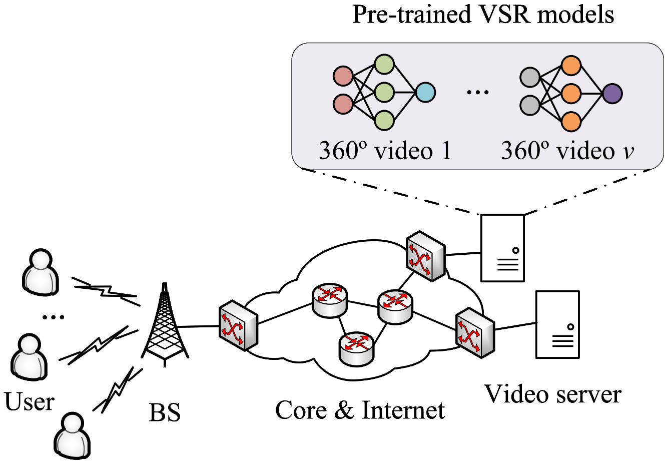


Fig. 3: The considered network scenario.

Content-aware VSR is considered to support video quality enhancement [18]. A particular DNN-based VSR model is pre-trained for each 360° video and stored in a remote server with the 360° video. When a user initiates a 360° video streaming request to a remote server, the server sends back the media presentation description (MPD) file and the pre-trained VSR model of the requested 360° video to the user. Due to the small sizes of an MPD file and a VSR model, e.g., 1 MB, the time for transmitting the MPD file and the pre-trained VSR model at the initial stage of a 360° video streaming session is neglected. Each VSR model leverages the computing capability of a user device (i.e., HMD) to reconstruct a video tile from a low-resolution (or low bitrate level) to a higher-resolution (or higher bitrate level), which incurs additional reconstruction time and device energy consumption. Let p_u be the device electricity power (in Watt) of user u. Due to the small focused area of a video tile and the inherent overfitting property of DNN, near-zero training error can be achieved, and a pre-trained VSR model for a particular 360° video can learn the features that map a low-resolution video tile to a higher-resolution one [20]. Thus, a video tile with an original bitrate level r_b' achieves the same perceived quality as the same tile that is reconstructed from a lower bitrate level to the bitrate level r_b' [19], [42].

B. Video Traffic Model

As shown in Fig. 4, each 360° video is temporally divided into multiple consecutive video chunks, with each video chunk

[1] Throughout this paper, we assume that all the mathematical sets are finite and their elements are sorted in an ascending order.

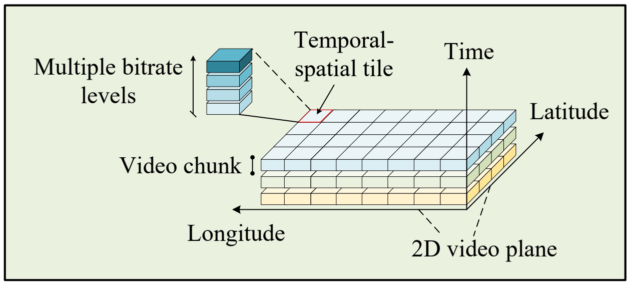


Fig. 4: Video traffic model.

of T_c seconds. The duration of a time slot is smaller than that of a video chunk (i.e., $\tau < T_c$), and T_c is assumed to be divisible by τ. Let $\mathcal{I}_u = \{1, 2, \cdots, |\mathcal{I}_u|\}$ be the set of video chunks of user u, indexed by $i \in \mathcal{I}_u$. Each video chunk is further spatially partitioned into a set of non-overlapping video tiles (e.g., 4×6), denoted by $\mathcal{J} = \{1, 2, \cdots, |\mathcal{J}|\}$ and indexed by $j \in \mathcal{J}$. To support adaptive tile-based 360° video streaming, constant bitrate (CBR) format is considered for video tile encoding [16], [56]. Each video tile is encoded into multiple bitrate levels denoted by $\mathcal{R} = \left\{r_1, r_2, \cdots, r_{|\mathcal{R}|}\right\}$ and indexed by $r_b \in \mathcal{R}$, corresponding to different video tile resolutions. A video tile with a higher bitrate level has a higher quality. Let $\mathcal{R}^\dagger = \{r_0\} \cup \mathcal{R}$ with $r_0 = 0$.

C. Two-step Adaptive Streaming Framework

Considering the interplay among video playback smoothness, perceived FoV quality, and FoV prediction accuracy in tile-based 360° video streaming, we propose a two-step adaptive streaming approach, as shown in Fig. 5. Each video chunk can go through two sequential steps, i.e., *chunk prefetching* and *enhancement*. In the prefetching step, a new panoramic chunk is prefetched, where the predictive FoV tiles are downloaded with a selected bitrate level, and the peripheral tiles outside the predictive FoV are downloaded with a basic bitrate level (i.e., r_1) to avoid blank screen. In the enhancement step, a prefetched chunk is selected to enhance the FoV quality based on the updated and more accurate FoV. With the two-step approach, video playback smoothness is improved by prefetching new video chunks, while the FoV quality of each prefetched chunk is enhanced with the updated FoV that is more likely to be watched by a user. For simplicity, we consider that each video chunk is enhanced at most once. Comparing the predictive FoV in the prefetching step and the updated FoV in the enhancement step, there are two types of video tiles, i.e., *miss-predicted tiles* and *hit tiles*. Miss-predicted tiles are those covered by the updated FoV but not by the predictive FoV of a video chunk and thus have a basic quality. Hit tiles are the ones covered by both the predictive and updated FoVs of a video chunk. As shown in Fig. 6, we consider transmission-driven and device-VSR-driven methods for the chunk enhancement. In the transmission-driven method, miss-predicted or hit tiles with an enhanced bitrate level are directly transmitted from a remote server to its intended user. In the device-VSR-driven method, the prefetched miss-predicted or hit tiles are locally reconstructed to a target enhanced bitrate level by the user device.

In order to track the streaming performance over time for a user, two auxiliary vectors are defined for user u at slot t: $\mathbf{A}_{t,u} = \left(A_{t,u,i}, i \in \mathcal{I}_u\right), u \in \mathcal{U}_t$ for the (average) FoV quality of each chunk, where $A_{t,u,i}$ denotes the FoV quality of chunk i at slot t; $\mathbf{e}_{t,u} = \left(e_{t,u,i}, i \in \mathcal{I}_u\right), u \in \mathcal{U}_t$ for the chunk enhancement status, where $e_{t,u,i} = 1$ indicates video chunk i has been enhanced before slot t and 0 otherwise. If a new video chunk is prefetched, the chunk index is given by $i^{\#} = \min\limits_{i \in \mathcal{I}_u} \{i, A_{t,u,i} = 0\}$. Let $\mathcal{J}^P_{t,u}$ be the set of predictive FoV tiles when a new video chunk is prefetched, and the assigned bitrate level of the predictive FoV tiles is denoted by $r^P_{t,u}$. Moreover, let $r^P_{u,i}$ and $\mathcal{J}^P_{u,i}$ record the bitrate level and the set of predictive FoV tiles, respectively, when video chunk i is prefetched. Thus, when video chunk $i^{\#}$ is prefetched, we have $r^P_{u,i^{\#}} = r^P_{t,u}$ and $\mathcal{J}^P_{u,i^{\#}} = \mathcal{J}^P_{t,u}$. Then, $A_{t+1,u,i^{\#}}$ is updated by

$$A_{t+1,u,i^{\#}} = q(r^P_{t,u}) \tag{1}$$

where $q\left(r_b\right) = \frac{r_b}{r_{|\mathcal{R}|}}$ [21].

When prefetched chunk $i \in \mathcal{B}_{t,u}$ in the current playback buffer is enhanced, the updated FoV of prefetched chunk i is denoted by $\mathcal{J}^E_{t,u,i}$. Comparing the predictive FoV when video chunk i is prefetched, i.e., $\mathcal{J}^P_{u,i}$, with the updated FoV, the sets of miss-predicted and hit tiles are denoted as $\mathcal{J}^M_{t,u,i} = \mathcal{J}^E_{t,u,i} \backslash \mathcal{J}^P_{u,i}$ and $\mathcal{J}^H_{t,u,i} = \mathcal{J}^E_{t,u,i} \cap \mathcal{J}^P_{u,i}$, respectively. Let $r^M_{t,u,i}$ and $r^H_{t,u,i}$ be the enhanced bitrate levels of miss-predicted and hit tiles, respectively, and we have

$$\begin{array}{l} r^M_{t,u,i} \geq r_1 \\ r^H_{t,u,i} \geq r^P_{u,i}. \end{array} \tag{2}$$

Then, we have $e_{t+1,u,i} = 1$, and the FoV quality of prefetched chunk i, i.e., $A_{t+1,u,i}$, is updated by

$$A_{t+1,u,i} = \frac{\left|\mathcal{J}^M_{t,u,i}\right| q(r^M_{t,u,i}) + \left|\mathcal{J}^H_{t,u,i}\right| q(r^H_{t,u,i})}{\left|\mathcal{J}^E_{t,u,i}\right|}. \tag{3}$$

Moreover, let $\rho^M_{t,u,i} = \{0, 1\}$ and $\rho^H_{t,u,i} = \{0, 1\}$ be the enhancement method indicators for the miss-predicted and hit tiles of prefetched chunk i, respectively, where $\rho^M_{t,u,i}/\rho^H_{t,u,i} = 0$ for the transmission-driven method and $\rho^M_{t,u,i}/\rho^H_{t,u,i} = 1$ for the device-VSR-driven method.

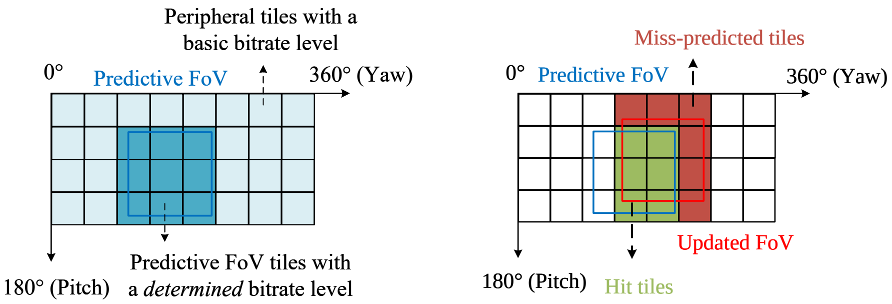
D. Communication Model

At each time slot, the BS allocates orthogonal bandwidth among users to support chunk prefetching and/or transmission-driven chunk enhancement. Let $w_{t,u} = [0, 1], u \in \mathcal{U}_t$ be the fraction of bandwidth allocated to user u at slot t, with the wireless transmission rate $\kappa_{t,u}$ given by

$$\kappa_{t,u} = w_{t,u} W \log_2(1 + \xi_{t,u}) \tag{4}$$

where $\xi_{t,u}$ denotes the signal-to-noise ratio (SNR). We assume that the wireless link between the BS and a user is the bottleneck during E2E video data transmission from a remote server to the user.

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2025.3617381



(a) Prefetching a new panoramic chunk (b) Enhancing the FoV quality of a prefetched chunk

Fig. 5: The proposed two-step adaptive streaming framework.

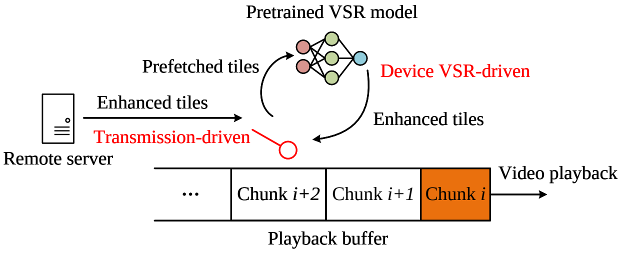


Fig. 6: Enhancement methods.

The total fraction of bandwidth allocated to all users at any slot t should not exceed 1, given by

$$\sum_{u\in\mathcal{U}_t} w_{t,u} \leq 1, \forall t. \tag{5}$$

E. Delay Model

Next, we analyze the total experienced delay for user $u \in \mathcal{U}_t$ when a new video chunk is prefetched or when a prefetched chunk is enhanced at slot t. When a new video chunk is prefetched, the delay for transmitting the chunk from a remote server to the user, denoted by $d^P_{t,u}$, is given by

$$d^P_{t,u} = \mathbf{1}\left(r^P_{t,u} \neq r_0\right) \frac{\left|\mathcal{J}\backslash\mathcal{J}^P_{t,u}\right| s(r_1) + \left|\mathcal{J}^P_{t,u}\right| s(r^P_{t,u})}{\kappa_{t,u}} \tag{6}$$

where $\mathbf{1}\left(a\right)$ is an identity function which equals 1 if condition a is true and 0 otherwise and $s\left(\cdot\right)$ returns the size in Mbits of a video tile given a bitrate level r_b with $s\left(r_b\right) = r_b T_c$. In the numerator of (6), the first and the second terms represent the total sizes of peripheral tiles and predictive FoV tiles, respectively.

When prefetched chunk i is enhanced, the experienced delay, denoted by $d^E_{t,u}$, depends on both the enhancement methods and the enhanced bitrate levels for the miss-predicted and hit tiles, given by

$$d^E_{t,u} = \mathbf{1}(\rho^H_{t,u,i} = \rho^M_{t,u,i}) \cdot d^{E,A}_{t,u} + \mathbf{1}(\rho^H_{t,u,i} \neq \rho^M_{t,u,i}) \cdot d^{E,B}_{t,u} \tag{7}$$

where $d^{E,A}_{t,u}$ and $d^{E,B}_{t,u}$ are given in (8) and (9), respectively. In (7) and (8), $d^{E,A}_{t,u}$ calculates the total delay when the miss-predicted and hit tiles are enhanced via the same method, $d^{E,B}_{t,u}$ calculates the total delay when the miss-predicted and hit tiles are enhanced via different methods in parallel, and $\phi_u(r_b, r'_b)$ represents the time for reconstructing a video tile from bitrate level r_b to a higher bitrate level r'_b by user device, where $r'_b \geq r_b$ with $\phi_u(r_b, r_b) = 0$ [34].

IV. Problem Formulation

A. User Viewing QoE

User viewing QoE consists of three components: average perceived FoV quality, average video stall time, and average temporal quality smoothness across all video chunks of a user [10], [57], [58]. The average temporal quality smoothness captures the average perceived FoV quality variations between consecutive video chunks. Formulating user viewing QoE is challenging: First, with the proposed two-step adaptive streaming approach, each video chunk can go through the two sequential steps of prefetching and enhancement, which happen at two different time slots. Second, the average perceived FoV quality and the average temporal quality smoothness are two QoE components that are measured on the basis of video chunks, whereas the average video stall time is a QoE component measured based on the dynamics of a user's playback buffer over time. In the following, we leverage a time-difference approach to analyze each QoE component individually.

(1) Average Perceived FoV Quality

Let $Q^1_{t,u,i}, i \in \mathcal{I}_u$ be the perceived FoV quality for video chunk i of u at slot t with FoV quality $A_{t,u,i}$. The impact of FoV prediction accuracy on the FoV quality perceived by a user needs to be considered. In the chunk prefetching step, a high bitrate level can be assigned to predictive FoV tiles when the current playback buffer occupancy of a user is small, that is when the FoV prediction accuracy is high with a small prediction time gap. On the other hand, it is more effective to enhance a prefetched chunk closer to its playback time, as the updated FoV is more accurate and more likely to be the actual FoV a user will watch during the playback of the chunk. We define an effective factor vector for user u, denoted by $\mathbf{f}_{t,u} = (f_{t,u,i}, i \in \mathcal{I}_u)$, to characterize the impact of FoV prediction accuracy when a new video chunk is prefetched or a prefetched chunk is enhanced, which is updated over time.

$$\begin{aligned} d_{t,u}^{E,A} = (1-\rho_{t,u,i}^{H}) &\left(\frac{\mathbf{1}(r_{t,u,i}^{M} > r_1) \left|\mathcal{J}_{t,u,i}^{M}\right| s(r_{t,u,i}^{M}) + \mathbf{1}(r_{t,u,i}^{H} > r_{u,i}^{P}) \left|\mathcal{J}_{t,u,i}^{H}\right| s(r_{t,u,i}^{H})}{\kappa_{t,u}} \right) \\ &+ \rho_{t,u,i}^{H} \left(\left|\mathcal{J}_{t,u,i}^{M}\right| \phi_u(r_1, r_{t,u,i}^{M}) + \left|\mathcal{J}_{t,u,i}^{H}\right| \phi_u(r_{u,i}^{P}, r_{t,u,i}^{H}) \right) \end{aligned} \tag{8}$$

$$d_{t,u}^{E,B} = \max \left[\begin{array}{l} (1-\rho_{t,u,i}^{H}) \frac{\mathbf{1}(r_{t,u,i}^{H} > r_{u,i}^{P}) \left|\mathcal{J}_{t,u,i}^{H}\right| s(r_{t,u,i}^{H})}{\kappa_{t,u}}, \rho_{t,u,i}^{H} \left|\mathcal{J}_{t,u,i}^{H}\right| \phi_u(r_{u,i}^{P}, r_{t,u,i}^{H}), \\ (1-\rho_{t,u,i}^{M}) \frac{\mathbf{1}(r_{t,u,i}^{M} > r_1) \left|\mathcal{J}_{t,u,i}^{M}\right| s(r_{t,u,i}^{M})}{\kappa_{t,u}}, \rho_{t,u,i}^{M} \left|\mathcal{J}_{t,u,i}^{M}\right| \phi_u(r_1, r_{t,u,i}^{M}) \end{array} \right] \tag{9}$$

Specifically, when a new video chunk $i^{\#}$ is prefetched at slot t, $f_{t+1,u,i^{\#}}$ is updates as

$$f_{t+1,u,i^{\#}} = 1 - \frac{O_{t,u}}{B} \tag{10}$$

where B is a constant for normalization.

When prefetched chunk $i \in \mathcal{B}_{t,u}$ is enhanced, $f_{t+1,u,i}$ is updated as

$$f_{t+1,u,i} = 1 - \frac{\left(\mathcal{B}_{t,u}^{-1}(i) - 1\right)^{+} T_c + \varepsilon_{t,u}}{B} \tag{11}$$

where $(x)^{+} = \max(x, 0)$, $\varepsilon_{t,u}$ denotes the unplayed time of the video chunk currently being played, and $\mathcal{B}_{t,u}^{-1}(i)$ indicates a reverse function that returns the index of element i in $\mathcal{B}_{t,u}$.

Then, the perceived FoV quality for video chunk i is given by $Q_{t,u,i}^{1} = A_{t,u,i} f_{t,u,i}$. The average perceived FoV quality, denoted by $Q_{t,u}^{1}$, is given by

$$Q_{t,u}^{1} = \frac{1}{|\mathcal{I}_u|} \sum_{i \in \mathcal{I}_u} Q_{t,u,i}^{1}, u \in \mathcal{U}_t. \tag{12}$$

(2) Average Video Stall Time

Let $Q_{t,u}^{2}$ be the average video stall time of user u at slot t. When a new video chunk of user u is prefetched at slot t, the video stall time at slot t is given by $(\tau - O_{t,u})^{+}$, and the average video stall time is updated as $Q_{t+1,u}^{2} = Q_{t,u}^{2} + \frac{1}{|\mathcal{I}_u|} (\tau - O_{t,u})^{+}$. The playback buffer occupancy is updated as

$$O_{t+1,u} = (O_{t,u} - \tau)^{+} + T_c. \tag{13}$$

When prefetched chunk $i \in \mathcal{B}_{t,u}$ is enhanced, the video stall time is 0 since FoV quality enhancement does not affect the smooth playback of the chunk, thus $Q_{t+1,u}^{2} = Q_{t,u}^{2}$. The playback buffer occupancy is updated by

$$O_{t+1,u} = (O_{t,u} - \tau)^{+}. \tag{14}$$

Finally, when neither chunk prefetching nor enhancement is performed, the video stall time at slot t is $(\tau - O_{t,u})^{+}$, and the average video stall time is updated by $Q_{t+1,u}^{2} = Q_{t,u}^{2} + \frac{1}{|\mathcal{I}_u|} (\tau - O_{t,u})^{+}$. The playback buffer occupancy is updated as $O_{t+1,u} = (O_{t,u} - \tau)^{+}$.

(3) Average Temporal Quality Smoothness

The average temporal quality smoothness of user $u \in \mathcal{U}_t$ at slot t, denoted by $Q_{t,u}^{3}$, is given by

$$Q_{t,u}^{3} = \frac{1}{|\mathcal{I}_u| - 1} \sum_{i=2}^{|\mathcal{I}_u|} \left| Q_{t,u,i}^{1} - Q_{t,u,i-1}^{1} \right|. \tag{15}$$

Let $G_{t,u}^{1}$, $G_{t,u}^{2}$, and $G_{t,u}^{3}$ represent the incremental gains in terms of different QoE components of user $u \in \mathcal{U}_t$ at slot t, respectively, given by

$$\begin{cases} G_{t,u}^{1} = Q_{t+1,u}^{1} - Q_{t,u}^{1} \\ G_{t,u}^{2} = Q_{t+1,u}^{2} - Q_{t,u}^{2} \\ G_{t,u}^{3} = Q_{t+1,u}^{3} - Q_{t,u}^{3}. \end{cases} \tag{16}$$

The incremental QoE gain of user $u \in \mathcal{U}_t$ at slot t is given by

$$G_{t,u} = \alpha G_{t,u}^{1} - \beta G_{t,u}^{2} - \gamma G_{t,u}^{3} \tag{17}$$

where $\alpha, \beta, \gamma \in [0, 1]$ are the importance coefficients regarding different QoE components, which can be flexibly adjusted by preference. The viewing QoE of user u is then given by $\sum_{t} \mathbf{1}(u \in \mathcal{U}_t) G_{t,u}$.

B. Best Candidate Chunk

As a comprehensive evaluation of viewing experience, it is challenging to optimize the user viewing QoE due to dynamics of user viewing behaviors (i.e., changing viewing orientation and the number of video tiles covered by an FoV) and video streaming requests (i.e., the time instant when a streaming request is initiated and the requested 360° video). In the following, we develop a solution to enhance multi-user viewing QoE under user device energy constraints.

Fig. 7 shows an example playback buffer of a user $u \in \mathcal{U}_t$ at slot t, where chunk i is the video chunk currently being played, chunks $i+1$ - $i+3$ are prefetched chunks that have not been played, and chunk $i+4$ is the to-be-prefetched chunk. The prefetched chunks and the to-be-prefetched chunk are referred to as *'candidate chunks'*. Our main idea is to first estimate the achieved largest possible incremental QoE gain, denoted by $\Delta_{t,u,i}$, for each candidate chunk. Specifically, for each candidate chunk i, the achieved largest possible perceived FoV quality, denoted by $\hat{Q}_{t+1,u,i}^{1}$, is estimated based on the following two principles: 1) The incremental gain in terms of average perceived FoV quality must be no less than the penalty of average temporal quality smoothness, which is given in (18). Note that only the adjacent chunks that have already been enhanced before slot t are accounted for the

$$f\left(Q^1_{t+1,u,i}\right) = \alpha\frac{1}{|\mathcal{I}_u|}(Q^1_{t+1,u,i} - Q^1_{t,u,i}) - \frac{1}{|\mathcal{I}_u|-1}\gamma\left[\begin{array}{l}\left(\left|Q^1_{t+1,u,i} - Q^1_{t+1,u,i-1}\right| - \left|Q^1_{t,u,i} - Q^1_{t,u,i-1}\right|\right)e_{t,u,i-1}\\ + \left(\left|Q^1_{t+1,u,i+1} - Q^1_{t+1,u,i}\right| - \left|Q^1_{t,u,i+1} - Q^1_{t,u,i}\right|\right)e_{t,u,i+1}\end{array}\right] \geq 0 \quad (18)$$

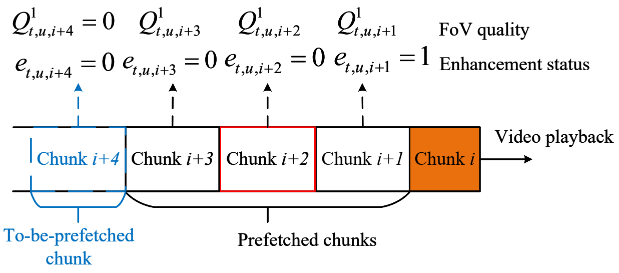


Fig. 7: An example playback buffer of a user $u \in \mathcal{U}_t$ at slot t.

penalty of average temporal quality smoothness, since the adjacent chunks that have not been enhanced at slot t (i.e., $e_{t,u,i} = 0$) have the chance of being enhanced in future slots. For example, in Fig. 7, when prefetched chunk $i+2$ is enhanced at slot t, only the adjacent chunk $i+1$ with $e_{t,u,i+1} = 1$ is considered in the estimation of the achieved largest possible perceived FoV quality for chunk $i+2$; 2) For each candidate chunk i, the perceived FoV quality must satisfy $Q^1_{t,u,i} \leq Q^1_{t+1,u,i} \leq f_{t+1,u,i}$. Then, the perceived FoV quality of each candidate chunk i can be maximized as in $(\mathbf{P}_0)^2$.

$$(\mathbf{P}_0): \max_{Q^1_{t+1,u,i}} \quad f\left(Q^1_{t+1,u,i}\right) \quad (19)$$

$$\text{s.t. } Q^1_{t,u,i} \leq Q^1_{t+1,u,i} \leq f_{t+1,u,i} \quad (20)$$

$$f\left(Q^1_{t+1,u,i}\right) \geq 0. \quad (21)$$

Proposition 1. *The optimum of* $(\mathbf{P}_0)$ *is obtained at the boundary values of* $Q^1_{t+1,u,i}$.

Proof. The proof of Proposition 1 is provided in Appendix. □

According to Proposition 1, the largest perceived FoV quality for each candidate chunk i of user u at slot t, i.e., $\hat{Q}^1_{t+1,u,i}$, is estimated. Then, the best candidate chunk that achieves the largest incremental QoE gain is given by

$$i^*_{t,u} = \operatorname*{arcmax}_i \Delta_{t,u,i} = \operatorname*{arcmax}_i \left\{ f\left(\hat{Q}^1_{t+1,u,i}\right) - \beta\frac{1}{|\mathcal{I}_u|}\left(\tau - O_{t,u}\right)^+ \right\}. \quad (22)$$

If the best candidate chunk is the to-be-prefetched chunk, then chunk prefetching is performed at slot t, and the user is termed as a *prefetching user*. Otherwise, chunk enhancement is performed at slot t, and the user is termed as an *enhancement user*. Hereafter, we use i instead of i^* to indicate the best candidate chunk for brevity.

C. Problem Formulation

A single-slot problem is formulated to allocate bandwidth among multiple users, select the bitrate level of predictive FoV tiles for the best candidate chunk of each prefetching user, and determine the enhanced bitrate levels and enhancement methods of the miss-predicted and hit tiles for the best candidate chunk of each enhancement user. Let $\mathcal{U}^P_t$ and $\mathcal{U}^E_t$ be the sets of prefetching and enhancement users at slot t, respectively. The total delay for chunk prefetching or enhancement needs to be smaller than the slot length, given by

$$d^P_{t,u} \leq \tau, u \in \mathcal{U}^P_t \quad (23)$$

$$d^E_{t,u} \leq \tau, u \in \mathcal{U}^E_t. \quad (24)$$

Moreover, device energy is consumed when the best candidate chunk of an enhancement user is enhanced through the device-VSR-driven method, which is critical to consider due to an HMD's limited battery capacity. Let $E_{t,u}$ be the incremental average (normalized) device energy consumption of user u at slot t, given in (25), where $E_{t,u} = 0$ for any $u \in \mathcal{U}^P_t$. At each slot, the objective is to maximize the total incremental QoE gain while minimizing the total incremental average device energy consumption to achieve energy-efficient multi-user viewing QoE optimization. The single-slot problem is formulated as

$$(\mathbf{P}_1): \max_{\substack{w_{t,u}, r^P_{t,u}, \\ r^H_{t,u,i}, r^M_{t,u,i}, \rho^H_{t,u,i}, \rho^M_{t,u,i}}} \sum_{u \in \mathcal{U}^P_t} \Psi^P_u + \sum_{u \in \mathcal{U}^E_t} \Psi^E_u \quad (26)$$

$$\text{s.t. } (5), (23), (24)$$

$$r^P_{t,u} \in \mathcal{R}^\dagger, u \in \mathcal{U}^P_t \quad (27)$$

$$r^H_{t,u,i} \in \mathcal{R}, r^H_{t,u,i} \geq r^P_{u,i}, u \in \mathcal{U}^E_t \quad (28)$$

$$r^M_{t,u,i} \in \mathcal{R}, r^M_{t,u,i} \geq r_1, u \in \mathcal{U}^E_t \quad (29)$$

$$\rho^H_{t,u,i}, \rho^M_{t,u,i} = \{0, 1\}, u \in \mathcal{U}^E_t \quad (30)$$

where $\Psi^P_u = f\left(Q^1_{t+1,u,i}\right) - \beta\frac{1}{|\mathcal{I}_u|}\left(\tau - O_{t,u}\right)^+$, $\Psi^E_u = f\left(Q^1_{t+1,u,i}\right) - \beta\frac{1}{|\mathcal{I}_u|}\left(\tau - O_{t,u}\right)^+ - \delta E_{t,u}$, and $\delta = [0, 1]$ is the importance coefficient regarding device energy consumption.

Note that $(\mathbf{P}_1)$ is a mixed-integer non-linear programming (MINLP), which is NP-hard [59]. In the next section, we develop a PSO-based iterative solution to solve the problem.

V. PSO-Based Iterative Solution

A. Problem Decomposition

Given the bandwidth allocation decisions $w_{t,u}, u \in \mathcal{U}_t$, the bitrate level and/or enhancement method selection decisions for a user are independent of those for the other users. Thus, $(\mathbf{P}_1)$ can be decomposed into multiple independent per-user subproblems, expressed as $(\mathbf{P}_2-1)$ and $(\mathbf{P}_2-2)$, where $(\mathbf{P}_2-1)$ represents the subproblem for each prefetching user $u \in \mathcal{U}^P_t$

[2] A tunable parameter, $\epsilon > 0$, can be introduced in (19) as $f\left(Q^1_{t+1,u,i}\right) - \epsilon$ for prefetched chunk $i \in \mathcal{B}_{t,u}$ to prioritize video playback smoothness over perceived FoV quality.

$$E_{t,u} = \frac{\rho^M_{t,u,i}\left|\mathcal{J}^M_{t,u,i}\right|\phi(r_1, r^M_{t,u,i}) + \rho^H_{t,u,i}\left|\mathcal{J}^H_{t,u,i}\right|\phi(r^P_{u,i}, r^H_{t,u,i})}{\tau\left|\mathcal{I}_u\right|} \tag{25}$$

at slot t, and $(\mathbf{P}_2 - 2)$ represents the subproblem for each enhancement user $u \in \mathcal{U}^E_t$ at slot t.

$$(\mathbf{P}_2{-}1): \max_{r^P_{t,u}} \ \Psi^P_u \tag{31}$$

$$\text{s.t. } (23), (27)$$

$$(\mathbf{P}_2{-}2): \max_{r^H_{t,u,i}, r^M_{t,u,i}, \rho^H_{t,u,i}, \rho^M_{t,u,i}} \ \Psi^E_u \tag{32}$$

$$\text{s.t. } (24), (28), (29), (30)$$

Given $w_{t,u}, u \in \mathcal{U}_t$, the objective of $(\mathbf{P}_1)$ is obtained by solving the subproblem of each user according to the user type (i.e., prefetching or enhancement user). As both $(\mathbf{P}_2 - 1)$ and $(\mathbf{P}_2 - 2)$ have only discrete decision variables and the size of the bitrate level set $|\mathcal{R}|$ is small (e.g., 5), the sizes of the solution spaces for $(\mathbf{P}_2 - 1)$ and $(\mathbf{P}_2 - 2)$ are $\left|\mathcal{R}^\dagger\right|$ and $4\,|\mathcal{R}| \times |\mathcal{R}|$, respectively. Therefore, we enumerate all feasible combinations of decisions and find the one that obtains the maximal objective value to determine the optimal decisions of $(\mathbf{P}_2 - 1)$ and $(\mathbf{P}_2 - 2)$.

B. Algorithm Design

We propose a PSO-based solution to solve $(\mathbf{P}_1)$, where the bandwidth allocation decisions are iteratively optimized through the PSO algorithm, and the bitrate level and/or the enhancement method selection decisions are made by solving the subproblem of each user (i.e., $(\mathbf{P}_2 - 1)$ or $(\mathbf{P}_2 - 2)$). The flowchart of the proposed PSO-based solution is shown in Fig. 8. Specifically, we consider a particle swarm of size N. The position of particle n, denoted by $\mathbf{x}_n = \left(x_{n,1}, x_{n,2}, \cdots, x_{n,|\mathcal{U}_t|}\right)$, is a multi-dimensional vector with dimension $D = |\mathcal{U}_t|$, representing a feasible bandwidth allocation among users, i.e., $\mathbf{x}_n = \{w_{t,u}, u \in \mathcal{U}_t\}$. Let K be the total number of iterations. The inertia weight is denoted by ϖ. Two acceleration coefficients, also termed as cognitive and social factors, are denoted by c_1 and c_2, respectively. In the initialization stage of the PSO algorithm, the positions of N particles are first initialized with the Dirichlet distribution, which meets the constraint (5). The velocity of particle n, denoted by $\mathbf{v}_n = \left(v_{n,1}, v_{n,2}, \cdots, v_{n,|\mathcal{U}_t|}\right)$, is initialized with each dimension $v_{n,d} \in [v_{\min}, v_{\max}]$, where $v_{\min}$ and $v_{\max}$ are the minimal and maximal velocities, respectively. The personal best position and fitness value (i.e., the objective value of $(\mathbf{P}_1)$) of each particle are initialized using its initial position and the corresponding fitness value, respectively. The global best position and fitness value within the swarm are initialized by the particle that has the largest personal best fitness value. Let the algorithm iteration index $k = 1$.

For each iteration k, the position and velocity of particle n are updated as (*Step 1*)

$$v^k_{n,d} = \varpi^k v^{k-1}_{n,d} + c_1\theta_1\left(x^P_{n,d} - x^{k-1}_{n,d}\right) + c_2\theta_2\left(x^G_d - x^{k-1}_{n,d}\right) \tag{33}$$

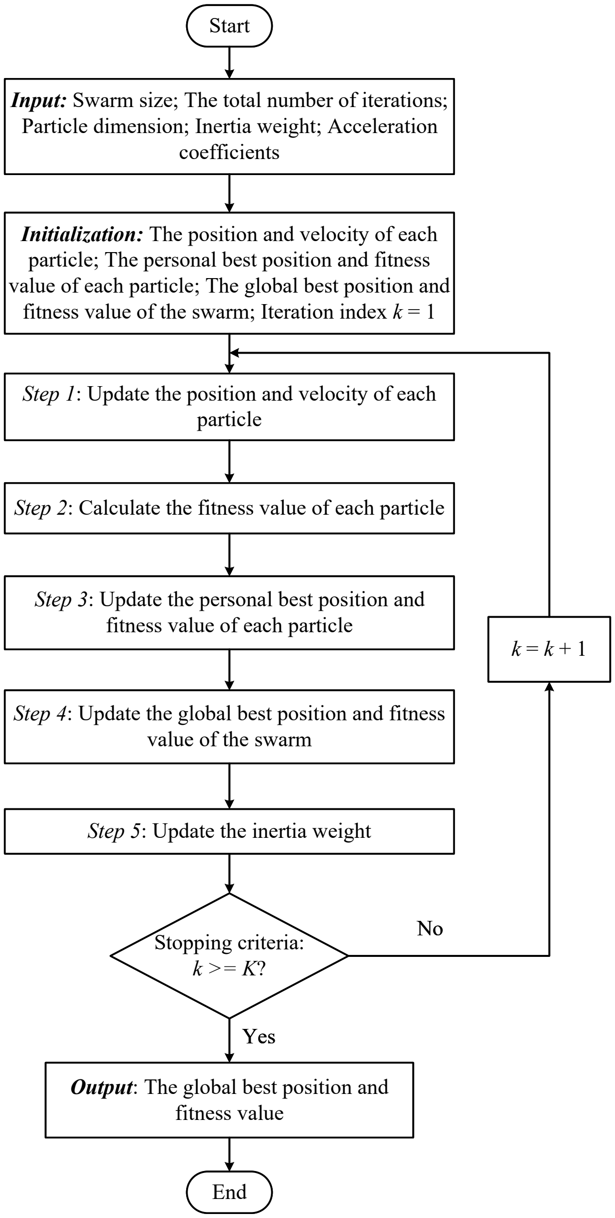

Start
Input: Swarm size; The total number of iterations; Particle dimension; Inertia weight; Acceleration coefficients
Initialization: The position and velocity of each particle; The personal best position and fitness value of each particle; The global best position and fitness value of the swarm; Iteration index k = 1
Step 1: Update the position and velocity of each particle
Step 2: Calculate the fitness value of each particle
Step 3: Update the personal best position and fitness value of each particle
Step 4: Update the global best position and fitness value of the swarm
Step 5: Update the inertia weight
Stopping criteria: k >= K?
No
$k = k + 1$
Yes
Output: The global best position and fitness value
End

Fig. 8: The flowchart of the proposed PSO-based solution.

$$x^k_{n,d} = \left(x^{k-1}_{n,d} + v^k_{n,d}\right)^+ \tag{34}$$

where $\theta_1, \theta_2 \in [0, 1]$ are two random numbers, $x^P_{n,d}$ is the d-th dimension of the personal best position of particle n, and x^G_d is the d-th dimension of the global best position within the swarm. If $\sum_{d=1}^{|\mathcal{U}_t|} x^k_{n,d} > 1$, $x^k_{n,d}$ is projected onto $x^k_{n,d} / \sum_{d=1}^{|\mathcal{U}_t|} x^k_{n,d}$ to satisfy the constraint (5).

Next, the fitness value of each particle is calculated, and the personal best position and fitness value are updated (*Steps*

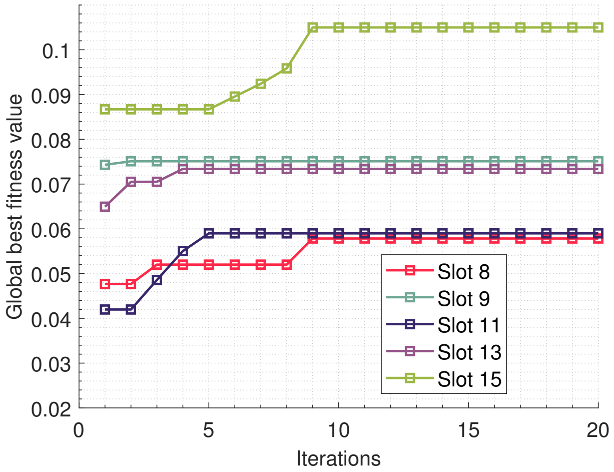


Fig. 9: The convergence performance of the proposed PSO-based solution at sample slots.

Algorithm 1: PSO-based Two-Step Adaptive 360° Video Streaming

```
1 Input: $\mathcal{U}_t$, $\mathcal{B}_{t,u}$, $O_{t,u}$, $\mathbf{A}_{t,u}$, $\mathbf{e}_{t,u}$, $\mathbf{f}_{t,u}$, $\mathcal{J}^P_{t,u}$, $\mathcal{J}^E_{t,u,i}$, $r^P_{u,i}$, $\mathcal{J}^P_{u,i}$
2 for slot $t$ do
3     Determine the best candidate chunk of each user $u \in \mathcal{U}_t$
4     Obtain the optimal bandwidth allocation decisions $w_{t,u}, u \in \mathcal{U}_t$
5     Solve $(\mathbf{P}_2 - 1)$ or $(\mathbf{P}_2 - 2)$ for each user $u \in \mathcal{U}_t$ based on the allocated bandwidth $w_{t,u}$ and the user type
6     Update $\mathbf{A}_{t,u}$, $\mathbf{e}_{t,u}$, $\mathbf{f}_{t,u}$, $r^P_{u,i}$, $\mathcal{J}^P_{u,i}$
7 Output: $w_{t,u}$, $r^P_{t,u}$, $r^H_{t,u,i}$, $r^M_{t,u,i}$, $\rho^H_{t,u,i}$, $\rho^M_{t,u,i}$
```

TABLE III: Main simulation parameters

Parameters	Values
Video chunk length T_c	1 s
Time slot length τ	0.25 s
Transmit power of the BS	43 dBm
Pass loss exponent	3.4
Noise power	10^{-13} W
Video tiling layout	4×8
FoV	$100^\circ \times 100^\circ$
Video chunk bitrate levels	$[2.5, 5, 8, 16, 40]$ Mbps
Importance coefficients (α, β, γ)	$\left(\frac{2}{3}, \frac{1}{6}, \frac{1}{6}\right)$, $\left(\frac{1}{6}, \frac{2}{3}, \frac{1}{6}\right)$, or $\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$
The minimal and maximal inertia weights $(\varpi_{\min}, \varpi_{\max})$	$(0.4, 1)$
Acceleration coefficients (c_1, c_2)	$(1.5, 1.5)$

2 and *3*). Then, the global best position and fitness value of the swarm are updated (*Step 4*). Finally, inertia weight ϖ is updated with an adaptive linear decreasing method to balance the global exploration and local exploitation capabilities of particles within the swarm (*Step 5*), given by

$$\varpi^k = \varpi_{\max} - (\varpi_{\max} - \varpi_{\min}) \frac{k}{K} \tag{35}$$

where $\varpi_{\min}$ and $\varpi_{\max}$ denote the minimal and maximal inertia weights, respectively. The global best position and fitness value of the swarm are output when the number of iterations reaches K.

Fig. 9 shows the convergence performance of the proposed PSO-based iterative solution at sample slots [60], [61]. The proposed PSO-based two-step algorithm is summarized in Algorithm 1. At slot t, the best candidate chunk of each user is first determined (*line 3*). Next, the proposed PSO-based solution is applied to obtain the optimal bandwidth allocation decisions, i.e., $w_{t,u}, u \in \mathcal{U}_t$ (*line 4*). Then, based on the allocated bandwidth $w_{t,u}$, the subproblem $(\mathbf{P}_2-1)$ or $(\mathbf{P}_2-2)$ is solved for user u based on the user type to obtain the optimal bitrate level and/or the enhancement method selection decisions, i.e., $r^P_{t,u}$ or $(r^H_{t,u,i}, r^M_{t,u,i}, \rho^H_{t,u,i}, \rho^M_{t,u,i})$ (*line 5*). At the end of each slot, $\mathbf{A}_{t,u}$, $\mathbf{e}_{t,u}$, $\mathbf{f}_{t,u}$, $r^P_{u,i}$, and $\mathcal{J}^P_{u,i}$ are updated for each user based on the determined optimal decisions (*line 6*). For the practical implementation of Algorithm 1, at any slot, each user first determines the best candidate chunk. Then, the current streaming performance and the related information of the best candidate chunk (e.g., the set of predictive FoV tiles or the sets of miss-predicted and hit tiles) are sent from the user to the BS. The BS determines the bandwidth allocation among different users. Finally, each user determines the other decisions based on the allocated bandwidth and user type, and sends a corresponding video content request to the intended video server.

The time complexity of Algorithm 1 consists of two parts. The best candidate chunk of each user at any slot t is first determined, which requires a running time of $O\left(|\mathcal{B}_{t,u}|\,|\mathcal{U}_t|\right)$. Then, the proposed PSO-based solution is applied, consuming at most $O\left(NK\,|\mathcal{U}_t|\,|\mathcal{R}|^2\right)$ time. Overall, the time complexity of Algorithm 1 is $O\left(|\mathcal{B}_{t,u}|\,|\mathcal{U}_t|\right) + O\left(NK\,|\mathcal{U}_t|\,|\mathcal{R}|^2\right)$.

VI. Performance Evaluation

In this section, simulation results are presented to validate the performance of our proposed two-step adaptive streaming scheme based on real data traces[3] [62]. All the experiments are conducted using Python 3.10.0 and FFmpeg[4]. FFmpeg is leveraged for video cropping (i.e., video chunk and tile), encoding, and FoV stitching, while tile-based adaptive 360° video streaming is simulated using Python. The main simulation parameters are given in Table III.

A. Data Trace Preprocessing

The selected data traces contain ten 360° videos and the viewing orientations of 50 subjects for each 360° video. The viewing trajectories are given in radian in terms of yaw (from $-\pi$ to π) and pitch (from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$).

As shown in Fig. 10, we consider a 4×8 tiling layout and an FoV of $100^\circ \times 100^\circ$. The panoramic video scene is spatially partitioned into 32 video tiles, each of which covers

[3] https://github.com/360VidStr/A-large-dataset-of-360-video-user-behaviour
[4] https://ffmpeg.org/

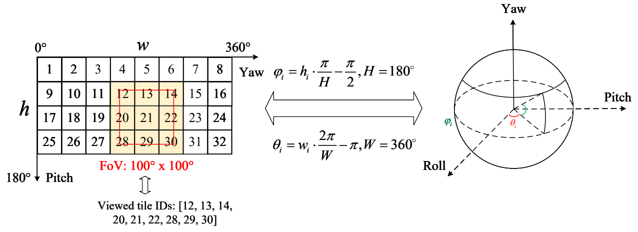
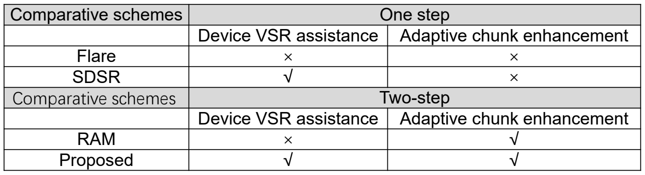

0°
w
360°
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
h
Yaw
180°
Pitch
FoV: 100° x 100°
Viewed tile IDs: [12, 13, 14, 20, 21, 22, 28, 29, 30]
$\varphi_i = h_i \cdot \frac{\pi}{H} - \frac{\pi}{2}, H = 180^\circ$
$\theta_i = w_i \cdot \frac{2\pi}{W} - \pi, W = 360^\circ$
Yaw
Pitch
Roll
φ_i
θ_i

Fig. 10: The considered video tiling layout with sphere-to-plane coordinate transformation.



Comparative schemes	One step	
	Device VSR assistance	Adaptive chunk enhancement
Flare	×	×
SDSR	√	×
Comparative schemes	Two-step	
	Device VSR assistance	Adaptive chunk enhancement
RAM	×	√
Proposed	√	√

Fig. 11: The considered benchmark schemes for performance comparison.

a $45^\circ \times 45^\circ$ square viewing span and is indexed in raster-scan order [9]. For a predicted viewpoint (i.e., yaw and pitch), we first conduct sphere-to-plane coordinate transformation, which maps the predicted yaw (θ_i) and pitch (φ_i) in the spherical coordinate system to the horizontal ($\omega_i \in [0^\circ, 360^\circ]$) and vertical ($h_i \in [0^\circ, 180^\circ]$) coordinates in the projected 2D video plane [7]. Then, the set of tiles covered by the predicted FoV is obtained.

B. Simulation Settings

We consider that $|\mathcal{U}|$ users are uniformly distributed within the coverage (500 m) of a BS and asynchronously request one of the two 360° videos (3840×2160) selected from the real data traces. The transmit power of the BS is 43 dBm. The channel gain between the BS and a user consists of path loss, log-normal shadow fading with a standard deviation of 8 dB, and Rayleigh-distributed fast fading with scale parameter $\frac{\sqrt{2}}{2}$ [63], [64]. Video chunk length $T_c = 1$ s, and the slot length $\tau = 0.25$ s [21], [54]. Following the recommended encoding settings by YouTube[5], each video chunk can be encoded into five bitrate levels: 480p (2.5 Mbps), 720p (5 Mbps), 1080p (8 Mbps), 2K (16 Mbps), and 4K (40 Mbps). Each video tile consumes $\frac{1}{32}$ bitrate of a video chunk [19]. Linear regression (LR) is adopted for FoV prediction using the recent past 3-s viewing orientations (i.e., pitch and yaw) at a sampling frequency of 5 Hz [16], [54]. According to empirical data, the importance coefficient regarding device energy consumption δ is set to 0.04.

We consider three existing streaming schemes as benchmarks for performance comparison, termed as Flare, SDSR, and RAM, respectively, as shown in Fig. 11. Specifically,

- Flare [65]: A typical tile-based streaming scheme. Video chunks are progressively downloaded based on the predictive FoVs;
- SDSR [19]: A device-VSR-assisted streaming scheme. The predictive FoV tiles of each video chunk are first downloaded and then locally enhanced by the user device. A pair of download and enhanced bitrate levels is adaptively determined to balance transmission time with reconstruction time. Device energy consumption is not considered;
- RAM [16]: A two-layer streaming scheme. At each slot, either a new video chunk is prefetched with a basic quality or a prefetched chunk in the current playback buffer is selected to be enhanced via only the transmission-driven method.

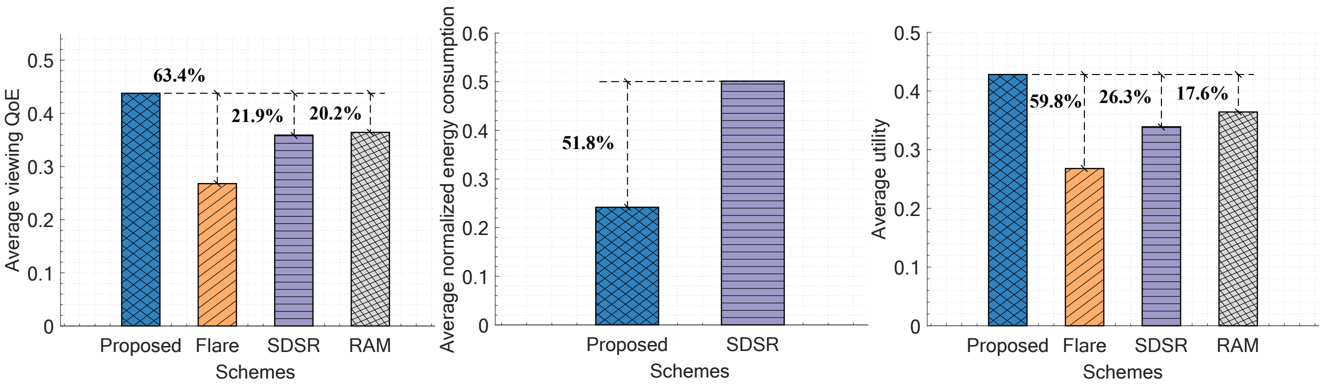
C. Simulation Results

Fig. 12 - Fig. 14 show the performance of different schemes in terms of average viewing QoE, average normalized (device) energy consumption, and average utility, which incorporates both user viewing QoE and device energy consumption, under various QoE objectives. The number of users is 4, and the total bandwidth of the BS is 4 MHz. The size of the particle swarm is 30, and the total number of iterations is 100. We consider three sets of configurations for importance coefficients (α, β, γ) to represent different QoE objectives, i.e., $(2/3, 1/6, 1/6)$ for prioritizing the average perceived FoV quality, $(1/6, 2/3, 1/6)$ for prioritizing the average video stall time, and $(1/3, 1/3, 1/3)$ for comprehensive consideration [20], [21].

Our observation on the performance comparison are as follows: First, SDSR achieves higher average viewing QoE than Flare due to the device-VSR-assisted (predictive) FoV quality enhancement when the total bandwidth of the BS is insufficient; Second, RAM achieves higher average viewing QoE than Flare due to the two-layer streaming scheme. Each video chunk's FoV quality is enhanced more effectively with a more accurate updated FoV, which is limited by the insufficient total bandwidth and may incur large temporal quality variations. In addition, each video chunk is prefetched with only a basic quality while neglecting the playback buffer dynamics. Bandwidth resources may not be efficiently utilized, especially when the allocated bandwidth to a user is large and the current playback buffer occupancy is small (i.e., the FoV prediction accuracy is high); Third, our proposed two-step adaptive streaming scheme achieves the highest average viewing QoE among the benchmarks and less average device energy consumption than SDSR under different QoE objectives. The proposed scheme captures the impact of FoV prediction accuracy on user perceived FoV quality. Bandwidth resources are efficiently utilized through adaptive bandwidth allocation among users under network condition and playback buffer dynamics. Device energy efficiency is achieved via adaptive enhancement method selection. Therefore, the proposed scheme outperforms the benchmarks and achieves the highest average utility under various QoE objectives.

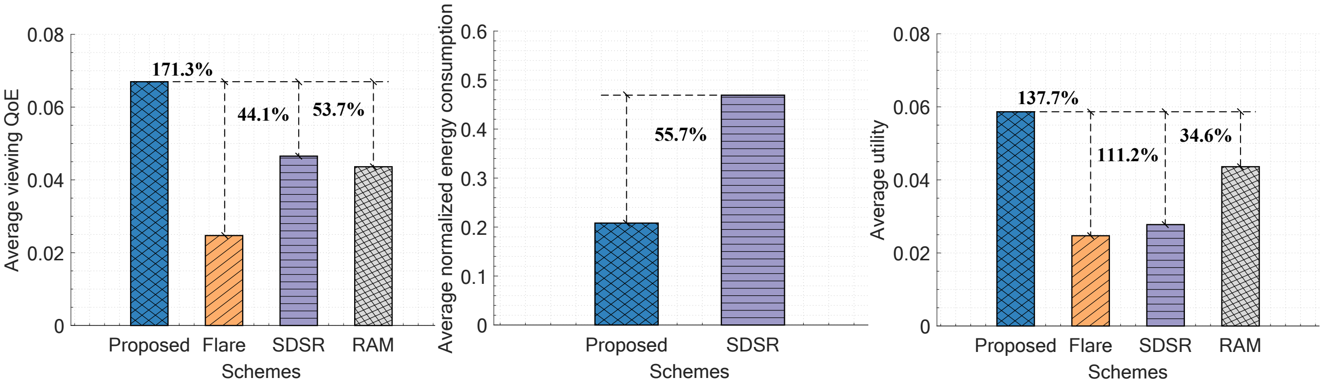
Fig. 15(a) - Fig. 15(c) show the average viewing QoE, average normalized energy consumption, and average utility of different schemes with varying total bandwidth of the BS, respectively. The number of users is 3 and $(\alpha, \beta, \gamma) = (1/3, 1/3, 1/3)$. The size of the particle swarm is 10, and the total number of iterations is 50. It can be seen that

[5] https://support.google.com/youtube/answer/1722171?hl=en



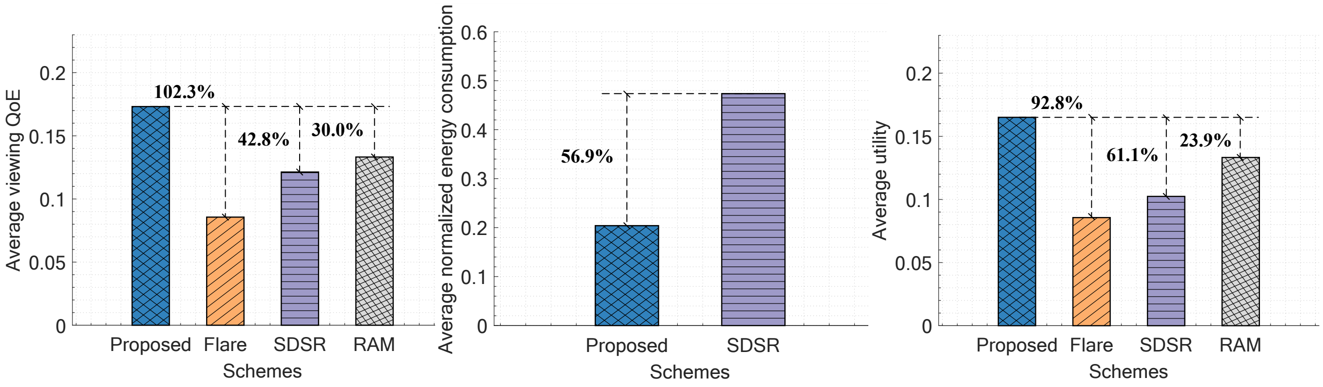
(a) Average viewing QoE (b) Average normalized energy consumption (c) Average utility

Fig. 12: Performance of different schemes with $(\alpha, \beta, \gamma) = (2/3, 1/6, 1/6)$.



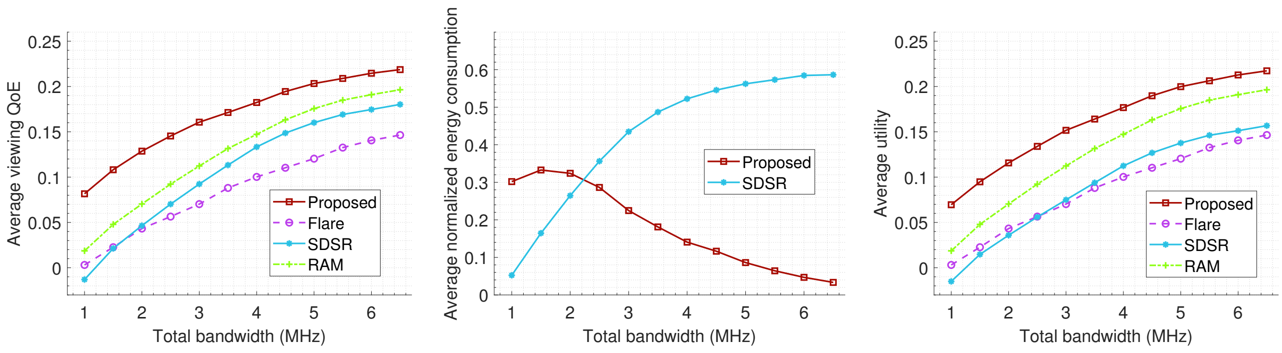
(a) Average viewing QoE (b) Average normalized energy consumption (c) Average utility

Fig. 13: Performance of different schemes with $(\alpha, \beta, \gamma) = (1/6, 2/3, 1/6)$.



(a) Average viewing QoE (b) Average normalized energy consumption (c) Average utility

Fig. 14: Performance of different schemes with $(\alpha, \beta, \gamma) = (1/3, 1/3, 1/3)$.



(a) Average viewing QoE (b) Average normalized energy consumption (c) Average utility

Fig. 15: Performance of different schemes with varying total bandwidth of the BS.

as the total bandwidth increases, the average viewing QoE and utility under different streaming schemes increase, where our proposed two-step adaptive scheme achieves the highest average viewing QoE and utility compared to the benchmarks. The average normalized energy consumption under our proposed scheme decreases with the total bandwidth, whereas the average normalized energy consumption increases with the total bandwidth under SDSR. SDSR and Flare are both one-step streaming schemes, which overlook the impact of FoV prediction accuracy on user perceived FoV quality. SDSR fully relies on the device-VSR-driven method for FoV quality enhancement without considering the device energy consumption, thus resulting in inefficient bandwidth utilization and large device energy consumption, especially when the total bandwidth is large. The performance of RAM is limited by the total bandwidth, as it fully relies on the transmission-driven method for FoV quality enhancement. Each video chunk is prefetched with only a basic quality even if the total bandwidth is large, leading to inefficient bandwidth usage.

As the total bandwidth of the BS increases, a larger bitrate level is selected for predictive FoV tiles when prefetching a new video chunk, which leads to smaller reconstruction time for enhancing a video tile to a target bitrate level with the device-VSR-driven method. More video tiles are likely to be enhanced with the transmission-driven method. The computing loads for device-VSR-driven chunk enhancement are reduced. Therefore, higher average viewing QoE and less device energy consumption are achieved with the proposed scheme when the total bandwidth increases.

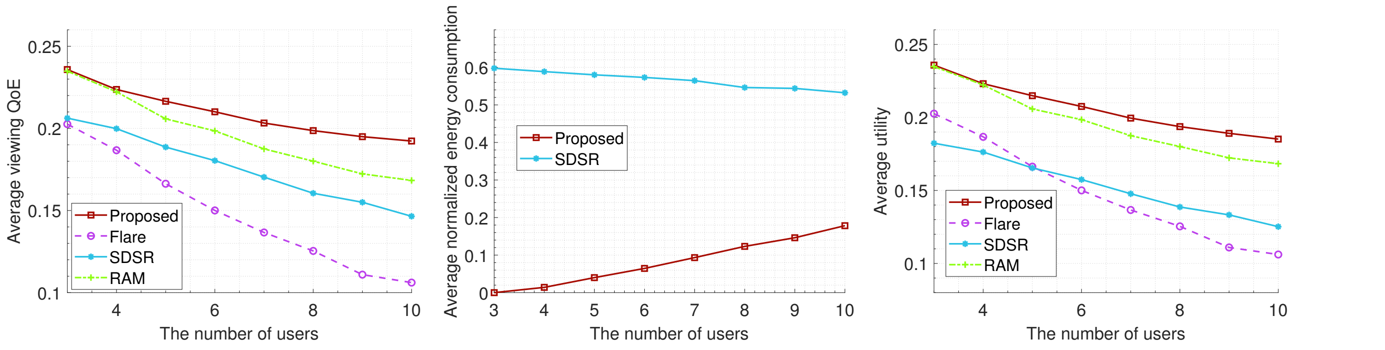
Fig. 16(a) - Fig. 16(c) show the average viewing QoE, average normalized energy consumption, and average utility of different schemes with varying numbers of users, respectively. The total bandwidth of the BS is 10 MHz and $(\alpha, \beta, \gamma) = (1/3, 1/3, 1/3)$. The size of the particle swarm is 30, and the total number of iterations is 100. We can see that as the number of users increases, the average viewing QoE and average utility under different schemes decrease, where our proposed two-step adaptive scheme achieves the highest average viewing QoE and utility compared to the benchmarks. The average normalized energy consumption under the proposed scheme increases with the number of users and is smaller than that under SDSR. For SDSR, when the number of users increases, it costs more time, including transmission and reconstruction time, to deliver a video tile with a given pair of download and enhanced bitrate levels. A smaller pair of download and enhanced bitrate levels is selected to satisfy the delay constraint, thus leading to less device energy consumption. In addition, as discussed earlier, Flare and SDSR neglect the impact of FoV prediction accuracy on user perceived FoV quality, and RAM depends on the total bandwidth, thus resulting in worse performance than our proposed scheme.

With the proposed scheme, when more users share the total bandwidth of the BS, a smaller bitrate level is selected for predictive FoV tiles when prefetching a new video chunk, which leads to larger reconstruction time for enhancing a video tile to a target enhanced bitrate level with the device-VSR-driven method. The achieved FoV quality improvement is much constrained with the transmission-driven method. More video tiles are likely to be enhanced via the device-VSR-driven method, resulting in higher computing loads. Thus, more device energy is consumed. On the other hand, the FoV quality of each video chunk is enhanced effectively based on the updated FoV with improved prediction accuracy. A best candidate chunk is determined for each user in any slot to achieve the largest total incremental QoE gain with efficient bandwidth usage. The transmission-driven and device-VSR-driven methods are adaptively enabled for FoV quality enhancement under network and playback buffer dynamics to achieve the device energy efficiency. Therefore, our proposed scheme achieves higher average viewing QoE and utility than the benchmarks under different numbers of users.

To provide visualization of the performance improvement with our proposed streaming scheme, Fig. 17 and Fig. 18 show snapshot images from the generated demo videos for comparison of perceived FoV qualities using the proposed scheme and one of the benchmark schemes (Flare). Specifically, we consider two types of 360° videos: slow-paced (Coastal pier) and fast-paced (Mega coaster). The total bandwidth of the BS is 2 MHz. The number of users is 4 and $(\alpha, \beta, \gamma) = (1/3, 1/3, 1/3)$. The size of the particle swarm is 4, and the total number of iterations is 30. The images show the cropped FoV of a particular user while watching the 360° videos. By comparing the image qualities generated by both schemes, shown in Fig. 17 and Fig. 18, we can see that with the proposed two-step adaptive streaming scheme, the user enjoys better video quality with higher-resolution FoV tiles than Flare. For better visualization, we highlight some major areas showing video quality improvements using red rectangular boxes in both figures. Please note that due to display resolution and space limit, we only visualize the image comparison of perceived FoV quality between our scheme and Flare. The visualized performance comparisons with all the benchmark schemes in terms of user viewing QoE are provided in the generated demo videos, submitted as supplemental materials along with this manuscript.

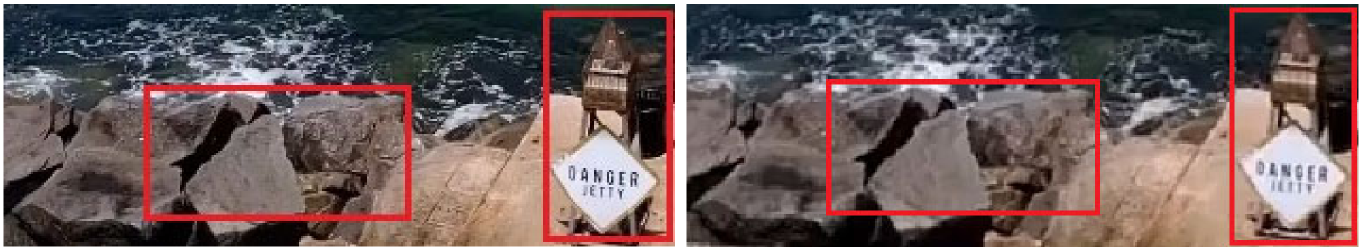
VII. Conclusion

In this paper, a two-step adaptive streaming framework has been proposed to enhance 360° video viewing experience. Each video chunk is first prefetched and then enhanced at a closer-to-playback time instant. Transmission-driven and device-VSR-driven methods are adaptively enabled to achieve energy-efficient FoV quality enhancement. User viewing QoE is characterized using a time-difference approach. A practical PSO-based algorithm is developed for multi-user viewing QoE optimization, through the decision-making of bandwidth allocation, bitrate level selection, and enhancement method selection. Simulation results validate the performance of our proposed framework over benchmark schemes. For future work, the proposed scheme will be extended to consider successive chunk enhancement, where each chunk can be enhanced multiple times before the playback. A more comprehensive QoE model incorporating the QoE component of spatial quality smoothness will be considered for more fine-grained viewing QoE optimization. In addition, the cooperation among edge computing/caching nodes and users will be



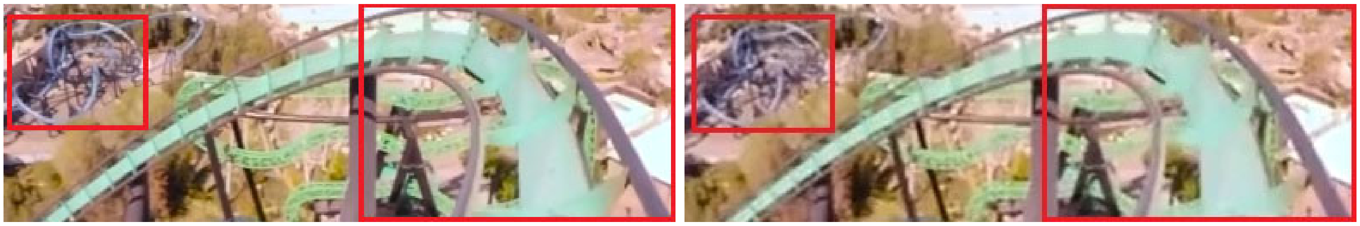
(a) Average viewing QoE (b) Average normalized energy consumption (c) Average utility

Fig. 16: Performance of different schemes with varying number of users.



(a) Proposed (b) Flare

Fig. 17: Cropped snapshots of the generated demo videos (Coastal pier, slow-paced) under the proposed scheme and Flare.



(a) Proposed (b) Flare

Fig. 18: Cropped snapshots of the generated demo videos (Mega coaster, fast-paced) under the proposed scheme and Flare.

investigated to support more advanced 360° video streaming service provisioning.

Appendix: Proof of Proposition 1

For each candidate chunk i, $Q^1_{t,u,i}$, $Q^1_{t+1,u,i-1}$, $Q^1_{t,u,i-1}$, $Q^1_{t+1,u,i+1}$, and $Q^1_{t,u,i+1}$ are known as constants, with $Q^1_{t+1,u,i-1} = Q^1_{t,u,i-1}$ and $Q^1_{t+1,u,i+1} = Q^1_{t,u,i+1}$. For brevity, let $Q^1_{t+1,u,i} = x$, $Q^1_{t+1,u,i-1} = Q^1_{t,u,i-1} = C_1$, $Q^1_{t,u,i} = C_2$, $Q^1_{t+1,u,i+1} = Q^1_{t,u,i+1} = C_3$, and (18) is rewritten as

$$\begin{aligned} f(x) = \alpha\tfrac{1}{|\mathcal{I}_u|}(x-C_2) - \tfrac{1}{|\mathcal{I}_u|-1}\gamma\big[&(|x-C_1|-|C_2-C_1|)\\ &e_{t,u,i-1} + (|C_3-x|-|C_3-C_2|)\,e_{t,u,i+1}\big]. \end{aligned} \tag{A1}$$

We first consider the case when $e_{t,u,i-1} = e_{t,u,i+1} = 1$. In this case, we have

$$\begin{aligned} f(x) = f_A(x) - f_B(x) = \alpha\frac{1}{|\mathcal{I}_u|}(x-C_2) - \frac{1}{|\mathcal{I}_u|-1}\gamma\\ \big[(|x-C_1|-|C_2-C_1|) + (|C_3-x|-|C_3-C_2|)\big] \end{aligned} \tag{A2}$$

with $f'_A(x) = \alpha\frac{1}{|\mathcal{I}_u|} > 0$. Next, we analyze the monotonicity of $f_B(x)$, which has the following two cases.

(1) $C_1 \le C_3$

If $C_1 \le x \le C_3$, $f'_B(x) = 0$ and $f'(x) = \alpha\frac{1}{|\mathcal{I}_u|} > 0$. If $x \le C_1$, $f'_B(x) = -\frac{2}{|\mathcal{I}_u|-1}\gamma$ and $f'(x) = \alpha\frac{1}{|\mathcal{I}_u|} + \frac{2}{|\mathcal{I}_u|-1}\gamma > 0$. If $x \ge C_3$, $f'_B(x) = \frac{2}{|\mathcal{I}_u|-1}\gamma$ and $f'(x) = \alpha\frac{1}{|\mathcal{I}_u|} - \frac{2}{|\mathcal{I}_u|-1}\gamma$, which can be either larger than, smaller than, or equal to 0. Therefore, when $C_1 \le C_3$, $f(x)$ monotonically increases with x when $x \le C_3$ and monotonically increases or decreases with x when $x \ge C_3$. Considering the domain of x, i.e., $C_2 \le x \le f_{t+1,u,i}$, the maximum of $f(x)$ is obtained by $\max(f(C_2), f(f_{t+1,u,i}))$ with $f(C_2) = 0$.

(2) $C_1 \ge C_3$

Similar monotonicity analysis can be applied to the case when $C_1 \ge C_3$, which is omitted for brevity.

When $e_{t,u,i-1} = 0$ and/or $e_{t,u,i+1} = 0$, it is a special case of $f(x)$. Similar monotonicity analysis can be applied, which is omitted for brevity.

References

[1] A. Yaqoob, T. Bi, and G.-M. Muntean, "A survey on adaptive 360 video streaming: Solutions, challenges and opportunities," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2801–2838, 2020.

[2] S. Z. A. Ansari, V. K. Shukla, K. Saxena, and B. Filomeno, "Implementing virtual reality in entertainment industry," in *Proc. Int. Conf. Cyber Intelligence and Information Retrieval*, 2022, pp. 561–570.

[3] J. Kim, K. Kim, and W. Kim, “Impact of immersive virtual reality content using 360-degree videos in undergraduate education,” *IEEE Trans. on Learning Technol.*, vol. 15, no. 1, pp. 137–149, 2022.

[4] M. Intelligence, “Virtual reality (VR) market–growth, trends, Covid-19 impact, and forecasts (2021–2026),” 2021.

[5] J. Van Der Hooft, H. Amirpour, M. T. Vega, Y. Sanchez, R. Schatz, T. Schierl, and C. Timmerer, “A tutorial on immersive video delivery: From omnidirectional video to holography,” *IEEE Commun. Surveys Tuts.*, vol. 25, no. 2, pp. 1336–1375, 2023.

[6] Y. Guan, C. Zheng, X. Zhang, Z. Guo, and J. Jiang, “Pano: Optimizing 360 video streaming with a better understanding of quality perception,” in *Proc. of the ACM Special Interest Group on Data Commun.*, 2019, pp. 394–407.

[7] Y. Wei, Q. J. Ye, K. Qu, W. Zhuang, and X. S. Shen, “Transmission protocol customization for on-demand tile-based 360° VR video streaming,” in *Proc. 2024 IEEE/CIC Int. Conf. on Commun. in China (ICCC)*, pp. 1069–1074.

[8] Y. Wang, J. Li, Z. Li, S. Shang, and Y. Liu, “Synergistic temporal-spatial user-aware viewport prediction for optimal adaptive 360-degree video streaming,” *IEEE Trans. on Broadcasting*, vol. 70, no. 2, pp. 453–467, 2024.

[9] Y. Wei, Q. Ye, K. Qu, W. Zhuang, and X. Shen, “Customized transmission protocol for tile-based 360° VR video streaming over core network slices,” *IEEE Trans. Netw.*, vol. 33, no. 1, pp. 340–354, 2025.

[10] J. Zeng, X. Zhou, and K. Li, “MADRL-based joint edge caching and bitrate selection for multicategory 360 video streaming,” *IEEE Internet Things J.*, vol. 11, no. 1, pp. 584–596, 2023.

[11] D. V. Nguyen, H. T. Tran, A. T. Pham, and T. C. Thang, “An optimal tile-based approach for viewport-adaptive 360-degree video streaming,” *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 1, pp. 29–42, 2019.

[12] N. Kan, J. Zou, C. Li, W. Dai, and H. Xiong, “RAPT360: Reinforcement learning-based rate adaptation for 360-degree video streaming with adaptive prediction and tiling,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 1607–1623, 2021.

[13] F. Duanmu, E. Kurdoglu, S. A. Hosseini, Y. Liu, and Y. Wang, “Prioritized buffer control in two-tier 360 video streaming,” in *Proc. of the Workshop on Virtual Reality and Augmented Reality Network*, 2017, pp. 13–18.

[14] L. Sun, F. Duanmu, Y. Liu, Y. Wang, Y. Ye, H. Shi, and D. Dai, “A two-tier system for on-demand streaming of 360 degree video over dynamic networks,” *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 1, pp. 43–57, 2019.

[15] J. Song, F. Yang, W. Zhang, W. Zou, Y. Fan, and P. Di, “A fast FoV-switching DASH system based on tiling mechanism for practical omnidirectional video services,” *IEEE Trans. on multimedia*, vol. 22, no. 9, pp. 2366–2381, 2019.

[16] H. Zhang, Y. Ban, Z. Guo, K. Chen, and X. Zhang, “RAM360: Robust adaptive multi-layer 360° video streaming with Lyapunov optimization,” *IEEE Trans. on Multimedia*, vol. 25, pp. 4225–4239, 2023.

[17] A. A. Baniya, T.-K. Lee, P. W. Eklund, and S. Aryal, “A survey of deep learning video super-resolution,” *IEEE Trans. on Emerg. Topics in Computational Intelligence*, vol. 8, no. 4, pp. 2655–2676, 2024.

[18] J. Shi, L. Pu, X. Yuan, Q. Gong, and J. Xu, “Sophon: Super-resolution enhanced 360 video streaming with visual saliency-aware prefetch,” in *Proc. of the 30th ACM Int. Conf. on Multimedia*, 2022, pp. 3124–3133.

[19] B. Chai, J. Chen, Z. Luo, Z. Wang, M. Hu, Y. Zhou, and D. Wu, “SDSR: Optimizing metaverse video streaming via saliency-driven dynamic super-resolution,” *IEEE J. Sel. Areas Commun.*, vol. 42, no. 4, pp. 978–989, 2023.

[20] J. Chen, M. Hu, Z. Luo, Z. Wang, and D. Wu, “SR360: Boosting 360-degree video streaming with super-resolution,” in *Proc. of the 30th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2020, pp. 1–6.

[21] J. Zeng, X. Zhou, and K. Li, “Toward high-quality low-latency 360° video streaming with edge–client collaborative caching and super-resolution,” *IEEE Internet Things J.*, vol. 11, no. 17, pp. 29 020–29 034, 2024.

[22] J.-L. Lin, Y.-H. Lee, C.-H. Shih, S.-Y. Lin, H.-C. Lin, S.-K. Chang, P. Wang, L. Liu, and C.-C. Ju, “Efficient projection and coding tools for 360° video,” *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 1, pp. 84–97, 2019.

[23] J. M. Boyce, Y. Ye, J. Chen, and A. K. Ramasubramonian, “Overview of SHVC: Scalable extensions of the high efficiency video coding standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 20–34, 2015.

[24] G. Tech, Y. Chen, K. Müller, J.-R. Ohm, A. Vetro, and Y.-K. Wang, “Overview of the multiview and 3D extensions of high efficiency video coding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 35–49, 2016.

[25] S.-C. Yen, C.-L. Fan, and C.-H. Hsu, “Streaming 360° videos to head-mounted virtual reality using DASH over QUIC transport protocol,” in *Proc. of the 24th ACM Workshop on Packet Video*, 2019, pp. 7–12.

[26] Z. Huang, P. Yang, N. Zhang, F. Lyu, Q. Li, W. Wu, and X. S. Shen, “QoE-driven mobile 360 video streaming: Predictive view generation and dynamic tile selection,” in *Proc. IEEE/CIC Int. Conf. on Commun. in China (ICCC)*, 2021, pp. 1113–1118.

[27] Y. Xu, Y. Dong, J. Wu, Z. Sun, Z. Shi, J. Yu, and S. Gao, “Gaze prediction in dynamic 360 immersive videos,” in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2018, pp. 5333–5342.

[28] Z. Ye, Q. Li, X. Ma, D. Zhao, Y. Jiang, L. Ma, B. Yi, and G.-M. Muntean, “VRCT: A viewport reconstruction-based 360 video caching solution for tile-adaptive streaming,” *IEEE Trans. on Broadcasting*, vol. 69, no. 3, pp. 691–703, 2023.

[29] Q. Cheng, H. Shan, W. Zhuang, L. Yu, Z. Zhang, and T. Q. S. Quek, “Design and analysis of MEC- and proactive caching-based 360° mobile VR video streaming,” *IEEE Trans. on Multimedia*, vol. 24, pp. 1529–1544, 2022.

[30] A. Nguyen, Z. Yan, and K. Nahrstedt, “Your attention is unique: Detecting 360-degree video saliency in head-mounted display for head movement prediction,” in *Proc. of the 26th ACM Int. Conf. on Multimedia*, 2018, pp. 1190–1198.

[31] P. Maniotis, E. Bourtsoulatze, and N. Thomos, “Tile-based joint caching and delivery of 360° videos in heterogeneous networks,” *IEEE Trans. on Multimedia*, vol. 22, no. 9, pp. 2382–2395, 2020.

[32] Z. Yu, J. Liu, C. Wang, and Q. Yang, “Bandit learning-based edge caching for 360-degree video streaming with switching cost,” *IEEE Access*, vol. 10, pp. 80 714–80 726, 2022.

[33] T. Yang, Z. Tan, Y. Xu, and S. Cai, “Collaborative edge caching and transcoding for 360° video streaming based on deep reinforcement learning,” *IEEE Internet Things J.*, vol. 9, no. 24, pp. 25 551–25 564, 2022.

[34] H. Guo, F. Wang, W. Zhang, Y. Zhu, L. Cui, J. Liu, F. R. Yu, and L. Zhang, “Joint adaptation for mobile 360-degree video streaming and enhancement,” *IEEE Trans. Mobile Comput.*, 2025, early access, Apr. 1, 2025, doi: 10.1109/TMC.2025.3555322.

[35] X. Zhang, X. Hu, L. Zhong, S. Shirmohammadi, and L. Zhang, “Cooperative tile-based 360 panoramic streaming in heterogeneous networks using scalable video coding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 1, pp. 217–231, 2018.

[36] K. Long, Y. Cui, C. Ye, and Z. Liu, “Optimal wireless streaming of multi-quality 360 VR video by exploiting natural, relative smoothness-enabled, and transcoding-enabled multicast opportunities,” *IEEE Trans. on Multimedia*, vol. 23, pp. 3670–3683, 2021.

[37] H. Yuan, S. Zhao, J. Hou, X. Wei, and S. Kwong, “Spatial and temporal consistency-aware dynamic adaptive streaming for 360-degree videos,” *IEEE J. Sel. Topics in Signal Processing*, vol. 14, no. 1, pp. 177–193, 2019.

[38] M. Mahmoud, S. Rizou, A. S. Panayides, N. V. Kantartzis, G. K. Karagiannidis, P. I. Lazaridis, and Z. D. Zaharis, “A survey on optimizing mobile delivery of 360° videos: Edge caching and multicasting,” *IEEE Access*, vol. 11, pp. 68 925–68 942, 2023.

[39] Y. Jin, J. Liu, F. Wang, and S. Cui, “Ebublio: Edge-assisted multiuser 360° video streaming,” *IEEE Internet Things J.*, vol. 10, no. 17, pp. 15 408–15 419, 2023.

[40] 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 in Signal Processing*, vol. 17, no. 5, pp. 1131–1146, 2023.

[41] H. Xiao, C. Xu, Z. Feng, R. Ding, S. Yang, L. Zhong, J. Liang, and G.-M. Muntean, “A transcoding-enabled 360 VR video caching and delivery framework for edge-enhanced next-generation wireless networks,” *IEEE J. Sel. Areas Commun.*, vol. 40, no. 5, pp. 1615–1631, 2022.

[42] A. Zhang, Q. Li, Y. Chen, X. Ma, L. Zou, Y. Jiang, Z. Xu, and G.-M. Muntean, “Video super-resolution and caching—An edge-assisted adaptive video streaming solution,” *IEEE Trans. on Broadcasting*, vol. 67, no. 4, pp. 799–812, 2021.

[43] H. Yeo, Y. Jung, J. Kim, J. Shin, and D. Han, “Neural adaptive content-aware internet video delivery,” in *Proc. 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, pp. 645–661.

[44] Y. Zhang, Y. Zhang, Y. Wu, Y. Tao, K. Bian, P. Zhou, L. Song, and H. Tuo, “Improving quality of experience by adaptive video streaming with super-resolution,” in *Proc. IEEE INFOCOM 2020 - IEEE Conf. on Computer Commun.*, pp. 1957–1966.

[45] X. Zhang, H. Xu, L. Zou, J. Duan, C. Wu, Y. Xue, Z. Chen, and X. Chen, “Rosevin: Employing resource-and rate-adaptive edge super-resolution

for video streaming," in *Proc. IEEE INFOCOM 2024 - IEEE Conf. on Computer Commun.*, pp. 491–500.

[46] Q. Li, Y. Chen, A. Zhang, Y. Jiang, L. Zou, Z. Xu, and G.-M. Muntean, "A super-resolution flexible video coding solution for improving live streaming quality," *IEEE Trans. on Multimedia*, vol. 25, pp. 6341–6355, 2023.

[47] W. Jing, C. Liu, H. Cai, X. Wen, Z. Lu, Z. Wang, and H. Zhang, "MEC-based super-resolution enhanced adaptive video streaming optimization for mobile networks with satellite backhaul," *IEEE Trans. Netw. Service Manag.*, vol. 21, no. 3, pp. 2977–2991, 2024.

[48] M. Choi, W. J. Yun, S. B. Son, S. Park, and J. Kim, "Joint delay-sensitive and power-efficient quality control of dynamic video streaming using adaptive super-resolution," *IEEE Trans. on Green Commun. and Netw.*, vol. 8, no. 1, pp. 103–117, 2023.

[49] S. Wang, J. Yang, and S. Bi, "Adaptive video streaming in multi-tier computing networks: Joint edge transcoding and client enhancement," *IEEE Trans. Mobile Comput.*, vol. 23, no. 4, pp. 2657–2670, 2024.

[50] M. Dasari, A. Bhattacharya, S. Vargas, P. Sahu, A. Balasubramanian, and S. R. Das, "Streaming 360-degree videos using super-resolution," in *Proc. IEEE INFOCOM 2020 - IEEE Conf. on Computer Commun.*, pp. 1977–1986.

[51] H. Yeo, C. J. Chong, Y. Jung, J. Ye, and D. Han, "Nemo: enabling neural-enhanced video streaming on commodity mobile devices," in *Proc. of the 26th Annual Int. Conf. on Mobile Comput. and Netw.*, 2020, pp. 1–14.

[52] A. Sarkar, J. Murray, M. Dasari, M. Zink, and K. Nahrstedt, "L3bou: Low latency, low bandwidth, optimized super-resolution backhaul for 360-degree video streaming," in *Proc. 2021 IEEE Int. Symposium on Multimedia (ISM)*, pp. 138–147.

[53] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 136–144.

[54] L. Zhang, H. Zhou, H. Wang, and L. Cui, "TBSR: Tile-based 360° video streaming with super-resolution on commodity mobile devices," in *Proc. IEEE INFOCOM 2024 - IEEE Conf. on Computer Commun.*, pp. 501–510.

[55] L. Xie, Z. Xu, Y. Ban, X. Zhang, and Z. Guo, "360ProbDASH: Improving QoE of 360 video streaming using tile-based HTTP adaptive streaming," in *Proc. of the 25th ACM Int. Conf. on Multimedia*, 2017, pp. 315–323.

[56] H. Zhao, B. Zheng, S. Yuan, H. Zhang, C. Yan, L. Li, and G. Slabaugh, "CBREN: Convolutional neural networks for constant bit rate video quality enhancement," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 7, pp. 4138–4149, 2021.

[57] S. Kumar, J. Jin, Y.-N. Dong *et al.*, "Seer: Learning-based 360° video streaming for MEC-equipped cellular networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 6, pp. 3308–3319, 2023.

[58] X. Tan, S. Wang, X. Xu, Q. Zheng, J. Yang, and S. Chen, "DACOD360: Deadline-aware content delivery for 360-degree video streaming over MEC networks," *IEEE Trans. on Multimedia*, vol. 26, pp. 4168–4182, 2024.

[59] S. Burer and A. N. Letchford, "Non-convex mixed-integer nonlinear programming: A survey," *Surveys in Operations Research and Management Science*, vol. 17, no. 2, pp. 97–106, 2012.

[60] D. Tarekegn Nigatu, T. Gemechu Dinka, and S. Luleseged Tilahun, "Convergence analysis of particle swarm optimization algorithms for different constriction factors," *Frontiers in Applied Mathematics and Statistics*, vol. 10, p. 1304268, 2024.

[61] H. Huang, J. Qiu, and K. Riedl, "On the global convergence of particle swarm optimization methods," *Applied Mathematics & Optimization*, vol. 88, no. 2, p. 30, 2023.

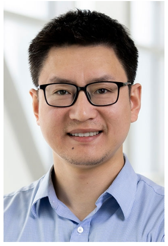
[62] W.-C. Lo, C.-L. Fan, J. Lee, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, "360 video viewing dataset in head-mounted virtual reality," in *Proc. of the 8th ACM on Multimedia Systems Conf.*, 2017, pp. 211–216.

[63] N. Gao, G. Liu, M. Feng, X. Hua, and T. Jiang, "Non-orthogonal multiple access enhanced scalable 360-degree video multicast," *IEEE Trans. on Multimedia*, vol. 26, pp. 8488–8503, 2024.

[64] X. Ye, K. Qu, W. Zhuang, and X. Shen, "Accuracy-aware cooperative sensing and computing for connected autonomous vehicles," *IEEE Trans. Mobile Comput.*, vol. 23, no. 8, pp. 8193–8207, 2023.

[65] F. Qian, B. Han, Q. Xiao, and V. Gopalakrishnan, "Flare: Practical viewport-adaptive 360-degree video streaming for mobile devices," in *Proc. of the 24th Annual Int. Conf. on Mobile Comput. and Netw.*, 2018, pp. 99–114.

Yannan Wei (S'24, IEEE) receives his Ph.D. degree in Electrical and Computer Engineering from the University of Waterloo, Canada, in 2025. His research interests include network intelligence, resource management, and service provisioning in future telecommunication systems.



Qiang (John) Ye (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, ON, Canada, in 2016. Since 2023, he has been an Assistant Professor with the Department of Electrical and Software Engineering, University of Calgary, AB, Canada.

He is/was a general, publication, publicity, TPC, or symposium co-chair for different reputable international conferences and workshops. He also serves/served as the IEEE VTS Region 7 Chapter Coordinator from 2024, the IEEE ComSoc Southern Alberta Chapter Vice Chair from 2024, and the VTS Regions 1-7 Chapters Coordinator (2022-2023). Dr. Ye serves as an Associate Editor for prestigious IEEE journals, such as IEEE IoT-J, TVT, TCCN, and OJ-COMS. He received the Best Paper Award in the IEEE ICCC in 2024 and the IEEE TCCN Exemplary Editor Award in 2023. Dr. Ye received the Early Career Research Excellence Award, Schulich School of Engineering, University of Calgary, in 2024. He has been selected as an IEEE ComSoc Distinguished Lecturer for the class of 2025-2026.

Weihua Zhuang (Fellow, IEEE) received the B.Sc. and M.Sc. degrees from Dalian Marine University, Dalian, China, and the Ph.D. degree from the University of New Brunswick, Canada, all in electrical engineering. She is a University Professor and a University Research Chair in Wireless Communication Networks with the University of Waterloo, Canada. Her research focuses on network architecture, algorithms, and protocols, and service provisioning in future communication systems. She is an Elected Member of the Board of Governors and the Past President of the IEEE Vehicular Technology Society. She is a Fellow of Royal Society of Canada, Canadian Academy of Engineering, and Engineering Institute of Canada.

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, Canada. His research focuses on network resource management, wireless network security, Internet of Things, AI for networks, and vehicular networks. Dr. Shen 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.

Dr. Shen is the Past President of the IEEE Communications Society. He was the Vice President for Technical & Educational Activities, Vice President for Publications, Member-at-Large on the Board of Governors, Chair of the Distinguished Lecturer Selection Committee, and Member of IEEE Fellow Selection Committee of the ComSoc.