Journal of Systems Architecture 162 (2025) 103401

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

Joint optimization of layering and power allocation for scalable VR video in 6G networks based on Deep Reinforcement Learning ☆

Junchao Yang [a], Hui Zhang [b], Wenxin Jiao [a], Zhiwei Guo [a], Fayez Alqahtani [c], Amr Tolba [d], Yu Shen [a,*]

[a] School of Artificial Intelligence, Chongqing Technology and Business University, Chongqing 400067, China
[b] Chongqing Key Laboratory of Public Big Data Security Technology, Chongqing College of Mobile Communication, Jiari Road, Chongqing, 401520, China
[c] Software Engineering Department, College of Computer and Information Sciences, King Saud University, Riyadh 12372, Saudi Arabia
[d] Computer Science Department, Community College, King Saud University, Riyadh 11437, Saudi Arabia

ARTICLE INFO

Keywords:
Scalable video encoding
Virtual reality
6G networks
Resource scheduling
Deep reinforcement learning

ABSTRACT

With the advancement and application of virtual reality (VR) technology, there is a growing demand for network bandwidth and computational capabilities. To address the challenges of high bandwidth requirements, low latency demands, and intensive computational tasks in VR video transmission, this paper proposes a joint optimization method for layering and power allocation based on Deep Reinforcement Learning (DRL). The method focuses on the transmission of scalable VR videos in 6G networks, utilizing DRL to achieve a cloud-edge-end collaborative transmission framework, where Tile-based scalable VR video is proactively cached to the MEC nodes, and Asynchronous Advantage Actor-Critic (A3C) algorithm is adopted to jointly optimize dual-connected link resources, edge computing resources, and user terminal computing resources. Through simulation experiments, the effectiveness of the proposed algorithms was validated. The results show that compared to baseline algorithms and state-of-the-art methods, the proposed A3C algorithm effectively improves the average quality of experience (QoE) for VR users and maintains low latency under various sub-6G and millimeter wave link capacities. Furthermore, with increased Mobile Edge Computing (MEC) computing power and User Equipment (UE) computing capabilities, the proposed method can further improve QoE and reduce latency.

1. Introduction

Since 2020, the world has been severely hit by the COVID-19, which has greatly stimulated the development of the non-contact economy, including the virtual reality industry [1]. Global tech giants began to strategically position themselves in the metaverse from 2021, such as Horizon a VR social platform released by Meta, and Baidu from China also accelerates the launch of its metaverse social platform "Xi Rang". VR has emerged as a critical gateway to the metaverse, characterized by its ability to provide users with immersive experiences. With the deep integration of 5G and cloud computing, it will have a profound and extensive transformative impact on the process of economic and social development [2]. Global VR headset shipments reached 11.1 million units in 2021, marking a pivotal moment with the Oculus Quest 2 as the first consumer VR device in the consumer market. According to International Data Corporation (IDC) prediction in 2024, the global VR/AR market is expected to exceed 100 billion Dollars [3].

The VR technology encounters several critical challenges, including stringent latency and bandwidth requirements, high computational demands, and inefficient content delivery. These factors must be addressed to ensure a smooth and immersive user experience. To provide a truly immersive experience that aligns with human sensory perception, VR demands ultra-high definition (UHD) resolution, panoramic images, and multi-angle information. This necessitates VR video content that contains up to ten times more data than conventional video, with a high frame rate typically 60 frames per second (fps) to prevent motion sickness due to simulation discrepancies [4]. However, the inherent inefficiencies in VR video transmission arise because only a portion of the panoramic content is rendered on the user's headset

☆ This work was funded by the Researchers Supporting Project Number (RSPD2025R681), King Saud University, Riyadh, Saudi Arabia; by the Chongqing Municipal Education Commission projects under grant KJQN202200829, KJQN202300844 and KJQN202200833.

* Corresponding author.
E-mail addresses: yangjc@ctbu.edu.cn (J. Yang), zhuicq@yeah.net (H. Zhang), jiaowenxin@ctbu.edu.cn (W. Jiao), zwguo@ctbu.edu.cn (Z. Guo), Fhalqahtani@ksu.edu.sa (F. Alqahtani), atolba@ksu.edu.sa (A. Tolba), shenyu@ctbu.edu.cn (Y. Shen).

https://doi.org/10.1016/j.sysarc.2025.103401
Received 21 October 2024; Received in revised form 11 February 2025; Accepted 7 March 2025
Available online 17 March 2025

at any given time, leading to a significant under-utilization of bandwidth resources. In traditional video streaming, Scalable Video Coding (SVC) is employed to accommodate varying network conditions and provide different levels of video quality [1,5]. SVC enables hierarchical encoding across time, space, and quality, allowing adaptive adjustment in frame rates, image resolution, and quality. This is achieved through multi-layer video encoding, where each layer represents a different spatial resolution or Signal-to-Noise Ratio (SNR) quality level, including a base layer and multiple enhancement layers. The latest video compression standard, Versatile Video Coding (VVC), has improved scalable encoding techniques for VR video, enhancing its ability to efficiently compress and transmit immersive content. Scalable encoding leverages the spatial characteristics of VR video, optimizing transmission efficiency and adapting to varying bandwidth conditions, but further research is required to fully exploit scalable encoding for VR video compression and adaptive transmission. Latency is another critical factor for VR performance. Given the high sensitivity of VR systems, maintaining a latency range of 17-20 ms is essential to ensure a high-quality user experience. Beyond this threshold, achieving strong interactive modes becomes extremely challenging, as network latency must remain within the millisecond range to maintain real-time responsiveness [6]. Furthermore, the sheer volume of data associated with VR content leads to significant processing complexity. In the early stages of VR development, decoding and rendering VR video required a high-performance PC, which was connected to a headset via a wired connection. However, this wired approach severely impacts the VR experience, restricting user mobility and immersion. The adoption of Qualcomm's XR2 chipset, along with the development of 6-degree-of-freedom (DOF) all-in-one VR headsets, has emerged as a mainstream solution for mobile VR applications. These mobile VR devices integrate graphics processing units (GPUs) and multi-core processors, are gaining popularity. However, the computational power of these mobile devices is still insufficient to meet the intensive demands of high-quality VR video rendering, especially as VR applications continue to evolve toward more complex and immersive experiences. Therefore, further advancements in mobile VR hardware and video encoding techniques are essential to address these challenges and enable seamless, high-performance VR experiences.

Most prior research on the transmission of virtual reality (VR) content has predominantly treated VR videos as high-definition (HD) videos with augmented informational content. The primary focus of these studies has been on minimizing bandwidth consumption through various strategies, including tile-based adaptive transmission [1], viewpoint-based adaptive transmission, and scalable video coding (SVC)-based adaptive transmission. While these methodologies have achieved certain improvements in reducing bandwidth consumption, they exhibit several critical limitations: (1) The user's perspective requires strict feedback latency, with optimal latency being less than 20 ms; however, current solutions often fail to meet this threshold. (2) There is insufficient consideration of intensive computational task offloading, as existing approaches heavily depend on high-performance PC devices for the intensive computational task, which poses challenges for lightweight mobile VR applications. This highlights the need for innovative solutions that address both latency and computational efficiency in VR content transmission.

In addressing the intensive computational demands of virtual reality (VR) applications, numerous researchers have conceptualized this challenge as a communication, storage, and computation trade-off optimization problem [7,8]. For example, the work [9] investigates VR video transmission in 5G networks based on MEC, under the constraints of latency and energy consumption, the research aims to minimize the required transmission rate by optimizing computation offloading and caching strategies. Further contributions, as detailed in [3,10], and [11], have modeled the communication, caching, and computation during VR content delivery as a trade-off optimization problem. These studies have effectively addressed decision-making processes with respect to caching and offloading strategies during the transmission of VR content, employing optimization techniques and reinforcement learning methodologies [12,13]. Furthermore, the research presented in [14] utilizes recurrent neural networks for user viewpoint prediction, facilitating video rendering tasks at the edge based on viewpoint prediction to enhance low-latency, lightweight VR experiences. To achieve adaptive transmission, work [15] introduces the storage of high-bitrate VR video versions on edge nodes, which are subsequently transcoded using the processing capabilities of these nodes. The study [16] contributes to the design of a low-latency VR transmission system by executing computational tasks such as video rendering and segmentation at the edge nodes. Moreover, the critical role of edge nodes in content caching and processing within augmented reality (AR) and VR systems is discussed in [17], which also highlights emerging prospects and challenges in the field. To improve content delivery efficiency, the work [18] proposes encoding VR content that is cached at edge nodes using network coding. Furthermore, [19] takes into account the time overhead of computation and transmission in VR content delivery, and studies the relationship between user experience, communication, and computation. The conclusion is that improving the accuracy of the user's viewport can enhance the user experience. Despite the significant computational tasks involved in VR content delivery, as evidenced by the aforementioned studies, many researches have approached the problem predominantly as a computation offloading issue [20–22]. This often neglects the unique characteristics of the VR, inadvertently introducing additional latency associated with computational offloading [23]. Therefore, a more comprehensive strategy is needed to address the challenges in VR content transmission.

The cloud-edge architecture driven by network operators and cloud service providers provides the architectural foundation for VR video transmission, where edge computing is an important breakthrough to achieve high-efficiency and low-latency VR video adaptive transmission. Mobile Edge Computing (MEC) as an important paradigm for 5G, provides cache and computational resources and is widely adopted in traditional video transmission for content caching and video transcoding [24–27]. To effectively leverage the limited caching and computational capabilities of MEC while delivering high-quality VR content, it is imperative to implement a collaborative resource scheduling framework that integrates both cloud-edge and edge-end collaboration, which is essential for adaptive transmission of VR video content with high-efficiency and low-latency. One viable strategy is to utilize cloud-edge collaboration for VR video caching [28], which minimizes end-to-end latency for users. However, conventional caching strategies that predominantly rely on content popularity metrics, as referenced in [29], often fall short in addressing the diverse and specific viewpoint requirements of VR users. Consequently, the further research is needed to focus on VR video caching methodologies that cater to these differentiated user requirement within the constraints of limited cache capacity at the edge. Furthermore, employing edge servers for offloading intensive computational tasks from terminal devices can significantly mitigate the computational overload on user devices, thereby further reducing latency associated with computation [30,31]. Nevertheless, simply offloading dense computational tasks to the edge, as presented in [32], does not adequately satisfy the stringent low-latency demands of VR applications. Therefore, it is crucial to conduct in-depth research on the design of edge and terminal computing power allocation strategies that align with latency constraints, optimizing the user experience in accordance with the unique characteristics of VR computational tasks. This probably ensures that the delivery of VR contents is both efficient and responsive to user expectations.

In summary, our analysis has identified several critical challenges in the domain of virtual reality (VR) transmission: (1) Transmission Mechanism Limitations: Current researches of VR video delivery strategies predominantly emphasize the reduction of bandwidth consumption,

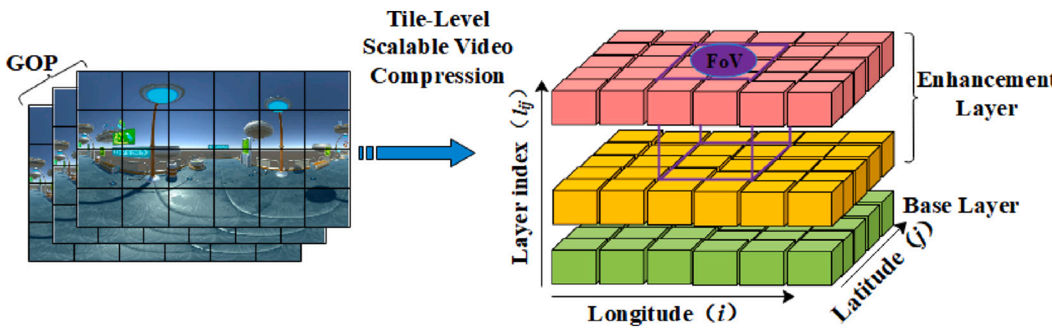


Fig. 1. Tile-based scalable coding for 360-degree video.

minimization of service latency, and offloading of intensive computational tasks. However, these studies often neglect to fully exploit the spatio-temporal characteristics of VR video content. Furthermore, there is a lack of comprehensive design of VR content delivery scheme that simultaneously achieve low latency and high efficiency within the framework of cloud–edge architecture. (2) Intensive Computing Offloading Optimization: Existing research primarily focuses on optimizing computational tasks offloading for VR applications according to the stringent latency requirements associated with immersive experiences. Nonetheless, there is a significant gap in the optimization of computing power allocation that takes into account the unique characteristics and demands of VR computing tasks. Addressing these issues require a more holistic approach that integrates advanced optimization techniques and leverages the inherent properties of VR content to enhance the overall performance and user experience in VR transmission systems.

This paper proposes a collaborative framework for scalable coding virtual reality (VR) video transmission based on a cloud-edge architecture, facilitating efficient transmission through edge caching and dual connection methodologies. Recognizing the variability in user demand for specific VR video viewpoint quality, we introduce a Tile-based scalable encoding scheme that compresses and encodes VR video [33–35]. This scheme effectively leverages the spatio-temporal characteristics of VR content, with the base layer delivering baseline video quality while the enhancement layer provides flexible quality improvements tailored to the user's viewpoint. To mitigate computational latency associated with decoding and rendering processes on VR headset devices, which typically possess limited computational capabilities, this research investigates the deployment of edge servers for decoding the Tile set at the enhancement layer. The use of millimeter-wave links for transmitting the decoded streams is explored, and the combination strategy of the scalable encoded Tile and the decoded raw video is proposed for transmission, thereby enhancing overall system performance. Under the constraints of maximum end-to-end latency, this study aims to optimize video quality specific to user viewpoints. We conduct a joint optimization of Tile layer selection and the allocation of computing power at both the edge and terminal for decoding. To achieve this object, a Deep Reinforcement Learning (DRL)-based optimization algorithm is proposed which jointly optimizes dual-connected link resources, edge computing resources, and user terminal computing resources.

The main contributions of this paper are summarized as follows:

- A framework for the joint optimization of tile layer selection and the allocation of computing power at the edge and terminal is established, where tile-based scalable VR video is proactively cached to the MEC nodes according to the popularity of VR tiles. This framework is designed to meet the maximum end-to-end latency constraints while optimizing video quality of the user's viewpoint.
- The paper contributes an optimization algorithm based on Deep Reinforcement Learning (DRL), specifically the Asynchronous Advantage Actor-Critic (A3C) algorithm, to jointly optimize dual-connected link resources, edge computing resources, and user terminal computing resources. This algorithm is shown to be effective in achieving high-quality, low-latency VR video transmission.
- Comprehensive simulation experiments are conducted to validate the effectiveness of the proposed algorithms and framework. The results demonstrate the ability of the proposed A3C algorithm to improve the average Quality of Experience (QoE) for VR users and reduce latency under various network conditions and computational capacities.
- The proposed methods are shown to scale well with the increasing number of VR users and to efficiently adapt to different network capacities and computational powers, highlighting their potential for real-world deployment in next-generation VR streaming services.

The remainder of the paper is organized as follows: In Section 2, the system model is explained, where a scalable encoding based collaborative framework for mobile VR adaptive transmission is introduced. The problem formulation and the DRL based solution for the joint optimization layer selection and power allocation are presented Section 3. The evaluation for the proposed framework and the algorithm are presented in Section 4. and the discussion is given in Section 5. Lastly, a summary of the paper is provided in Section 6.

2. System model

Tiling was introduced in the High Efficiency Video Coding (HEVC) standard to facilitate the parallel processing of conventional panoramic video in chunks using multi-core hardware [2]. In recent years, the benefits of Tile-based encoding of 360-degree video have been widely demonstrated, which enable to adaptively transmit Tile at different bit rates depending on the user's viewpoint. Recently, researchers have focused on Tile-based scalable encoding for VR video, experimentally demonstrating its ability to better adapt to VR user viewpoints and network bandwidth in spatio and temporal dimensions.

Fig. 1 shows the illustration of Tile-based scalable encoding for VR video, a set of original 360-degree video frames (Group of Picture, GOP) is generated into 2D panoramic frames by Equirectangular Projection (ERP) and spatially divided into a series of Tiles, denoted as (i, j) along its longitude and latitude dimensions. The set of Tiles of a group of (GOP) video frames at the same spatial location is compressed into a number of video layers with higher quality fidelity. The first layer of the compressed set of Tiles is usually referred to as the base layer and the remaining layers are the enhancement layers. The reconstruction fidelity of each set of Tiles is increased as the compressed layers are progressively decoded [5,14].

The base content can be constructed from a base layer L_b of the set of Tiles containing the user's viewpoint to prevent possible mismatches between enhancement content and actual user's viewpoint which caused by rapid movement of the user's head, thus ensuring that the content within the viewpoint can be constructed continuously. The enhancement content layer L_e, on the other hand, uses a smaller set of

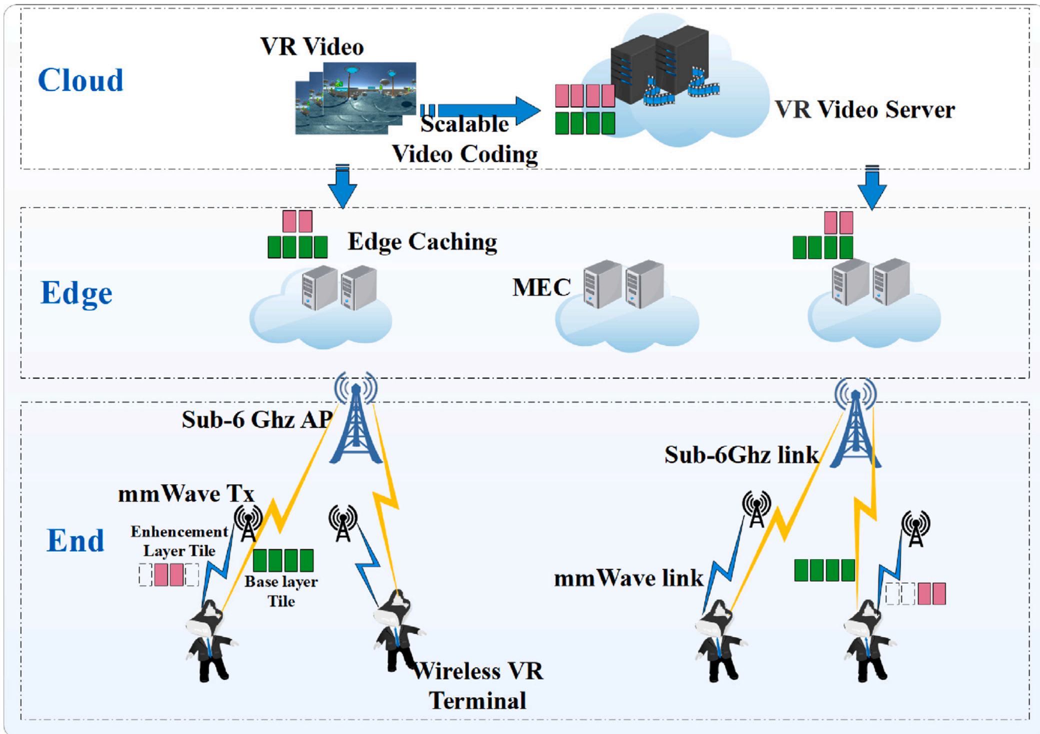


Fig. 2. The proposed transmission framework for scalable encoding VR video with cloud-edge-end collaboration.

Tiles containing only the user's viewpoint for quality enhancement to maximize the quality of the video within the viewpoint [26,36]. Therefore, constructing base and enhancement content based on scalable Tile sets involves the selection of Tile sets and the selection of the number of enhancement layers (i.e. layer selection). Under the constraints of wireless resources and latency, choosing the appropriate enhancement layer and Tile set directly affect the user's quality of experience. We will give an in-depth analysis on this issue in this paper, which will be described in the sections of the system framework and problem formulation.

The proposed framework for efficient transmission of scalable coded VR video with cloud-edge-end collaboration is shown in Fig. 2. The framework includes wireless VR terminal, Access network, edge server, and cloud server.

Wireless VR terminal: In this paper, a mainstream lightweight VR device is considered, which performs processing, display, interaction, and communication tasks. It is equipped with mainstream chips that support 5G and beyond network connections, sensors that enable 6 DOF tracking, and effective graphics rendering and 3D reconstruction [25,26,37].

Access networks: In this paper a 6G network VR application scenario is considered where traditional sub-6 GHz wireless access points are deployed around the user and dense high frequency wireless (e.g., millimeter wave) transmitters are deployed closer to the user (offices, gaming competition rooms) [20–22]. Wireless VR terminals are connected to the network via Dual connection technology, in addition, all types of access points are linked to edge servers via fibre.

Edge servers: This paper considers edge servers deployed at wireless access points with caching and computing resources. Among them, the AI engine is deployed on each edge node in the form of a virtual machine/container, which is responsible for the caching of VR content based on AI decisions.

Cloud server: In this paper, VR content server is deployed in the cloud, which is mainly responsible for ERP projection of raw 360-degree video frames [38], Tile-based scalable encoding, and proactively caching the corresponding scalable encoded Tiles to the edge node according to the caching decision made by the edge node.

Obviously, caching the encoded 360-degree video content to the edge is the key to meet the low latency requirement, and selecting the appropriate Tiles and number of layers for caching will directly affect the user's experience [39]. Based on the proactively viewpoint prediction, the edge server could cache and update the Tile of each layer of each GOP according to the user's request model, the user's viewpoint information, and the edge node's cache space. The edge server is responsible for maximizing the quality of the video within the current user's viewpoint by continuously updating the edge cache content, and the caching problem has been studied in our recent work [40]. Therefore, we consider that part of the decoding tasks are offloaded on the edge nodes, and with the goal of maximizing the video quality in the user's viewpoint, the problem is formulated as an optimization of the allocation of the edge computing resources and terminal computing resources under the latency constraints.

3. Joint optimization of layer selection and power allocation

3.1. Problem formulation

In current VR applications, the client device is mainly respond to decoding computation and dynamic rendering of the user's current viewpoint [41], however the limited computing power of the headset device causes additional computational latency, so utilizing more powerful GPU computing power of the edge server can alleviate the computational requirements of the mobile VR headset to optimize the immersive fidelity of the delivery, whilst meeting the stringent latency requirements of the application. In this paper, the enhancement content layer is considered streaming to VR device partially as raw data through the high frequency link with abundant transmission bandwidth, the partial encoded Tile in combination with the decoded raw video is transmitted to client device in order to achieve further performance gains. And the strategy is presented in Fig. 3.

Given $\mathbf{U} = \{1, \dots, u, \dots, U\}$ VR users, $\mathbf{N} = \{1, \dots, n, \dots, N\}$ 360-degree videos, each 360-degree video is encoded into $\mathbf{G} = \{1, \dots, g, \dots, G\}$ groups of images in the time domain, and each group of images

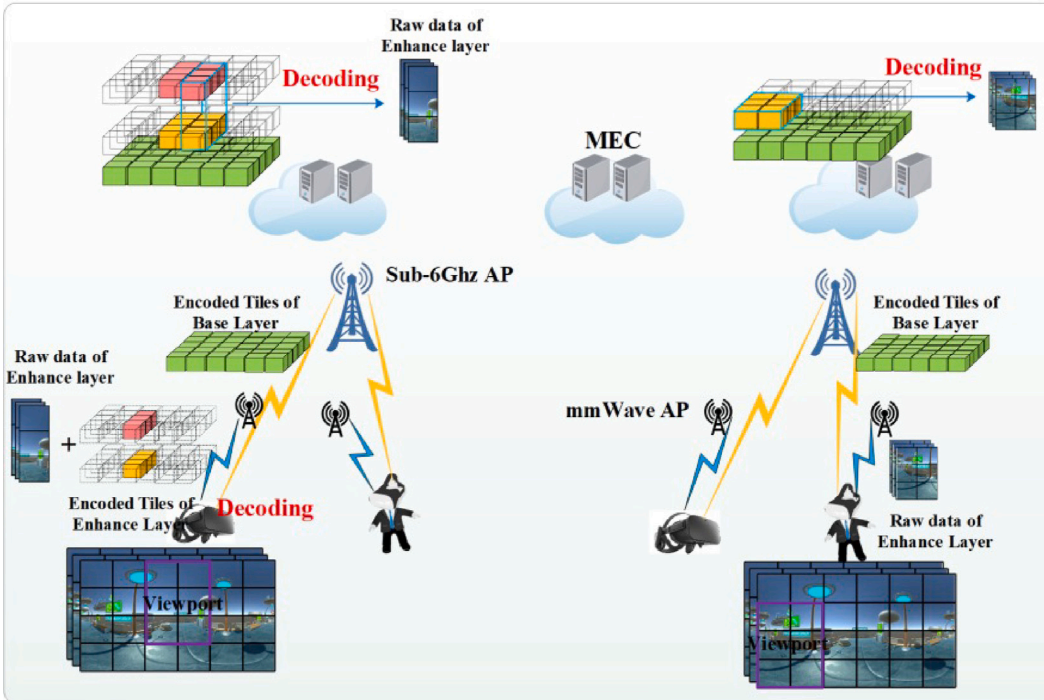


Fig. 3. The illustration of joint optimization problem of layering and power Allocation for VR video streaming.

is encoded into $\mathbf{L} = \{1,\ldots,l,\ldots,L\}$ layers based on scalable coding, and each layer is spatially partitioned into $\mathbf{M} = \{1,\ldots,m,\ldots,M\}$ Tiles. In addition, $\mathbf{K} = \{1,\ldots,k,\ldots,K\}$ edge servers are considered in the system. Based on the prediction of the user's viewpoint, the edge server selects a set of Tiles with non-zero probability of viewing for a group of images (GOP) to construct the base layer content, and the duration of the group of images is denoted as $\Delta\tau$, and the time latency for transmitting the base layer content to the user's headset via the Sub-6G link is denoted as $\tau_b^{u,tr}$, where c_b^u is the transmission rate of the Sub-6G link of the user u and $D_{n,g,m}^{u,b}$ is the bit rate of the base layer Tile.

$$\tau_b^{u,tr} = \sum_{m\in M_b^u} D_{n,g,m}^{u,b}\Delta\tau \Big/ c_b^u \tag{1}$$

The set of Tiles used for user u's viewpoint quality enhancement is denoted as M_e^u, the set of Tiles $M_{e,r}^u$ decoded at the edge server, and the remaining set of Tiles $M_{e,w}^u$, therefore, the latency for decoding at the edge server is denoted as $\tau_{e,r}^{u,de}$, where $z^{k,u}$ is the computation power allocated to user u by the edge server, $D_{n,g,m}^{u,e}$ is the encoding rate of the enhancement layer Tiles.

$$\tau_{e,r}^{u,de} = \sum_{m\in M_{e,r}^u} D_{n,g,m}^{u,e}\Delta\tau / z^{k,u} \tag{2}$$

The transmission latency of the enhancement content is denoted as $\tau_e^{u,tr}$, which is transmitted through the user's High Frequency (HF) millimeter wave link, and the latency of this part is mainly the transmission latency $\tau_{e,r}^{u,tr}$ of the set of Tiles $M_{e,r}^u$ decoded at the edge server (i.e. the raw data) and the transmission latency $\tau_{e,w}^{u,tr}$ of the set of Tiles $M_{e,w}^u$ un-decoded at the edge server. Where c_e^u is the transmission rate of the user u's HF millimeter wave link, $\eta(D_{n,g,m}^{u,e})$ is the rate of the enhancement layer Tile after decoding at the edge server, $\eta(\cdot)$ is a function of the rate of the Tile before decoding, which is obtained by polynomial fitting.

$$\tau_e^{u,tr} = \Big(\sum_{m\in M_{e,r}^u} \eta(D_{n,g,m}^{u,e})\Delta\tau + \sum_{m\in M_{e,w}^u} D_{n,g,m}^{u,e}\Delta\tau\Big)/c_e^u \tag{3}$$

$$\tau_{e,r}^{u,tr} = \sum_{m\in M_{e,r}^u} \eta(D_{n,g,m}^{u,e})\Delta\tau / c_e^u \tag{4}$$

$$\tau_{e,w}^{u,tr} = \sum_{m\in M_{e,w}^u} D_{n,g,m}^{u,e}\Delta\tau / c_e^u \tag{5}$$

The latency of the base layer Tile set decoding at the user terminal is denoted as $\tau_b^{u,de}$, where z_b^u is the computation power of the base layer encoding performed by user u.

$$\tau_b^{u,de} = \sum_{m\in M_b^u} D_{n,g,m}^{u,b}\Delta\tau / z_b^u \tag{6}$$

The latency of the enhancement Tile set $M_{e,w}^u$ to be decoded at the user terminal is denoted as $\tau_{e,w}^{u,de}$, where z_e^u is the computation power of the user u to perform the enhancement layer encoding.

$$\tau_{e,w}^{u,de} = \sum_{m\in M_{e,w}^u} D_{n,g,m}^{u,e}\Delta\tau / z_e^u \tag{7}$$

As shown in Fig. 4, the time latency of each part of the scalable video to the user terminal based on the edge-end cooperative decoding is shown, where the enhancement layer Tile set $M_{e,r}^u$ is decoded at the edge server while the Tile set $M_{e,w}^u$ is transmitted to the user terminal via the millimeter wave link, and the Tile set $M_{e,r}^u$ is transmitted to the user terminal via the millimeter wave link only after the time $\max(\tau_{e,r}^{u,de}, \tau_{e,w}^{u,tr})$ has been exceeded, so the time latency of this part can be expressed as $\max(\tau_{e,r}^{u,de}, \tau_{e,w}^{u,tr}) + \tau_{e,r}^{u,tr}$. In the case of the Tile set $M_{e,w}^u$ decoded at the terminal and the base layer Tile set M_b^u, the decoding calculation is performed only after the base layer Tile set M_b^u is transmitted to the user terminal via the Sub-6G link, after time $\tau_b^{u,tr}$, and the time latency can be expressed as $\tau_b^{u,tr} + \tau_{e,w}^{u,de}$, $\tau_b^{u,tr} + \tau_b^{u,de}$, respectively. To ensure the user u's experience, the latency of these three parts needs to be less than the maximum end-to-end latency τ.

Based on above analysis, the joint optimization of Tile layering and computing power allocation based on edge-end collaboration can be modeled as maximizing the expected video quality of users' viewpoint based on the viewpoint prediction. The edge server needs to select the appropriate Tile set $M_{e,r}^u$ and allocate the corresponding computing resources $z^{k,u}$for edge decoding, as well as the code rate$D_{n,g,m}^{u,b}$, $D_{n,g,m}^{u,e}$ of the base and enhancement Tile sets; the user terminal need to allocate the appropriate computation power according to the decoding demand of the base and enhancement layer Tile setsz_b^u, z_e^u. Where $f_{n,g,m}^{k,u}$ is the predicted user's viewpoint, λ and β are the video quality contribution coefficients of the base layer and enhancement layer, respectively. $Q(\cdot)$ is the video quality function, which is the function of tile data rate.

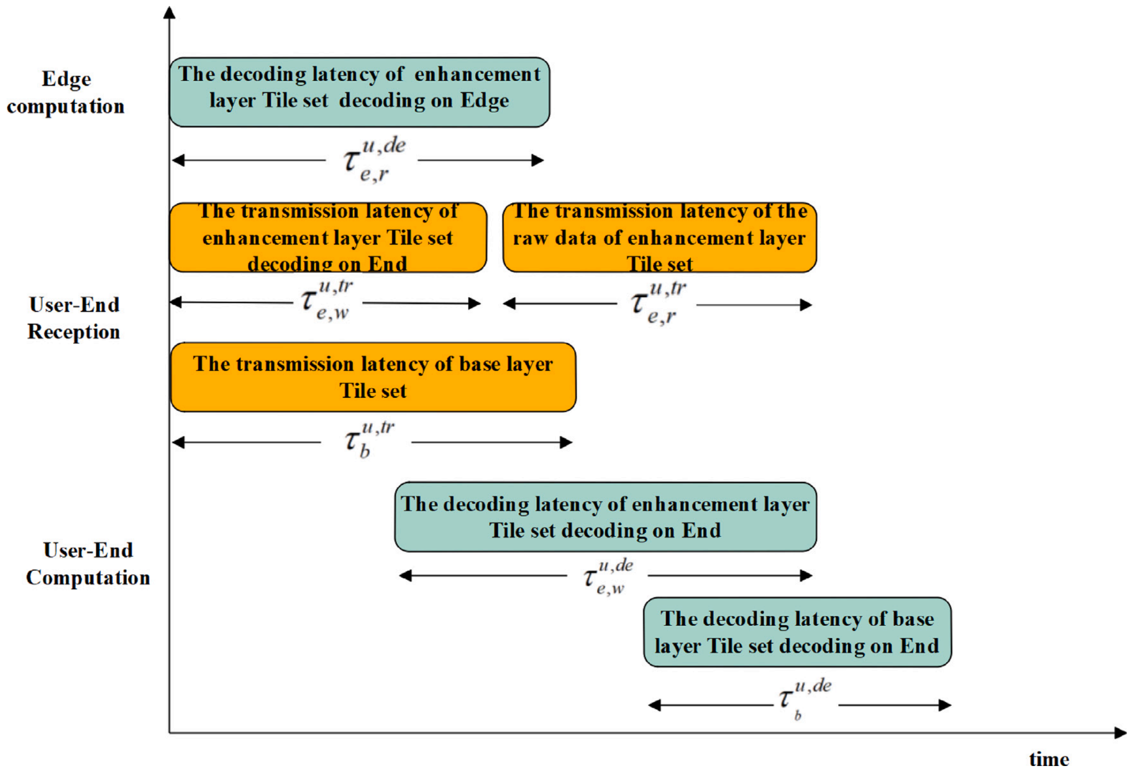


Fig. 4. VR user latency decomposition based on edge-end cooperative decoding.

C1 is the end-to-end maximum latency constraint of the enhancement layer Tile set decoded at the edge server; C2 is the end-to-end maximum latency constraint of the enhancement layer Tile set decoded at the user terminal; C3 is the maximum latency constraint of the base layer Tile set transmitted to the user terminal; C4 is the computing power constraint of the edge computing server; C5 is the computing power constraint of the user terminal.

$$
\begin{aligned}
&P1: \\
&\max_{\substack{M^u_{e,r},\, z^{k,u},\\ \{R^{u,b}_{n,g,m},\, R^{u,e}_{n,g,m}\},\, \{z^u_b,\, z^u_e\}}} \sum_k^K \sum_u^U \sum_n^N \sum_g^G \sum_m^M f^{k,u}_{n,g,m}\left(\lambda Q\left(D^{u,b}_{n,g,m}\right)\right. \\
&\qquad\qquad \left. + \beta Q\left(D^{u,e}_{n,g,m}\right)\right) \\
&\qquad s.t.\ C1, C2, C3, C4, C5
\end{aligned} \tag{8}
$$

$$C1: max(\tau^{u,de}_{e,r}, \tau^{u,tr}_{e,w}) + \tau^{u,tr}_{e,r} \le \tau \tag{9}$$

$$C2: \tau^{u,tr}_{b} + \tau^{u,de}_{e,w} \le \tau \tag{10}$$

$$C3: \tau^{u,tr}_{b} + \tau^{u,de}_{b} \le \tau \tag{11}$$

$$C4: \sum_{u\in U} z^{k,u} \le Z^k \tag{12}$$

$$C5: z^u_b + z^u_e \le Z^u \tag{13}$$

where the $Q(\cdot)$ function is used to measure the quality of VR video transmission. The QoE value of the tile can be expressed as Eq. (14), where $D^u_{n,g,m}$ and $D^{MAX}_{n,g,m}$ represent the bitrate of the g,mth tile of uth user and the maximum bitrate of the g,mth tile of the nth video, respectively. α and ϵ are the coefficients of the QoE function. The QoE function is a convex function of the video bitrate, and the first-order derivative of the function decreases as the video bitrate increases. These characteristics of the QoE function can well model the user's viewing experience. For tile-based VR videos, the QoE value of a tile describes its contribution to the user's overall viewing experience at a given bitrate.

$$Q^u_{n,g,m} = \begin{cases} \alpha \log(\epsilon \frac{D^u_{n,g,m}}{D^{MAX}_{n,g,m}}),\ D^u_{n,g,m} > 0 \\ 0,\ D^u_{n,g,m} = 0 \end{cases} \tag{14}$$

3.2. DRL based solution

The optimization problem $P1$ is a typical Mixed Integer Linear Programming (MILP) problem with high dimensional parameters, which makes it difficult to find the optimal solution in practical applications [15,33].

The objective of this paper is to optimize the total QoE of VR users under a fixed latency constraint by determining the optimal tile layering in the quality dimension, the optimal set of enhancement tiles in the spatial dimension, and the optimal allocation of MEC resources for decoding portions of the tile set on the MEC. In each time slot, the MEC seeks to maximize the long-term cumulative QoE in accordance with the policy π under the VR latency constraint. This policy maps the current system state S_t to the probability distribution over potential actions in the action space A_t. Thus, for a specific MEC, the objective of resource allocation is to maximize the total QoE of VR users by strategically allocating network and computing resources. The underlying optimization problem (denoted as $P1$) is reformulated as an optimization problem ($P2$), which aims to effectively balance the quality of the video experience with the resource constraints, while ensuring the end-to-end latency requirements of VR applications.

$$
\begin{aligned}
P2: \max\nolimits_{\pi(A_t|S_t)} &\textstyle\sum_u^U \sum_n^N \sum_m^M \sum_t^\infty f^{k,u}_{n,m}(t)\left(\lambda Q\left(D^{u,b}_{n,m}(t)\right)\right. \\
&\left. + \beta Q\left(D^{u,e}_{n,m}(t)\right)\right) \\
&s.t.\ C1, C2, C3, C4, C5
\end{aligned} \tag{15}
$$

The above optimization problem ($P2$) is a typical Partially Observable Markov Decision Process (POMDP) problem, which is difficult to find an optimal solution for practical applications. The computational resource allocation problem for edge nodes can be modeled as a typical

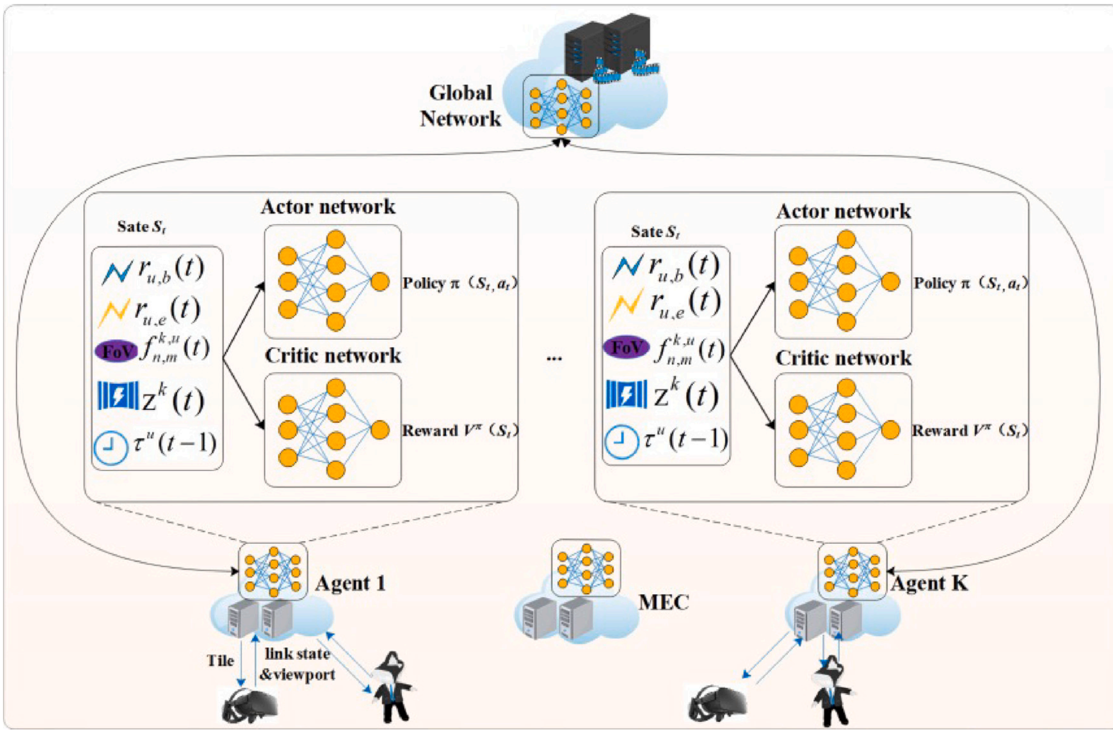


Fig. 5. DRL based solution of layer selection and power allocation for scalable VR video streaming.

Markov Decision Process (MDP), where each edge node can be optimally rewarded by interacting with the environment. Therefore, Deep Reinforcement Learning (DRL) is adopted in this paper to solve the problem. Due to the variability of user content requests and different user viewpoints, which results in a large state–action space, convergence difficulties will occur using traditional reinforcement learning algorithms. Meanwhile, the variability and randomness among users also make the probability of state transfer more difficult to obtain as well.

Based on the analysis presented above, this paper adopts Asynchronous Advantage Actor-Critic (A3C), an asynchronous reinforcement learning algorithm with excellent performance, which learns through distributed multiple agents interacting with the environment individually and periodically sharing parameters with other agents through a global network. The algorithm learns by distributing multiple agents to interact with the environment separately, and periodically shares parameters with other agents through a Global Network to guide its own interaction with the environment. In addition, the natural distributed architecture of edge servers and cloud servers is more suitable for deploying A3C algorithms. The edge node agents continuously learn the resource allocation policy to improve user experience through optimal resource allocation under different network states and computing power constraints, and share and update the globally optimal policy through the Global Network of the cloud servers (as shown in Fig. 5).

State space: the state space of the edge node agent k is defined as:

$$\mathbf{S}^k_t = \{f^{k,u}_{n,m}(t), r_{u,b}(t), r_{u,e}(t), z^k(t), \tau^u(t-1)\} \tag{16}$$

where, $f^{k,u}_{n,m}(t)$ is the predicted user's viewpoint at tth time slot, $r_{u,b}(t)$, $r_{u,e}(t)$ is the bandwidth state of the Sub-6G link and millimeter-wave link, respectively, $z^k(t)$ is the current cache space of agent k, and $\tau^u(t-1)$ is the latency of the current user at previous time slot.

Action space: the action space of the agent k is defined as:

$$\mathbf{A}^k_t = \{l^{k,u}_{n,m}(t), z^{k,u}(t)\} \tag{17}$$

where, $l^{k,u}_{n,m}$ is whether the m Tile of the g image group of the n th 360-degree video is cached to the edge server k, and the number of layers of its cache is denoted as l.

Reward function: the goal of each edge agent is to optimize the video quality of the user's viewpoint by caching layered VR video content at the edge, thus, the reward function can be defined as:

$$R^k_t = \sum_u^U \sum_n^N \sum_m^M f^{k,u}_{n,m}(t)\left(\lambda Q\left(D^{u,b}_{n,m}(t)\right) + \beta Q\left(D^{u,e}_{n,m}(t)\right)\right) \tag{18}$$

Actor network: Agents act according to a policy after receiving a state, which is a probability distribution over actions: $\pi(s_t, a_t) = P(a_t|s_t) \rightarrow [0, 1]$. In real application, due to the diversity of users' request and the variability of different user viewpoints, resulting in a large state–action space, there are large mounts of $(s_t,\ a_t)$ pairs, and convergence difficulties will occur using traditional reinforcement learning algorithms. In order to solve this issue, we employ a neural network (i.e., Actor network) to generate the policy since it receives data straight from observation without the artificial features, and it simply control the policy parameters. The main objective of DRL agents is to maximize the expected cumulative discounted reward from the environment. Using the policy gradient approach, the A3C algorithm is adopted to train its policy. The gradient of the cumulative discounted reward for its policy parameter w is computed as follows:

$$\nabla_w \mathrm{E}_{\pi_w}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] = \mathrm{E}_{\pi_w}\left[\nabla_w \log \pi_w(s,a) A^{\pi_w}(s,a)\right] \tag{19}$$

where γ is the discount factor. The advantage function $A^{\pi_w}(s,a)$ represents the difference between the expected reward of the action taken from the strategy π_w and the expected total reward of taking action a deterministically in state s. The strategy function $\pi_w(s,a)$ adopts a softmax function as its activation function. An entropy regularization term is added to Actor's update rule in A3C, which is derived from the A2C algorithm [42] and aids Actor the converge toward a more effective strategy. In conclusion, Actor's cumulative gradient update is:

$$w \leftarrow w + \sum_t \nabla_w \log \pi_w(s_t, a_t) A(s_t, a_t) + c\nabla_w H(\pi(s_t; w)) \tag{20}$$

Critic network: The Critic network is a component that supports the training of the Actor network in reinforcement learning frameworks, particularly within Actor-Critic architectures [15]. It plays a crucial role by evaluating the actions taken by the Actor network and providing feedback that facilitates the improvement of the policy. In an Actor-Critic model, the Actor network is responsible for selecting actions based on the current state of the environment, whereas the Critic network estimates the value function, typically the state-value function or the action-value function. The Critic evaluates the quality of the actions taken by the Actor by calculating the temporal difference (TD) error, which represents the discrepancy between the predicted value and the actual reward received from the environment.

$$\delta_t = r_t + \gamma V_{AC}(s_{t+1}; w_{AC}) - V_{AC}(s_t; w_{AC}) \tag{21}$$

$$w_v \leftarrow w_v - \sum_t \nabla_{w_v}(r_t + \gamma V^{\pi_w}(s_{t+1}; w_v) - V^{\pi_w}(s_t; w_v))^2 \tag{22}$$

During the testing phase, the Actor network alone generates the final policy by performing actions based on the current state s. However, in order to effectively assess and improve this policy during training, the Critic network is required to estimate the value function, $V^{\pi_w}(s)$, which quantifies the expected total reward for following the policy π_w from state s. The value function $V^{\pi_w}(s)$ serves as a baseline that enables the Critic to compute the advantage function $A(s_t, a_t)$, defined as the difference between the expected return of taking action a_t in state s_t and the value function at s_t:

$$A(s_t, a_t) = Q^{\pi_w}(s_t, a_t) - V^{\pi_w}(s_t) \tag{23}$$

where $Q^{\pi_w}(s_t, a_t)$ represents the action-value function, estimating the expected total reward starting from s_t and taking action a_t, and $V^{\pi_w}(s_t)$ is the value function, as described above. The Critic network plays a pivotal role in this process by estimating the value function $V^{\pi_w}(s)$ through feedback obtained from the environment, typically via empirically observed rewards. The Critic's estimate is crucial for guiding the Actor network in refining its policy. The Critic provides an evaluation of the policy's performance, which is essential for updating the policy in a direction that maximizes cumulative rewards. To train the Critic network, the parameters w_v are updated by minimizing a mean square error (MSE) loss function. The MSE loss quantifies the discrepancy between the predicted value $V^{\pi_w}(s)$ and the target value, typically the return or the observed reward plus the discounted value of the subsequent state:

$$\mathcal{L}_{\text{Critic}} = \mathbb{E}\left[(r_t + \gamma V^{\pi_w}(s_{t+1}) - V^{\pi_w}(s_t))^2\right] \tag{24}$$

where r_t is the reward received at time slot t, $V^{\pi_w}(s_{t+1})$ is the value function estimate for the next state s_{t+1}, and γ is the discount factor. By iteratively minimizing this loss, the Critic network updates its parameters w_v, improving the accuracy of its value function estimation. In turn, the Critic network provides better feedback to the Actor network, enabling it to adjust its policy and maximize the expected long-term reward. And the detail processes of the A3C algorithm are presented as Algorithm 1.

Algorithm 1 A3C algorithm

1 : **Initialize** The learning rate lr_{actor}, lr_{critic} and the hyper-parameters w_{AC} , w_{CR} of actor and critic network, respectively. And the discount coefficient γ.
2 : **INPUT:** The network state S of the MEC agent.
3 : **for** all t in $[1, T]$ **do**
4 : According to $\pi(A_t|S_t; w)$, select the action $A \in \mathbf{A}$.
5 : The MEC select the required tile set and allocated the proper power for decoding and streaming them to VR users through the sub-6G and mmWave link due to the selected action A_t
6 : MEC calculate the immediate reward R_t and obtain the environment state S_{t+1}
7 : Store transition S_t, A_t, R_t, S_{t+1}
8 : Calculate TD error δ_t according to (21)
9 : Update the parameters w_{CR} of the Critic network
10 : Update the parameters w_{AC} of the Actor network
11 : **end For**

4. Experimental evaluation

In this paper, HTC vive is used for experimental evaluation and an eye tracker(aSee) recorded the real movement of mobile VR users. Firstly, the real network environment is simulated through the network simulation platform (NS-3). For the simulation, the 4K resolution VR video source from the dataset provided by work [10] is utilized. We adopted 4×8 tiles in the simulation, and tile-based compression encoding is performed using the open-source HEVC encoder Kvazaar. The intelligent adaptive transmission decision engine makes policy deployment and optimization decisions through the network condition and user state information.

The experimental environment is designed to evaluate the performance of VR video streaming in a multi-user, edge-computing setting. The number of VR users is set 8 to 16, and the system is considered with 6 Multi-Access Edge Computing (MEC) servers. Each MEC server is equipped with 5 GHz GPU computation capacity, while each VR terminal has 2 GHz of GPU computation. The 3 VR videos are adopted in the simulation, each divided into 60 Groups of Pictures (GOP), with the video content spatially tiled in a 4×8 grid, and it supports up to 3 layers of scalable encoding, with the base layer encoded at 2 Mbps and enhancement layers ranging from 0.5 Mbps to 1 Mbps. Communication is facilitated through sub-6 GHz and mmWave links, offering data rates between 100-200 Mbps and 600-800 Mbps, respectively. The maximum tolerable latency is set to 60 ms. The video quality contribution from the base and enhancement layers is weighted at 0.7 and 0.3, respectively, ensuring efficient quality management during streaming. This configuration provides a comprehensive framework for assessing VR video delivery and computational resource allocation in edge computing environments. The detailed simulation parameters are presented in Table 1.

In the simulation experiment, the baseline algorithm AC(Actor-Critic), and State of Art algorithm A3C (Asynchronous Advantage Actor-Critic) are used to solve the formulated POMDP problem [42]. Then the corresponding VR video content is distributed to the VR terminal devices based on the optimization decision results, and finally the user VR video quality is calculated based on the user eye tracker data and the corresponding video content, then the user VR experience is evaluated, and the Peak Signal-to-Noise ratio (PSNR) [36] is used to measure the QoE of VR users.

For user viewpoint prediction, we adopted the cross-user viewpoint prediction, users' behavioral similarity and users' head movement trajectory are jointly used for viewpoint prediction, which is presented in our recent work [43]. In addition, the work [1] provides the viewpoint trajectory of 48 users while watching the VR video, which is used for our viewpoint prediction method. And the detail setting used in the simulation is summarized in Table 1. In order to evaluate the proposed reinforcement learning based scheme in this paper, it is compared with several typical strategies for VR video transmission in terms of convergence, performance of total QoE and average latency, respectively.

(1) Baseline: The user viewpoint is predicted based on GRU model, and the latest video streaming standard MPEG DASH (Dynamic Adaptive Streaming over HTTP) is adopted, and the VR video chunk segment is set as 1 s, to deliver the VR content over sub-6 GHz link with the same system constraints, while the decoding task is completed on the VR terminal only.

Table 1
Parameters of simulation.

Parameters	Value
Number of VR user U	8∼16
Number of MEC servers K	6
Number of VR video N	3
GOP of VR video G	60
Tiling M	4*8
Maximum layer of scalable encoding L	3
The length of GOP $\Delta\tau$	1 s
GPU computation capacity of MEC z^k	5 GHz
GPU computation capacity of VR terminal z^u	2 GHz
Data rate of sub-6 GHz communication link	100–200 Mbps
Data rate of mmWave communication links	600–800 Mbps
Coefficient of video quality contribution λ, β	0.7, 0.3
Coefficient of long term reward discount γ	0.9
Encoding rate of base layer tile R^b	2 Mbps
Encoding rate of enhancement layer tile R^b	0.5–1 Mbps
Maximum tolerable latency τ	60 ms

(2) Proposed-Fixed: dual connection (mmWave link and sub-6 GHz link) and edge computing technologies are applied, However, each user's raw GoP tile set is chosen at random. The compute resource allocation at the edge server and users, as well as the data rate allocation for the encoded enhancement layer tiles, are empirically set in a fixed manner.

(3) Proposed-AC without viewpoint prediction(VP): Here, dual connection (mmWave link and sub-6 GHz link) and edge computing technologies are applied, However, each user's raw GoP tile set is chosen at random. The AC algorithm determines the data rate allocation for the encoded enhancement layer tiles as well as the compute resource allocation at the edge server and users.

(4) Proposed-AC with viewpoint prediction: Dual connection (mmWave link and sub-6 GHz link) and edge computing technologies are applied, and raw GoP tile set is selected for each user based on viewpoint prediction. The data rate allocation for encoded enhancement layer tiles and computation resource allocation at the users and the edge server are obtained by AC algorithm.

(5) Proposed-A3C without viewpoint prediction: Dual connection (mmWave link and sub-6 GHz link) and edge computing technologies are applied, However, each user's raw GoP tile set is chosen at random. The data rate allocation for encoded enhancement layer tiles and computation resource allocation at the users and the edge server are obtained by the A3C algorithm.

(6) Proposed-A3C with viewpoint prediction: Dual connection (mmWave link and sub-6 GHz link) and edge computing technologies are applied, and raw GoP tile set is selected for each user based on viewpoint prediction. The A3C algorithm determines the computation resource allocation at the edge server and user as well as the data rate allocation for encoded enhancement layer tiles.

Fig. 6 displays the cross-user viewpoint prediction performance. Among these, the method adopted K Nearest Neighbor (KNN) viewpoint prediction in work [44] (KNN-based viewpoint prediction, KVP), the GRU (Gated Recurrent Unit) prediction method based on head movement data and the traditional linear regression (LR) prediction method based on historical viewpoint trajectory are chosen for comparison. In order to ensure a fair comparison, the cross-user viewpoint prediction approach suggested in the paper's K value, $K = 5$, is consistent with the K value found in the work [44]. The accuracy trend and prediction time of several viewpoint prediction algorithms are displayed in Fig. 6. It is clear that the overall trend of prediction accuracy for the four methodologies decreases over time. However, the prediction accuracy based solely on the historical trajectory of the viewpoint is the lowest, especially when the prediction period is long, and the accuracy drops quickly because the user's viewpoint moves in accordance with the video content. When the prediction time is short, the KVP approach maintains a high accuracy. Similar to the linear regression approach, it experiences a sharp decline in prediction accuracy over time. The cross-user viewpoint prediction approach outperforms the KVP method by at least 6% on average, and when the prediction duration is long (5 s), the prediction accuracy can remain over 80%.

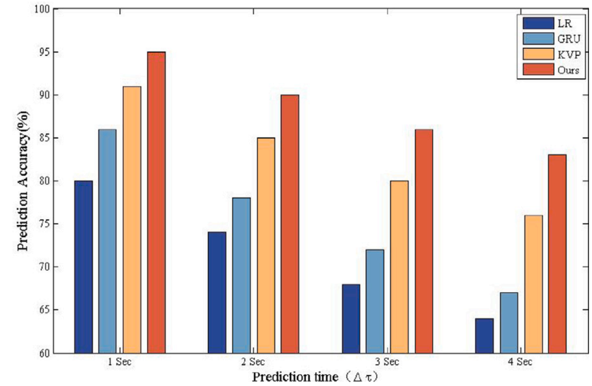


Fig. 6. The performance of cross-user viewpoint prediction for scalable VR video streaming.

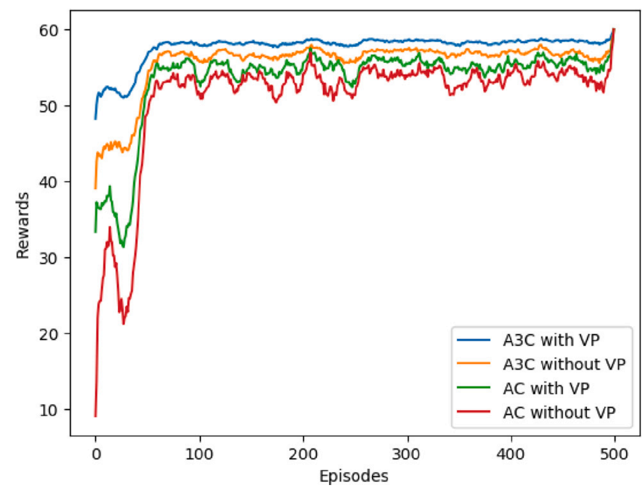
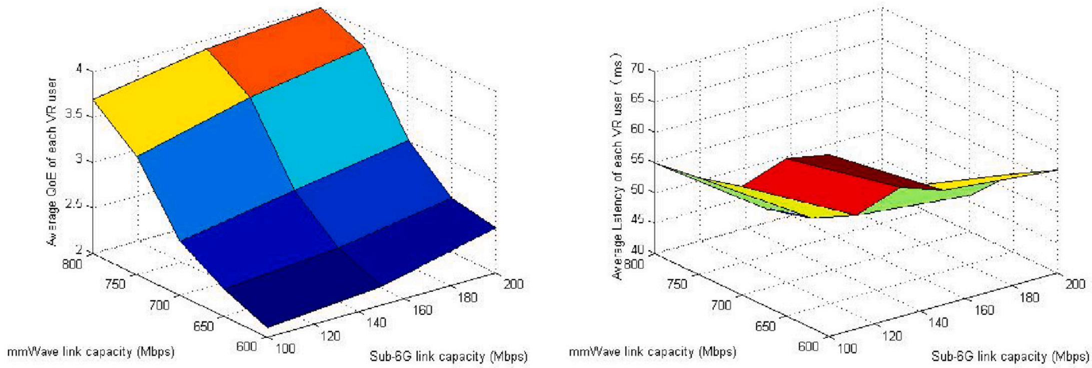


Fig. 7. The total reward of training process at each episode with A3C/AC learning algorithm.

Fig. 7 shows the total reward of training process at each episode with A3C/AC learning algorithm. The total reward is an important metric as it reflects the effectiveness of the algorithms in learning the optimal policy for resource allocation and content delivery in the VR streaming scenario. From the figure, we can see that the A3C algorithm exhibit a faster learning curve, indicating that it can adapt more quickly to the environment and improve its decision-making process over time, while the AC algorithm show a slower convergence rate, which could be attributed to its simpler update mechanism compared to A3C. The figure also reveals the stability and consistency of the learning process, with fluctuations in the reward indicating the exploration and exploitation phases of the learning process. The A3C algorithm, with its distributed learning approach, appears to be more effective in optimizing the total reward, suggesting a better adaptation to the dynamic VR streaming environment. Notice that the performance of A3C/AC learning algorithm work with and without VP also shows that the VP improve the reward of DRL based algorithm, since with the better knowledge of user viewpoint, the optimization algorithm selects the corresponding content (especially, the enhancement layer tiles) for the specific VR user, which improves video quality in the field of viewpoint, i.e the reward defined in the solution section.

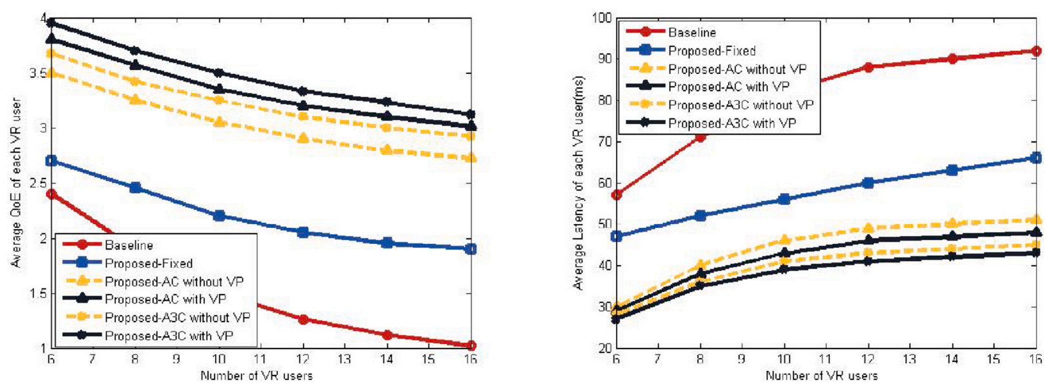
Fig. 8 presents the average QoE and latency of VR User with our proposed A3C algorithm with VP under different sub-6G and mmWave link



(a) The average QoE of VR User with our proposed A3C algorithm with VP under different sub-6G and mmWave link capacity

(b) The average latency of VR User with our proposed A3C algorithm with VP under different sub-6G and mmWave link capacity

Fig. 8. The average QoE and latency of VR User with our proposed A3C algorithm with VP under different sub-6G and mmWave link capacity.



(a) The average QoE of VR User with different algorithms with increasing of number of VR users

(b) The average latency of VR User with different algorithms with increasing of number of VR users

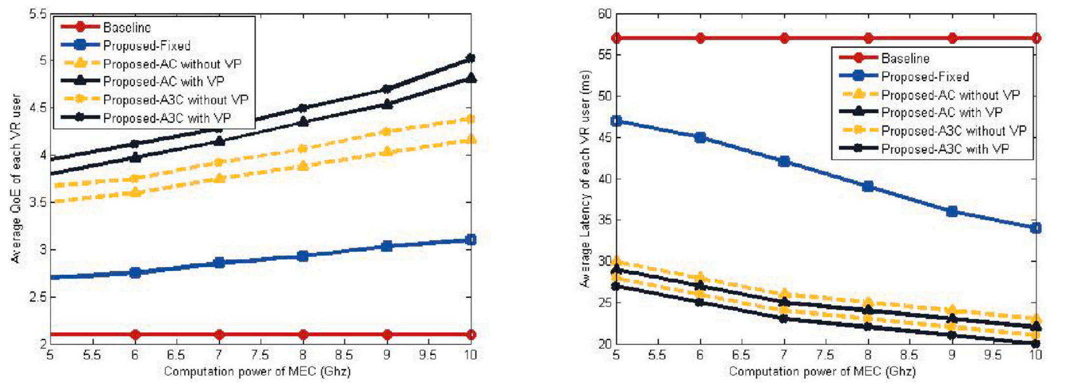
Fig. 9. The average QoE and latency of VR User with different algorithms with increasing of number of VR users.

capacity. As the capacity of both sub-6G and mmWave links increases, there is a corresponding increase in the average QoE, indicating that higher data rates can support better video quality and user experience. The latency, on the other hand, would ideally decrease with higher link capacities, as more data can be transmitted in less time, reducing waiting times for content delivery. The A3C algorithm's performance in terms of QoE and latency may be consistent across different link capacities, demonstrating its robustness and effectiveness in various network conditions. From Fig. 8(b), the results shows the A3C algorithm successfully optimizes resource allocation to deliver a high QoE even under varying network conditions, with the best performance achieved under higher link capacities. From Fig. 8(b), we can see that the latency is also effectively managed by the A3C algorithm, with lower latency values observed as link capacities increase, contributing to a smoother and more responsive VR experience.

Fig. 9 shows the average QoE and latency of VR User with different algorithms with increasing number of VR users. From Fig. 9(a), we can see that with the increasing number of VR users, the system's resources become more strained, which could lead to a decrease in QoE if not managed effectively. From Fig. 9(a), the A3C algorithm might maintain or only slightly reduce the average QoE, indicating its ability to distribute resources efficiently among users, ensuring a fair and satisfactory experience for all. Fig. 9(b) shows the latency performance with increase users, the latency should remain within acceptable limits for a good VR experience. The results show that the A3C algorithm's impact on latency management is notable, with the results suggesting that it can prevent significant latency increases as the number of users increases, which is crucial for maintaining a high-quality VR experience. In addition, the result also shows the QoE and the average latency of the algorithms without the VP (i.e, proposed-AC without VP and proposed-A3C without VP) has average 6.2% and 3.6% gap between the algorithms with the VP (i.e, proposed-AC with VP and proposed-A3C with VP), which proves that the DRL based algorithms perform better with proactive knowledge of user viewpoint.

The average QoE and latency of VR User with different algorithms with increasing MEC computation power is presented in Fig. 10. As shown in Fig. 10(a), an increase of MEC computation power leads to a significant improvement in average QoE, allowing for more complex processing and higher video quality. Notice that the QoE and latency of the baseline scheme are fixed, since the baseline scheme does not apply decoding at the MEC nodes. As shown in Fig. 10(b), latency of other algorithms are reduced with greater computation power, enabling faster processing and decision-making, which is essential for maintaining a responsive VR experience. The A3C algorithm leverages the enhanced computation power to optimize user experience, demonstrating its capability to adapt to different computational environments.

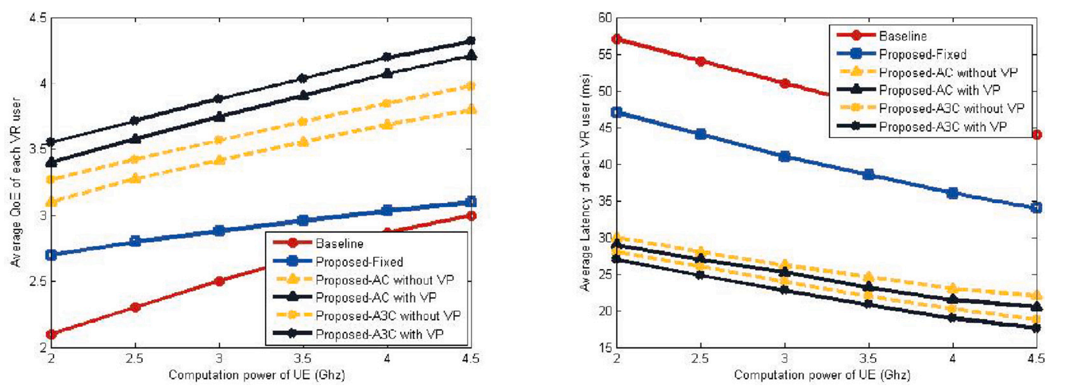
The average QoE and latency of VR User with different algorithms with increasing UE computation power is shown in Fig. 11. As shown in Fig. 11(a), the increase of UE computation power results in a notable increase in average QoE, as the user equipment can handle more demanding tasks locally, reducing the reliance on network transmission. As shown in Fig. 11(b), The latency decreases as UE computation power increases, suggesting that local processing at the user's device can mitigate network-related latency. The A3C algorithm effectively coordinates between the MEC and UE computation power to achieve an



(a) The average QoE of VR User with different algorithms with increasing of MEC computation power

(b) The average latency of VR User with different algorithms with increasing of MEC computation power

Fig. 10. The average QoE and latency of VR User with different algorithms with increasing MEC computation power.



(a) The average QoE of VR User with different algorithms with increasing of UE computation power

(b) The average latency of VR User with different algorithms with increasing of UE computation power

Fig. 11. The average QoE and latency of VR User with different algorithms with increasing UE computation power.

optimal balance, ensuring a high-quality VR experience with minimal latency.

5. Discussion

The experimental evaluation conducted in this study has provided valuable insights into the performance of the proposed A3C and AC algorithms in the scalable VR video streaming. Based on the evaluation, the discussion is summarized as follows:

A. Learning Efficiency and Adaptability

The A3C algorithm has demonstrated superior learning efficiency compared to the AC algorithm. The asynchronous nature of A3C, which involves multiple agents learning independently and sharing experiences periodically, has proven to be effective in quickly adapting to the dynamic environment of VR video streaming. The faster convergence and higher total rewards achieved by A3C indicate that it can effectively optimize resource allocation to improve the overall streaming performance.

B. Impact of Network Link Capacities

The performance metrics, including QoE and latency, are significantly influenced by the capacities of the sub-6G and mmWave links. As expected, higher link capacities lead to enhanced QoE and reduced latency, allowing for the transmission of more data-rich VR content with minimal latency. The A3C algorithm's ability to adapt to varying link capacities showcases its robustness and potential for real-world application in diverse network scenarios.

C. Scalability with Increasing number of VR User

One of the critical findings of this study is the scalability of the A3C algorithm in handling an increasing number of VR users. Despite the growing demand for resources, the algorithm manages to maintain a high level of QoE and controlled latency. This scalability is crucial for the future of VR services, which are expected to serve a large user base simultaneously.

D. Implications for VR Streaming Services

The results of this study have significant implications for the development of VR streaming services. The A3C algorithm's ability to optimize resource allocation and maintain low latency under various conditions makes it a promising candidate for implementation in next-generation VR platforms. For fast implementation of the proposed method, the algorithm can be deployed on each edge node in the form of a virtual machine/container.

E. Future Research Directions

While the A3C algorithm has shown promising results, there are areas for further research and improvement. Future work could focus on refining the algorithm to handle more complex scenarios, such as those with fluctuating network conditions or a wider variety of VR content types. Additionally, exploring the integration of the A3C algorithm with other emerging technologies, such as network slicing and advanced caching strategies, may lead to more efficient and reliable VR streaming solutions [7,8,12,42].

6. Conclusion

This paper addresses the challenges in mobile VR video transmission by proposing a joint optimization method for layering and computation allocation based on DRL. Through simulation experiments, we have demonstrated that the proposed method effectively utilizes network and computational resources while ensuring user experience. The A3C algorithm performs well under different network conditions and computational capabilities, particularly in resource-constrained scenarios, significantly improving the average QoE for VR users and reducing latency.

CRediT authorship contribution statement

Junchao Yang: Writing – original draft, Methodology. **Hui Zhang:** Data curation, Conceptualization. **Wenxin Jiao:** Funding acquisition, Formal analysis. **Zhiwei Guo:** Writing – review & editing, Visualization. **Fayez Alqahtani:** Project administration, Investigation. **Amr Tolba:** Software, Resources. **Yu Shen:** Validation, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Han Xiao, Changqiao Xu, Zichen Feng, Renjie Ding, Shujie Yang, Lujie Zhong, Jie Liang, Gabriel-Miro Muntean, A transcoding-enabled 360° VR video caching and delivery framework for edge-enhanced next-generation wireless networks, IEEE J. Sel. Areas Commun. 40 (5) (2022) 1615–1631.

[2] Liang Zhao, Zhenguo Bi, Ammar Hawbani, Keping Yu, Yan Zhang, Mohsen Guizani, ELITE: An intelligent digital twin-based hierarchical routing scheme for softwarized vehicular networks, IEEE Trans. Mob. Comput. (2022).

[3] Guntur Ravindra, Wei Tsang Ooi, On tile assignment for region-of-interest video streaming in a wireless LAN, in: Baochun Li, Shervin Shirmohammadi (Eds.), Network and Operating System Support for Digital Audio and Video Workshop, NOSSDAV '12, Toronto, on, Canada, June 7-8, 2012, ACM, 2012, pp. 59–64.

[4] Ana De Abreu, Cagri Ozcinar, Aljosa Smolic, Look around you: Saliency maps for omnidirectional images in VR applications, in: Ninth International Conference on Quality of Multimedia Experience, QoMEX 2017, Erfurt, Germany, May 31 - June 2, 2017, IEEE, 2017, pp. 1–6.

[5] Sharare Zehtabian, Mina Razghandi, Ladislau Bölöni, Damla Turgut, Predictive caching for AR/VR experiences in a household scenario, in: International Conference on Computing, Networking and Communications, ICNC 2020, Big Island, HI, USA, February 17-20, 2020, IEEE, 2020, pp. 591–595.

[6] Chen Chen, Zhan Liao, Ying Ju, Ci He, Keping Yu, Shaohua Wan, Hierarchical domain-based multi-controller deployment strategy in SDN-enabled space-air-ground integrated network, IEEE Trans. Aerosp. Electron. Syst. (2022).

[7] Kai Jiang, Yue Cao, Yujie Song, Huan Zhou, Shaohua Wan, Xu Zhang, Asynchronous federated and reinforcement learning for mobility-aware edge caching in IoV, IEEE Internet Things J. 11 (9) (2024) 15334–15347.

[8] Zhenyang Shu, Xiaoheng Deng, Leilei Wang, Jinsong Gui, Shaohua Wan, Honggang Zhang, Geyong Min, Relay-assisted edge computing framework for dynamic resource allocation and multiple-access task processing in Digital Divide Regions, IEEE Internet Things J. 11 (21) (2024) 35724–35741.

[9] Afshin Taghavi Nasrabadi, Anahita Mahzari, Joseph D. Beshay, Ravi Prakash, Adaptive 360-degree video streaming using scalable video coding, in: Qiong Liu, Rainer Lienhart, Haohong Wang, Sheng-Wei "Kuan-Ta" Chen, Susanne Boll, Yi-Ping Phoebe Chen, Gerald Friedland, Jia Li, Shuicheng Yan (Eds.), Proceedings of the 2017 ACM on Multimedia Conference, MM 2017, Mountain View, CA, USA, October 23-27, 2017, ACM, 2017, pp. 1689–1697.

[10] Chenglei Wu, Zhihao Tan, Zhi Wang, Shiqiang Yang, A dataset for exploring user behaviors in VR spherical video streaming, in: Proceedings of the 8th ACM on Multimedia Systems Conference, MMSys 2017, Taipei, Taiwan, June 20-23, 2017, ACM, 2017, pp. 193–198.

[11] Jounsup Park, Jenq-Neng Hwang, Hung-Yu Wei, Cross-layer optimization for VR video multicast systems, in: IEEE Global Communications Conference, GLOBECOM 2018, Abu Dhabi, United Arab Emirates, December 9-13, 2018, IEEE, 2018, pp. 206–212.

[12] Ying Chen, Jie Zhao, Jintao Hu, Shaohua Wan, Jiwei Huang, Distributed task offloading and resource purchasing in NOMA-enabled mobile edge computing: Hierarchical game theoretical approaches, ACM Trans. Embed. Comput. Syst. 23 (1) (2024) 2:1–2:28.

[13] Na Su, Jun-Bo Wang, Yijian Chen, Hongkang Yu, Changfeng Ding, Yijin Pan, Joint rendering offloading and resource allocation optimization for MEC-assisted VR systems, IEEE Wirel. Commun. Lett. 13 (4) (2024) 949–953.

[14] Georgios Papaioannou, Iordanis Koutsopoulos, Tile-based caching optimization for 360° videos, in: Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing, Mobihoc 2019, Catania, Italy, July 2-5, 2019, ACM, 2019, pp. 171–180.

[15] Fengxian Guo, F. Richard Yu, Heli Zhang, Hong Ji, Victor C.M. Leung, Xi Li, An adaptive wireless virtual reality framework in future wireless networks: A distributed learning approach, IEEE Trans. Veh. Technol. 69 (8) (2020) 8514–8528.

[16] Lei Feng, Zhixiang Yang, Yang Yang, Xiaoyu Que, Kai Zhang, Smart mode selection using online reinforcement learning for VR broadband broadcasting in D2D assisted 5G HetNets, IEEE Trans. Broadcast. 66 (2) (2020) 600–611.

[17] Haodan Zhang, Yixuan Ban, Zongming Guo, Ken Chen, Xinggong Zhang, RAM360: robust adaptive multi-layer 360 ° video streaming with Lyapunov optimization, IEEE Trans. Multimedia 25 (2023) 4225–4239.

[18] Sabyasachi Gupta, Jacob Chakareski, Petar Popovski, Mmwave networking and edge computing for scalable 360° video multi-user virtual reality, IEEE Trans. Image Process. 32 (2023) 377–391.

[19] Jacob Chakareski, Mahmudur Khan, Murat Yuksel, Toward enabling next-generation societal virtual reality applications for virtual human teleportation: A novel future system concept and computation-communication-signal representation trade-offs, IEEE Signal Process. Mag. 39 (5) (2022) 22–41.

[20] Fanyi Duanmu, Eymen Kurdoglu, S. Amir Hosseini, Yong Liu, Yao Wang, Prioritized buffer control in two-tier 360 video streaming, in: Proceedings of the Workshop on Virtual Reality and Augmented Reality Network, VR/AR Network@SIGCOMM 2017, Los Angeles, CA, USA, August 25, 2017, ACM, 2017, pp. 13–18.

[21] Mathias Almquist, Viktor Almquist, Vengatanathan Krishnamoorthi, Niklas Carlsson, Derek L. Eager, The prefetch aggressiveness tradeoff in 360° video streaming, in: Pablo César, Michael Zink, Niall Murray (Eds.), Proceedings of the 9th ACM Multimedia Systems Conference, MMSys 2018, Amsterdam, the Netherlands, June 12-15, 2018, ACM, 2018, pp. 258–269.

[22] Madhukar Budagavi, John Furton, Guoxin Jin, Ankur Saxena, Jeffrey Wilkinson, Andrew Dickerson, 360 degrees video coding using region adaptive smoothing, in: 2015 IEEE International Conference on Image Processing, ICIP 2015, Quebec City, QC, Canada, September 27-30, 2015, IEEE, 2015, pp. 750–754.

[23] Shichao Xia, Zhixiu Yao, Yun Li, Shiwen Mao, Online distributed offloading and computing resource management with energy harvesting for heterogeneous MEC-enabled IoT, IEEE Trans. Wirel. Commun. 20 (10) (2021) 6743–6757.

[24] Yuanwei Zhu, Yakun Huang, Xiuquan Qiao, Zhijie Tan, Boyuan Bai, Huadong Ma, Schahram Dustdar, A semantic-aware transmission with adaptive control scheme for volumetric video service, IEEE Trans. Multimedia 25 (2023) 7160–7172.

[25] Pantelis Maniotis, Nikolaos Thomos, Viewport-aware deep reinforcement learning approach for 360° video caching, IEEE Trans. Multimedia 24 (2022) 386–399.

[26] Niklas Carlsson, Derek L. Eager, Cross-user similarities in viewing behavior for 360° video and caching implications, ACM Trans. Multimedia Comput. Commun. Appl. 19 (5) (2023) 152:1–152:24.

[27] Nikita Smirnov, Sven Tomforde, Real-time rate control of WebRTC video streams in 5G networks: Improving quality of experience with deep reinforcement learning, J. Syst. Archit. 148 (2024) 103066.

[28] Hang Zhang, Jinsong Wang, Zening Zhao, Zhao Zhao, A survey of edge caching security: Framework, methods, and challenges, J. Syst. Archit. 158 (2025) 103306.

[29] Junchao Yang, Feng Lin, Chinmay Chakraborty, Keping Yu, Zhiwei Guo, Anh-Tu Nguyen, Joel J.P.C. Rodrigues, A parallel intelligence-driven resource scheduling scheme for digital twins-based intelligent vehicular systems, IEEE Trans. Intell. Veh. 8 (4) (2023) 2770–2785.

[30] Yanwei Liu, Jinxia Liu, Antonios Argyriou, Liming Wang, Zhen Xu, Rendering-aware VR video caching over multi-cell MEC networks, IEEE Trans. Veh. Technol. 70 (3) (2021) 2728–2742.

[31] Alok Choudhury, Manojit Ghose, Akhirul Islam, Yogita, Machine learning-based computation offloading in multi-access edge computing: A survey, J. Syst. Archit. 148 (2024) 103090.

[32] Patrice Rondao-Alface, Jean-François Macq, Nico Verzijp, Interactive omnidirectional video delivery: A bandwidth-effective approach, Bell Labs Tech. J. 16 (4) (2012) 135–147.

[33] Jianmei Dai, Zhilong Zhang, Shiwen Mao, Danpu Liu, A view synthesis-based 360° VR caching system over MEC-enabled C-RAN, IEEE Trans. Circuits Syst. Video Technol. 30 (10) (2020) 3843–3855.

[34] Mugen Peng, Tony Q.S. Quek, Guoqiang Mao, Zhiguo Ding, Chonggang Wang, Artificial-intelligence-driven fog radio access networks: Recent advances and future trends, IEEE Wirel. Commun. 27 (2) (2020) 12–13.

[35] Pantelis Maniotis, Eirina Bourtsoulatze, Nikolaos Thomos, Tile-based joint caching and delivery of 360 videos in heterogeneous networks, IEEE Trans. Multimedia 22 (9) (2020) 2382–2395.

[36] Taoyu Yang, Zengjie Tan, Yuanyuan Xu, Shuwen Cai, Collaborative edge caching and transcoding for 360° video streaming based on deep reinforcement learning, IEEE Internet Things J. 9 (24) (2022) 25551–25564.

[37] Ziwen Ye, Qing Li, Xiaoteng Ma, Dan Zhao, Yong Jiang, Lianbo Ma, Bo Yi, Gabriel-Miro Muntean, VRCT: A viewport reconstruction-based 360° video caching solution for tile-adaptive streaming, IEEE Trans. Broadcast. 69 (3) (2023) 691–703.

[38] Yixuan Ban, Lan Xie, Zhimin Xu, Xinggong Zhang, Zongming Guo, Yue Wang, CUB360: Exploiting cross-users behaviors for viewport prediction in 360 video adaptive streaming, in: 2018 IEEE International Conference on Multimedia and Expo, ICME 2018, San Diego, CA, USA, July 23-27, 2018, IEEE Computer Society, 2018, pp. 1–6.

[39] Vincent Sitzmann, Ana Serrano, Amy Pavel, Maneesh Agrawala, Diego Gutierrez, Belén Masiá, Gordon Wetzstein, Saliency in VR: how do people explore virtual environments? IEEE Trans. Vis. Comput. Graph. 24 (4) (2018) 1633–1642.

[40] Junchao Yang, Zhiwei Guo, Jiangtao Luo, Yu Shen, Keping Yu, Cloud-edge-end collaborative caching based on graph learning for cyber-physical virtual reality, IEEE Syst. J. 17 (4) (2023) 5097–5108.

[41] Qi Cheng, Hangguan Shan, Weihua Zhuang, Lu Yu, Zhaoyang Zhang, Tony Q.S. Quek, Design and analysis of MEC- and proactive caching-based 360° mobile VR video streaming, IEEE Trans. Multimedia 24 (2022) 1529–1544.

[42] Chunlin Li, Kun Jiang, Yong Zhang, Lincheng Jiang, Youlong Luo, Shaohua Wan, Deep reinforcement learning-based mining task offloading scheme for intelligent connected vehicles in UAV-aided MEC, ACM Trans. Des. Autom. Electr. Syst. 29 (3) (2024) 54:1–54:29.

[43] Junchao Yang, Ali Kashif Bashir, Zhiwei Guo, Keping Yu, Mohsen Guizani, Intelligent cache and buffer optimization for mobile VR adaptive transmission in 5G edge computing networks, Digit. Commun. Netw. 10 (5) (2024) 1234–1244.

[44] Siyi Liao, Jun Wu, Jianhua Li, Kostromitin Konstantin, Information-centric massive IoT-based ubiquitous connected VR/AR in 6G: A proposed caching consensus approach, IEEE Internet Things J. 8 (7) (2021) 5172–5184.