Joint Optimization of Network Resources with Multi-Agent Reinforcement Learning in Multi-User Virtual Reality Networks

Xinyu Wan, *Student Member, IEEE* and Abbas Jamalipour, *Fellow, IEEE*

***Abstract*—Multi-user virtual reality (VR) networks face significant challenges in wireless environments due to their demanding requirements for low latency, high bandwidth, and efficient resource allocation across concurrent users with heterogeneous needs. These challenges are further complicated in dynamic environments where users move and interact, creating varying load patterns and potential resource contention. Traditional optimization approaches often struggle to balance the competing objectives of maximizing users' quality of experience (QoE) while efficiently utilizing limited network resources. This paper utilizes a comprehensive framework employing Proximal Policy Optimization-based Multi-Agent Markov Decision Process (PPO-MAMDP) to jointly optimize communication, computing, and caching resources in multi-user wireless VR networks. Our approach formulates the problem as a social welfare maximization that balances VR user QoE with small base station (SBS) resource efficiency through coordinated decision-making across multiple agents. The framework incorporates actor-critic networks with generalized advantage estimation (GAE) for enhanced training stability and faster convergence when navigating complex decision spaces. Simulation results demonstrate that the proposed PPO-MAMDP algorithm outperforms baseline approaches across multiple metrics, achieving higher performance in both user experience and system efficiency. The algorithm demonstrates robust adaptability and improved stability as network density increases. Our approach uniquely maintains balanced performance across both system and user metrics, making it particularly suitable for next-generation multi-user VR applications.**

***Index Terms*—wireless communication, virtual reality, tile streaming, resource allocation, multi-user**

I. Introduction

VIRTUAL Reality (VR) technology has gained significant attention with rapid development across entertainment, healthcare, tourism, and education sectors [1]. The global VR market is expected to grow substantially, driven by advances in both hardware and network infrastructure. VR applications face unique challenges compared to conventional communication services, especially regarding strict latency and high-resolution video quality requirements [2] needed to maintain user immersion and prevent motion sickness [3].

Recent advances address these challenges through multiple technical approaches. Viewport-based streaming techniques [4] transmit only the user's visible field of view (FoV) rather than the entire 360-degree environment, evolving into sophisticated tile-based streaming with optimized network routing [5]. Viewport prediction using deep learning [6] and edge computing [7] enables proactive content delivery. Dynamic transcoding [8], [9] balances quality expectations against bandwidth constraints through real-time bitrate adaptation. Energy efficiency has been addressed through intelligent resource scheduling and adaptive foveated rendering [10], [11] balancing visual quality with power consumption. Multi-user VR introduces additional complexity where latency differences impact collaboration and fairness [12], requiring synchronized experiences through techniques like buffer-nadir-based multicast [13]. Infrastructure management involves small base stations (SBSs) handling bandwidth allocation [14], energy efficiency, and cache capacity [15], with recent hierarchical multicast approaches [16] optimizing resource sharing. Deep reinforcement learning (DRL) has progressed from Deep Q-Network (DQN) to sophisticated frameworks including Deep Deterministic Policy Gradient (DDPG) [17], Proximal Policy Optimization (PPO), and Double Deep Q-Network (DDQN) [18] for resource allocation. Multi-agent reinforcement learning (MARL) approaches [19], [20] demonstrate potential for handling complex distributed optimization.

Despite these advances, current research largely focuses on individual users' quality of experience (QoE), with system-wide performance in multi-user scenarios remaining relatively unexplored. Existing approaches typically optimize isolated network aspects such asbandwidth allocation, computation offloading, or cache managementf that fails to capture critical interdependencies between resources. Single-agent reinforcement learning methods struggle with multi-user coordination where individual decisions affect shared resources. While MARL frameworks exist, they typically focus on either user-centric QoE metrics or system-level efficiency rather than explicitly balancing both objectives. Furthermore, the fairness implications of latency variations in multi-user scenarios have received limited attention, particularly regarding inter-user delay synchronization that is essential for seamless collaborative VR experiences. Finally, the effective integration of device-to-device (D2D) communication with centralized resource allocation frameworks remains inadequately explored.

In this context, this paper proposes a comprehensive multi-agent reinforcement learning framework for resource allocation in wireless VR networks. The framework jointly optimizes communication, computation, caching, rendering location, and quality adaptation while balancing system-wide resource utilization with individual user experience. The major contributions are summarized as follows:

- A joint optimization framework combining MARL with

The authors are with the School of Electrical and Computer Engineering, The University of Sydney, Sydney NSW 2006, Australia. E-mail: xinyu.wan@sydney.edu.au; a.jamalipour@ieee.org

a greedy matching algorithm for D2D communication is proposed. The framework coordinates bandwidth allocation, cache management, rendering decisions, and quality adaptation across five resource dimensions.
- An innovative PPO-based MARL model is developed that flexibly adapts to different user requirements and network conditions. The model employs heterogeneous action heads for mixed discrete-continuous action spaces, dual-timescale optimization, and group-based fairness mechanisms incorporating inter-user delay synchronization.
- A social welfare maximization formulation is introduced that explicitly balances user QoE metrics such as video quality, inter-user delay, energy consumption, and SBS utility factors including resource utilization efficiency, effectively capturing the complex relationship between user satisfaction and system sustainability.
- A greedy matching algorithm for D2D communication is designed that efficiently pairs content providers with requesters based on distance, content availability, and channel quality, enabling peer-to-peer content sharing integrated with centralized MARL resource allocation.

The rest of this paper is organized as follows. Section II reviews related work in wireless VR networks and distinguishes our contributions from existing approaches. Section III presents the system model and problem formulation. Section IV details the proposed PPO-based MARL framework and solution methodology. Section V provides comprehensive simulation results and performance analysis under varying network conditions. Section VI concludes the paper and discusses future research directions.

II. Related Work

In this section, we review related research across key areas of VR network optimization, including tile-based streaming approaches, dynamic transcoding and quality adaptation, energy-efficient resource management, and multi-user coordination strategies. We also examine the integration of machine learning techniques, particularly reinforcement learning methods, in addressing complex VR network challenges.

A. Multi-Agent Reinforcement Learning in VR Networks

MARL effectively addresses VR network optimization through distributed coordination. [21] proposes distributed task scheduling using multi-agent PPO, treating edge networks as cooperative systems. [34] introduces MARLISE employing DQN and PPO variants for dynamic resource scaling. [20] specifically addresses asymmetric transmission through asynchronous hybrid reinforcement learning for VR streaming.

Advanced architectures enable sophisticated coordination among multiple agents. [35] presents digital twin frameworks leveraging multi-agent DRL for heterogeneous tasks, while [36] incorporates graph attention mechanisms enhancing topology awareness in networks. [37] introduces Asynchronous Actors Hybrid Critic for optimizing computation offloading and channel assignment in asymmetric transmissions.

Trust and collaboration mechanisms address complex interactions. [22] explores trust-based collaboration for collaborative rendering among edge servers. Single-agent approaches provide foundational techniques: [19] proposes DQN-based resource allocation for XR applications, [18] investigates DDQN for network slicing, and [17] employs DDPG for offloading optimization. However, multi-agent frameworks better capture the distributed nature of multi-user VR networks.

B. Edge Computing Infrastructure and Resource Optimization

Mobile edge computing enables untethered VR experiences. [9] proposes frameworks leveraging edge computing for adaptive content delivery through real-time transcoding. [38] extends this with scalable multi-layer 360° video tiling and viewport-adaptive allocation. Dynamic service placement strategies are critical: [39] proposes EDSP-Edge jointly optimizing network access, service placement, and video resolution, while [40] addresses social VR with algorithms considering computational requirements and social interactions.

Joint optimization frameworks address broader challenges. [41] combines infrastructure and peer-to-peer communication modes. [42] introduces digital twin concepts for predictive allocation. [43] proposes knowledge-driven belief propagation replacing complex operations with neural networks. [44] presents graph-based joint computing and communication scheduling, demonstrating integrated representation that reduces delay. Cross-layer optimization combines physical and application decisions. However, existing frameworks typically optimize resources subsets rather than joint optimization across all types including rendering location and quality.

Energy optimization remains critical for power-constrained devices. [23] formulates joint rendering offloading balancing computational and communication energy. [11] develops deep reinforcement learning for adaptive foveated rendering. [24] proposes tile-based collaborative rendering for UAV-enabled VR reducing energy under latency constraints. [45] applies model partitioning to VR rendering pipelines. [46] introduces energy-efficient viewport prediction for point clouds. [47] presents computing power networking frameworks enabling resource flow using proximal policy optimization.

Dynamic resource allocation leverages various advanced DRL techniques. [13] develops cost-efficient content delivery with dynamic transcoding. The authors in [48] demonstrate DQN-based approaches for near-optimal network slicing while [49] proposes unified solutions for overfilling, offloading, and subband allocation. [50] addresses QoE-aware volumetric video caching through regularization-based optimization.

C. Multi-User Coordination and Content Delivery

Multi-user VR requires sophisticated coordination for synchronization and fairness. [12] investigates quality optimization for collaborative VR systems. [29] develops hierarchical multicast techniques exploiting shared FoVs. [30] addresses multiplayer interactive games jointly optimizing computing, bandwidth, and processing while minimizing inter-player delay. [51] introduces hierarchical federated DRL for cell-free multi-group broadcast. [52] proposes deadline-aware scheduling prioritizing frames by perceptual importance, while [53] develops cross-layer congestion control for 5G edge networks.

TABLE I: Our Contributions in Contrast to the State-of-the-Art

	Our work	[21], [22]	[17], [20]	[11], [23], [24]	[25], [26]	[9], [13]	[27], [28]	[18], [19]	[12], [29], [30]	[31]–[33]
Multi-User VR Systems	✓	✓	✓	×	×	✓	✓	×	✓	✓
Communication Resource Allocation	✓	✓	✓	×	✓	✓	✓	✓	✓	✓
Computation Resource Management	✓	✓	✓	✓	✓	✓	×	✓	×	×
Energy Consumption Optimization	✓	×	×	✓	×	×	×	×	×	×
Content Caching Strategy	✓	✓	×	×	✓	×	×	×	×	✓
Quality Adaptation	✓	×	×	×	✓	✓	✓	×	✓	✓
Rendering Location Strategy	✓	✓	×	✓	✓	×	×	×	✓	✓
Joint Optimization Framework	✓	✓	✓	✓	✓	✓	✓	✓	✓	×
QoE/QoS Optimization	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

D2D communication can effectively reduce system load. [31] investigates sidelink-aided multicast for multi-quality tiled transmission. [33] proposes DIBR-based collaborative computation for multiplayer VR, enabling view synthesis through cooperative rendering among users. [32] develops edge-device collaborative frameworks for multi-user interactive VR, jointly optimizing rendering positions and bandwidth. However, existing approaches focus on transmission protocols or rendering coordination, with limited exploration of efficient peer matching integrated with centralized resource allocation.

Tile-based streaming and viewport prediction remain fundamental. [26] addresses privacy concerns through camouflaged tile requests. [25] combines GRU-based FoV prediction with PPO for tile reuse. Meta-learning and contextual bandits [54], [55] demonstrate improved robustness. [56] explores 6G integration with software-defined networking. System-wide optimization addresses comprehensive challenges. [27] tackles fairness through max-min QoE optimization. [28] develops human-centric utility measures. [57], [58] demonstrate digital twin integration and attention-based QoE optimization. [59] presents distributed learning for metaverse over wireless networks with variants addressing network heterogeneity.

Despite progress, existing approaches face fundamental limitations. First, frameworks typically optimize individual aspects or limited resource combinations— [13], [48] address bandwidth and slicing, [11], [23] focus on computation and energy while few jointly optimize all five dimensions: communication, computation, caching, rendering location, and quality. Second, although MARL has been applied [21], [34], existing systems focus on either user-centric QoE or system efficiency, lacking frameworks explicitly balancing both through social welfare maximization. Third, while works [27], [28] consider QoE fairness, they overlook inter-user delay synchronization critical for collaborative experiences. Finally, while D2D shows promise [31], [53], integration of efficient peer matching with centralized MARL frameworks remains unexplored. Different from above works, our research addresses these through a comprehensive PPO-based multi-agent framework jointly optimizing five dimensions while explicitly balancing user QoE and system efficiency through social welfare maximization, integrating inter-user delay fairness with greedy matching for D2D content sharing. Main contributions are summarized in Table I.

III. System Model

A. Network Model

We consider a wireless VR network environment including a cloud server, a total number of M SBSs, and N number of

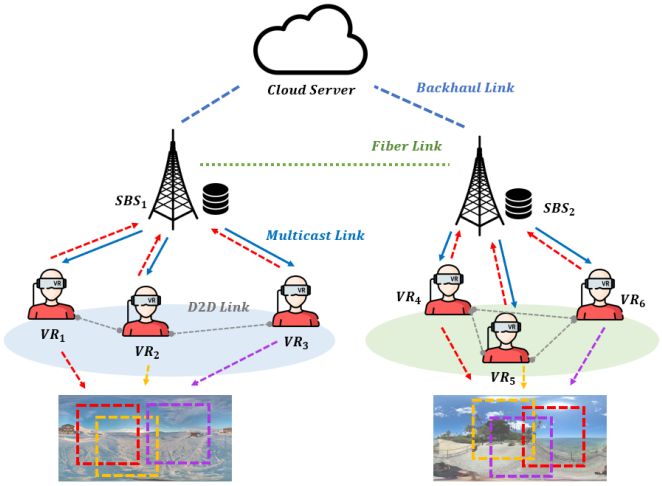


Fig. 1: System model of a multi-user VR streaming network

VR users residing in the network, as shown in Fig. 1. Let VR_i represent the i^{th} VR user, $i = \{1, 2, 3, ..., \mathcal{N}\}$, SBS_j represent the j^{th} SBS, $j = \{1, 2, 3, ..., \mathcal{M}\}$ in the network where each SBS is equipped with computation, bandwidth, and storage resources. In this scenario, multiple users will join the same VR-watching session for group streaming and the users in the same session will choose to watch a VR video $u \in \mathcal{U}$ where the video resources are initially available at the cloud server. After the user starts requesting for u^{th} video for streaming, the requested viewport content will be either transmitted from the cloud server through a backhaul link or transmitted from another SBS with available video resources cached through a fiber link to the associated SBS and forwarded to the user. As the paper mainly focuses on improving the QoS of multi-user scenarios, users who join the same VR streaming session will be grouped remotely and virtually for multicasting.

B. Streaming Model

The VR videos can be divided in both spatial and temporal dimensions, and VR videos are streamed to VR users as segments, each containing a fixed time interval. Each segment is split into tiles for proactive streaming and bandwidth saving. Let users' FoV at the t^{th} segment be represented by the set $\boldsymbol{V} = \{\boldsymbol{V}_1, \boldsymbol{V}_2, ..., \boldsymbol{V}_t\}$ where $\boldsymbol{V}_{t,i,u,m} \in \{0, 1\}$ and $\boldsymbol{V}_{t,i,u,m} = 1$ denotes the m^{th} tile of the u^{th} video is watched by VR_i at t^{th} segment. To meet the different viewing requirements of VR users, transcoding can be performed at each SBS to encode video tiles into different quality versions. Let $\mathcal{D} = \{1, 2, ..., D\}$ denote the set of available quality

versions, where $d \in \mathcal{D}$ represents a specific quality level and D indicates the total number of quality levels in the system. Higher values of d correspond to higher video quality and larger tile sizes. The size of the m^{th} tile for u^{th} video denoted by $R^d_{u,m}$ increases with a larger value of d (i.e., $R^d_{u,m} < R^{d+1}_{u,m}$).

During the streaming process of the t^{th} segment, the FoV tile requested for the $(t+1)^{th}$ segment will be predicted and prefetched by users. The prefetched tiles will be stored in the local cache, denoted by $\boldsymbol{C}^d_{t,i,u,m} \in \{0,1\}$. Here, $\boldsymbol{C}^d_{t,i,u,m} = 1$ indicates that the d^{th} version of the m^{th} tile has been cached on the local device when VR_i starts watching the t^{th} segment of the u^{th} video. We assume that all the tile cached at the SBS and transmitted from the cloud server will be at the highest quality, while transcoding will be performed to degrade the tile quality to cope with varying wireless channel conditions from users' side. If all tiles required for the t^{th} segment are available in the local cache, i.e., $\boldsymbol{C}_{t,i,u} \succeq \boldsymbol{V}_{t,i,u}$, VR_i can seamlessly play the segment. Otherwise, VR_i must request the missing tiles from the SBS and the quality of tiles will be adjusted based on varying channel conditions. Prefetching begins with the central tile of the predicted FoV and proceeds with the remaining tiles in a clockwise order. To optimize bandwidth usage, these prefetched tiles are compared with previously cached tiles to avoid redundant transmissions. Since local cache sizes are limited, only frequently requested tiles are cached and used for comparison. Based on the quality of tiles, users' watching reward can be determined through the average quality of watching tiles as below:

$$\mathrm{r}_{t,i,u} = \frac{1}{\sum_{m=1}^{V_u} y_{t,i,m}} \sum_{m \in V_u} y_{t,i,m} \cdot q_m(d) \tag{1}$$

where $y_{t,i,m} = 1$ indicates the m^{th} tile is requested at t^{th} segment and $y_{t,i,m} \in \{0,1\}$, $q_m(d)$ is an evaluation function based on version d that higher value of d generates high function value.

C. Caching Model

Since users' tile requests may overlap spatially within the same streaming group and between their consecutive tile requirements, popular tile caching is applied to further reduce the transmission delay. Each SBS is equipped with the storage capacity indicated by C_{SBS_j}, which represents the maximum storage available at j^{th} SBS. Similarly, each VR user VR_i has a cache size denoted as C_{VR_i} that represents the maximum amount of data that can be stored locally by VR_i. As SBS possesses higher storage capability, we assume that:

$$C_{\text{SBS}_j} > C_{\text{VR}_i}, \quad \forall j \in \mathcal{M}, i \in \mathcal{N} \tag{2}$$

In this network, both VR users and SBSs can store content in their respective caches to improve system performance by reducing latency and bandwidth usage. VR video resources requested by a VR user can be fetched from the SBS, the user's local cache, or another user's local cache. The decision to store a tile in a specific cache is determined by factors including available cache sizes C_{VR_i} and C_{SBS_j}, content request frequency, and channel conditions and transmission rates between VR users and associated SBSs. These factors determine the optimal caching strategy that minimizes access latency while efficiently utilizing limited storage resources.

The cache state of the m^{th} tile from the u^{th} video at SBS_j is indicated by $c_j^{u,m}$, where $c_j^{u,m} = 1$ denotes that the tile is cached at SBS_j. Similarly, the cache state of the m^{th} tile from the u^{th} video at VR_i is indicated by $c_i^{u,m}$. This binary representation enables the system to track the distribution of cached content throughout the network, facilitating efficient content retrieval and resource allocation decisions.

D. Communication Model

In the proposed wireless network, the cloud server and SBSs are connected through high-speed fiber backhaul links, which also enable base-station-to-base-station (B2B) communications between SBSs. SBSs communicate with VR users through wireless links, while VR users can communicate directly through D2D wireless connections.

1) Path Loss Model: We adopt the channel models specified in 3GPP TR 38.901 [60], which provides standardized models for various scenarios. Our network employs two different path loss models: the Urban Micro Street Canyon (i.e., UMi-Street Canyon) model for Base Station to Device (B2D) communications and the Indoor Office model for D2D communications.

a) B2D Path Loss Model: For B2D communications, the path loss is calculated as:

$$PL^{\text{B2D}} = \begin{cases} PL_{\text{LOS}}(d) & \text{for LOS} \\ PL_{\text{NLOS}}(d) & \text{for NLOS} \end{cases} \tag{3}$$

where for line-of-sight (LOS) conditions:

$$PL_{\text{LOS}}(d) = 32.4 + 21\log_{10}(d) + 20\log_{10}(f_c) + X_{\sigma_{\text{LOS}}} \tag{4}$$

and for non-line-of-sight (NLOS) conditions:

$$\begin{aligned} PL_{\text{NLOS}}(d) &= \max(PL_{\text{LOS}}(d), PL'_{\text{NLOS}}(d)) \\ PL'_{\text{NLOS}}(d) &= 35.3\log_{10}(d_{3D}) + 22.4 + 21.3\log_{10}(f_c) \\ &\quad -0.3(h_{UT} - 1.5) + X_{\sigma_{\text{NLOS}}} \end{aligned} \tag{5}$$

where d is the relative distance in meters, f_c is the carrier frequency in GHz, h_{UT} is the height of the user terminal in meters, $X_{\sigma_{\text{LOS}}} \sim \mathcal{N}(0, 4^2)$ and $X_{\sigma_{\text{NLOS}}} \sim \mathcal{N}(0, 7.82^2)$ are the shadow fading in dB with standard deviations of 4 dB and 7.82 dB for LOS and NLOS conditions respectively [60].

The LOS probability is reference from [60]:

$$P_{\text{LOS}}(d) = \begin{cases} 1 & d \le 18 \text{ m} \\ \frac{18}{d} + \exp\left(-\frac{d}{36}\right)\left(1 - \frac{18}{d}\right) & d > 18 \text{ m} \end{cases} \tag{6}$$

b) D2D Path Loss Model: For D2D communications, we utilize the Indoor Office path loss model [60]:

$$PL^{\text{D2D}} = \begin{cases} PL^{\text{In}}_{\text{LOS}}(d) & \text{for LOS} \\ PL^{\text{In}}_{\text{NLOS}}(d) & \text{for NLOS} \end{cases} \tag{7}$$

where for LOS conditions:

$$\begin{aligned} PL^{\text{In}}_{\text{LOS}}(d) &= 32.4 + 17.3\log_{10}(d_{3D}) + 20\log_{10}(f_c) \\ &\quad + X_{\sigma^{\text{In}}_{\text{LOS}}} \end{aligned} \tag{8}$$

and for NLOS conditions:

$$\begin{aligned} PL^{\text{In}}_{\text{NLOS}}(d) =& \max(PL^{\text{In}}_{\text{LOS}}(d), PL'^{\text{In}}_{\text{NLOS}}(d)) \\ PL'^{\text{In}}_{\text{NLOS}}(d) =& 38.3\log_{10}(d_{3D}) + 17.3 \\ &+ 24.9\log_{10}(f_c) + X_{\sigma^{\text{In}}_{\text{NLOS}}} \end{aligned} \tag{9}$$

where $X_{\sigma^{\text{In}}_{\text{LOS}}} \sim \mathcal{N}(0, 3^2)$ and $X_{\sigma^{\text{In}}_{\text{NLOS}}} \sim \mathcal{N}(0, 8.03^2)$ represent the shadow fading in dB with standard deviations of 3 dB and 8.03 dB for LOS and NLOS conditions respectively [60].

The LOS probability for Indoor Office follows [60]:

$$P^{\text{In}}_{\text{LOS}}(d) = \begin{cases} 1 & d \leq 1.2 \text{ m} \\ \exp\left(-\frac{d-1.2}{4.7}\right) & d > 1.2 \text{ m} \end{cases} \tag{10}$$

c) Channel Gain Calculation: The channel gain is derived from the path loss using $g = 10^{-PL/10}$. Additionally, small-scale fading is modeled using Rician fading for LOS and Rayleigh fading for NLOS conditions [60]:

$$h = \begin{cases} \sqrt{\frac{K}{K+1}} + \sqrt{\frac{1}{2(K+1)}}(X + jY) & \text{for LOS (Rician)} \\ \sqrt{\frac{1}{2}}(X + jY) & \text{for NLOS (Rayleigh)} \end{cases} \tag{11}$$

where K is the Rician K-factor which is set to 9 dB for UMi and 7 dB for Indoor scenario [61] [62], and X, Y are independent normal random variables.

2) Transmission Delay Model:

a) B2B Transmission: The SBSs are inter-connected through fiber links, providing high-speed, interference-free transmission channels. These connections are not subject to the wireless path loss models described above. The transmission delay between SBS_j and SBS_j^+ for m^{th} tile from u^{th} video is denoted by $T^{B2B}_{j,j^+,u,m}$ and expressed as follows:

$$T^{B2B}_{j,j^+,u,m} = \frac{\sum_{u=1}^{U}\sum_{m=1}^{V_u} y_{u,m} \cdot R^d_{u,m}}{c_{fiber}}, \tag{12}$$

where $y_{u,m}$ indicates whether the m^{th} tile is requested for the u^{th} video, and c_{fiber} refers to the dedicated fiber transmission rate between SBS_j and SBS_j^+.

b) B2D Transmission: For B2D communications, the system allocates bandwidth resources through resource blocks (RBs), and the downlink transmission rate uses the B2D path loss model (PL^{B2D}) shown before and can be expressed as:

$$c_{i,j} = \sum_{k=1}^{s_{i,j}} w_R \cdot \log_2\left(1 + \frac{P^B_j g^{B2D}_{i,j}}{\sum_{\substack{l=1 \\ l\neq j}}^{M} P^B_l g^{B2D}_{i,l} + \sigma^2}\right), \tag{13}$$

where $s_{i,j}$ represents the number of RBs allocated to VR_i by SBS_j, and w_R denotes the bandwidth of each RB. P^B_j is the transmit power from SBS_j, while $g^{B2D}_{i,j} = 10^{-PL^{\text{B2D}}_{i,j}/10}$ represents the channel gain derived from the B2D path loss model. The denominator accounts for co-channel interference from other base stations and noise.

The B2D transmission delay is calculated as:

$$T^{B2D}_{i,j,u,m} = \frac{\sum_{u=1}^{U}\sum_{m=1}^{V_u} y_{i,u,m} \cdot R^d_{u,m}}{c_{i,j}} \tag{14}$$

where the RB allocation scheme must satisfy $\sum_{i=1}^{N} s_{i,j} \leq \mathcal{S}, \forall j \in \mathcal{M}$ and $s_{i,j} \in \mathbb{Z}^+, \forall i \in \mathcal{N}, j \in \mathcal{M}$.

c) D2D Transmission: For D2D communications, we employ a frequency reuse scheme where the dedicated band is spatially shared among user pairs. Although each active D2D pair is allocated bandwidth w_D/N_D, co-channel interference occurs when multiple pairs are within each other's interference range, which is captured by the interference term. Using the D2D path loss model PL^{D2D}, the D2D transmission rate between VR_i and VR_{i^+} is expressed as:

$$c_{i,i^+} = \frac{w_D}{N_D} \cdot \log_2\left(1 + \frac{P^D_i g^{D2D}_{i,i^+}}{\sum_{\substack{l=1 \\ l\neq i}}^{N} P^D_l g^{D2D}_{i,l} + \sigma^2}\right), \tag{15}$$

where w_D represents the total bandwidth allocated for D2D communications and N_D is the number of concurrent D2D pairs. P^D_i denotes the transmit power of VR_i, and $g^{D2D}_{i,i^+} = 10^{-PL^{\text{D2D}}_{i,i^+}/10}$ is the channel gain derived from the D2D path loss model. The term $\sum_{\substack{l=1 \\ l\neq i}}^{N} P^D_l g^{D2D}_{i,l}$ accounts for the interference from other D2D pairs.

The D2D transmission delay is calculated as:

$$T^{D2D}_{i,i^+,u,m} = \frac{\sum_{u=1}^{U}\sum_{m=1}^{V_u} y_{i,u,m} \cdot R^d_{u,m}}{c_{i,i^+}} \tag{16}$$

3) Rendering Delay Model: The rendering delay depends on whether the rendering is performed remotely at the SBS or locally on the VR device.

a) Remote Rendering (at SBS): For remote rendering at SBS_j, the rendering delay is expressed as:

$$T^{Ren}_{j,u,m} = \frac{\sum_{u=1}^{U}\sum_{m=1}^{V_u} y_{i,u,m} \cdot R^d_{u,m} \cdot \overline{\omega}}{f_j} \tag{17}$$

where f_j is the computational capacity of SBS_j in cycles per second, and $\overline{\omega}$ is the computational complexity per bit.

b) Local Rendering (at VR device): For local rendering at VR_i, the rendering delay is given by:

$$T^{Ren}_{i,u,m} = \frac{\sum_{u=1}^{U}\sum_{m=1}^{V_u} y_{i,u,m} \cdot R^d_{u,m} \cdot \overline{\omega}}{f_i} \tag{18}$$

where f_i is the computational capacity of VR_i.

4) Queueing Delay Model: In multi-user VR networks, multiple VR devices may simultaneously request resources from the same SBS, resulting in queueing delays. The queueing delay model accounts for scheduling and processing based on priority assignment. For each device VR_i associated with SBS_j, the queueing delay is determined by:

$$T^{queue}_{i,j} = \sum_{\substack{k=1 \\ p_{k,j} > p_{i,j}}}^{N_j} T^{proc}_{k,j} \tag{19}$$

where N_j is the number of devices associated with SBS_j, $p_{i,j}$ represents the priority of VR_i at SBS_j, and $T^{proc}_{k,j}$ is the processing time required for VR_k at SBS_j. Devices are scheduled according to their priority values, with higher priority devices

processed first. The priority values $p_{i,j}$ are provided by the PPO-MAMDP agent given the current state condition.

The processing time $T_{i,j}^{proc}$ depends on the rendering mode:

$$T_{i,j}^{proc} = \begin{cases} \frac{\sum_{u=1}^{U}\sum_{m=1}^{V_u} y_{i,u,m} \cdot R_{u,m}^{d} \cdot \overline{\omega}}{f_j} & \text{for remote rendering} \\ 0 & \text{for local rendering} \end{cases} \tag{20}$$

where f_j represents the computational capacity of SBS$_j$. For local rendering, the processing time at the SBS is zero, as rendering is performed on the device itself.

The system employs a dynamic priority assignment that can be adjusted in each time segment to optimize overall network performance with the constraint $\sum_{i=1}^{N_j} p_{i,j} = 1, \forall j \in \mathcal{M}$.

Queueing delay is particularly significant for scenarios involving remote rendering such as Scenarios 1 and 3 in Table II, where SBS computational resources become a potential bottleneck. For these scenarios, the total delay expression must be modified to include the queueing component:

For Scenario 1 (B2D with Remote Rendering):

$$T_{total} = T_{i,j,u,m}^{B2D} + T_{j,u,m}^{Render} + T_{i,j}^{queue} \tag{21}$$

For Scenario 3 (B2B $\rightarrow$ B2D with Remote Rendering):

$$T_{total} = T_{j,j^+,u,m}^{B2B} + T_{j,u,m}^{Render} + T_{i,j,u,m}^{B2D} + T_{i,j}^{queue} \tag{22}$$

5) Total Delay Model for Different Scenarios: Table II summarizes the delay components for different transmission and rendering scenarios in our VR network. Each scenario combines specific transmission paths and rendering modes, resulting in different delay and energy consumption components.

E. Computation Model

The computation model includes the transcoding performed at the SBS as well as the viewport prediction performed at the local VR devices. Based on the abovementioned assumption that all the tiles arrived at the SBS's buffer will be of the highest quality, transcoding procedure can be taken to cope with users' different network conditions. The time required to transcode one tile for VR$_i$ from version d to $(d-1)$ is denoted by $T_{u,m,d,d'}^{trsc}$ and is expressed as follows:

$$T_{i,d,d'}^{trsc} = \frac{\hat{f} \cdot T_t}{f_{i,j}} \cdot \sum_{m=1}^{V_u} y_{i,u,m} \tag{23}$$

where $\hat{f}$ denotes the computational transcoding overhead across one quality level which is typically considered to be 20 GHz [9] and T_t is the duration of one time step.

In the proposed system, a Deep Neural Network model is employed to perform the viewport prediction task. This model is trained to predict users' FoV at the $(t+1)^{th}$ segment using the tracking data collected at the t^{th} segment.

F. Energy Model

The energy model considers perspectives from both SBSs and VR devices. For SBS, the energy consumption includes the computation and transmission energy consumption. The computation energy consumption includes both transcoding and rendering part. The transcoding energy consumption at SBS$_j$ for converting one tile from version d to $(d-1)$ required at t^{th} segment is denoted by $E_{i,d,d',t}^{trsc}$ as expressed as follows:

$$E_{i,d,d',t}^{trsc} = \sum_{i \in \mathcal{N}} \overline{\mu} \cdot \hat{f} \cdot T_t \sum_{m=1}^{V_u} y_{i,u,m} \tag{24}$$

where $\overline{\mu}$ is the energy consumption per cycle.

The rendering energy consumption at SBS$_j$ for rendering t^{th} segment is denoted by $E_{i,j}^{ren}$ as expressed below:

$$E_{i,j,t}^{ren} = \sum_{i \in \mathcal{N}} \overline{\mu} \cdot s_{i,t} \sum_{u=1}^{U}\sum_{m=1}^{V_u} y_{i,u,m} \cdot R_{u,m}^{d} \cdot \overline{\omega} \tag{25}$$

where $s_{i,t}$ represents the computational resource allocation factor for VR$_i$ at time segment t, determining the proportion of SBS capacity dedicated to rendering.

The transmission energy consumption includes remote rendering, local rendering, and peer transmission cases. For remote rendering, the transmission energy consumption required between SBS$_j$ and VR$_i$ at the t^{th} segment is:

$$E_{i,j,t}^{tran,Re} = \frac{P_j^B \cdot \sum_{u=1}^{U}\sum_{m=1}^{V_u} \alpha_p \cdot y_{i,u,m} \cdot R_{u,m}^{d}}{c_{i,j}} \tag{26}$$

where α_p is a projection variable that convert 2D content to 3D videos in terms of data size.

While for local rendering case where SBS$_j$ only need to transmit tiles, the transmission energy consumption is:

$$E_{i,j,t}^{tran,Lo} = \frac{P_j^B \cdot \sum_{u=1}^{U}\sum_{m=1}^{V_u} y_{i,u,m} \cdot R_{u,m}^{d}}{c_{i,j}} \tag{27}$$

where $c_{i,j}$ can be replaced with c_{j,j^+} for peer transmission energy consumption $E_{j,j^+,t}^{tran,B2B}$ between SBS$_j$ and SBS$_{j^+}$.

From VR devices' perspective, the transmission, computation, and rendering energy consumption are considered in the proposed system. The transmission energy consumption between VR$_i$ and VR$_{i^+}$ can be denoted by:

$$E_{i,i^+,t}^{tran,D2D} = \frac{P_j^D \cdot \sum_{u=1}^{U}\sum_{m=1}^{V_u} y_{i,u,m} \cdot R_{u,m}^{d}}{c_{i,i^+}} \tag{28}$$

The computation energy consumption required for viewport prediction model at VR$_i$ for the t^{th} segment is:

$$E_{i,t}^{cpt} = \overline{\eta} f_i^2 \cdot s_t \overline{\omega} \tag{29}$$

where $\overline{\eta}$ is the energy coefficient which is the effective switched capacitance for each VR device and s_t is the size of collected training data at the t^{th} segment.

The rendering energy consumption required for local rendering the t^{th} segment at VR$_i$ can be expressed as follows:

$$E_{i,t}^{ren} = \overline{\eta} f_i^2 \cdot \sum_{u=1}^{U}\sum_{m=1}^{V_u} y_{i,u,m} \cdot R_{u,m}^{d} \overline{\omega} \tag{30}$$

TABLE II: Transmission and Rendering Scenarios in VR Network

Sc.	Path	Rendering	Components	Total Delay	Energy Consumption
Base Station to Device Scenarios					
1	B2D	Remote (SBS)	B2D transmission of rendered frames; SBS rendering with computation resources f_j; Transcoding at SBS as needed	$T^{B2D}_{i,j,u,m}+T^{Ren}_{j,u,m}+T^{queue}_{i,j}+T^{trsc}_{i,d,d'}$	$E^{tran,Re}_{i,j,t}+E^{ren}_{i,j,t}+E^{trsc}_{i,d,d',t}$
2	B2D	Local (Device)	B2D transmission of video tiles; Device rendering with computation resources f_i; Transcoding at SBS as needed	$T^{B2D}_{i,j,u,m}+T^{Ren}_{i,u,m}+T^{trsc}_{i,d,d'}$	$E^{tran,Lo}_{i,j,t}+E^{ren}_{i,t}+E^{trsc}_{i,d,d',t}$
Base Station to Base Station Scenarios					
3	B2B→B2D	Remote (SBS)	B2B transmission via fiber link; SBS rendering with f_j; B2D transmission of rendered frames; Transcoding at SBS as needed	$T^{B2B}_{j,j^+,u,m}+T^{Ren}_{j,u,m}+T^{B2D}_{i,j,u,m}+T^{queue}_{i,j}+T^{trsc}_{i,d,d'}$	$E^{tran,B2B}_{j,j^+,t}+E^{ren}_{i,j,t}+E^{tran,Re}_{i,j,t}+E^{trsc}_{i,d,d',t}$
4	B2B→B2D	Local (Device)	B2B transmission via fiber link; B2D transmission of video tiles; Device rendering with f_i; Transcoding at SBS as needed	$T^{B2B}_{j,j^+,u,m}+T^{B2D}_{i,j,u,m}+T^{Ren}_{i,u,m}+T^{trsc}_{i,d,d'}$	$E^{tran,B2B}_{j,j^+,t}+E^{tran,Lo}_{i,j,t}+E^{ren}_{i,t}+E^{trsc}_{i,d,d',t}$
Device to Device Scenarios					
5	D2D	Local (Device)	D2D transmission between VR users; Device rendering with f_i; Uses D2D bandwidth w_D	$T^{D2D}_{i,i^+,u,m}+T^{Ren}_{i,u,m}$	$E^{tran,D2D}_{i,i^+,t}+E^{ren}_{i,t}$
6	B2B→B2D→D2D	Local (Device)	B2B transmission via fiber; B2D for some tiles; D2D for remaining tiles; Device rendering with f_i; Transcoding at SBS as needed	$\max(T^{B2B}_{j,j^+,u,m}+T^{B2D}_{i,j,u,m}, T^{D2D}_{i,i^+,u,m})+T^{Ren}_{i,u,m}+T^{trsc}_{i,d,d'}$	$E^{tran,B2B}_{j,j^+,t}+E^{tran,Lo}_{i,j,t}+E^{tran,D2D}_{i,i^+,t}+E^{ren}_{i,t}+E^{trsc}_{i,d,d',t}$

Notes: (1) Energy calculated separately for SBS (transmission, rendering, transcoding) and VR devices (local rendering, D2D). (2) $T^{queue}_{i,j}$ reflects priority-based processing with values $p_{i,j}$ provided by PPO-MAMDP. (3) $T^{trsc}_{i,d,d'}$ represents transcoding delay from quality level d to d'. (4) Scenario 6 uses $\max()$ for parallel B2D and D2D transmissions. (5) B2D allocates resource blocks $s_{i,j}$; D2D uses bandwidth w_D/N_D per pair. (6) Path loss models: UMi-Street Canyon for B2D, Indoor Office for D2D communications.

IV. Problem Formulation and Methodology

To improve the viewing experience in multi-player scenarios, a maximization problem is formulated based on inter-user delay, watching reward and energy consumption by jointly optimizing bandwidth allocation, cache and rendering decisions, computation resource allocation and quality selection.

A. Inter-user Delay in Multicast Group

In multi-player scenarios, inter-user delay in the same VR streaming group is critical for providing fair and immersive experience. We define the inter-user delay as the end-to-end (E2E) delay difference between two users, including transmission and rendering delays. The inter-user delay between VR_i and VR_{i^+} is denoted by $\Delta D_{i,i^+}$ and expressed as:

$$\Delta D_{i,i^+} = D_i^{\text{E2E}} - D_{i^+}^{\text{E2E}}, \quad \forall i, i^+ \in \mathcal{N} \tag{31}$$

where D_i^{E2E} is the E2E delay for VR_i which depends on the service routes including:

- Remote rendering requested tiles at SBSs and transmit directly to VR users:

$$D_i^{\text{E2E}} = T^{B2B}_{j,j^+,u,m}+T^{Re}_i+T^{B2D}_{i,j}, \quad \forall i \in \mathcal{N}, \forall j, j^+ \in \mathcal{M} \tag{32}$$

- Remote transmits all requested tiles to local VR devices and performs local rendering:

$$D_i^{\text{E2E}} = T^{B2B}_{j,j^+,u,m}+T^{B2D}_{i,j}+T^{Lo}_i, \quad \forall i \in \mathcal{N}, \forall j, j^+ \in \mathcal{M} \tag{33}$$

- Remote transmits partial tiles and receives partial tiles through D2D communication with local rendering:

$$D_i^{\text{E2E}} = \max\{T^{B2B}_{j,j^+,u,m}+T^{B2D}_{i,j}, T^{D2D}_{i,i^+}\}+T^{Lo}_i, \quad \forall i, i^+ \in \mathcal{N}, \forall j, j^+ \in \mathcal{M} \tag{34}$$

  where we consider the transmission from SBS to VR_i and the communication between VR devices can be performed in parallel so that that maximum delay will be counted toward the E2E delay. What's more, the transcoding process will incur additional delay and will be contribute to the E2E delay as required.

We define the inter-user delay for VR_i as the delay difference between one's E2E delay and the minimum E2E delay in the group which is denoted by ΔD_i as shown below:

$$\Delta D_i = D_i^{\text{E2E}} - \min_{i \in \mathcal{N}} D_i^{\text{E2E}} \tag{35}$$

B. Joint Quality of Service Metrics

To consider users' quality of service from different aspects, inter-user delay, average watching reward, and energy consumption at VR devices are considered as joint quality of service metrics in multi-player scenarios. The combined utility function of VR_i during the t^{th} segment is:

$$U_{i,t} = \mu_1 * r_{t,i,u} - \mu_2 * \Delta D_i - \mu_3 * E^{total}_{i,t}, \tag{36}$$

where μ_1, μ_2, μ_3 are the hyperparameters for preference adjustment based on users' requirements. $r_{t,i,u}$ is the reward gained from watching VR video with different quality, ΔD_i is the deduction of immersive experience due to inter-user delay within the same streaming group. The total energy consumption is denoted by $E^{total}_{i,t}$ which is the sum of the computation, transmission, and rendering energy consumption.

What's more, considering the SBSs' utility in case of energy consumption, bandwidth allocation, computation resources, and cache capacity limitation, the joint utility of SBSs' is also taken into account and the corresponding joint utility for SBS_j during the t^{th} segment can be written as:

$$U_{j,t} = \mu_4 \cdot E^{total}_{j,t} + \mu_5 \cdot \frac{\sum b^B_{i,j}}{b^B_j} + \mu_6 \cdot \frac{\sum f_{i,j}}{f_j} + \mu_7 \cdot \frac{\sum c^{u,m}_j}{c_j}, \tag{37}$$

where $\mu_4, \mu_5, \mu_6, \mu_7$ are hyperparameters for SBS preference adjustment. $E^{total}_{j,t}$ is the total energy consumption at SBS_j

including transcoding, rendering, and transmission. b_j^B, $f_{i,j}$, and c_j represent bandwidth capacity, computational resources allocated to user i, and cache capacity of SBS$_j$ respectively. The terms $\frac{\sum b_{i,j}^B}{b_j^B}$, $\frac{\sum f_{i,j}}{f_j}$, and $\frac{\sum c_j^{u,m}}{c_j}$ represent the proportions of bandwidth, computation, and cache utilization respectively.

C. Problem Formulation

To provide better QoE, we maximize the utility function composed of inter-user delay, watching reward based on average tile quality, and energy consumption from prediction and rendering tasks. The maximization problem is formulated in (38) with constraints: (38a) local epoch number τ_i cannot exceed global round T; (38b) transmit power limitations for users and SBSs; (38c) association constraint limiting one user per SBS; (38d)-(38e) bandwidth limitations for B2D and D2D communications; (38f) positive hyperparameters; (38g) maximum delay constraint ζ; (38h) binary cache indicators.

From the expression shown above, we can tell that maximizing users' utility over the long term depends on the proper allocation of bandwidth resources for both SBS and device communications, adequate allocation of computation resources for remote and local rendering tasks, appropriate selection of popular tiles to be cached at SBSs and local VR devices as well as the appropriate decision on the quality of tiles to be streamed to users. Thus, the objective is formulated as a multi-variable maximization problem.

$$\text{Maximize} \quad \sum_{t=1}^{T}\left\{\sum_{i=1}^{\mathcal{N}} U_{i,t} - \sum_{j=1}^{\mathcal{M}} U_{j,t}\right\} \tag{38}$$

$$\text{s.t.} \quad \tau_i \le T \quad \forall i \in \mathcal{N}, \forall t \in T \tag{38a}$$

$$0 \le P_i^D \le P_j^B \le P_{max}, \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{M} \tag{38b}$$

$$\sum_{j=1}^{\mathcal{M}} a_{i,j}^t \le 1, \tag{38c}$$

$$\sum_{i\in\mathcal{N}, j\in\mathcal{M}} b_{i,j}^B \le B^B, \tag{38d}$$

$$\sum_{i, i^+\in\mathcal{N}, i\neq i^+} b_{i,i^+}^D \le B^D, \tag{38e}$$

$$\mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6, \mu_7 > 0, \tag{38f}$$

$$D_i^{E2E} \le \zeta, \forall i \in \mathcal{U}, \tag{38g}$$

$$c_j^{u,m}, c_i^{u,m} \in 0, 1, \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{M} \tag{38h}$$

D. PPO-Based MA-MDP Framework

To address the dynamic resource allocation problem in multi-user VR networks, we adopt a multi-agent Markov Decision Process (MA-MDP) framework optimized using PPO. In this framework, each SBS functions as an independent agent making decisions based on local observations while affecting the shared network environment. PPO is selected for its superior training stability in complex multi-agent scenarios through the clipped objective function, which prevents destructive policy updates across multiple coordinating agents.

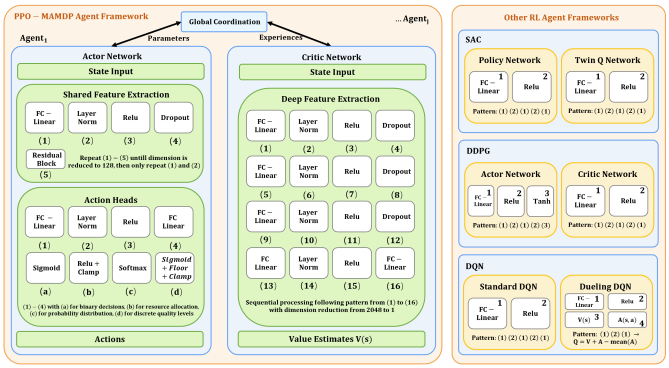


Fig. 2: PPO-MAMADP framework with comparison to other RL frameworks

MARL naturally scales with network size and better captures the distributed nature of wireless networks where centralized control is impractical due to communication overhead and latency constraints. This approach enables simultaneous coordination of multiple SBSs while making decisions based on local observations and global system performance.

1) Framework Architecture: Figure.2 illustrates our PPO-MAMDP agent framework, which consists of two primary components: the agent framework and the VR network environment. Within the agent framework, multiple agents coordinate through a global coordination mechanism that facilitates parameter sharing and experience exchange. Each agent is equipped with an actor network that determines actions and a critic network that evaluates state values. The VR network environment represents the physical infrastructure where SBSs with RBs, computation resources, and caches connect to multiple VR devices through B2D links, while also supporting B2B links between SBSs and D2D links between VR devices.

2) MDP Formulation: We define the MDP as a 4-tuple $\mathcal{K} = \{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}\}$, representing states, actions, transition probabilities, and rewards. The state space $\mathcal{S} = \{S_{\text{SBS}}, S_{\text{VR}}\}$ comprises SBS states $S_{\text{SBS}} = \{S_B, S_C, S_H, S_U\}$ and VR device states $S_{\text{VR}} = \{S_b, S_c, S_r, S_t, S_d\}$. SBS states include bandwidth state S_B, computation resource state S_C, cache state S_H, and user state S_U. VR device states include device bandwidth S_b, device cache S_c, tile receive S_r, tile transmit S_t and location states S_d. The action space $\mathcal{A} = \{A_A, A_B, A_C, A_H, A_Q, A_R\}$ consists of association decisions A_A, bandwidth allocation A_B, computation resource allocation A_C, caching decisions A_H, quality selection A_Q, and rendering location decisions A_R. The combined reward function reflects the joint optimization objectives, and is shown in the equation below:

$$\begin{aligned}\mathcal{R} = &\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{M}} \mu_1 \cdot R_R - \mu_2 \cdot R_D - \mu_3 \cdot R_E^i \\ &-(\mu_4 \cdot R_E^j + \mu_5 \cdot R_B + \mu_6 \cdot R_C + \mu_7 \cdot R_H)\end{aligned} \tag{39}$$

where R_R is watching quality reward, R_D is inter-user delay penalty, R_E^i and R_E^j represent energy consumption at devices and SBSs, and R_B, R_C, R_H denote bandwidth, computation, and cache utilization rewards. As shown in Figure 2, these metrics flow from the VR network environment to the reward function, which influences the actor networks of each agent.

3) PPO Architecture and Optimization: Our implementation uses a simplified actor-critic architecture as depicted in Figure .2. The actor network maps states to actions through shared feature extraction layers with residual connections and specialized heads for each action dimension. Heads include association, resource block allocation, computation priority, cache decision, rendering decision, and quality selection. The critic network estimates state values using feature normalization and fully connected layers with scalar output.

The PPO algorithm optimizes policy using surrogate objectives with clipped probability ratios:

$$L^{\pi}(\theta) = \mathbb{E}_t \left[\min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t) \right] \quad (40)$$

We use GAE to compute advantages:

$$A_t^{\text{GAE}} = \sum_{t'=t}^{T'} (\gamma\lambda)^{t'-t} \delta_{t'} \quad (41)$$

where $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$.

The total loss includes an entropy bonus for exploration and a separate value function loss:

$$L^{\text{total}} = L^{\pi}(\theta) + c_{\text{entropy}} \cdot H[\pi_\theta], \quad L^{V}(\phi) = \mathbb{E}_t[(V_\phi(s_t) - R_t)^2] \quad (42)$$

The PPO-MAMDP algorithm employs key implementation strategies for multi-agent learning in VR networks. Network parameters are updated every B time steps when $t \bmod B = 0$ and $t > 0$. This mini-batch approach ensures regular updates while maintaining stability and efficiency. All agents receive the same global observation, enabling coordinated decision-making. Each agent maintains its experience buffer, cleared every I_{clear} episodes to ensure data freshness. Agent actions are collected and processed through the MDP layer for consistency. The batch size balances sample efficiency and computational overhead while ensuring adaptation to dynamic conditions. The complete PPO-MAMDP training procedure is formalized in Algorithm 1. The implementation includes multiple update epochs per batch of experience, gradient clipping to prevent large policy updates, quality-weighted action MSE for stability, emphasis on quality selection components through weighted losses, and batch normalization of advantages before updates. As illustrated in Figure 2, the global coordination component facilitates parameter sharing and experience exchange between agents, enabling more efficient learning across the system. This PPO-MAMDP framework effectively balances multiple objectives in the VR network, adapting to dynamic user requests and resource constraints while maximizing overall QoE.

Algorithm 1: PPO-MAMDP Algorithm

Input: Agents set $\mathcal{M}$; State space $\mathcal{S}$; Action space $\mathcal{A}$; Reward function $\mathcal{R}$; Hyperparameters γ, λ, ϵ; Batch size $B = 32$; Buffer clear interval $I_{clear} = 10$
Output: Optimized policies π_θ for all agents
Initialization: Initialize actor networks π_θ, critic networks V_ϕ, and experience buffers for each agent;
for *episode = 1 to* E **do**
  Reset environment and get initial state s_0;
  for $t = 1$ *to* T_{max} **do**
    Get global observation: $obs_{global} \leftarrow$ MDP.get_global_observation(s_t);
    Initialize action array $actions \leftarrow \mathbf{0}^{M \times d_{action}}$;
    foreach *agent* $j \in \mathcal{M}$ **do**
      Select action $a_j \sim \pi_\theta(a_j | obs_{global})$;
      $actions[j] \leftarrow a_j$;
    Process actions: $a_{processed} \leftarrow$ MDP.process_action(s_t, $actions$);
    Execute action $a_{processed}$ in environment;
    Receive reward r_t and next state s_{t+1};
    Get next global observation $obs_{next} \leftarrow$ MDP.get_global_observation(s_{t+1});
    foreach *agent* $j \in \mathcal{M}$ **do**
      Store experience $(obs_{global}, actions[j], r_t, obs_{next}, done)$ in agent's buffer;
    if $t \bmod B = 0$ ***and*** $t > 0$ **then**
      foreach *agent* $j \in \mathcal{M}$ **do**
        Sample mini-batch of size B from agent's buffer;
        Compute returns and advantages using GAE: $A_t^{\text{GAE}} = \sum_{l=0}^{T'} (\gamma\lambda)^l \delta_{t+l}$, where $\delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$;
        Update critic network V_ϕ by minimizing: $L^V(\phi) = (V_\phi(s_t) - R_t)^2$;
        Update actor network π_θ by maximizing: $L^{\pi}(\theta) = \min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1+\epsilon)A_t) + c_{\text{entropy}} H[\pi_\theta]$;
        Apply gradient clipping to both networks;
    $s_t \leftarrow s_{t+1}$;
    if *done* **then**
      break;
  if *episode* $\bmod I_{clear} = 0$ **then**
    foreach *agent* $j \in \mathcal{M}$ **do**
      Clear agent's experience buffer;
return Optimized policy networks π_θ for all agents;

E. Greedy Matching in D2D Communication

Apart from the PPO-MAMDP structure where all the SBSs are taking control over system resource management and inter-device communication, the tile selection and streaming between VR devices should also be considered as part of the system design. As the selection scheme should be performed along with the PPO-MAMDP process, a low-complexity algorithm is required to match the tile streaming between peer VR devices. The greedy matching algorithm is chosen for its simplicity and practical feasibility in VR networks with limited devices. It also remains manageable for scalable VR scenarios and provides deterministic pairing decisions without iterative optimization. This complements MARL's complex long-term resource allocation with straightforward short-term D2D pairing, balancing system performance with implementation simplicity. The matching theory [63] is widely used in D2D content sharing scenarios to optimally match content providers with demanders with requested content, and the greedy matching algorithm is detailed in Algorithm 2.

The matching utility function between VR devices is defined as the combination of transmission delay, tile availability as well as tile quality. The availability of requested tiles and their quality is dependent on the content cached at each VR device.

Algorithm 2: Greedy Matching Algorithm

Input: Set of devices $\mathcal{D}$, matching utility matrix $U_{\mathcal{M}}[i][i^+]$ where $i, i^+ \in \mathcal{D}$ and $i \neq i^+$.
Output: Matching set $\mathcal{M}$.
Step 1: Initialize
Initialize matched set $\mathcal{S} = \emptyset$ and matching result $\mathcal{M} = \emptyset$.
Step 2: Sort Pairs by Matching Utility
Construct all possible pairs of devices (i, i^+) with $i \neq i^+$.
Sort pairs (i, i^+) by utility $U_{\mathcal{M}}[i][i^+]$ in descending order.
Step 3: Greedy Matching
foreach *pair* (i, i^+) *in the sorted list* **do**
  if $i \notin \mathcal{S}$ ***and*** $i^+ \notin \mathcal{S}$ **then**
    Add (i, i^+) to $\mathcal{M}$.
    Add i and i^+ to $\mathcal{S}$.
Step 4: Output Matching
Return $\mathcal{M}$.

The overall matching utility function between VR_i and VR_{i^+} is expressed as follows:

$$U_{\mathcal{M}}[i][i^+] = \kappa_1 \cdot s_{j,u,m} + \kappa_2 \cdot d_{i,u,m} - \kappa_3 \cdot T^{D2D}_{i,i^+} - \kappa_4 \cdot E^{tran,D2D}_{i,i^+} \tag{43}$$

where $\kappa_1, \kappa_2, \kappa_3, \kappa_4$ are the hyperparameters determining the weighting of four corresponding factors. Based on real-time processing requirements, we set $\kappa_1 = 0.4$ and $\kappa_2 = 0.3$ to prioritize tile availability and channel quality using real time data, while $\kappa_3 = 0.05$ and $\kappa_4 = 0.05$ are assigned minimal weights as delay and energy metrics require historical data incompatible with real-time VR requirements. $s_{j,u,m}$ indicates the availability of m^{th} tile for u^{th} video at the provider device, and $d_{i,u,m}$ represents the quality level of the requested tile.

V. Simulations Analysis

A. System Parameters

Our simulation comprises 5 SBSs and up to 50 VR devices supporting 10 VR videos, each divided into 200 tiles across 1800 time segments for 60-second content. We utilize a public VR dataset [64] for realistic tile request patterns based on authentic viewport transitions. Video sizes range between 0.25-0.8 GB [64] with 5 quality levels. Storage configurations include SBS cache capacities of 8-32 GB for content caching and VR device working cache of 1-6 GB for tile buffering. The PPO-MAMDP and benchmark agents undergo pretraining evaluation across {0, 250, 500, 1000, 1500, 2000} steps with 5 independent trials per configuration, evaluated for 10 episodes with 100 time steps each with identical initialization conditions to ensure fair algorithmic comparison. Performance metrics are defined as follows: Average Reward represents the joint utility optimization from Equation (38), VR Utility quantifies normalized user satisfaction through video quality scores, the maximum tolerable E2E delay is 20 ms according to VR requirements, and energy efficiency metrics, SBS Utility measures infrastructure efficiency via bandwidth utilization ratios and computational resource allocation efficiency.

TABLE III: Simulation and Training Parameters

Parameter	Value
SBS downlink transmit power (P_j^B)	0.2 W [65]
VR device D2D transmit power (P_i^D)	0.1 W [66]
Carrier frequency (f_c)	28 GHz [67]
AWGN Noise power (σ^2)	1e-14 W [68]
B2D bandwidth allocation per SBS:	$\sim$200 MHz [69]
Resource block bandwidth (w_R)	1.44 MHz [70]
D2D communication bandwidth (w_D)	50 MHz [71]
SBS CPU frequency (f_j)	2.5-4 GHz [72]
VR device CPU frequency (f_i)	1.5-3 GHz
Computation Complexity ($\overline{\omega}$)	10 cycles/bit [73]
PPO-MAMDP learning rate	3e-4
PPO-MAMDP batch Size B	32
Reward discount factor (γ)	0.99
GAE advantage parameter (λ)	0.95
PPO policy clipping range (ϵ)	0.2

TABLE IV: Performance Across Different Pretraining Steps

Steps	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
Mean Performance						
0	0.6850	0.6887	0.6725	0.6889	0.6716	0.6813 ± 0.0087
250	0.7020	0.6853	0.7067	0.7201	0.6905	0.7009 ± 0.0134
500	0.6898	0.6741	0.7169	0.7100	0.6734	0.6928 ± 0.0200
1000	0.6745	0.6851	0.6577	0.6889	0.7099	0.6832 ± 0.0193
1500	0.7085	0.7041	0.7045	0.6719	0.6702	0.6918 ± 0.0186
2000	0.7271	0.7200	0.6284	0.6846	0.7070	0.6934 ± 0.0398
AUC						
0	67.96	68.31	66.67	68.32	66.63	67.58 ± 0.84
250	69.59	67.95	70.08	71.38	68.45	69.49 ± 1.33
500	68.40	66.84	71.06	70.37	66.78	68.69 ± 2.00
1000	66.81	67.90	65.22	68.31	70.35	67.72 ± 1.92
1500	70.20	69.79	69.84	66.62	66.41	68.57 ± 1.85
2000	72.08	71.35	62.25	67.88	70.09	68.73 ± 3.96
Peak Performance						
0	0.7183	0.7293	0.7069	0.7376	0.7034	0.7191 ± 0.0141
250	0.7401	0.7099	0.7375	0.7452	0.7402	0.7346 ± 0.0143
500	0.7221	0.7108	0.7529	0.7536	0.7099	0.7299 ± 0.0218
1000	0.7031	0.7024	0.6936	0.7381	0.7512	0.7177 ± 0.0248
1500	0.7412	0.7275	0.7342	0.6991	0.7031	0.7210 ± 0.0182
2000	0.7558	0.7589	0.6609	0.7262	0.7357	0.7275 ± 0.0394

B. Computational Complexity Analysis

To evaluate the practical feasibility of our approach, we analyze the computational complexity of PPO-MAMDP compared to baseline algorithms.

Training Complexity: PPO-MAMDP exhibits training complexity of $O(M \times T \times E \times H^2)$, where M represents the number of agents (SBSs), T denotes training steps, E indicates epochs per update, and H is the neural network hidden layer size. While this scales linearly with agent count compared to single-agent methods ($O(T \times E \times H^2)$ for DDPG/SAC), the multi-agent framework enables parallel training and achieves faster convergence through coordinated learning, effectively amortizing the additional computational cost.

Inference Complexity: During deployment, PPO-MAMDP requires $O(M \times H^2)$ operations per decision step compared to $O(H^2)$ for single-agent baselines. For our experimental configuration with $M = 5$ SBSs and $H = 64$ hidden units, this translates to approximately 20,480 operations per time step, executable in microseconds on standard hardware.

PPO-MAMDP's computational overhead is justified by its superior performance across multiple evaluation metrics and its unique capability for balanced multi-objective optimization essential for practical VR network management.

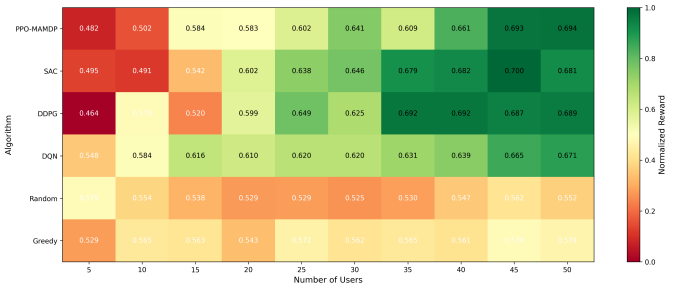


Fig. 3: Reward heatmap for different algorithms over varying number of users

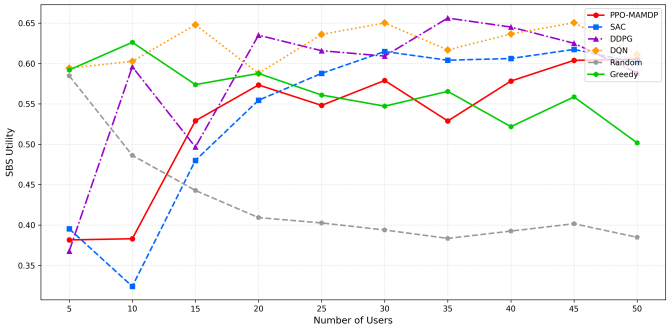


Fig. 4: SBS utility for different algorithms over varying number of users

C. Results Analysis

All algorithms undergo identical training with 5 independent runs per configuration. Average Reward represents the optimization objective from equation (38), VR Utility quantifies user experience through equation (36), and SBS Utility measures infrastructure efficiency using equation (37). Statistical significance is evaluated using 95% confidence intervals.

To evaluate pretraining impact on PPO-MAMDP performance, we conducted experiments across multiple configurations as presented in Table IV. The 250 pretraining steps configuration delivers superior results, achieving the highest average performance of 0.7009 and area under the curve of 69.49. This configuration maximizes performance while demonstrating reproducibility across experimental iterations.

Comparative analysis of different pretraining regimes yields several key insights. PPO-MAMDP without pretraining consistently underperforms with mean performance of 0.6813, confirming pretraining's critical role for complex VR resource allocation. The 500 steps configuration emerges as a compelling alternative for constrained computational resources, delivering peak performance of 0.7299 at modest cost.

The 1000 steps configuration exhibits notable performance variability from 0.6577 to 0.7099, making it less reliable for the dynamic wireless environments. The 1500 and 2000 steps configurations show diminishing returns and increased variability, with 2000 steps demonstrating highest inconsistency with standard deviation of 0.0398, suggesting a trade-off between extended pretraining and generalization capability.

We evaluate PPO-MAMDP against several widely-used reinforcement learning algorithms and baseline approaches:

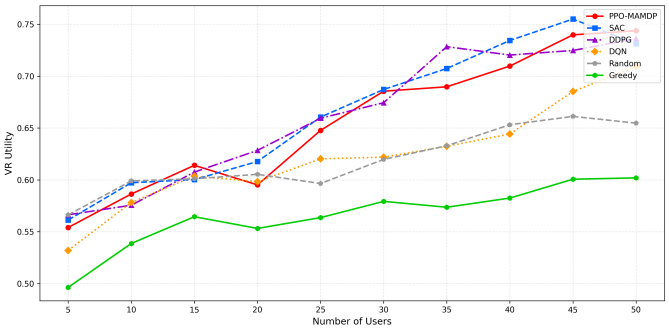


Fig. 5: VR utility for different algorithms over varying number of users

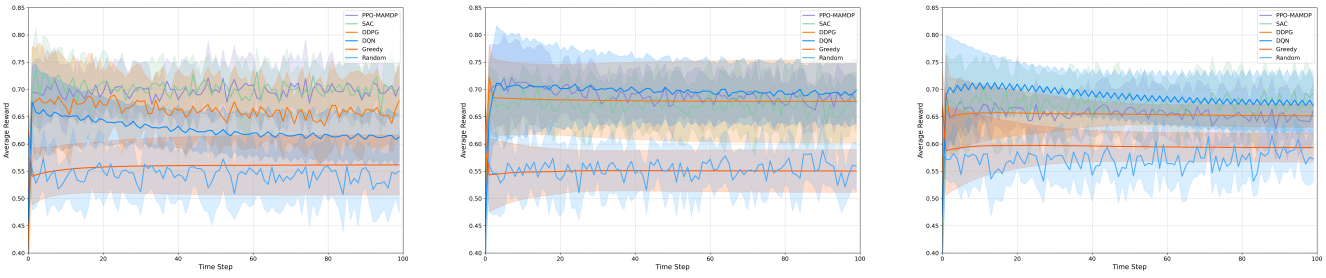
DDPG, using Q-learning with policy gradients for continuous actions; SAC, balances reward maximization with exploration; DQN, uses experience replay for stable learning; Random policy to establish performance floor; and Greedy policy with highest immediate reward. This provides a comprehensive evaluation framework across different learning approaches.

As illustrated in Fig. 6, performance trajectories across different pretraining configurations reveal PPO-MAMDP's distinctive advantages. Even at initialization (Fig. 6 (a)), PPO-MAMDP establishes a performance lead over baseline algorithms. This advantage becomes more pronounced with increasing pretraining (Fig. 6 (b,c)), showing performance improvements and narrower confidence intervals, indicating enhanced policy robustness. The 250 steps configuration (Fig. 6 (b)) shows promising results with competitive performance and reduced volatility compared to alternatives.

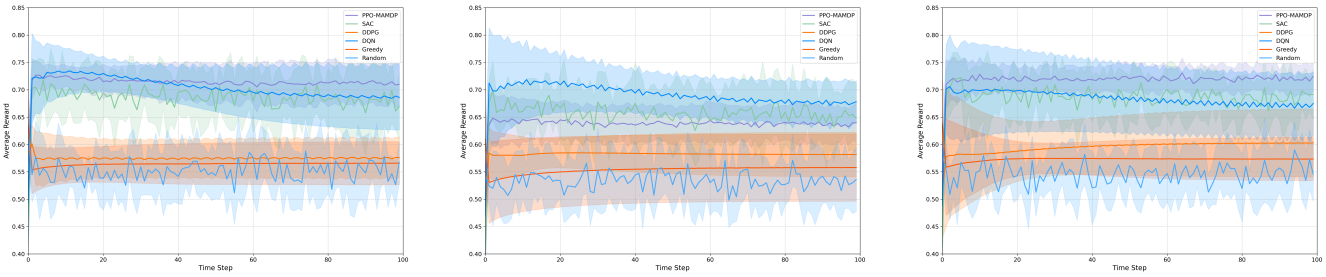
The intermediate pretraining regime (Fig. 6 (d)) demonstrates notable performance variability characterizing the 1000 steps configuration, though PPO-MAMDP maintains performance leadership despite this inconsistency. At 1500 and 2000 steps (Fig. 6 (e,f)), while our approach maintains its advantage, the confidence intervals confirm diminishing returns beyond optimal pretraining, aligning with our quantitative analysis. Throughout all configurations, PPO-MAMDP generally outperforms alternatives with narrower confidence intervals, demonstrating superior capability in high-dimensional VR environments. This validates our PPO-MAMDP framework's effectiveness, particularly after pretraining at 250-500 steps, where it demonstrates robust performance and adaptability.

To further validate our approach, we analyze PPO-MAMDP performance across varying user densities as shown in Fig. 5 and Fig. 4. In Fig. 5, PPO-MAMDP demonstrates consistent upward trajectory in VR utility as users increase, closely matching SAC in high-density scenarios while outperforming DDPG, DQN, Random, and Greedy policies. This indicates PPO-MAMDP effectively leverages increased system complexity to optimize resource allocation, showing particular strength in dense deployments where resource competition is intense. The performance gap with simpler approaches such as Random and Greedy algorithms widens significantly at higher user counts, highlighting superior scaling capabilities.

Examining Fig. 4, we observe that PPO-MAMDP achieves a balanced trade-off between SBS and VR utilities. While DQN



(a) 0 pretraining steps (b) 250 pretraining steps (c) 500 pretraining steps



(d) 1000 pretraining steps (e) 1500 pretraining steps (f) 2000 pretraining steps

Fig. 6: Evaluation of reward performance over different agents with varying pretraining steps.

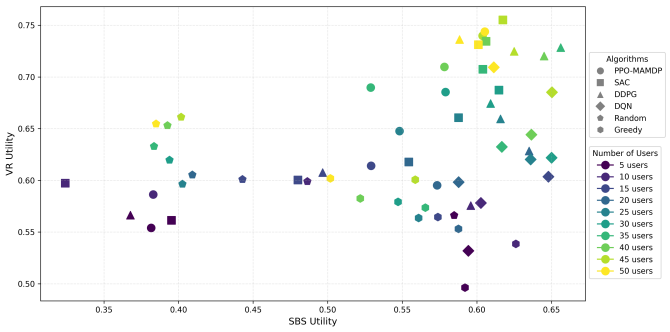


Fig. 7: Relationship between SBS utility and VR utility

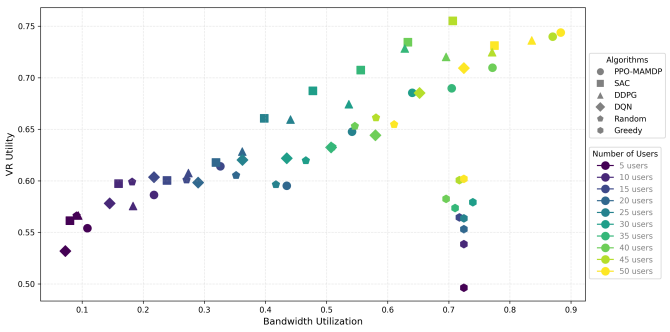


Fig. 8: Relationship between VR utility and bandwidth utilization

and DDPG exhibit marginally higher SBS utility values, this comes at the cost of their VR performance as demonstrated in Fig. 5. PPO-MAMDP shows steady improvement in SBS utility as the user count increases from 10 to 50, culminating in a performance comparable to that of SAC at maximum density. The initial lower SBS utility at 5-10 users followed by rapid improvement suggests that PPO-MAMDP's policy effectively adapts its resource allocation strategy as it encounters more complex environments. This strategic balancing between SBS and VR utilities demonstrates PPO-MAMDP's capability to maintain holistic system performance while prioritizing the primary objective of VR experience optimization, making it particularly well-suited for practical VR network deployments where multi-objective optimization is essential.

To further explore PPO-MAMDP's multi-dimensional optimization capabilities, we analyze its performance across resource utilization and utility trade-offs as illustrated in Fig. 7 and Fig. 8. Fig. 7 reveals critical insights into PPO-MAMDP's balanced optimization strategy, achieving high VR utility and competitive SBS utility in high-density scenarios. This upper-right quadrant positioning demonstrates PPO-MAMDP's effectiveness in managing trade-offs between VR user experience and base station resource efficiency. While other algorithms like DDPG occasionally achieve marginally higher VR utility or SAC reaches slightly better SBS utility at certain points, PPO-MAMDP consistently delivers balanced performance across both metrics as user density increases.

The resource efficiency of PPO-MAMDP is further validated in Fig. 8, where a clear positive correlation is observed between bandwidth utilization and VR utility. PPO-MAMDP exhibits optimal operating points in the 0.7-0.9 bandwidth utilization range, achieving VR utility values of 0.70-0.74 for high user counts (40-50). This demonstrates the algorithm's capability to efficiently leverage available bandwidth resources without over-utilization. Notably, PPO-MAMDP maintains this efficiency in high-density deployments where resource competition is most intense, showing adaptive resource allocation compared to baseline approaches. The clustering of high-user PPO-MAMDP data points in the upper-right region confirms that our approach scales effectively with increasing

user density, making it particularly valuable for real-world VR deployments where maximizing experience quality under bandwidth constraints is essential. This bandwidth-utility relationship further validates our algorithm's capability to navigate complex multi-dimensional optimization spaces while maintaining performance across varying network conditions.

The comprehensive evaluation of PPO-MAMDP across varying user scales validates its effectiveness for resource allocation in multi-user wireless VR networks. Our approach successfully balances competing objectives while demonstrating superior capabilities in: maximizing social welfare through dynamic resource management; adapting to heterogeneous user requirements with optimal pretraining configurations; jointly optimizing QoE metrics and system utility; and facilitating efficient resource sharing in high-density environments. These results confirm that our PPO-MAMDP framework provides a robust solution for wireless VR networks where both system efficiency and user experience are considered, offering advantages over existing reinforcement learning approaches.

VI. Conclusion

In this paper, we presented a PPO-based MARL framework for resource allocation in wireless VR networks that maximizes social welfare by jointly optimizing communication, computing, caching resources, rendering service routes decision as well as associations between SBSs and VR users. Our experimental results demonstrate that PPO-MAMDP outperforms baseline approaches across multiple metrics while maintaining balanced performance between user experience and system efficiency. While the framework shows promising results, limitations include assumptions of global state information availability, synchronized decision-making, and VR-only environments, while real-world implementations would face information collection delays, training latency, inter-agent communication overhead, and mixed device scenarios where SBSs serve heterogeneous traffic. Future work will address distributed implementation with realistic delays, mixed device environment handling, and large-scale real-world validation to enhance practical applicability.

References

[1] Y. Ma, K. Ota, and M. Dong, "Qoe optimization for virtual reality services in multi-ris-assisted terahertz wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 3, pp. 538–551, 2024.

[2] J. Lee, H. Lu, and Y. Chen, "Robust wireless vr video transmission based on overlapped fovs," in *ICC 2023 - IEEE International Conference on Communications*, pp. 3084–3089, 2023.

[3] W. Chen, Q. Song, P. Lin, L. Guo, and A. Jamalipour, "Proactive 3c resource allocation for wireless virtual reality using deep reinforcement learning," in *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, 2021.

[4] Huawei Technologies Co., Ltd., "Whitepaper on the VR-oriented bearer network requirement," tech. rep., Huawei Technologies Co., Ltd., 2016. [Accessed 17-02-2025].

[5] M. Xu, Y. Zhou, and Y. Chen, "A monte carlo tree search-based routing scheme with vr video qoe guarantees in sdns," in *2024 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–5, 2024.

[6] M. Setayesh and V. W. Wong, "Viewport prediction, bitrate selection, and beamforming design for thz-enabled 360-degree video streaming," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2024.

[7] M. Xu, W. C. Ng, W. Y. B. Lim, J. Kang, Z. Xiong, D. Niyato, Q. Yang, X. Shen, and C. Miao, "A full dive into realizing the edge-enabled metaverse: Visions, enabling technologies, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 656–700, 2023.

[8] R. Zhang, J. Liu, F. Liu, T. Huang, Q. Tang, S. Wang, and F. R. Yu, "Buffer-aware virtual reality video streaming with personalized and private viewport prediction," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 2, pp. 694–709, 2022.

[9] 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 Journal on Selected Areas in Communications*, vol. 40, no. 5, pp. 1615–1631, 2022.

[10] Y. Yang, L. Feng, Y. Sun, Y. Li, W. Li, and M. A. Imran, "Multi-cluster cooperative offloading for vr task: A marl approach with graph embedding," *IEEE Transactions on Mobile Computing*, vol. 23, no. 9, pp. 8773–8788, 2024.

[11] B. W. Nyamtiga, D. K. P. Asiedu, A. A. Hermawan, Y. F. Luckyarno, and J.-H. Yun, "Adaptive foveated rendering and offloading in an edge-assisted virtual reality system," *IEEE Access*, vol. 12, pp. 17308–17327, 2024.

[12] Z. Chen, H. Zhu, L. Song, D. He, and B. Xia, "Wireless multiplayer interactive virtual reality game systems with edge computing: Modeling and optimization," *IEEE Transactions on Wireless Communications*, vol. 21, no. 11, pp. 9684–9699, 2022.

[13] L. Zhong, X. Chen, C. Xu, Y. Ma, M. Wang, Y. Zhao, and G.-M. Muntean, "A multi-user cost-efficient crowd-assisted vr content delivery solution in 5g-and-beyond heterogeneous networks," *IEEE Transactions on Mobile Computing*, vol. 22, no. 8, pp. 4405–4421, 2023.

[14] M. Chen, W. Saad, C. Yin, and M. Debbah, "Data correlation-aware resource management in wireless virtual reality (vr): An echo state transfer learning approach," *IEEE Transactions on Communications*, vol. 67, no. 6, pp. 4267–4280, 2019.

[15] 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 Transactions on Multimedia*, vol. 24, pp. 1529–1544, 2022.

[16] N. Q. Hieu, N. H. Chu, D. T. Hoang, D. N. Nguyen, and E. Dutkiewicz, "A unified resource allocation framework for virtual reality streaming over wireless networks," in *ICC 2023 - IEEE International Conference on Communications*, pp. 3042–3047, 2023.

[17] X. Xu and Y. Song, "A deep reinforcement learning-based optimal computation offloading scheme for vr video transmission in mobile edge networks," *IEEE Access*, vol. 11, pp. 122772–122781, 2023.

[18] Y. Zhou, X. Li, S. Lv, G. He, M. Shi, and X. Chen, "A network slicing elastic switching algorithm for vr devices based on ddqn," in *2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring)*, pp. 1–5, 2024.

[19] B. Feng, "A deep reinforcement learning-based resource allocation mechanism for xr applications*," in *2023 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pp. 1–6, 2023.

[20] W. Yu, T. J. Chua, and J. Zhao, "Asynchronous hybrid reinforcement learning for latency and reliability optimization in the metaverse over wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 7, pp. 2138–2157, 2023.

[21] Z. Li, H. Zhang, X. Li, H. Ji, and V. C. Leung, "Distributed task scheduling for mec-assisted virtual reality: A fully-cooperative multiagent perspective," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 7, pp. 10572–10586, 2024.

[22] Y. Xu, H. Zhang, X. Li, F. R. Yu, V. C. Leung, and H. Ji, "Trusted collaboration for mec-enabled vr video streaming: A multi-agent reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 9, pp. 12167–12180, 2023.

[23] N. Su, J.-B. Wang, Y. Chen, H. Yu, C. Ding, and Y. Pan, "Joint rendering offloading and resource allocation optimization for mec-assisted vr systems," *IEEE Wireless Communications Letters*, vol. 13, no. 4, pp. 949–953, 2024.

[24] Y. Xu, J. An, C. Zhou, H. Xu, and Z. Han, "Dynamic energy management for uav-enabled vr systems: A tile-based collaboration approach," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 11, pp. 17668–17683, 2024.

[25] C. Liu, K. Wang, H. Zhang, X. Li, and H. Ji, "Rendered tile reuse scheme based on fov prediction for mec-assisted wireless vr service," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 3, pp. 1709–1721, 2023.

[26] X. Wei and C. Yang, "Fov privacy-aware vr streaming," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1515–1520, 2022.

[27] J. Feng, L. Liu, X. Hou, Q. Pei, and C. Wu, "Qoe fairness resource allocation in digital twin-enabled wireless virtual reality systems," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 11, pp. 3355–3368, 2023.

[28] J. Zhao, L. Qian, and W. Yu, "Human-centric resource allocation in the metaverse over wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 3, pp. 514–537, 2024.

[29] N. Su, J.-B. Wang, X. Zhang, C. Chang, Y. Pan, Y. Chen, H. Yu, and J. Wang, "Rate splitting for mobile edge computing assisted multiuser virtual reality systems," *IEEE Transactions on Communications*, vol. 73, no. 1, pp. 303–316, 2025.

[30] Z. Chen, H. Zhu, L. Song, D. He, and B. Xia, "Wireless multiplayer interactive virtual reality game systems with edge computing: Modeling and optimization," *IEEE Transactions on Wireless Communications*, vol. 21, no. 11, pp. 9684–9699, 2022.

[31] J. Dai, G. Yue, S. Mao, and D. Liu, "Sidelink-aided multiquality tiled 360° virtual reality video multicast," *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4584–4597, 2022.

[32] C. Xu, Z. Chen, M. Tao, and W. Zhang, "Edge-device collaborative rendering for wireless multi-user interactive virtual reality in metaverse," in *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, pp. 3542–3547, 2023.

[33] H. Shen, X. Li, H. Ji, and H. Zhang, "Dibr-based collaborative computation in edge network for multiplayer online vr game," in *2023 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1451–1456, 2023.

[34] J. Prodanov, B. Bertalanič, C. Fortuna, S.-K. Chou, M. B. Jurič, R. Sanchez-Iborra, and J. Hribar, "Multi-agent reinforcement learning-based in-place scaling engine for edge-cloud systems," in *2025 IEEE 18th International Conference on Cloud Computing (CLOUD)*, pp. 32–42, 2025.

[35] C. Xu, Z. Tang, H. Yu, P. Zeng, and L. Kong, "Digital twin-driven collaborative scheduling for heterogeneous task and edge-end resource via multi-agent deep reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 10, pp. 3056–3069, 2023.

[36] Z. Yao, S. Xia, Y. Li, and G. Wu, "Cooperative task offloading and service caching for digital twin edge networks: A graph attention multi-agent reinforcement learning approach," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 11, pp. 3401–3413, 2023.

[37] W. Yu, T. C. Jie, and J. Zhao, "Multi-agent deep reinforcement learning for digital twin over 6g wireless communication in the metaverse," in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, 2023.

[38] F.-Y. Chao, C. Ozcinar, and A. Smolic, "Transformer-based long-term viewport prediction in 360° video: Scanpath is all you need," in *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6, 2021.

[39] X. Chi, H. Chen, G. Li, Z. Ni, N. Jiang, and F. Xia, "Edsp-edge: Efficient dynamic edge service entity placement for mobile virtual reality systems," *IEEE Transactions on Wireless Communications*, vol. 23, no. 4, pp. 2771–2783, 2024.

[40] Y. Lin, Z. Gao, H. Du, D. Niyato, J. Kang, A. Jamalipour, and X. S. Shen, "A unified framework for integrating semantic communication and ai-generated content in metaverse," *IEEE Network*, vol. 38, no. 4, pp. 174–181, 2024.

[41] B. Gao, D. Sheng, L. Zhang, Q. Qi, B. He, Z. Zhuang, and J. Wang, "Star-vp: Improving long-term viewport prediction in 360° videos via space-aligned and time-varying fusion," MM '24, (New York, NY, USA), p. 5556–5565, Association for Computing Machinery, 2024.

[42] T. Chen, F. Tan, J. Ai, X. Xiong, C. Wu, and X. Ren, "Joint optimization of task offloading and service placement for digital twin empowered mobile edge computing," CNCIT '24, (New York, NY, USA), p. 132–137, Association for Computing Machinery, 2024.

[43] G. Qi, R. Sun, N. Cheng, W. Quan, H. Zhou, Z. Su, and C. Li, "Knowledge-driven rendering task offloading strategy for virtual reality in mec-enabled wireless networks," in *2024 IEEE 35th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1–6, 2024.

[44] F. Liu, H. Li, P. Wang, K. Shi, and Y. Hu, "Graph based joint computing and communication scheduling for virtual reality applications," in *2023 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2023.

[45] J. Li, W. Liang, Y. Li, Z. Xu, X. Jia, and S. Guo, "Throughput maximization of delay-aware dnn inference in edge computing by exploring dnn model partitioning and inference parallelism," *IEEE Transactions on Mobile Computing*, vol. 22, no. 5, pp. 3017–3030, 2023.

[46] J. Li, Z. Li, Q. Li, W. Sun, W. Li, H. Wang, and Z. Liu, "Demo: Landscape: Saliency and trajectory based viewport prediction in point cloud video streaming," in *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*, MobiSys '23, (New York, NY, USA), p. 604–605, Association for Computing Machinery, 2023.

[47] L. Lin, Y. Chen, Z. Zhou, P. Li, and J. Xiong, "When metaverse meets computing power networking: An energy-efficient framework for service placement," *IEEE Wireless Communications*, vol. 30, no. 5, pp. 76–85, 2023.

[48] Y. Chiang, C.-H. Hsu, G.-H. Chen, and H.-Y. Wei, "Deep q-learning-based dynamic network slicing and task offloading in edge network," *IEEE Transactions on Network and Service Management*, vol. 20, no. 1, pp. 369–384, 2023.

[49] M. Iftikar, D. Kwaku Pobi Asiedu, T. Nishio, and J.-H. Yun, "Deep reinforcement learning-based overfill rendering, offloading, and sub-band allocation for edge-assisted vr system," *IEEE Access*, vol. 12, pp. 149147–149161, 2024.

[50] Y. Pei, M. Li, X. Huang, and X. Shen, "Qoe-aware volumetric video caching and rendering for mobile extended reality services," *IEEE Internet of Things Journal*, vol. 12, no. 12, pp. 21852–21865, 2025.

[51] F. Hu, Y. Deng, and A. H. Aghvami, "Cooperative multigroup broadcast 360° video delivery network: A hierarchical federated deep reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 4009–4024, 2022.

[52] S. Liu, Y. Yu, X. Lian, Y. Feng, C. She, P. L. Yeoh, L. Guo, B. Vucetic, and Y. Li, "Dependent task scheduling and offloading for minimizing deadline violation ratio in mobile edge computing networks," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 2, pp. 538–554, 2023.

[53] W. Yang, W. Du, B. Zhao, Y. Ren, J. Sun, and X. Zhou, "Cross-layer assisted early congestion control for cloud vr applications in 5g edge networks," in *2024 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2024.

[54] C.-Y. Chen and H.-Y. Hsieh, "Cross-frame resource allocation with context-aware qoe estimation for 360° video streaming in wireless virtual reality," *IEEE Transactions on Wireless Communications*, vol. 22, no. 11, pp. 7887–7901, 2023.

[55] Y. Jiang, K. Poularakis, D. Kiedanski, S. Kompella, and L. Tassiulas, "Robust and resource-efficient machine learning aided viewport prediction in virtual reality," in *2022 IEEE International Conference on Big Data (Big Data)*, pp. 1002–1013, 2022.

[56] J. Li, L. Han, C. Zhang, Q. Li, and Z. Liu, "Spherical convolution empowered viewport prediction in 360 video multicast with limited fov feedback," vol. 19, Jan. 2023.

[57] J. Song, Q. Song, Y. Kang, L. Guo, and A. Jamalipour, "Qoe-driven distributed resource optimization for mixed reality in dynamic tdd systems," *IEEE Transactions on Communications*, vol. 70, no. 11, pp. 7294–7306, 2022.

[58] J. Yu, A. Y. Alhilal, T. Zhou, P. Hui, and D. H. K. Tsang, "Attention-based qoe-aware digital twin empowered edge computing for immersive virtual reality," *IEEE Transactions on Wireless Communications*, vol. 23, no. 9, pp. 11276–11290, 2024.

[59] X. Liu and Y. Liu, "Distributed learning for metaverse over wireless networks," *IEEE Communications Magazine*, vol. 61, no. 9, pp. 40–46, 2023.

[60] 3GPP, "Study on channel model for frequencies from 0.5 to 100 GHz," Technical Report TR 38.901, 3rd Generation Partnership Project (3GPP), 3 2024. Release 18.

[61] M. K. Samimi, G. R. MacCartney, S. Sun, and T. S. Rappaport, "28 ghz millimeter-wave ultrawideband small-scale fading models in wireless channels," in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, pp. 1–6, 2016.

[62] J.-J. Park, M.-D. Kim, H.-K. Chung, and W. Kim, "Ricean k-factor analysis of indoor channel measurements at 3.7 ghz," in *2010 5th International ICST Conference on Communications and Networking in China*, pp. 1–5, 2010.

[63] D. Wu, L. Zhou, Y. Cai, H.-C. Chao, and Y. Qian, "Physical–social-aware d2d content sharing networks: A provider–demander matching game," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 8, pp. 7538–7549, 2018.

[64] 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 *Proceedings of the 8th ACM on Multimedia Systems Conference*, MM-

Sys'17, (New York, NY, USA), p. 211–216, Association for Computing Machinery, 2017.
[65] S. Sarkar, R. K. Ganti, and M. Haenggi, "Optimal base station density for power efficiency in cellular networks," in *2014 IEEE International Conference on Communications (ICC)*, pp. 4054–4059, 2014.
[66] H. T. Nguyen, H. D. Tuan, T. Q. Duong, H. V. Poor, and W.-J. Hwang, "Joint d2d assignment, bandwidth and power allocation in cognitive uav-enabled networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 3, pp. 1084–1095, 2020.
[67] J. Mashino, K. Satoh, S. Suyama, Y. Inoue, and Y. Okumura, "5g experimental trial of 28 ghz band super wideband transmission using beam tracking in super high mobility environment," in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, pp. 1–5, 2017.
[68] T. Dang, C. Liu, and M. Peng, "Low-latency mobile virtual reality content delivery for unmanned aerial vehicle-enabled wireless networks with energy constraints," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 2, pp. 2189–2201, 2023.
[69] N. Explained, "Nr bandwidth," 2023.
[70] ETSI, "5G; NR; User Equipment (UE) conformance specification; Radio transmission and reception; Part 2: Range 2 Standalone," Technical Specification TS 138 521-2, European Telecommunications Standards Institute (ETSI), 07 2021. Release 16.
[71] E. Barri and C. Bouras, *Efficient Mechanism of eNB Bandwidth in D2D Communication in 5G*, pp. 310–319. 03 2020.
[72] B. William, A. Hermawan, Y. Luckyarno, T.-W. Kim, D.-Y. Jung, J. Kwak, and J.-H. Yun, "Edge-computing-assisted virtual reality computation offloading: An empirical study," *IEEE Access*, vol. 10, pp. 95892–95907, 09 2022.
[73] Z. Huang and V. Friderikos, "Network resource optimization for multi-view streaming mobile augmented reality," in *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*, pp. 1–7, 2022.

PLACE PHOTO HERE

Xinyu Wan Biography text here.

PLACE PHOTO HERE

Abbas Jamalipour Biography text here.