Feeling of Presence Maximization: mmWave-Enabled Virtual Reality Meets Deep Reinforcement Learning

Peng Yang, *Member, IEEE*, Tony Q. S. Quek, *Fellow, IEEE*, Jingxuan Chen, Chaoqun You, *Member, IEEE*, and Xianbin Cao, *Senior Member, IEEE*

***Abstract*— This paper investigates the problem of providing ultra-reliable and power-efficient virtual reality (VR) experiences for wireless mobile users. To ensure reliable ultra-high-definition (UHD) video frame delivery to mobile users and enhance their immersive visual experiences, a coordinated multipoint (CoMP) transmission technique and millimeter wave (mmWave) communications are exploited. Owing to user movement and time-varying wireless channels, the wireless VR experience enhancement problem is formulated as a sequence-dependent and mixed-integer problem with a goal of maximizing users' feeling of presence (FoP) in the virtual world, subject to power consumption constraints on access points (APs) and users' head-mounted displays (HMDs). The problem, however, is hard to be directly solved due to the lack of users' accurate tracking information and the sequence-dependent and mixed-integer characteristics. To overcome this challenge, we develop a parallel echo state network (ESN) learning method to predict users' tracking information by training fresh and historical tracking samples separately collected by APs. With the learnt results, we propose a deep reinforcement learning (DRL) based optimization algorithm to solve the formulated problem. In this algorithm, we implement deep neural networks (DNNs) as a scalable solution to produce integer decision variables and solve a continuous power control problem to criticize the integer decision variables. Finally, the performance of the proposed algorithm is compared with various benchmark algorithms, and the impact of different design parameters is also discussed. Simulation results demonstrate that the proposed algorithm is more 4.14% power-efficient than the benchmark algorithms.**

***Index Terms*— Virtual reality, coordinated multipoint transmission, feeling of presence, parallel echo state network, deep reinforcement learning.**

I. Introduction

VIRTUAL reality (VR) applications have attracted tremendous interest in various fields, including entertainment, education, manufacturing, transportation, healthcare, and many other consumer-oriented services [1]. These applications exhibit enormous potential in the next generation of multimedia content envisioned by enterprises and consumers through providing richer and more engaging, and immersive experiences. According to market research [2], the VR ecosystem is predicted to be an 80 billion market by 2025, roughly the size of the desktop PC market today.

However, several major challenges need to be overcomed such that businesses and consumers can get fully on board with VR technology [3], one of which is to provide compelling content. To this aim, the resolution of provided content must be guaranteed. In VR applications, VR wearers can either view objects up close or across a wide field of view (FoV) via head-mounted or goggle-type displays (HMDs). As a result, very subtle defects such as poorly rendering pixels at any point on an HMD may be observed by a user up close, which may degrade users' truly visual experiences. To create visually realistic images across the HMD, it must have more display pixels per eye, which indicates that ultra-high-definition (UHD) video frame transmission must be enabled for VR applications. However, the transmission of UHD video frames typically requires $4-5$ times the system bandwidth occupied for delivering a regular high-definition (HD) video [4], [5]. Further, to achieve good user visual experiences, the motion-to-photon latency should be ultra-low (e.g., $10-25$ ms) [6]–[8]. High motion-to-photon values will send conflicting signals to the Vestibulo-ocular reflex (VOR) and then might cause dizziness or motion sickness.

Hence, today's high-end VR systems such as Oculus Rift [9] and HTC Vive [10] that offer high quality and accurate positional tracking remain tethered to deliver UHD VR video frames while satisfying the stringent low latency requirement. Nevertheless, wired VR display may degrade users' seamless visual experiences due to the constraint on the movement of users. Besides, a tethered VR headset presents a potential tripping hazard for users. Therefore, to provide ultimate VR

Manuscript received 9 July 2021; revised 30 December 2021 and 21 March 2022; accepted 31 May 2022. Date of publication 16 June 2022; date of current version 11 November 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 91738301 and Grant 61827901; in part by the National Research Foundation, Singapore and Infocomm Media Development Authority, under its Future Communications Research and Development Programme; and in part by the Singapore University of Technology and Design (SUTD) Growth Plan Grant for AI. The associate editor coordinating the review of this article and approving it for publication was A. El Gamal. *(Corresponding author: Xianbin Cao.)*

Peng Yang, Jingxuan Chen, and Xianbin Cao are with the School of Electronic and Information Engineering, Beihang University, Beijing 100083, China, and also with the Key Laboratory of Advanced Technology, Near Space Information System (Beihang University), Ministry of Industry and Information Technology of China, Beijing 100083, China (e-mail: xbcao@buaa.edu.cn).

Tony Q. S. Quek and Chaoqun You are with the Information Systems Technology and Design, Singapore University of Technology and Design, Singapore 487372.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TWC.2022.3181674.

Digital Object Identifier 10.1109/TWC.2022.3181674

experiences, VR systems or at least the headset component should be untethered [6].

Recently, the investigation on wireless VR has attracted numerous attention from both industry and academe; of particular interest is how to a) develop mobile (wireless and lightweight) HMDs, b) how to enable seamless and immersive VR experiences on mobile HMDs in a bandwidth-efficiency manner, while satisfying ultra-low latency requirements.

A. Related Work

On the aspect of designing lightweight VR HMDs, considering heavy image processing tasks, which are usually insufficient in the graphics processing unit (GPU) of a local HMD, one might be persuaded to transfer the image processing from the local HMD to a cloud or network edge units (e.g., edge servers, base stations (BSs), and access points (APs)). For example, the work in [1] proposed to enable mobile VR with lightweight VR glasses by completing computation-intensive tasks (such as encoding and rendering) on a cloud/edge server and then delivering video streams to users. The framework of fog radio access networks, which could significantly relieve the computation burden by taking full advantages of the edge fog computing, was explored in [11] to facilitate the lightweight HMD design.

In terms of proposing VR solutions with improved bandwidth utilization, current studies focus on spatially dividing VR video frames into small segments (or called tiles), and only tiles within users' FoVs are delivered to users [12]–[22]. The FoV of a user is defined as the extent of the observable environment at any given time. By sending high-quality tiles in users' FoVs, the bandwidth utilization is improved. For example, the work in [18] proposed to transmit tiled 360 videos from a server (BS or AP) to multiple users and optimized the transmission time and power allocation to minimize the average transmission energy. By correlating the subjective requirements in the application layer with resource allocation in the PHY layer, the work in [20] investigated the issue of transmitting multi-quality tiled 360 videos to VR users in an power-efficient way. Besides, the work in [21] integrated scalable multi-layer 360 video tiling and optimal communication resource allocation to enable high-quality untethered VR streaming.

The aforementioned works either transmit relatively narrow user FoVs [12]–[14] or deliver video tiles via a single-antenna server [15]–[22]. Actually, wider FoV is rather important for a user to have immersive and presence experiences. By deploying multiple antennas at a server and designing efficient beamformers, UHD video frame transmission will be enabled, and users' visual experiences will be significantly enhanced. Additionally, a VR whitepaper by Huawei categorized 360 VR video systems into four levels [23], and the expected network requirements of some levels can be well satisfied by millimeter wave (mmWave) techniques. To this aim, advanced wireless communication techniques (e.g., multiple-input and multiple-output (MIMO) and mmWave), which can significantly improve data rates and reduce propagation latency, are explored in VR video transmission [4], [24]–[28]. For example, in [24] and [25], the authors studied the optimal streaming of a multi-quality tiled 360 VR video in a MIMO-orthogonal frequency division multiple access (OFDMA) system. In [26], the authors investigated the quality-of-experience (QoE) driven 360 VR video transmission in a multi-user massive MIMO system. The work in [4] utilized a mmWave-enabled communication architecture to support the panoramic and UHD VR video transmission. Aiming to improve users' immersive VR experiences in a wireless multi-user VR network, a mmWave multicast transmission framework was developed in [27]. Besides, the mmWave communication for ultra-reliable and low latency wireless VR was investigated in [28].

B. Motivation and Contributions

Although mmWave techniques can alleviate the current bottleneck for UHD video delivery, mmWave links are prone to the outage as they require line-of-sight (LoS) propagation. Various physical obstacles in the environment (including users' bodies) may completely break mmWave links and severely degrade MIMO channel quality as well [29]. As a result, the VR requirement for a perceptible image-quality degradation-free uniform experience cannot be accommodated. However, the mmWave VR-related works in [4], [27] and [28] and the MIMO VR-related works in [24]–[26] did not effectively investigate the crucial issue of guaranteeing the transmission reliability of VR video tiles. To tackle this issue, the coordinated multipoint (CoMP) transmission technique that can provide ultra-reliable connectivity from spatial diversity without relying on packet retransmission can be explored [30], [31]. For example, by cooperating, multiple APs can send the same 360 VR video tiles in parallel along multiple spatial paths. Since it will be unlikely that all spatial paths are degraded simultaneously, the requirement of VR applications for high reliability can be perfectly satisfied. Note that CoMP has been listed as a key feature in LTE-Advanced and 5G because of its ability to support delay-sensitive wireless applications (e.g., VR and live video streaming) [32]. Besides, it is extensively considered that proactive computing (e.g., image processing or frame rendering) enabled by machine learning methods is a crucial ability for a wireless VR network to mandate the stringent low latency requirement of UHD VR video transmission [1], [29], [33], [34]. Therefore, this paper investigates the issue of maximizing users' feeling of presence (FoP) in their virtual world in a mmWave-enabled VR network incorporating CoMP transmission and machine learning. The main contributions of this paper are summarized as follows:

- Owing to the user movement and the time-varying wireless channel conditions, we formulate the issue of maximizing users' FoP in virtual environments as a mixed-integer and sequential decision problem, subject to power consumption constraints on APs and users' HMDs. This problem is difficult to be directly solved by exploring conventional numerical optimization methods due to the lack of accurate users' tracking information (including users' locations and orientation angles) and mixed-integer and sequence-dependent characteristics.
- As users' historical tracking information is separately collected by diverse APs, a parallel echo state network

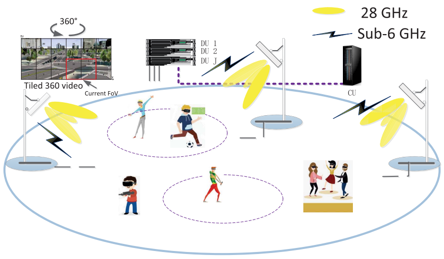


Fig. 1. A mmWave-enabled VR network incorporating CoMP transmission. $X = 6$ and $Y = 4$.

(ESN) learning method is designed to predict users' tracking information while accelerating the learning process.

- With the predicted results, we develop a deep reinforcement learning (DRL) based optimization algorithm to tackle the mixed-integer and sequential decision problem. Particularly, to avoid generating infeasible solutions by simultaneously optimizing all variables while alleviating the curse of dimensionality issue, the DRL-based optimization algorithm decomposes the formulated mixed-integer optimization problem into an integer association optimization problem and a continuous power control problem. Next, deep neural networks (DNNs) with continuous action output spaces followed by an action quantization scheme are implemented to solve the integer association problem. Given the association results, the power control problem is solved to criticize them and optimize the transmit power.
- Finally, the performance of the proposed DRL-based optimization algorithm is compared with various benchmark algorithms, and the impact of different design parameters is also discussed. Simulation results demonstrate the effectiveness of the proposed algorithm.

II. System Model and Problem Formulation

As shown in Fig. 1, we consider a mmWave-enabled VR network incorporating a CoMP transmission technique. This network includes a centralized unit (CU) connecting to J distributed units (DUs) via optical fiber links, a set $\mathcal{J}$ of J access points (APs) connected with the DUs, and a set of $\mathcal{U}$ of N ground mobile users wearing HMDs and watching 360 VR videos. To acquire immersive and interactive experiences, users will report their tracking information to their connected APs via reliable uplink communication links. Further, with collected users' tracking information, the CU will centrally simulate and construct virtual environments and coordinately transmit UHD VR videos to users via all APs in real time. To accomplish the task of enhancing users' immersive and interactive experiences in virtual environments, joint uplink and downlink communications should be considered. We assume that APs and users can work at both mmWave (exactly, 28 GHz)[1] and sub-6 GHz frequency bands, where the mmWave frequency band is reserved for downlink UHD VR video delivery, and the sub-6 GHz frequency band is allocated for uplink users' tracking information transmission. This is because an ultra-high data rate can be achieved on the mmWave frequency band, and sub-6 GHz can support reliable communications. Besides, to theoretically model the joint uplink and downlink communications, we suppose that the time domain is discretized into a sequence of time slots in the mmWave-enabled VR network and conduct the system modelling including tiled 360 video model, uplink and downlink transmission models, power consumption model, and FoP model.

A. Tiled 360 Video Model

For 360 video transmissions, tiling is adopted to improve transmission efficiency. For a 360 video, it is divided into $X \times Y$ rectangular tiles with X and Y denoting the numbers of tiles in each row and column, respectively [24], [35]. Owing to the impact of some factors (e.g., fluctuated channels and display resolutions of HMDs) on the quality of received videos by VR users, the dynamic adaptive streaming over HTTP (DASH) and dynamic video quality level selection schemes can be jointly explored to achieve smooth playback on the user side [36]. Note that a 360 video watched by a user includes multiple viewpoints or FoVs. A FoV may include multiple tiles, and a user can freely switch views when watching a 360 video. Nevertheless, a FoV that a user will view is closely correlated with the user's head orientation [24], [25]. Some methods such as prediction and empirical judgement can be leveraged to obtain users' head orientations. Here, we consider the case of determining users' head orientations based on the (historical) locations of users and APs.[2] Given information of users' head orientations, tiles in the corresponding FoVs can then be delivered to users via downlink transmission.

B. Uplink and Downlink Transmission Models

1) Uplink Transmission Model: Denote $\boldsymbol{x}_{it}^{\mathrm{3D}} = [x_{it}, y_{it}, h_i]^{\mathrm{T}}$ as the three dimensional (3D) Cartesian coordinate of the HMD worn by user i for all $i \in \mathcal{U}$ at time slot t, and $h_i \sim \mathcal{N}(\bar{h}, \sigma_h^2)$ is the user height. $[x_{it}, y_{it}]^{\mathrm{T}}$ is the two dimensional (2D) location of user i at time slot t. Denote $\boldsymbol{v}_j^{\mathrm{3D}} = [x_j, y_j, H_j]^{\mathrm{T}}$ as the 3D coordinate of the antenna of AP j, and H_j is the antenna height. Owing to the reliability requirement, users' data information (e.g., users' tracking information and profiles) is required to be successfully decoded by corresponding APs. We express the condition that an AP can successfully decode the received user data packets as follows

$$SNR_{ijt}^{\mathrm{ul}} = \frac{a_{ijt}^{\mathrm{ul}} p_{it} c_{ij} \hat{h}_{ijt}}{N_0 W^{\mathrm{ul}}/N} \geq \theta^{\mathrm{th}}, \quad \forall i, j, t, \tag{1}$$

where $a_{ijt}^{\mathrm{ul}} \in \{0, 1\}$ is an association variable indicating whether user i's uplink data packets can be successfully decoded by AP j at time slot t. The data packets can

[1] The 28 GHz frequency band is selected since it is currently a widely trialled/tested 5G band in the world.

[2] Certainly, the proposed learning method is also applicable to scenarios where users' head orientations need to be predicted if provided with real training data sets.

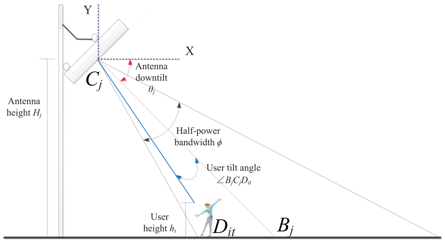

Y
X
Antenna downtilt θ_j
C_j
Antenna height H_j
Half-power bandwidth ϕ
User tilt angle $\angle B_jC_jD_{it}$
User height h_i
D_{it}
B_j

Fig. 2. Sectored antenna model of an AP.

be decoded if $a_{ijt}^{\text{ul}} = 1$; otherwise, $a_{ijt}^{\text{ul}} = 0$. p_{it} is the uplink transmit power of user i's HMD, c_{ij} is the Rayleigh channel gain, $\hat{h}_{ijt} = d_{ijt}^{-\alpha}(\boldsymbol{x}_{it}^{\text{3D}}, \boldsymbol{v}_j^{\text{3D}})$ is the uplink path-loss from user i to AP j with α being the fading exponent, $d_{ijt}(\cdot)$ denotes the Euclidean distance between user i and AP j, N_0 denotes the single-side noise spectral density, W^{ul} represents the uplink bandwidth. θ^{th} is the target signal-to-noise ratio (SNR) experienced at AP j for successfully decoding data packets from user i. Besides, considering the reliability requirement of uplink transmission and the stringent power constraint on HMDs, frequency division multiplexing (FDM) technique is adopted in this paper. The adoption of FDM technique can avoid the decoding failure resulting from uplink signal interferences and significantly reduce power consumption without compensating the signal-to-interference-plus-noise ratio (SINR) loss caused by uplink interferences.

Additionally, we assume that each user i can connect to at most one AP j via the uplink channel at each time slot t, i.e., $\sum_{j\in\mathcal{J}} a_{ijt}^{\text{ul}} \leq 1$, $\forall i$. This is reasonable because it is unnecessary for each AP to decode all users' data successfully at each time slot t. A user merely connects to an AP (e.g., the nearest AP if possible) will greatly reduce power consumption. Meanwhile, considering the stringent low latency requirements of VR applications and the time consumption of processing (e.g., decoding and checking) received user data packets, we assume that an AP can serve up to $\tilde{M}$ users during a time slot, i.e., $\sum_{i\in\mathcal{U}} a_{ijt}^{\text{ul}} \leq \tilde{M}$, $\forall j$.

2) Downlink Transmission Model: In the downlink transmission configuration, antenna arrays are deployed to perform directional beamforming. For analysis facilitation, a sectored antenna model [37], which consists of four components, i.e., the half-power beamwidth ϕ, the antenna downtilt angle θ_j $\forall j$, the antenna gain of the mainlobe G, and the antenna gain of the sidelobe g, shown in Fig. 2, is exploited to approximate actual array beam patterns. The antenna gain of the transmission link from AP j to user i is

$$f_{ijt} = \begin{cases} G & \angle B_jC_jD_{it} \leq \frac{\phi}{2}, \\ g & \text{otherwise,} \end{cases} \quad \forall i,j,t, \tag{2}$$

where $\angle B_jC_jD_{it}$ represents user i's tilt angle towards AP j, the location of the point 'B_j' can be determined by AP j's 2D coordinate $\boldsymbol{v}_j^{\text{2D}} = [x_j, y_j]^{\text{T}}$ and θ_j, the point 'D_{it}' represent user i's position, the point 'C_j' denotes the position of AP j's antenna.

For any AP j, the 2D coordinate $\boldsymbol{x}_{bj}^{\text{2D}} = [x_{bj}, y_{bj}]^{\text{T}}$ of point 'B_j' can be given by

$$x_{bj} = d_j(x_o - x_j)/r_j + x_j, \quad \forall j, \tag{3}$$

$$y_{bj} = d_j(y_o - y_j)/r_j + y_j, \quad \forall j, \tag{4}$$

where $d_j = H_j/\tan(\theta_j)$, $r_j = ||\boldsymbol{x}_o - \boldsymbol{v}_j^{\text{2D}}||_2$, and $\boldsymbol{x}_o = [x_o, y_o]^{\text{T}}$ is 2D coordinate of the center point of the considered communication area.

Then, user i's tilt angle towards AP j can be written as

$$\angle B_jC_jD_{it} = \arccos\left(\frac{\overrightarrow{C_jB_j} \cdot \overrightarrow{C_jD_{it}}}{\|C_jB_j\|_2 \|C_jD_{it}\|_2}\right), \quad \forall i,j,t, \tag{5}$$

where direction vectors $\overrightarrow{C_jB_j} = (x_{bj} - x_j, y_{bj} - y_j, -H_j)$ and $\overrightarrow{C_jD_{it}} = (x_{it} - x_j, y_{it} - y_j, h_i - H_j)$.

A mmWave link may be blocked if a user turns around; this is because the user wears an HMD in front of his/her forehead. Denote ϑ as the maximum angle within which an AP can experience LoS transmission towards its downlink associated users. For user i at time slot t, an indicator variable b_{ijt} introduced to indicate the blockage effect of user i's body is given by

$$b_{ijt} = \begin{cases} 1 & \angle(\vec{A}_{jit}, \vec{x}_{it}) > \vartheta, \\ 0 & \text{otherwise,} \end{cases} \quad \forall i,j,t, \tag{6}$$

where $\angle(\vec{A}_{jit}, \vec{x}_{it})$ represents the orientation angle of user i at time slot t, which can be determined by locations of both user i and AP j, $\vec{x}_{it} = (x_{it} - x_{it-1}, y_{it} - y_{it-1})$ is a direction vector. When $t = 1$, the direction vector $\vec{x}_{i1} = (x_{i1}, y_{i1})$. $\vec{A}_{jit} = (x_j - x_{it}, y_j - y_{it})$ is a direction vector between the AP j and user i.

Given $\vec{A}_{jit}$ and $\vec{x}_{it}$, we can calculate the orientation angle of user i that is also the angle between $\vec{A}_{jit}$ and $\vec{x}_{it}$ by

$$\angle(\vec{A}_{jit}, \vec{x}_{it}) = \arccos\left(\frac{\vec{A}_{jit} \cdot \vec{x}_{it}}{||\vec{A}_{jit}||_2 ||\vec{x}_{it}||_2}\right), \quad \forall i,j,t. \tag{7}$$

The channel gain coefficient h_{ijkt} of an LoS link and a non line-of-sight (NLoS) link between the k-th antenna element of AP j and user i at time slot t can take the form [37]

$$\begin{aligned} &10\text{log}_{10}(h_{ijkt}h_{ijkt}^{\text{H}}) \\ &= \begin{cases} 10\eta_{\text{LoS}}\text{log}_{10}(d_{ijt}(x_{it}^{\text{3D}}, v_j^{\text{3D}})) \\ +20\text{log}_{10}\left(\frac{4\pi f_c}{c}\right) + \\ 10\text{log}_{10}f_{ijt} + \mu_k^{\text{LoS}}, & b_{ijt} = 0 \\ 10\eta_{\text{NLoS}}\text{log}_{10}(d_{ijt}(x_{it}^{\text{3D}}, v_j^{\text{3D}})) \\ +20\text{log}_{10}\left(\frac{4\pi f_c}{c}\right) + \\ 10\text{log}_{10}f_{ijt} + \mu_k^{\text{NLoS}}, & b_{ijt} = 1 \quad \forall i,j,k,t, \end{cases} \end{aligned} \tag{8}$$

where f_c (in Hz) is the carrier frequency, c (in m/s) the light speed, η_{LoS} (in dB) and η_{NLoS} (in dB) the path-loss exponents of LoS and NLoS links, respectively, $\mu_{\text{LoS}} \sim \mathcal{CN}(0, \sigma_{\text{LoS}}^2)$ (in dB) and $\mu_{\text{NLoS}} \sim \mathcal{CN}(0, \sigma_{\text{NLoS}}^2)$ (in dB).

For any user i, to satisfy its immersive experience requirement, its downlink achievable data rate (denoted by r_{it}^{dl}) from cooperative APs should be no less than a data rate threshold γ^{th}, i.e.,

$$r_{it}^{\text{dl}} \geq \gamma^{\text{th}}, \quad \forall i, t. \tag{9}$$

Define $a_{it}^{\text{dl}} \in \{0, 1\}$ as an association variable indicating whether the user i's data rate requirement can be satisfied at time slot t. $a_{it}^{\text{dl}} = 1$ indicates that its data rate requirement can be satisfied; otherwise, $a_{it}^{\text{dl}} = 0$. Then, for any user i at time slot t, according to Shannon capacity formula and the principle of CoMP transmission, we can calculate r_{it}^{dl} by

$$r_{it}^{\text{dl}} = W^{\text{dl}} \log_2 \left(1 + \frac{a_{it}^{\text{dl}} \left| \sum_{j \in \mathcal{J}} \boldsymbol{h}_{ijt}^{\text{H}} \boldsymbol{g}_{ijt} \right|^2}{N_0 W^{\text{dl}} + I_{it}^{\text{dl}}}\right), \quad \forall i, t, \tag{10}$$

where $\boldsymbol{h}_{ijt} = [h_{ij1t}, \ldots, h_{ijKt}]^{\text{T}} \in \mathbb{C}^K$ is a channel gain coefficient vector with K denoting the number of antenna elements, $\boldsymbol{g}_{ijt} \in \mathbb{C}^K$ is the transmit beamformer pointed at user i from AP j, W^{dl} represents the downlink system bandwidth. Owing to the directional propagation, for user i, not all users will be its interfering users. It is regarded that users whose distances from user i are small than D^{th} will be user i's interfering users, where D^{th} is determined by antenna configuration of APs (e.g., antenna height and downtilt angle). Denote the set of interfering users of user i at time slot t by $\mathcal{M}_{it}$, then, we have $I_{it}^{\text{dl}} = \sum_{m \in \mathcal{M}_{it}} a_{mt}^{\text{dl}} \left| \sum_{j \in \mathcal{J}} \boldsymbol{h}_{mjt}^{\text{H}} \boldsymbol{g}_{mjt} \right|^2$.

C. Power Consumption Model

HMDs are generally battery-driven and constrained by the maximum instantaneous power. For any user i's HMD, define p_{it}^{tot} as its instantaneous power consumption including the transmit power and circuit power consumption (e.g., power consumption of mixers, frequency synthesizers and digital-to-analog converters) at time slot t, we then have

$$p_{it}^{\text{tot}} \leq \tilde{p}_i, \quad \forall i, t, \tag{11}$$

where $p_{it}^{\text{tot}} = p_{it} + p_i^c$, p_i^c denotes the HMD's circuit power consumption during a time slot, and $\tilde{p}_i$ is a constant. Without loss of generality, we assume that all users' HMDs are homogenous.

The instantaneous power consumption of each AP is also constrained. As CoMP transmission technique is explored, for any AP j, we can model its instantaneous power consumption at time slot t as the following

$$\sum\nolimits_{i \in \mathcal{U}} a_{it}^{\text{dl}} \boldsymbol{g}_{ijt}^{\text{H}} \boldsymbol{g}_{ijt} + E_j^c \leq \tilde{E}_j, \quad \forall j, t, \tag{12}$$

where E_j^c is a constant representing the circuit power consumption, $\tilde{E}_j$ is the maximum instantaneous power of AP j.

D. Feeling of Presence Model

In VR applications, FoP is defined as a state that VR users experience the sense of 'being there' or a full presence- or immersion-feeling of herself/himself in fictitious environments [38]–[40]. To model VR users' virtual experience, a breaks-in-presence (BIP) metric that considered video quality, latency and motion tracking was introduced in [29]. Different from BIP, we incorporate five key performance indicators (KPIs) proposed for VR applications to model FoP. These KPIs include video quality, smoothness, latency, motion tracking, power and thermal efficiency that can be improved by optimizing physical network resources. Yet, the first three KPIs are subjective and then seriously hinder the modeling of FoP. To tackle this issue, the objective indicators such as peak signal-to-noise ratio (PSNR) [41], logarithmic utility function [42] and inverse of data rate can be introduced. However, mathematical expressions of these indicators are complicated, which will lead to a highly challenging objective function. Fortunately, under the constraint on VR users' data rate requirements, one can observe that the maximization of $\sum_{i \in \mathcal{U}} a_{it}^{\text{dl}}$ will bring improvements in video quality, smoothness and latency to VR users. Besides, considering that the successful reception of uplink user tracking information is critical for precision motion tracking and the improvement of power and thermal efficiency desires the optimization of total power consumption of HMDs, we model the FoP experienced by users at time slot t as the following

$$\begin{aligned} F_t(\boldsymbol{a}_t^{\text{ul}}, \boldsymbol{a}_t^{\text{dl}}, \boldsymbol{p}_t^{\text{tot}}) = {} & B_t^{\text{ul}}\left(\boldsymbol{a}_t^{\text{ul}}\right) + B_t^{\text{dl}}\left(\boldsymbol{a}_t^{\text{dl}}\right) \\ & - \quad \sum\nolimits_{i \in \mathcal{U}} \sum\nolimits_{j \in \mathcal{J}} a_{ijt}^{\text{ul}} p_{it}^{\text{tot}} / \tilde{p}_i, \end{aligned} \tag{13}$$

where $\boldsymbol{p}_t^{\text{tot}} = [p_{1t}^{\text{tot}}, p_{2t}^{\text{tot}}, \ldots, p_{Nt}^{\text{tot}}]^{\text{T}}$, $B_t^{\text{ul}}\left(\boldsymbol{a}_t^{\text{ul}}\right) = \frac{1}{N} \sum\limits_{i \in \mathcal{U}} \sum\limits_{j \in \mathcal{J}} a_{ijt}^{\text{ul}}$ with $\boldsymbol{a}_t^{\text{ul}} = [a_{11t}^{\text{ul}}, \ldots, a_{ijt}^{\text{ul}}, \ldots, a_{NJt}^{\text{ul}}]^{\text{T}}$, $B_t^{\text{dl}}\left(\boldsymbol{a}_t^{\text{dl}}\right) = \frac{1}{N} \sum\limits_{i \in \mathcal{U}} a_{it}^{\text{dl}}$ with $\boldsymbol{a}_t^{\text{dl}} = [a_{1t}^{\text{dl}}, \ldots, a_{it}^{\text{dl}}, \ldots, a_{Nt}^{\text{dl}}]^{\text{T}}$.

E. Objective Function and Problem Formulation

To guarantee immersive and interactive VR experiences of users over a period of time slots, uplink user data packets should be successfully decoded, and downlink data rate requirements of users should be satisfied at each time slot. Additionally, reducing HMDs' power consumption indicates less heat generation, which can enhance users' VR experiences. Therefore, our goal is to maximize users' FoP over a period of time slots, subject to some physical resource constraints. Combining with the above analysis, we can formulate the problem of enhancing users' immersive experiences as below

$$\underset{\{\boldsymbol{a}_t^{\text{ul}}, \boldsymbol{a}_t^{\text{dl}}, \boldsymbol{p}_t, \boldsymbol{g}_{ijt}\}}{\text{maximize}} \quad \liminf_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} F_t(\boldsymbol{a}_t^{\text{ul}}, \boldsymbol{a}_t^{\text{dl}}, \boldsymbol{p}_t^{\text{tot}}) \tag{14a}$$

$$\text{s.t.} \quad \sum\nolimits_{j \in \mathcal{J}} a_{ijt}^{\text{ul}} \leq 1, \quad \forall i, t \tag{14b}$$

$$\sum\nolimits_{i \in \mathcal{U}} a_{ijt}^{\text{ul}} \leq \tilde{M}, \forall j, t \tag{14c}$$

$$a_{ijt}^{\text{ul}} \in \{0, 1\}, \quad \forall i, j, t \tag{14d}$$

$$a_{it}^{\text{dl}} \in \{0, 1\}, \quad \forall i, t \tag{14e}$$

$$0 \leq p_{it} \leq \tilde{p}_i - p_i^c, \quad \forall i, t \tag{14f}$$

$$(1), \ (9), \ (12), \tag{14g}$$

where $\boldsymbol{p}_t = [p_{1t}, p_{2t}, \ldots, p_{Nt}]^{\text{T}}$.

However, the solution to (14) is highly challenging due to the unknown users' tracking information at each time slot.

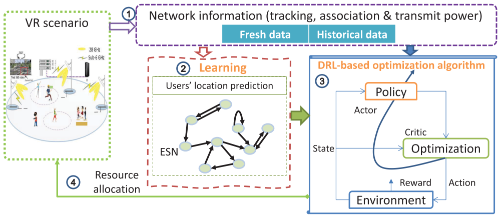


Fig. 3. Working diagram of a framework of solving (14).

Given users' tracking information, the solution to (14) is still NP-hard or even non-detectable. It can be confirmed that (14) is a mixed-integer non-linear programming (MINLP) problem as it simultaneously contains zero-one variables, continuous variables, and non-linear constraints. Further, we can know that (9) and (12) are non-convex with respect to (w.r.t) a_{it}^{dl} and $\boldsymbol{g}_{ijt}$, $\forall i$, j, by evaluating the Hessian matrix. To tackle the tricky problem, we develop a novel solution framework as depicted in Fig. 3. In this framework, we first propose to predict users' tracking information using a machine learning method. With the predicted results, we then develop a DRL-based optimization algorithm to solve the MINLP problem. The procedure of solving (14) is elaborated in the following sections.

III. Users' Location Prediction

As analyzed above, the efficient user-AP association and transmit power of both HMDs and APs are configured on the basis of the accurate perception of users' tracking information. If the association and transmit power are identified without knowledge of users' tracking information, users may have degraded VR experiences, and the working duration of users' HMDs may be dramatically shortened. Meanwhile, owing to the stringent low latency requirement, the user-AP association and transmit power should be proactively determined to enhance users' immersive and interactive VR experiences. Hence, APs must collect fresh and historical tracking information for users' tracking information prediction in future time slots. With predicted tracking information, the user-AP association and transmit power can be configured in advance. Certainly, from (7), we observe that users' orientation angles can be obtained by their and APs' locations; thus, we only predict users' locations in this section. Machine learning is convinced as a promising proposal to predict users' locations. In machine learning methods, the accuracy and completeness of sample collection are crucial for accurate model training. However, the user-AP association may vary with user movement, which indicates that location information of each user may scatter in multiple APs, and each AP may only collect partial location information of its associated users after a period of time. To tackle this issue, we develop a parallel machine learning method, which exploits J slave virtual machines (VMs) created in the CU to train learning models for each user, as shown in Fig. 4. Besides, for each AP, it will feed its locally collected location information to a slave VM for training. In this way, the prediction process can also be accelerated. With the predicted results, the CU can then proactively allocate system resources by solving (14).

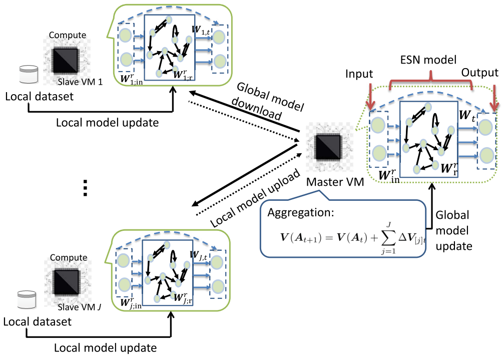


Fig. 4. Architecture of the parallel ESN learning method.

A. Echo State Network

In this section, the principle of echo state network (ESN) is exploited to train users' location prediction model as the ESN method can efficiently analyze the correlation of users' location information and quickly converge to obtain users' predicted locations [43].[3] It is noteworthy that there are some differences between the traditional ESN method and the developed parallel ESN learning method. The traditional ESN method is a centralized learning method with the requirement of the aggregation of all users' locations scattered in all APs, which is not required for the parallel ESN learning method. What's more, the traditional ESN method can only be used to conduct data prediction in a time slot while the parallel ESN learning method can predict users' locations in $M > 1$ time slots. An ESN is a recurrent neural network that can be partitioned into three components: input, ESN model, and output, as shown in Fig. 4. For any user $i \in \mathcal{U}$, the N_i-dimensional input vector $\boldsymbol{x}_{it} \in \mathbb{R}^{N_i}$ is fed to an N_r-dimensional reservoir whose internal state $\boldsymbol{s}_{i(t-1)} \in \mathbb{R}^{N_r}$ is updated according to the state equation

$$\boldsymbol{s}_{it} = \tanh\left(\boldsymbol{W}_{\text{in}}^{r}\boldsymbol{x}_{it} + \boldsymbol{W}_{\text{r}}^{r}\boldsymbol{s}_{i(t-1)}\right), \tag{15}$$

where $\boldsymbol{W}_{\text{in}}^{r} \in \mathbb{R}^{N_r \times N_i}$ and $\boldsymbol{W}_{\text{r}}^{r} \in \mathbb{R}^{N_r \times N_r}$ are randomly generated matrices with each matrix element locating in the interval $(0, 1)$.

The evaluated output of the ESN at time slot t is given by

$$\hat{\boldsymbol{y}}_{i(t+1)} = \boldsymbol{W}_{\text{in}}^{o}\boldsymbol{x}_{it} + \boldsymbol{W}_{\text{r}}^{o}\boldsymbol{s}_{it}, \tag{16}$$

where $\boldsymbol{W}_{\text{in}}^{o} \in \mathbb{R}^{N_o \times N_i}$, $\boldsymbol{W}_{\text{r}}^{o} \in \mathbb{R}^{N_o \times N_r}$ are trained based on collected training data samples.

To train the ESN model, suppose we are provided with a sequence of Q desired input-outputs pairs $\{(\boldsymbol{x}_{i1}, \boldsymbol{y}_{i1}), \ldots, (\boldsymbol{x}_{iQ}, \boldsymbol{y}_{iQ})\}$ of user i, where $\boldsymbol{y}_{it} \in \mathbb{R}^{N_o}$ is the target location of user i at time slot t. Define the hidden matrix $\boldsymbol{X}_{it}$ as

$$\boldsymbol{X}_{it} = \begin{bmatrix} \boldsymbol{x}_{i1} & \cdots & \boldsymbol{x}_{iQ} \\ \boldsymbol{s}_{i1} & & \boldsymbol{s}_{iQ} \end{bmatrix}. \tag{17}$$

[3] Note that, the long short-term memory (LSTM) network is not leveraged because the network structure of ESN is simpler and a parallel learning method is desired. The ESN can be extended to a parallel ESN learning method.

The optimal output weight matrix is then achieved by solving the following regularized least-square problem

$$\boldsymbol{W}_t^\star = \underset{\boldsymbol{W}_t \in \mathbb{R}^{(N_i+N_r)\times N_o}}{\arg\min} \frac{1}{Q} l\left(\boldsymbol{X}_{it}^{\mathrm{T}} \boldsymbol{W}_t\right) + \xi r(\boldsymbol{W}_t) \quad (18)$$

where $\boldsymbol{W}_t = [\boldsymbol{W}_{\mathrm{in}}^o \boldsymbol{W}_{\mathrm{r}}^o]^{\mathrm{T}}$, $\xi \in \mathbb{R}_+$ is a positive scalar known as regularization factor, the loss function $l(\boldsymbol{X}_{it}^{\mathrm{T}} \boldsymbol{W}_t) = \frac{1}{2}||\boldsymbol{X}_{it}^{\mathrm{T}} \boldsymbol{W}_t - \boldsymbol{Y}_{it}||_F^2$, the regulator $r(\boldsymbol{W}_t) = ||\boldsymbol{W}_t||_F^2$, and the target location matrix $\boldsymbol{Y}_{it} = [\boldsymbol{y}_{i1}^{\mathrm{T}}; \ldots; \boldsymbol{y}_{iQ}^{\mathrm{T}}] \in \mathbb{R}^{Q\times N_o}$.

B. Parallel ESN Learning Method for Users' Location Prediction

Based on the principle of the ESN method, we next elaborate on the procedure of the parallel ESN learning method for users' location prediction. To facilitate the analysis, we make the following assumptions on the regulator and the loss function.

Assumption 1: The function $r : \mathbb{R}^{m\times n} \to \mathbb{R}$ is ζ-strongly convex, i.e., $\forall i \in \{1, 2, \ldots, n\}$, $\forall \boldsymbol{X}$, and $\Delta \boldsymbol{X} \in \mathbb{R}^{m\times n}$, we have [44]

$$r(\boldsymbol{X} + \Delta\boldsymbol{X}) \geq r(\boldsymbol{X}) + \nabla r(\boldsymbol{X}) \odot \Delta\boldsymbol{X} + \zeta||\Delta\boldsymbol{X}||_F^2/2, \quad (19)$$

where $\nabla r(\cdot)$ denotes the gradient of $r(\cdot)$.

Assumption 2: The function $l : \mathbb{R} \to \mathbb{R}$ are $\frac{1}{\mu}$-smooth, i.e., $\forall i \in \{1, 2, \ldots, n\}$, $\forall x$, and $\Delta x \in \mathbb{R}$, we have

$$l(x + \Delta x) \leq l(x) + \nabla l(x)\ \Delta x + (\Delta x)^2/2\mu, \quad (20)$$

where $\nabla l(\cdot)$ represents the gradient of $l(\cdot)$.

According to Fenchel-Rockafeller duality, we can formulate the local dual optimization problem of (18) in the following way.

Lemma 1: For a set of J slave VMs and a typical user i, the dual problem of (18) can be written as follows

$$\underset{\boldsymbol{A}\in\mathbb{R}^{Q\times N_o}}{\mathrm{maximize}} \left\{ -\xi r^\star\left(\frac{1}{\xi Q}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\right) - \frac{1}{Q}\sum_{m=1}^{Q}\sum_{n=1}^{N_o} l^\star(-a_{mn}) \right\} \quad (21)$$

where

$$r^\star(\boldsymbol{C}) = \frac{1}{4}\sum\nolimits_{n=1}^{N_o} \boldsymbol{z}_n^{\mathrm{T}} \boldsymbol{C}\boldsymbol{C}^{\mathrm{T}} \boldsymbol{z}_n, \quad (22)$$

$$l^\star(-a_{mn}) = -a_{mn} y_{mn} + a_{mn}^2/2, \quad (23)$$

$\boldsymbol{A} \in \mathbb{R}^{Q\times N_o}$ is a Lagrangian multiplier matrix, $\boldsymbol{z}_n \in \mathbb{R}^{N_o}$ is a column vector with the n-th element being one and all other elements being zero, $\boldsymbol{X}$ is a lightened notation of $\boldsymbol{X}_{it} = \begin{bmatrix} \boldsymbol{x}_{i(t-1)} & \ldots & \boldsymbol{x}_{i(t-Q)} \\ \boldsymbol{s}_{i(t-1)} & & \boldsymbol{s}_{i(t-Q)} \end{bmatrix}$, and y_{mn} is an element of matrix $\boldsymbol{Y} = [\boldsymbol{y}_{it}^{\mathrm{T}}; \ldots; \boldsymbol{y}_{i(t-Q+1)}^{\mathrm{T}}]$ at the location of the m-th row and the n-th column.

Proof: Please refer to Appendix A in technical report [45]. ■

Denote the objective function of (21) as $D(\boldsymbol{A})$, and define $\boldsymbol{V}(\boldsymbol{A}) := \frac{1}{\xi Q}(\boldsymbol{X}\boldsymbol{A})^{\mathrm{T}} \in \mathbb{R}^{N_o\times(N_i+N_r)}$, we can then rewrite $D(\boldsymbol{A})$ as

$$D(\boldsymbol{A}) = -\xi r^\star(\boldsymbol{V}(\boldsymbol{A})) - \sum\nolimits_{j=1}^{J} R_j(\boldsymbol{A}_{[j]}), \quad (24)$$

where $R_j(\boldsymbol{A}_{[j]}) = \frac{1}{Q}\sum_{m\in\mathcal{Q}_j}\sum_{n=1}^{N_o} l^\star(-a_{mn})$, $\boldsymbol{A}_{[j]} = \hat{\boldsymbol{Z}}_j \boldsymbol{A}$, and $\hat{\boldsymbol{Z}}_j \in \mathbb{R}^{Q\times Q}$ is a square matrix with $J \times J$ blocks. In $\hat{\boldsymbol{Z}}_j$, the block in the j-th row and j-th column is a $q_j \times q_j$ identity matrix with q_j being the cardinality of a set $\mathcal{Q}_j$ and all other blocks are zero matrices, $\mathcal{Q}_j$ is an index set including the indices of Q data samples fed to slave VM j.

Then, for a given matrix $\boldsymbol{A}^t$, varying its value by $\Delta\boldsymbol{A}^t$ will change (24) as below

$$\begin{aligned} D(\boldsymbol{A}^t + \Delta\boldsymbol{A}^t) = &-\xi r^\star(\boldsymbol{V}(\boldsymbol{A}^t + \Delta\boldsymbol{A}^t)) \\ &- \sum\nolimits_{j=1}^{J} R_j(\boldsymbol{A}_{[j]}^t + \Delta\boldsymbol{A}_{[j]}^t), \end{aligned} \quad (25)$$

where $\Delta\boldsymbol{A}_{[j]}^t = \hat{\boldsymbol{Z}}_j \Delta\boldsymbol{A}^t$.

Note that the second term of the right-hand side (RHS) of (25) includes the local changes of each VM j, while the first term involves the global variations.

As $r(\cdot)$ is ζ-strongly convex, $r^\star(\cdot)$ is then $\frac{1}{\zeta}$-smooth [44]. Thus, we can calculate the upper bound of $r^\star(V(\boldsymbol{A}^t + \Delta\boldsymbol{A}^t))$ as follows

$$\begin{aligned} r^\star(V(\boldsymbol{A}^t + \Delta\boldsymbol{A}^t)) &\leq r^\star\left(\boldsymbol{V}(\boldsymbol{A}^t)\right) \\ &+ \frac{1}{\xi Q}\sum_{n=1}^{N_o} \boldsymbol{z}_n^{\mathrm{T}} \nabla r^\star(\boldsymbol{V}(\boldsymbol{A}^t))\boldsymbol{X}\Delta\boldsymbol{A}^t \boldsymbol{z}_n \\ &+ \frac{\kappa}{2(\xi Q)^2}\sum_{n=1}^{N_o} \left\|\boldsymbol{X}\Delta\boldsymbol{A}^t\boldsymbol{z}_n\right\|^2 \\ = r^\star\left(\boldsymbol{V}(\boldsymbol{A}^t)\right) &+ \frac{1}{\xi Q}\sum_{j=1}^{J}\sum_{n=1}^{N_o} \boldsymbol{z}_n^{\mathrm{T}} \nabla r^\star(\boldsymbol{V}(\boldsymbol{A}^t))\boldsymbol{X}_{[j]}\Delta\boldsymbol{A}_{[j]}^t \boldsymbol{z}_n \\ &+ \frac{\kappa}{2(\xi Q)^2}\sum_{j=1}^{J}\sum_{n=1}^{N_o} \left\|\boldsymbol{X}_{[j]}\Delta\boldsymbol{A}_{[j]}^t\boldsymbol{z}_n\right\|^2, \end{aligned} \quad (26)$$

where $\boldsymbol{X}_{[j]} = \boldsymbol{X}\hat{\boldsymbol{Z}}_j$, $\kappa > \frac{1}{\zeta}$ is a data dependent constant measuring the difficulty of the partition to the whole samples.

By substituting (26) into (25), we obtain

$$\begin{aligned} D(\boldsymbol{A}^t + \Delta\boldsymbol{A}^t) &\geq -\xi r^\star\left(\boldsymbol{V}(\boldsymbol{A}^t)\right) \\ &- \frac{1}{Q}\sum_{j=1}^{J}\sum_{n=1}^{N_o} \boldsymbol{z}_n^{\mathrm{T}} \nabla r^\star(\boldsymbol{V}(\boldsymbol{A}^t))\boldsymbol{X}_{[j]}\Delta\boldsymbol{A}_{[j]}^t \boldsymbol{z}_n \\ &- \frac{\kappa}{2\xi Q^2}\sum_{j=1}^{J}\sum_{n=1}^{N_o} \left\|\boldsymbol{X}_{[j]}\Delta\boldsymbol{A}_{[j]}^t\boldsymbol{z}_n\right\|^2 - \sum_{j=1}^{J} R_j(\boldsymbol{A}_{[j]}^t + \Delta\boldsymbol{A}_{[j]}^t). \end{aligned} \quad (27)$$

From (27), we observe that the problem of maximizing $D(\boldsymbol{A}^t + \Delta\boldsymbol{A}^t)$ can be decomposed into J subproblems, and J slave VMs can then be exploited to optimize these subproblems separately. If slave VM j can optimize $\Delta\boldsymbol{A}^t$ using its collected data samples by maximizing the RHS of (27), the resultant improvements can be aggregated to drive $\boldsymbol{D}(\boldsymbol{A}^t)$ toward the optimum. The detailed procedure is described below.

As shown in Fig. 4, during any communication round t, a master VM produces $\boldsymbol{V}(\boldsymbol{A}^t)$ using updates received at the last round and shares it with all slave VMs. The task at any slave VM j is to obtain $\Delta\boldsymbol{A}^t_{[j]}$ by maximizing the following problem

$$\begin{aligned}\Delta\boldsymbol{A}^{t\star}_{[j]} &= \mathop{\arg\max}_{\Delta\boldsymbol{A}^t_{[j]}\in\mathbb{R}^{Q\times N_o}} \Delta D_j\left(\Delta\boldsymbol{A}^t_{[j]};\boldsymbol{V}(\boldsymbol{A}^t),\boldsymbol{A}^t_{[j]}\right)\\ &= \mathop{\arg\max}_{\Delta\boldsymbol{A}^t_{[j]}\in\mathbb{R}^{Q\times N_o}}\Bigg\{-R_j\left(\boldsymbol{A}^t_{[j]}+\Delta\boldsymbol{A}^t_{[j]}\right)-\frac{\xi}{J}r^\star(\boldsymbol{V}(\boldsymbol{A}^t))\\ &\quad -\frac{1}{Q}\sum_{n=1}^{N_o}\boldsymbol{z}_n^{\mathrm{T}}\nabla r^\star(\boldsymbol{V}(\boldsymbol{A}^t))\boldsymbol{X}_{[j]}\Delta\boldsymbol{A}^t_{[j]}\boldsymbol{z}_n\\ &\quad -\frac{\kappa}{2\xi Q^2}\sum_{n=1}^{N_o}\left\|\boldsymbol{X}_{[j]}\Delta\boldsymbol{A}^t_{[j]}\boldsymbol{z}_n\right\|^2\Bigg\}. \end{aligned} \tag{28}$$

Calculate the derivative of $\Delta D_j\left(\Delta\boldsymbol{A}^t_{[j]};\boldsymbol{V}(\boldsymbol{A}^t),\boldsymbol{A}^t_{[j]}\right)$ over $\Delta\boldsymbol{A}^t_{[j]}$, and force the derivative result to be zero, we have

$$\begin{aligned}\Delta\boldsymbol{A}^{t\star}_{[j]} &= \left(\hat{\boldsymbol{Z}}_j+\frac{\kappa}{\xi Q}\boldsymbol{X}^{\mathrm{T}}_{[j]}\boldsymbol{X}_{[j]}\right)^{-1}\\ &\quad\times\left(\boldsymbol{Y}_{[j]}-\boldsymbol{A}^t_{[j]}-\frac{1}{2}\boldsymbol{X}^{\mathrm{T}}_{[j]}\boldsymbol{V}^{\mathrm{T}}(\boldsymbol{A}_t)\right),\end{aligned} \tag{29}$$

where $\boldsymbol{Y}_{[j]}=\hat{\boldsymbol{Z}}_j\boldsymbol{Y}$.

Next, slave VM j, $\forall j$, sends $\Delta\boldsymbol{V}^t_{[j]}=\frac{1}{\xi Q}(\boldsymbol{X}_{[j]}\Delta\boldsymbol{A}^{t\star}_{[j]})^{\mathrm{T}}$ to the master VM. The master VM updates the global model as $\boldsymbol{V}(\boldsymbol{A}^t+\Delta\boldsymbol{A}^t)=\boldsymbol{V}(\boldsymbol{A}^t)+\sum_{j=1}^J\Delta\boldsymbol{V}^t_{[j]}$. Finally, alteratively update $\boldsymbol{V}(\boldsymbol{A}^t)$ and $\{\Delta\boldsymbol{A}^{t\star}_{[j]}\}_{j=1}^J$ on the global and local sides, respectively. It is expected that the solution to the dual problem can be enhanced at every step and will converge after several iterations.

At time slot t, based on the above derivation, the parallel ESN learning method for predicting locations of user i, $\forall i$, in M time slots can be summarized in Algorithm 1.

Algorithm 1 Parallel ESN Learning for User Location Prediction

```
Initialization: Data samples of all slave VMs. For any slave VM $j$, it randomly initiates a starting point $\boldsymbol{A}^0_{[j]}\in\mathbb{R}^{Q\times N_o}$. The master VM collects $\frac{1}{\xi Q}(\boldsymbol{X}_{[j]}\boldsymbol{A}^0_{[j]})^{\mathrm{T}}$ from all slave VMs, generates $\boldsymbol{V}(\boldsymbol{A}^0)=\sum_{j=1}^J\frac{1}{\xi Q}(\boldsymbol{X}_{[j]}\boldsymbol{A}^0_{[j]})^{\mathrm{T}}$, and then share the model $\boldsymbol{V}(\boldsymbol{A}^0)$ with all slave VMs. Let $\kappa=J/\zeta$.
for $r=0:\bar{r}_{\max}-1$ do
  for each slave VM $j\in\{1,2,\ldots,J\}$ in parallel do
    Calculate $\Delta\boldsymbol{A}^{r\star}_{[j]}$ using (29), update and store the local Lagrangian multiplier
      $\boldsymbol{A}^{r+1}_{[j]}=\boldsymbol{A}^r_{[j]}+\Delta\boldsymbol{A}^{r\star}_{[j]}/(r+1).$   (30)
    Compute the following local model and send it to the master VM
      $\Delta\boldsymbol{V}^r_{[j]}=\left(\boldsymbol{X}_{[j]}\Delta\boldsymbol{A}^{r\star}_{[j]}\right)^{\mathrm{T}}/\xi Q.$   (31)
  end for
  Given local models, the master VM updates the global model as
    $\boldsymbol{V}(\boldsymbol{A}^{r+1})=\boldsymbol{V}(\boldsymbol{A}^r)+\sum_{j=1}^J\Delta\boldsymbol{V}^r_{[j]},$   (32)
  and then share the updated global model $\boldsymbol{V}(\boldsymbol{A}^{r+1})$ with all slave VMs.
end for
Let $\boldsymbol{W}^{\mathrm{T}}=\nabla r^\star(\boldsymbol{V}(\boldsymbol{A}^r))$ and predict user $i$'s location $\hat{\boldsymbol{y}}_{it}$ by (16). Meanwhile, by iteratively assigning $\boldsymbol{x}_{i(t+1)}=\hat{\boldsymbol{y}}_{it}$, each user $i$'s locations in $M$ time slots can be obtained.
Output: The predicted locations of user $i$, $\hat{\boldsymbol{Y}}_{it}=[\hat{\boldsymbol{y}}^{\mathrm{T}}_{i(t+1)};\ldots;\hat{\boldsymbol{y}}^{\mathrm{T}}_{i(t+M)}]$, $\forall i$.
```

IV. DRL-Based Optimization Algorithm

Given the predicted locations of all users, it is still challenging to solve the original problem owing to its non-linear and mixed-integer characteristics. Alternative optimization is extensively considered as an effective scheme of solving MINLP problems. Unfortunately, the popular alternative optimization scheme cannot be adopted in this paper. This is because the alternative optimization scheme is of often high computational complexity, and the original problem is also a sequential decision problem requiring an MINLP problem to be solved at each time slot. Remarkably, calling an optimization scheme with a high computational complexity at each time slot is unacceptable for latency-sensitive VR applications.

Reinforcement learning (RL) methods can be explored to solve sequential decision problems. For example, the works in [46] and [47] proposed (RL) methods to solve sequential decision problems with a discrete decision space and a continuous decision space, respectively. However, how to solve sequential decision problems simultaneously involving discrete and continuous decision variables (e.g., the problem (14)) is a significant and understudied problem.

In this paper, we propose a DRL-based optimization algorithm to solve (14). Specifically, we design a DNN joint with an action quantization scheme to produce a set of association actions of high diversity. Given the association actions, a continuous optimization problem is solved to criticize them and optimize the continuous variables. The detailed procedure is presented in the following subsections.

A. Vertical Decomposition

Define a vector $\boldsymbol{g}_{it}=[\boldsymbol{g}_{i1t};\ldots;\boldsymbol{g}_{ijt};\ldots;\boldsymbol{g}_{iJt}]\in\mathbb{C}^{JK}$ and a vector $\boldsymbol{h}_{it}=[f_{i1t}\boldsymbol{h}_{i1t};\ldots;f_{ijt}\boldsymbol{h}_{ijt};\ \ldots;f_{iJt}\boldsymbol{h}_{iJt}]\in\mathbb{C}^{JK}$, $\forall i,\ t$. Let matrix $\boldsymbol{G}_{it}=\boldsymbol{g}_{it}\boldsymbol{g}_{it}^{\mathrm{T}}$ and matrix $\boldsymbol{H}_{it}=\boldsymbol{h}_{it}\boldsymbol{h}_{it}^{\mathrm{T}}$. As $\mathrm{tr}(\boldsymbol{AB})=\mathrm{tr}(\boldsymbol{BA})$ for matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of compatible dimensions, the signal power received by user $i\in\mathcal{U}$ can be expressed as $|\sum_{j\in\mathcal{J}}f_{it}\boldsymbol{h}_{it}^{\mathrm{T}}\boldsymbol{g}_{ijt}|^2=|\boldsymbol{h}_{it}^{\mathrm{T}}\boldsymbol{g}_{it}|^2=\left(\boldsymbol{h}_{it}^{\mathrm{T}}\boldsymbol{g}_{it}\right)^{\mathrm{T}}\boldsymbol{h}_{it}^{\mathrm{T}}\boldsymbol{g}_{it}=\mathrm{tr}(\boldsymbol{g}_{it}^{\mathrm{T}}\boldsymbol{h}_{it}\boldsymbol{h}_{it}^{\mathrm{T}}\boldsymbol{g}_{it})=\mathrm{tr}(\boldsymbol{h}_{it}\boldsymbol{h}_{it}^{\mathrm{T}}\boldsymbol{g}_{it}\boldsymbol{g}_{it}^{\mathrm{T}})=\mathrm{tr}(\boldsymbol{H}_{it}\boldsymbol{G}_{it})$. Likewise, by introducing a square matrix $\boldsymbol{Z}_j\in\mathbb{R}^{JK\times JK}$ with $J\times J$ blocks, the transmit power for serving users can be written as $\boldsymbol{g}_{ijt}^{\mathrm{T}}\boldsymbol{g}_{ijt}=\mathrm{tr}(\boldsymbol{Z}_j\boldsymbol{G}_{it})$. Besides, each block in $\boldsymbol{Z}_j$ is a $K\times K$ matrix. In $\boldsymbol{Z}_j$, the block in the j-th row and j-th column is a $K\times K$ identity matrix, and all other blocks are zero matrices. Then, by applying $\boldsymbol{G}_{it}=\boldsymbol{g}_{it}\boldsymbol{g}_{it}^{\mathrm{T}}\Leftrightarrow\boldsymbol{G}_{it}\succeq 0$ and $\mathrm{rank}(\boldsymbol{G}_{it})\leq 1$, we can convert (14) to the following problem

$$\underset{\{\boldsymbol{a}_t^{\text{ul}},\boldsymbol{a}_t^{\text{dl}},\boldsymbol{p}_t,\boldsymbol{G}_{it}\}}{\text{maximize}} \quad \liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} F_t(\boldsymbol{a}_t^{\text{ul}},\boldsymbol{a}_t^{\text{dl}},\boldsymbol{p}_t^{\text{tot}}) \tag{33a}$$

$$\text{s.t. } \log_2\left(1+\frac{a_{it}^{\text{dl}}\text{tr}(\boldsymbol{H}_{it}\boldsymbol{G}_{it})}{N_0W^{\text{dl}}+\sum_{m\in\mathcal{M}_{it}}a_{mt}^{\text{dl}}\text{tr}(\boldsymbol{H}_{mt}\boldsymbol{G}_{mt})}\right) \tag{33b}$$

$$\geq \gamma^{\text{th}}/W^{\text{dl}}, \quad \forall i,t$$

$$\sum\nolimits_{i\in\mathcal{U}} a_{it}^{\text{dl}}\text{tr}(\boldsymbol{Z}_j\boldsymbol{G}_{it})+\tilde{E}_j\leq E_j, \quad \forall j,t \tag{33c}$$

$$\boldsymbol{G}_{it}\succeq 0, \quad \forall i,t \tag{33d}$$

$$\text{rank}(\boldsymbol{G}_{it})\leq 1, \quad \forall i,t \tag{33e}$$

$$(1),\ (14\text{b})\text{-}(14\text{f}). \tag{33f}$$

Like (14), (33) is difficult to be directly solved; thus, we first vertically decompose it into the following two subproblems.

- Uplink optimization subproblem: The uplink optimization subproblem is formulated as

$$\underset{\{\boldsymbol{a}_t^{\text{ul}},\boldsymbol{p}_t\}}{\text{maximize}} \liminf_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}(B_t^{\text{ul}}\left(\boldsymbol{a}_t^{\text{ul}}\right)-\sum_{i\in\mathcal{U}}\sum_{j\in\mathcal{J}}\frac{a_{ijt}^{\text{ul}}p_{it}^{\text{tot}}}{\tilde{p}_i}) \tag{34a}$$

$$\text{s.t.} \qquad (1), (14\text{b})\text{-}(14\text{d}), (14\text{f}). \tag{34b}$$

- Downlink optimization subproblem: The downlink optimization subproblem can be formulated as follows

$$\underset{\{\boldsymbol{a}_t^{\text{dl}},\boldsymbol{G}_{it}\}}{\text{maximize}} \quad \liminf_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}B_t^{\text{dl}}\left(\boldsymbol{a}_t^{\text{dl}}\right) \tag{35a}$$

$$\text{s.t.} \qquad (14\text{e}), (33\text{b})\text{-}(33\text{e}). \tag{35b}$$

Next, we propose to solve the two subproblems separately by exploring DRL approaches.

B. Solution to the Uplink Optimization Subproblem

(34) is confirmed to be a mixed-integer and sequence-dependent optimization subproblem. Fig. 5 shows a DRL approach of solving (34). In this figure, a DNN is trained to produce continuous actions. The continuous actions are then quantized into a group of association (or discrete) actions. Given the association actions, we solve an optimization problem to select an association action maximizing the reward. Next, we describe the designing process of solving (34) using a DRL-based uplink optimization method in detail.

1) Action, State, and Reward Design: First, we elaborate on the design of the state space, action space, and reward function of the DRL-based method. The HMDs' transmit power and the varying channel gains caused by users' movement and/or time-varying wireless channel environments have a significant impact on whether uplink transmission signals can be successfully decoded by APs. In addition, each AP has a limited ability to decode uplink transmission signals simultaneously. Therefore, we design the state space, action space, and reward function of the DRL-based method as the following.

- **state space $\boldsymbol{s}_t^{\text{ul}}$:** $\boldsymbol{s}_t^{\text{ul}}=[\boldsymbol{m}_t;\hat{\boldsymbol{h}}_t^{\text{ul}};\boldsymbol{p}_t]$ is a column vector, where $m_{jt}\in\boldsymbol{m}_t\in\mathbb{R}^J$, $\forall j$, denotes the number of users successfully accessing to AP j at time slot t. Besides, the state space involves the path-loss from user i to AP j, $\hat{h}_{ijt}\in\hat{\boldsymbol{h}}_t^{\text{ul}}\in\mathbb{R}^{NJ}$, $\forall i, j, t$, and the transmit power of user i's HMD at time slot t, $p_{it}\in\boldsymbol{p}_t\in\mathbb{R}^N$, $\forall i, t$.

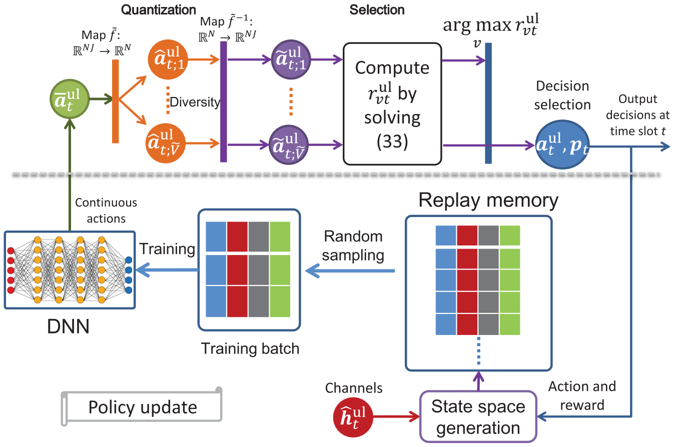

Quantization
Map $\tilde{f}$: $\mathbb{R}^{NJ}\to\mathbb{R}^N$
Map $\tilde{f}^{-1}$: $\mathbb{R}^N\to\mathbb{R}^{NJ}$
Selection
$\arg\max_v r_{vt}^{\text{ul}}$
$\bar{\boldsymbol{a}}_t^{\text{ul}}$
$\hat{\boldsymbol{a}}_{t;1}^{\text{ul}}$
Diversity
$\hat{\boldsymbol{a}}_{t;\tilde{V}}^{\text{ul}}$
$\tilde{\boldsymbol{a}}_{t;1}^{\text{ul}}$
$\tilde{\boldsymbol{a}}_{t;\tilde{V}}^{\text{ul}}$
Compute r_{vt}^{ul} by solving (33)
Decision selection
Output decisions at time slot t
$\boldsymbol{a}_t^{\text{ul}},\boldsymbol{p}_t$
Continuous actions
Training
DNN
Training batch
Random sampling
Replay memory
Policy update
Channels
$\hat{\boldsymbol{h}}_t^{\text{ul}}$
State space generation
Action and reward

Fig. 5. A DRL approach of association and transmit power optimization.

- **action space $\boldsymbol{a}_t^{\text{ul}}$:** $\boldsymbol{a}_t^{\text{ul}}=[a_{11t}^{\text{ul}},\ldots,a_{1Jt}^{\text{ul}},\ldots,a_{NJt}^{\text{ul}}]^{\text{T}}\in\mathbb{R}^{NJ}$ with $a_{ijt}^{\text{ul}}\in\{0,1\}$. The action of the DRL-based method is to deliver users' data information to their associated APs.
- **reward r_t^{ul}:** given $\boldsymbol{a}_t^{\text{ul}}$, the reward r_t^{ul} is the objective function value of the following power control subproblem.

$$r_t^{\text{ul}}=\underset{\boldsymbol{p}_t}{\text{maximize}}\ B_t^{\text{ul}}(\boldsymbol{a}_t^{\text{ul}})-\sum\nolimits_{i\in\mathcal{U}}\sum\nolimits_{j\in\mathcal{J}}a_{ijt}^{\text{ul}}p_{it}^{\text{tot}}/\tilde{p}_i \tag{36a}$$

$$\text{s.t.} \quad (1), \quad (14\text{f}). \tag{36b}$$

2) Training Process of the DNN: For the DNN module $\bar{\boldsymbol{a}}_t^{\text{ul}}=\mu(\boldsymbol{s}_t^{\text{ul}}|\theta_t^{\mu})$ shown in Fig. 5, where $\bar{\boldsymbol{a}}_t^{\text{ul}}=[\bar{\boldsymbol{a}}_{1t}^{\text{ul}};\ldots;\bar{\boldsymbol{a}}_{Nt}^{\text{ul}}]$ and θ_t^{μ} represents network parameters, we explore a two-layer fully-connected feedforward neural network with network parameters being initialized by a Xavier initialization scheme. There are N_1^{μ} and N_2^{μ} neurons in the 1st and 2nd hidden layers of the constructed DNN, respectively. Here, we adopt the ReLU function as the activation function in these hidden layers. For the output layer, a sigmoid activation function is leveraged such that relaxed association variables satisfy $\bar{a}_{ijt}^{\text{ul}}\in(0,1)$. In the action-exploration phase, the exploration noise ϵN_f is added to the output layer of the DNN, where $\epsilon\in(0,1)$ decays over time and $N_f\sim\mathcal{N}(0,\sigma^2)$.

To train the DNN effectively, the experience replay technique is exploited. This is because there are two special characteristics in the process of enhancing users' fictitious experiences: 1) the collected input state values $\boldsymbol{s}_t^{\text{ul}}$ incrementally arrive as users move to new positions, instead of all made available at the beginning of the training; 2) APs consecutively collect state values indicating that the collected state values may be closely correlated. The DNN may oscillate or diverge without breaking the correlation among the input state values. Specifically, at each training epoch t, a new training sample $(\boldsymbol{s}_t^{\text{ul}},\boldsymbol{a}_t^{\text{ul}},\boldsymbol{s}_{t+1}^{\text{ul}})$ is added to the replay memory. When the memory is filled, the newly generated sample replaces the oldest one. We randomly choose a minibatch of training samples $\{(\boldsymbol{s}_\tau^{\rm ul}, \boldsymbol{a}_\tau^{\rm ul}, \boldsymbol{s}_{\tau+1}^{\rm ul}) | \tau \in \mathcal{T}_t\}$ from the replay memory, where $\mathcal{T}_t$ is a set of training epoch indices. The network parameters θ_t^μ are trained using the ADAM method [48] to reduce the averaged cross-entropy loss

$$L(\theta_t^\mu) = -\frac{1}{|\mathcal{T}_t|}\sum\nolimits_{\tau \in \mathcal{T}_t}((\boldsymbol{a}_\tau^{\rm ul})^{\rm T} \log \bar{\boldsymbol{a}}_\tau^{\rm ul} + (1-\boldsymbol{a}_\tau^{\rm ul})^{\rm T}\log(1-\bar{\boldsymbol{a}}_\tau^{\rm ul})). \quad (37)$$

As evaluated in the simulation, we can train the DNN every T_{ti} epochs after collecting a sufficient number of new data samples.

3) Action Quantization and Selection Method: In the previous subsection, we design a continuous policy function and generate a continuous action space. However, a discrete action space is required in this paper. To this aim, the generated continuous action should be quantized, as shown in Fig. 5. A quantized action will directly determine the feasibility of the optimization subproblem and then the convergence performance of the DRL-based optimization method. To improve the convergence performance, we should increase the diversity of the quantized action set, which includes all quantized actions. Specifically, we quantize the continuous action $\bar{\boldsymbol{a}}_t^{\rm ul}$ to obtain $\tilde{V} \in \{1, 2, \ldots, 2^N\}$ groups of association actions and denote by $\bar{\boldsymbol{a}}_{t;v}^{\rm ul}$ the v-th group of actions. Given $\bar{\boldsymbol{a}}_{it;v}^{\rm ul}$, (36) is reduced to a linear programming problem, and we can derive its closed-form solution as below

$$p_{it} = \begin{cases} \sum\limits_j \frac{a_{ijt}^{\rm ul}\theta^{\rm th}N_0W^{\rm ul}}{Nf_i\hat{h}_{ijt}}, & \sum\limits_j \frac{a_{ijt}^{\rm ul}\theta^{\rm th}N_0W^{\rm ul}}{Nf_i\hat{h}_{ijt}} \le \tilde{p}_i - p_i^c, \\ 0, & \text{otherwise.} \end{cases} \quad (38)$$

Besides, a great $\tilde{V}$ will result in higher diversity in the quantized action set but a higher computational complexity, and vice versa. To balance the performance and complexity, we set $\tilde{V} = N$ and propose a lightweight action quantization and selection method. The detailed steps of quantizing and selecting association actions are given in Algorithm 2.

Algorithm 2 Action Quantization and Selection

```
Input: The output action of the uplink DNN, $\bar{\boldsymbol{a}}_t^{\rm ul}$.
Arrange $\bar{\boldsymbol{a}}_t^{\rm ul}$ as a matrix of size $N \times J$ and generate a vector $\hat{\boldsymbol{a}}_t^{\rm ul} = \{\max[\bar{a}_{i1t}^{\rm ul}, \ldots, \bar{a}_{iJt}^{\rm ul}], \forall i\}$.
Generate the reference action vector $\bar{\boldsymbol{b}}_t = [\bar{b}_{1t}, \ldots, \bar{b}_{vt}, \ldots, \bar{b}_{\tilde{V}t}]^{\rm T}$ by sorting the absolute value of all elements of $\hat{\boldsymbol{a}}_t^{\rm ul}$ in ascending order.
For any user $i$, generate the 1st group of association actions by
    $\hat{a}_{it;1}^{\rm ul} = \begin{cases} 1, \ \hat{a}_{it}^{\rm ul} > 0.5, \\ 0, \ \hat{a}_{it}^{\rm ul} \le 0.5. \end{cases}$   (39)
For any user $i$, generate the remaining $\tilde{V} - 1$ groups of association actions by
    $\hat{a}_{it;v}^{\rm ul} = \begin{cases} 1, \ \hat{a}_{it}^{\rm ul} > \bar{b}_{(v-1)t}, \ v = 2, \ldots, \tilde{V}, \\ 0, \ \hat{a}_{it}^{\rm ul} \le \bar{b}_{(v-1)t}, \ v = 2, \ldots, \tilde{V}. \end{cases}$   (40)
For each group of association actions $v \in \{1, 2, \ldots, \tilde{V}\}$, user $i$, and AP $j$, set
    $\tilde{a}_{ijt;v}^{\rm ul} = \begin{cases} 1, \ \hat{a}_{it;v}^{\rm ul} = 1, j = j^\star, \\ 0, \quad \text{otherwise}. \end{cases}$   (41)
    where, $j^\star = \arg\max\limits_j[\bar{a}_{i1t}^{\rm ul}, \ldots, \bar{a}_{iJt}^{\rm ul}]$.
For each group of association actions $v \in \{1, 2, \ldots, \tilde{V}\}$, given the vector $\tilde{\boldsymbol{a}}_{t;v}^{\rm ul} = [\tilde{a}_{i1t;v}^{\rm ul}, \ldots, \tilde{a}_{iJt;v}^{\rm ul}]_i^{\rm T}, \forall i$, solve (36) to obtain $r_{vt}^{\rm ul}$.
Select the association action $\boldsymbol{a}_t^{\rm ul} = \arg\max_{\{\tilde{a}_{ijt;v}^{\rm ul}\}} r_{vt}^{\rm ul}$.
Output: The association action $\boldsymbol{a}_t^{\rm ul}$.
```

Summarily, the proposed DRL-based uplink optimization method can be presented in Algorithm 3.

C. Solution to the Downlink Optimization Subproblem

Like (34), (35) is also a mixed-integer and sequence-dependent optimization problem. Therefore, the procedure of solving (35) is similar to that of solving (34), and we do not present the detailed steps of the DRL-based downlink optimization method in this subsection for brevity. However, there are differences in some aspects, for example, the design of action and state space and the reward function. For the DRL-based downlink optimization method, we design its action space, state space, and the reward function as the following.

- **state space $\boldsymbol{s}_t^{\rm dl}$:** $\boldsymbol{s}_t^{\rm dl} = [\boldsymbol{o}_t; \boldsymbol{I}_t^{\rm dl}; \boldsymbol{h}_t; \boldsymbol{g}_t]$ is a column vector, where $o_{jt} \in \boldsymbol{o}_t \in \mathbb{R}^J$ indicates the number of users to which AP j transmits VR video tiles, $I_{imt} \in \mathbb{R}^{N \times N} \in \boldsymbol{I}_t^{\rm dl}$ denotes whether user m is the interfering user of user i, $h_{ijkt} \in \boldsymbol{h}_t \in \mathbb{C}^{NJK}$, and $g_{ijkt} \in \boldsymbol{g}_t \in \mathbb{C}^{NJK}$.
- **action space $\boldsymbol{a}_t^{\rm dl}$:** $\boldsymbol{a}_t^{\rm dl} = [a_{1t}^{\rm dl}, \ldots, a_{it}^{\rm dl}, \ldots, a_{Nt}^{\rm dl}]^{\rm T}$ with $a_{it}^{\rm dl} \in \{0, 1\}$. The action of the DRL-based method at time slot t is to transmit VR video tiles to corresponding users.
- **reward $r_t^{\rm dl}$:** given $\boldsymbol{a}_t^{\rm dl}$, the reward $r_t^{\rm dl}$ is the objective function value of the following power control subproblem.

$$r_t^{\rm dl} = \underset{\boldsymbol{G}_{it}}{\text{maximize}} \ B_t^{\rm dl}\left(\boldsymbol{a}_t^{\rm dl}\right) \quad (42a)$$

$$\text{s.t.} \quad \text{(33b)-(33e)}. \quad (42b)$$

To solve (42), Algorithm 2 can be adopted to obtain the downlink association action $\boldsymbol{a}_t^{\rm dl}$. However, given $\boldsymbol{a}_t^{\rm dl}$, it is still hard to solve (42) as (42) is a non-convex programming problem with the existence of the non-convex low-rank constraint (33e). To handle the non-convexity, a semidefinite relaxation (SDR) scheme is exploited. The idea of the SDR scheme is to directly drop out the non-convex low-rank constraint. After dropping the constraint (33e), it can confirm that (42) becomes a standard convex semidefinite programming (SDP) problem. This is because (33b) and (33c) are linear constraints w.r.t $\boldsymbol{G}_{it}$ and (42a) is a constant objective function. We can then explore some optimization tools such as MOSEK to solve the standard convex SDP problem effectively. However, owing to the relaxation, power matrices $\{\boldsymbol{G}_{it}\}$ obtained by mitigating (42) without low-rank constraints will not satisfy

Algorithm 3 DRL-Based Uplink Optimization

```
1: Initialize: The maximum number of episodes $N_{epi}$, the maximum number of epochs per episode $N_{epo}$, initial exploration decaying rate $\epsilon$, DNN $\mu(\boldsymbol{s}_t^{\mathrm{ul}}|\theta_t^{\mu})$ with network parameters $\theta_t^{\mu}$, initial reward $r_0^{\mathrm{ul}} = 1$, and users' randomly initialized transmit power.
2: Initialize: Replay memory with capacity $C$, minibatch size $|\mathcal{T}_t|$, and DNN training interval $T_{\mathrm{ti}}$.
3: for each episode in $\{1, \ldots, N_{epi}\}$ do
4:   Calculate the state space according to locations of APs and users and users' randomly initialized transmit power.
5:   for each epoch $\bar{t} = 1, \ldots, N_{epo}$ do
6:     Select a relaxed action vector $\bar{\boldsymbol{a}}_{\bar{t}}^{\mathrm{ul}} = \mu(\boldsymbol{s}_{\bar{t}}^{\mathrm{ul}}|\theta_{\bar{t}}^{\mu}) + \epsilon N_f$, where $\epsilon$ decays over time.
7:     Call Algorithm 2 to choose the association action $\boldsymbol{a}_{\bar{t}}^{\mathrm{ul}}$.
8:     if $\boldsymbol{a}_{\bar{t}}^{\mathrm{ul}}$ results in the violation of constraints in (34) then
9:       Cancel the action and update the reward by $r_{\bar{t}}^{\mathrm{ul}} = r_{\bar{t}}^{\mathrm{ul}} - \varpi|r_{\bar{t}-1}^{\mathrm{ul}}|$.
10:    else
11:      Execute the action and observe the subsequent state $\boldsymbol{s}_{\bar{t}+1}^{\mathrm{ul}}$.
12:    end if
13:    Store the transition $(\boldsymbol{s}_{\bar{t}}^{\mathrm{ul}}, \boldsymbol{a}_{\bar{t}}^{\mathrm{ul}}, \boldsymbol{s}_{\bar{t}+1}^{\mathrm{ul}})$ in the memory.
14:    If $\bar{t} \geq |\mathcal{T}_t|$, sample a random minibatch of $|\mathcal{T}_t|$ transitions $(\boldsymbol{s}_m^{\mathrm{ul}}, \boldsymbol{a}_m^{\mathrm{ul}}, \boldsymbol{s}_{m+1}^{\mathrm{ul}})$ from the memory.
15:    If $\bar{t} \mod T_{\mathrm{ti}} == 0$, update the network parameters $\theta_{\bar{t}}^{\mu}$ by minimizing the loss function $L(\theta_{\bar{t}}^{\mu})$ using the ADAM method.
16:  end for
17: end for
```

the low-rank constraint in general. This is due to the fact that the (convex) feasible set of the relaxed (42) is a superset of the (non-convex) feasible set of (42). The following lemma reveals the tightness of exploring the SDR scheme.

Lemma 2: For any user i at time slot t, denote by $\boldsymbol{G}_{it}^{\star}$ the solution to (42). If $\mathcal{M}_{it} = \emptyset$, then the SDR for $\boldsymbol{G}_{it}$ in (42) is tight, that is, $\mathrm{rank}(\boldsymbol{G}_{it}^{\star}) \leq 1$; otherwise, we can not claim $\mathrm{rank}(\boldsymbol{G}_{it}^{\star}) \leq 1$.

Proof: The Karush-Kuhn-Tucker (KKT) conditions can be explored to prove the tightness of resorting to the SDR scheme. Nevertheless, we omit the detailed proof for brevity as a similar proof can be found in Appendix of the work [49]. ■

With the conclusion in Lemma 2, we can recover beamformers from the obtained power matrices. If $\mathrm{rank}(\boldsymbol{G}_{it}^{\star}) \leq 1$, $\forall i$, then execute eigenvalue decomposition on $\boldsymbol{G}_{it}^{\star}$, and the principal component is the optimal beamformer $\boldsymbol{g}_{it}^{\star}$; otherwise, some manipulations such as a randomization/scale scheme [50] should be performed on $\boldsymbol{G}_{it}^{\star}$ to impose the low-rank constraint.

Note that (42) should be solved for $\tilde{V}$ times at each time slot. To speed up the computation, they can be optimized in parallel. Moreover, it is tolerable to complete the computation within the interval $[t, t+M]$ as users' locations in M time slots are obtained.

Finally, we can summarize the DRL-based optimization algorithm of mitigating the problem of enhancing users' VR experiences in Algorithm 4.

Algorithm 4 DRL-Based Optimization Algorithm

```
1: Initialization: Run initialization steps of Algorithms 1, 2, and 3, and initialize the ESN training interval $T_{\mathrm{pr}}$.
2: Call Algorithm 3 to pre-train the uplink DNN $\mu(\boldsymbol{s}_t^{\mathrm{ul}}|\theta_t^{\mu})$. Likewise, pre-train the downlink DNN $\mu(\boldsymbol{s}_t^{\mathrm{dl}}|\theta_t^{Q})$.
3: Run steps 2-8 of Algorithm 1 to pre-train ESN models.
4: for each time slot $t = 1, 2, \ldots, T$ do
5:   Run step 9 of Algorithm 1 to obtain predicted location $\hat{\boldsymbol{y}}_{i(t+M)}$ of each user $i$.
6:   Run steps 6-12 of Algorithm 3 to obtain uplink association action $\boldsymbol{a}_{t+M}^{\mathrm{ul}}$ and transmit power $\boldsymbol{p}_{t+M}$. Likewise, optimize the downlink association action $\boldsymbol{a}_{t+M}^{\mathrm{dl}}$ and transmit beamformer $\boldsymbol{g}_{i(t+M)}$ for each user $i$.
7:   if $t \mod T_{\mathrm{pr}} == 0$ then
8:     Steps 2-8 of Algorithm 1.
9:   end if
10: end for
```

V. Simulation and Performance Evaluation

A. Comparison Algorithms and Parameter Setting

To verify the effectiveness of the proposed algorithm, we compare it with four benchmark algorithms: 1) k-nearest neighbors (KNN) based action quantization algorithm: The unique difference between the KNN-based algorithm and the proposed algorithm lies in the scheme of quantizing uplink and downlink action spaces. For the KNN-based algorithm, it adopts the KNN method [51] to quantize both uplink and downlink action spaces; 2) DROO algorithm: Different from the proposed algorithm, DROO leverages the order-preserving quantization method [51] to quantize both uplink and downlink action spaces; 3) Heuristic algorithm: The heuristic algorithm leverages the greedy admission algorithm in [52] to determine $\boldsymbol{a}_t^{\mathrm{ul}}$ and $\boldsymbol{a}_t^{\mathrm{dl}}$ at each time slot t. Besides, the user consuming less power in this algorithm will establish the connection with an AP(s) on priority; 4) ESN-RL algorithm [53]: It differs from the proposed algorithm in two aspects. First, it explores a centralized ESN method to perform mobility prediction for each user. Second, it leverages a RL method to generate uplink and downlink actions without performing an action quantization and selection scheme.

To test the practicality of the developed parallel ESN learning method, realistic user movement datasets are generated via Google Map. Particularly, for a user, we randomly select its starting position and ending position on the campus of Singapore University of Technology and Design (SUTD). Given two endpoints, we use Google Map to generate the user's 2D trajectory. Next, we linearly zoom all N users' trajectories into the communication area of size 0.5×0.5 km^2.

Additionally, the parameters related to APs and downlink transmission channels are listed as follows: the number of APs $J = 3$, the number of antenna elements $K = 2$, the antenna gain $G = 5$ dB, $g = 1$ dB, $\phi = \pi/3$, $\vartheta = \pi/2$, $W^{\rm dl} = 800$ MHz, $\gamma^{\rm th} = 1$ Gb/s, $\eta_{\rm LoS} = 2.0$, $\eta_{\rm NLoS} = 2.4$, $\sigma^2_{\rm LoS} = 5.3$, $\sigma^2_{\rm NLoS} = 5.27$, $D^{\rm th} = 50$ m, $x_o = y_o = 250$ m, $\theta_j = \pi/3$, $\tilde{E}_j = 40$ dBm, $E^c_j = 30$ dBm, $H_j = 5.5$ m, $\forall j$ [29]. User and uplink transmission channel-related parameters are shown as below: uplink system bandwidth $W^{\rm ul} = 200$ MHz, $\theta^{\rm th} = 200$, $\bar{h} = 1.8$ m, $\sigma^2_h = 0.05$ m, $\alpha = 5$, $c_{ij} = 0.3$, $p^c_i = 23$ dBm, $\tilde{p}_i = 27$ dBm, $\forall i, j$.

Set other learning-correlated parameters as below: $\zeta = 1$, $\xi = 0.25$, $\bar{r}_{\max} = 1000$, the sample number $Q = 6$, the number of future time slots $M = 8$, $N_i = 2, \forall i$, $N_o = 2$, $N_r = 300$, and $T_{\rm pr} = 5$. For both uplink DNN and downlink DNN, the first hidden layer has 120 neurons, and the second hidden layer has 80 neurons. The replay memory capacity $C = 1e{+}6$, $N_{epi} = 10$, $N_{epo} = 1000$, $\varpi = 10$, $\sigma^2 = 0.36$, $\epsilon = 0.99$. More system parameters are listed as follows: carrier frequency $f_c = 28$ GHz, light of speed $c = 3.0e{+}8$ m/s, noise power spectral density $N_0 = -167$ dBm/Hz, and $T = 5000$ time slots.

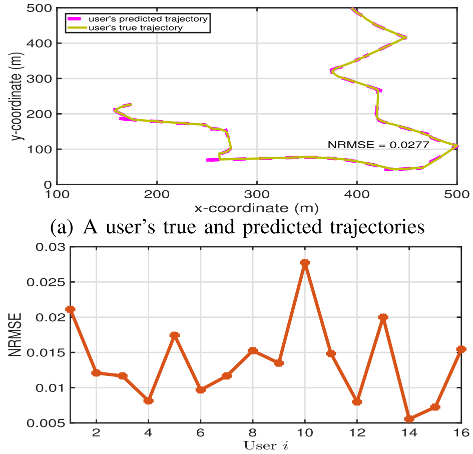
B. Performance Evaluation

To comprehensively understand the accuracy and the availability of the developed learning and optimization methods, we illustrate their performance results. In this simulation, we first let the AP number $J = 3$ and the mobile user number $N = 16$.

To validate the accuracy of the parallel ESN learning method on predicting mobile users' locations, we plot the actual trajectory of a randomly selected mobile user and its correspondingly predicted trajectory in Fig. 6(a). In Fig. 6(b), the accuracy, which is measured by the normalized root mean-squared error (NRMSE) [43], of predicted trajectories of 16 mobile users is plotted. From Fig. 6, we can observe that: i) when the orientation angles of users will not change fast, the learning method can exactly predict users' locations. When users change their moving directions quickly, the method loses their true trajectories. However, the method will re-capture users' tracks after training ESN models based on newly collected users' location samples; ii) the obtained NRMSE of the predicted trajectories of all mobile users will not be greater than 0.03. Therefore, we may conclude that the developed parallel ESN learning method can be utilized to predict mobile users' locations.

Next, to evaluate the performance of the proposed DRL-based optimization algorithm comprehensively, we illustrate the impact of some DRL-related crucial parameters such as minibatch size, training interval, and learning rate on the convergence performance of the proposed algorithm. DNN training loss and moving average reward, which is the average of the achieved rewards over the last 50 epochs, are leveraged as the evaluation indicators.

Fig. 7 plots the tendency of the DNN training loss and the achieved moving average reward of the proposed algorithm under diverse minibatch sizes. This figure illustrates that: i) a great minibatch size value will cause the DNN to converge slowly or even not. As shown in Fig. 7(a), $L(\theta^{\mu}_{465}) = 0.1885$ when we set $|\mathcal{T}_t| = 512$. Yet, $L(\theta^{\mu}_{465}) = 0.1023$ when we let $|\mathcal{T}_t| = 64$. The result in Fig. 7(b) shows that DNN does not converge after 10000 epochs when $|\mathcal{T}_t| = 2048$. This is because a great $|\mathcal{T}_t|$ indicates overtraining, resulting in the local minima and degraded convergence performance. Further, a large minibatch size value consumes more training time at each training epoch. Therefore, we set the training minibatch size $|\mathcal{T}_t| = 64$ in the simulation; ii) when $|\mathcal{T}_t| = 64$, $r^{\rm ul}_{\bar{t}}$ and $r^{\rm dl}_{\bar{t}}$ gradually increase and stabilize at around 0.7141 and 0.9375, respectively. The fluctuation is mainly caused by the random sampling of training data and user movement.



(a) A user's true and predicted trajectories

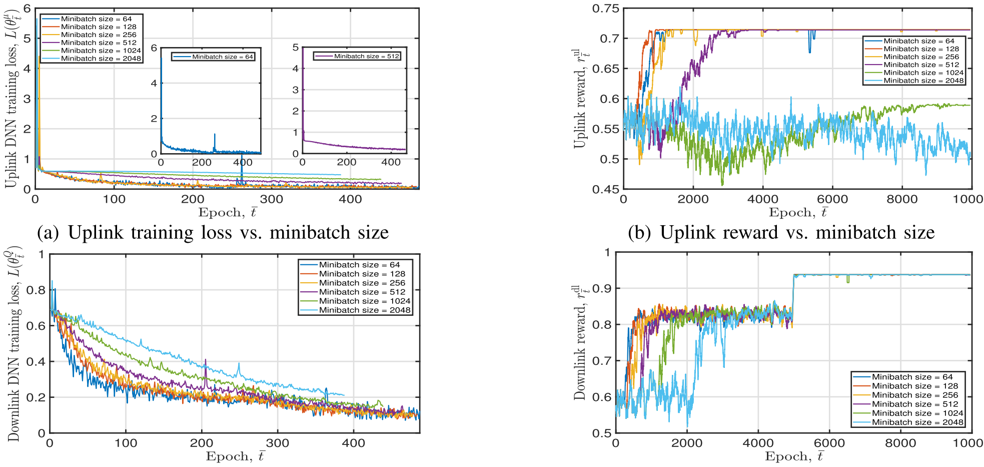
(b) NRMSE of predicted trajectories of N users

Fig. 6. Prediction accuracy of the parallel ESN learning method.

Fig. 8 illustrates the tendency of obtained uplink and downlink DNN training losses and moving average rewards under diverse training interval values. From this figure, we can observe that a small training interval value indicates faster convergence speed. For example, if we set the training interval $T_{\rm ti} = 5$, the obtained $r^{\rm ul}_{\bar{t}}$ converges to 0.7156 when epoch $\bar{t} > 439$. If we let the training interval $T_{\rm ti} = 100$, $r^{\rm ul}_{\bar{t}}$ converges to 0.7149 when epoch $\bar{t} > 4975$, as shown in Fig. 8(b). However, it is unnecessary to train and update the DNN frequently, which will bring more frequent policy updates, if the DNN can converge. Thus, to achieve the trade-off between the convergence speed and the policy update speed, we set $T_{\rm ti} = 20$ in the simulation.

Fig. 9 depicts the tendency of achieved DNN training loss and moving average reward of the proposed algorithm under different learning rate configurations. From this figure, we have the following observations: i) for the uplink DNN, when given a small learning rate value, it may converge to the local optimum or even not; ii) for the downlink DNN, both a small and a great learning rate value will degrade convergence performance. Therefore, when training the uplink DNN, we set the learning rate $l^{\rm ul}_r = 0.1$, which can lead to good convergence performance. For instance, $r^{\rm ul}_{\bar{t}}$ converges to 0.7141 when epoch $\bar{t} \geq 1300$ and the variance of $r^{\rm ul}_{\bar{t}}$ gradually decreases to zero with an increasing epoch $\bar{t}$. We set the learning rate $l^{\rm dl}_r = 0.01$ when training the downlink DNN. Given this parameter setting, the obtained $L(\theta^{Q}_{\bar{t}})$ is smaller than 0.2 after training for 200 epochs.

At last, we verify the superiority of the proposed algorithm by comparing it with other comparison algorithms. Particularly, we plot the achieved objective function values



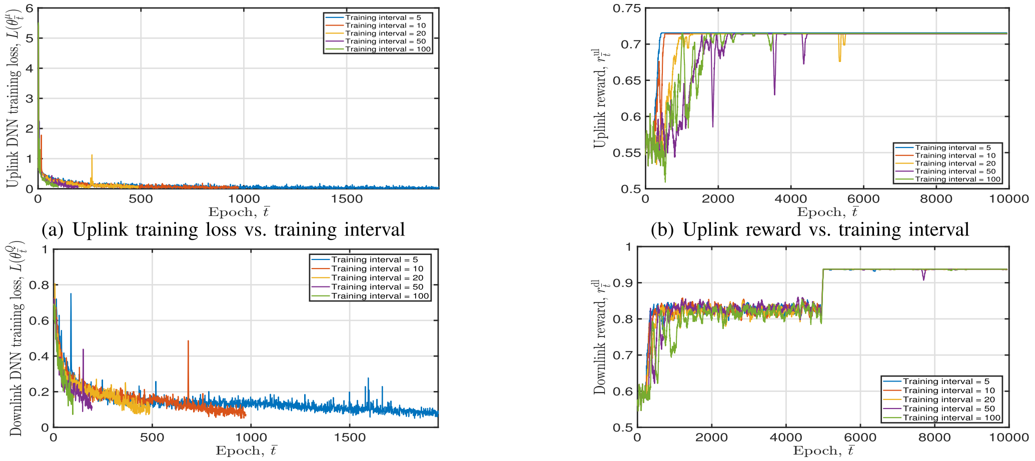
(a) Uplink training loss vs. minibatch size

(b) Uplink reward vs. minibatch size

(c) Downlink training loss vs. minibatch size

(d) Downlink reward vs. minibatch size

Fig. 7. The impact of minibatch size $|\mathcal{T}_t|$ on the convergence performance of the proposed algorithm.



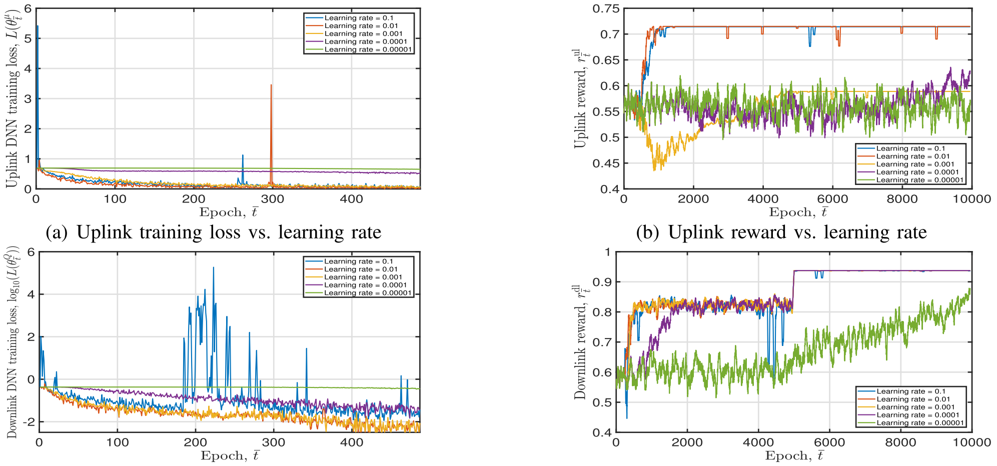
(a) Uplink training loss vs. training interval

(b) Uplink reward vs. training interval

(c) Downlink training loss vs. training interval

(d) Downlink reward vs. training interval

Fig. 8. The impact of DNN training interval T_{ti} on the convergence performance of the proposed algorithm.



(a) Uplink training loss vs. learning rate

(b) Uplink reward vs. learning rate

(c) Downlink training loss vs. learning rate

(d) Downlink reward vs. learning rate

Fig. 9. The impact of learning rates l_r^{ul} and l_r^{dl} on the convergence performance of the proposed algorithm.

of all comparison algorithms under varying number of mobile users $N \in \{8, 12, 16, 20\}$ in Fig. 10. Before the evaluation, the proposed algorithm and the other two action quantization algorithms have been trained with 10000 independent wireless channel realizations, and their downlink and uplink action quantization policies have converged. This is reasonable because we are more interested in the long-term operation performance for field deployment. Besides, we let the service ability of an AP $\tilde{M}$ vary with N with the $(N, \tilde{M})$ pair being $(8, 3)$, $(12, 5)$, $(16, 6)$, and $(20, 7)$.

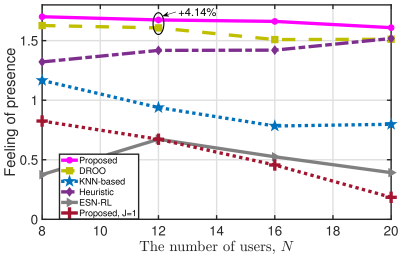


Fig. 10. Comparison of obtained FoP values of all algorithms.

We have the following observations from this figure: i) the proposed algorithm achieves the greatest FoP value. For the DROO algorithm, it gains a smaller FoP value than the proposed algorithm; for example, the achieved FoP value of DROO is 4.14% less than that of the proposed algorithm. For the KNN-based algorithm, it obtains a smaller FoP value than the proposed algorithm and DROO because it offers a smaller diversity in the produced uplink and downlink association action set; ii) except for heuristic and ESN-RL algorithms, the achieved FoP values of the other comparison algorithms decrease with the number of users owing to the increasing total power consumption. For the heuristic algorithm, its obtained FoP value increases with N mainly because more users can successfully access to APs. It's difficult to conclude the tendency of the obtained FoP value of ESN-RL algorithm because it does not explore an action quantization and selection scheme; iii) besides, when $J = 1$, the proposed algorithm achieves a small FoP value, which, in turn, verifies that CoMP can significantly enhance VR users' immersive experiences.

VI. Conclusion

This paper investigated the problem of enhancing VR visual experiences for mobile users and formulated the problem as a sequence-dependent problem aiming at maximizing users' feeling of presence in VR environments while minimizing the total power consumption of users' HMDs. This problem was confirmed to be a mixed-integer and non-convex optimization problem, the solution of which also needed accurate users' tracking information. To solve this problem effectively, we developed a parallel ESN learning method to predict users' tracking information, with which a DRL-based optimization algorithm was proposed. Specifically, this algorithm first decomposed the formulated problem into an association subproblem and a power control subproblem. Then, a DNN joint with an action quantization scheme was implemented as a scalable solution that learnt association variables from experience. Next, the power control subproblem with an SDR scheme being explored to tackle its non-convexity was leveraged to criticize the association variables. Finally, simulation results were provided to verify the accuracy of the learning method and showed that the proposed algorithm could improve the power efficiency by at least 4.14% compared with various benchmark algorithms.

References

[1] X. Hou, S. Dey, J. Zhang, and M. Budagavi, "Predictive adaptive streaming to enable mobile 360-degree and VR experiences," *IEEE Trans. Multimedia*, vol. 23, pp. 716–731, 2021.

[2] H. Bellini. (Feb. 2016). *The Real Deal With Virtual and Augmented Reality*. [Online]. Available: https://www.goldmansachs.com/insights/pages/virtual-and-augmented-reality.html

[3] C. Wiltz. (Apr. 2017). *5 Major Challenges for VR to Overcome*. [Online]. Available: https://www.designnews.com/electronics-test/5-major-challenges-vr-overcome

[4] Y. Liu, J. Liu, A. Argyriou, and S. Ci, "MEC-assisted panoramic VR video streaming over millimeter wave mobile networks," *IEEE Trans. Multimedia*, vol. 21, no. 5, pp. 1302–1316, May 2019.

[5] J. Dai, Z. Zhang, S. Mao, and D. Liu, "A view synthesis-based 360° VR caching system over MEC-enabled C-RAN," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 10, pp. 3843–3855, Oct. 2020.

[6] Z. Lai, Y. C. Hu, Y. Cui, L. Sun, N. Dai, and H.-S. Lee, "Furion: Engineering high-quality immersive virtual reality on today's mobile devices," *IEEE Trans. Mobile Comput.*, vol. 19, no. 7, pp. 1586–1602, Jul. 2020.

[7] X. Hou, Y. Lu, and S. Dey, "Wireless VR/AR with edge/cloud computing," in *Proc. ICCCN*, 2017, pp. 1–8.

[8] Qualcomm. *Making Immersive Virtual Reality Possible in Mobile*. Accessed: Oct. 2018. [Online]. Available: https://www.qualcomm.com/media/documents/files/making-immersive-virtual-reality-possible-in-mobile.pdf

[9] Oculus. (2018). *Mobile VR Media Overview*. Accessed: Sep. 2018. [Online]. Available: https://www.oculus.com/

[10] HTC. (2018). *HTC Vive*. Accessed: Sep. 2018. [Online]. Available: https://www.vive.com/us/

[11] T. Dang and M. Peng, "Joint radio communication, caching, and computing design for mobile virtual reality delivery in fog radio access networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 7, pp. 1594–1607, Jul. 2019.

[12] R. Ju *et al.*, "Ultra wide view based panoramic VR streaming," in *Proc. Workshop Virtual Reality Augmented Reality Netw.*, Aug. 2017, pp. 19–23.

[13] S. Mangiante, G. Klas, A. Navon, Z. GuanHua, J. Ran, and M. D. Silva, "VR is on the edge: How to deliver 360° videos in mobile networks," in *Proc. Workshop Virtual Reality Augmented Reality Netw.*, Aug. 2017, pp. 30–35.

[14] V. R. Gaddam, M. Riegler, R. Eg, C. Griwodz, and P. Halvorsen, "Tiling in interactive panoramic video: Approaches and evaluation," *IEEE Trans. Multimedia*, vol. 18, no. 9, pp. 1819–1831, Sep. 2016.

[15] L. Xie, Z. Xu, Y. Ban, X. Zhang, and Z. Guo, "360ProbDASH: Improving QoE of 360 video streaming using tile-based http adaptive streaming," in *Proc. ACM Multimedia*, 2017, pp. 315–323.

[16] J. Zou, C. Li, C. Liu, Q. Yang, H. Xiong, and E. Steinbach, "Probabilistic tile visibility-based server-side rate adaptation for adaptive 360-degree video streaming," *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 1, pp. 161–176, Jan. 2020.

[17] C. Guo, Y. Cui, and Z. Liu, "Optimal multicast of tiled 360 VR video in OFDMA systems," *IEEE Commun. Lett.*, vol. 22, no. 12, pp. 2563–2566, Dec. 2018.

[18] C. Guo, Y. Cui, and Z. Liu, "Optimal multicast of tiled 360 VR video," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 145–148, Feb. 2019.

[19] N. Kan, J. Zou, K. Tang, C. Li, N. Liu, and H. Xiong, "Deep reinforcement learning-based rate adaptation for adaptive 360-degree video streaming," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 4030–4034.

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

[21] J. Chakareski, "Viewport-adaptive scalable multi-user virtual reality mobile-edge streaming," *IEEE Trans. Image Process.*, vol. 29, pp. 6330–6342, 2020.

[22] W. Huang *et al.*, "Utility-oriented resource allocation for 360-degree video transmission over heterogeneous networks," *Digit. Signal Process.*, vol. 84, pp. 1–14, Jan. 2019.

[23] Huawei. *Whitepaper on the VR-Oriented Bearer Network Requirement (2016)*. Accessed: Sep. 2016. [Online]. Available: http://www-file.huawei.com/~/media/CORPORATE/PDF/white%20paper/whitepaper-on-the-vr-oriented-bearer-network-requirement-en.pdf

[24] L. Zhao, Y. Cui, Z. Liu, Y. Zhang, and S. Yang, "Adaptive streaming of 360 videos with perfect, imperfect, and unknown FoV viewing probabilities in wireless networks," *IEEE Trans. Image Process.*, vol. 30, pp. 7744–7759, 2021.

[25] C. Guo, L. Zhao, Y. Cui, Z. Liu, and D. W. K. Ng, "Power-efficient wireless streaming of multi-quality tiled 360 VR video in MIMO-OFDMA systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 5408–5422, Aug. 2021.

[26] L. Teng *et al.*, “QoE driven VR 360° video massive MIMO transmission,” *IEEE Trans. Wireless Commun.*, vol. 21, no. 1, pp. 18–33, Jan. 2022.

[27] C. Perfecto, M. S. Elbamby, J. D. Ser, and M. Bennis, “Taming the latency in multi-user VR 360°: A QoE-aware deep learning-aided multicast framework,” *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 2491–2508, Apr. 2020.

[28] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, “Edge computing meets millimeter-wave enabled VR: Paving the way to cutting the cord,” in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2018, pp. 1–6.

[29] M. Chen, O. Semiari, W. Saad, X. Liu, and C. Yin, “Federated echo state learning for minimizing breaks in presence in wireless virtual reality networks,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 177–191, Jan. 2020.

[30] P. Yang, X. Xi, Y. Fu, T. Q. S. Quek, X. Cao, and D. Wu, “Multicast eMBB and bursty URLLC service multiplexing in a CoMP-enabled RAN,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 5, pp. 3061–3077, May 2021.

[31] Qualcomm. *How CoMP Can Extend 5G NR to High Capacity and Ultra-Reliable Communications*. Accessed: Jul. 11, 2018. [Online]. Available: https://www.qualcomm.com/media/documents/files/how-comp-can-extend-5g-nr-to-high-capacity-and-ultra-reliable_communications.pdf

[32] *Coordinated Multi-Point Operation for LTE Physical Layer Aspects (Release 11)*, document TR 36.819, 3GPP, Sep. 2013. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2498.

[33] Q. Cheng, H. Shan, W. Zhuang, L. Yu, Z. Zhang, and T. Q. S. Quek, “Design and analysis of MEC-and proactive caching-based 360° mobile VR video streaming,” *IEEE Trans. Multimedia*, vol. 24, pp. 1529–1544, 2022, doi: 10.1109/TMM.2021.3067205.

[34] Y. Sun, Z. Chen, M. Tao, and H. Liu, “Communications, caching, and computing for mobile virtual reality: Modeling and tradeoff,” *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 7573–7586, Nov. 2019.

[35] Y. Ban, Y. Zhang, H. Zhang, X. Zhang, and Z. Guo, “MA360: Multi-agent deep reinforcement learning based live 360-degree video streaming on edge,” in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2020, pp. 1–6.

[36] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, and P. Tran-Gia, “A survey on quality of experience of HTTP adaptive streaming,” *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 469–492, Sep. 2015.

[37] O. Semiari, W. Saad, M. Bennis, and Z. Dawy, “Inter-operator resource management for millimeter wave multi-hop backhaul networks,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 5258–5272, Aug. 2017.

[38] S. Bouchard, J. St-Jacques, G. Robillard, and P. Renaud, “Anxiety increases the feeling of presence in virtual reality,” *Presence, Teleoperators Virtual Environ.*, vol. 17, no. 4, pp. 376–391, Aug. 2008.

[39] M. S. L. Khan, A. Halawani, S. ur Rehman, and H. Li, “Action augmented real virtuality: A design for presence,” *IEEE Trans. Cognit. Develop. Syst.*, vol. 10, no. 4, pp. 961–972, Dec. 2018.

[40] L. B. Pedersen and R. Nordahl, “Experiencing presence in a virtual reality music video,” in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces Abstr. Workshops (VRW)*, Mar. 2021, pp. 86–89.

[41] X.-W. Tang, X.-L. Huang, and F. Hu, “QoE-driven UAV-enabled pseudo-analog wireless video broadcast: A joint optimization of power and trajectory,” *IEEE Trans. Multimedia*, vol. 23, pp. 2398–2412, 2020.

[42] C. Zhan and R. Huang, “Energy efficient adaptive video streaming with rotary-wing UAV,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 8040–8044, Jul. 2020.

[43] S. Scardapane, D. Wang, and M. Panella, “A decentralized training algorithm for echo state networks in distributed big data applications,” *Neural Netw.*, vol. 78, pp. 65–74, Jun. 2015.

[44] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, “Scheduling policies for federated learning in wireless networks,” *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, Jan. 2020.

[45] P. Yang, T. Q. S. Quek, J. Chen, C. You, and X. Cao, “Feeling of presence maximization: mmWave-enabled virtual reality meets deep reinforcement learning,” 2021, *arXiv:2107.01001*.

[46] M. Bennis, S. M. Perlaza, P. Blasco, Z. Han, and H. V. Poor, “Self-organization in small cell networks: A reinforcement learning approach,” *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 3202–3212, Jul. 2013.

[47] P. Yang, X. Cao, X. Xi, W. Du, Z. Xiao, and D. Wu, “Three-dimensional continuous movement control of drone cells for energy-efficient communication coverage,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6535–6546, Jul. 2019.

[48] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. ICLR*, 2015, pp. 1–15.

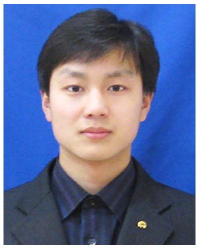
[49] P. Yang, X. Xi, T. Q. S. Quek, J. Chen, X. Cao, and D. Wu, “How should i orchestrate resources of my slices for bursty URLLC service provision?” *IEEE Trans. Commun.*, vol. 69, no. 2, pp. 1134–1146, Feb. 2021.

[50] Z.-Q. Luo, W.-K. Ma, A. M.-C. So, Y. Ye, and S. Zhang, “Semidefinite relaxation of quadratic optimization problems,” *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 20–34, Apr. 2010.

[51] L. Huang, S. Bi, and Y.-J.-A. Zhang, “Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks,” *IEEE Trans. Mobile Comput.*, vol. 19, no. 11, pp. 2581–2593, Nov. 2020.

[52] J. Tang, B. Shim, and T. Q. S. Quek, “Service multiplexing and revenue maximization in sliced C-RAN incorporated with URLLC and multicast eMBB,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 4, pp. 881–895, Apr. 2019.

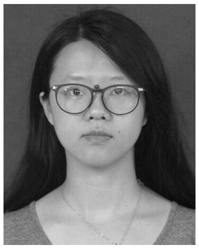
[53] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, “Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience,” *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1046–1061, May 2017.



Peng Yang (Member, IEEE) received the Ph.D degree in signal and information processing from Beihang University in 2018. Since 2021, he has been with Beihang University, where he is currently an Associate Professor. His current research topics include airborne communications and networking, network intelligence, network slicing, URLLC, and airborne video transmission.

Tony Q. S. Quek (Fellow, IEEE) is the Cheng Tsang Man Chair Professor with the Singapore University of Technology and Design (SUTD). He also serves as the Head of ISTD Pillar, the Sector Lead of the SUTD AI Program, and the Deputy Director of the SUTD-ZJU IDEA. His current research topics include wireless communications and networking, network intelligence, the Internet of Things, URLLC, and big data processing. He is currently the Editor of the IEEE Transactions on Wireless Communications and an Elected Member of the IEEE Signal Processing Society SPCOM Technical Committee.

Jingxuan Chen is currently pursuing the Ph.D. degree with Beihang University. His research interests include the intelligent transportation systems, mobile edge computing, and next-generation mobile cellular systems.



Chaoqun You received the B.S. and Ph.D. degrees in communication engineering from the University of Electronic Science and Technology of China (UESTC), in 2013 and 2020, respectively. She is a Post-Doctoral Research Fellow with the Singapore University of Technology and Design. Her research interests include data center networks, network function virtualization, and distributed machine learning.

Xianbin Cao (Senior Member, IEEE) is the Dean and a Professor with the School of Electronic and Information Engineering, Beihang University, Beijing, China. His current research interests include intelligent transportation systems, airspace transportation management, and intelligent computation. Currently, he is an Associate Editor of IEEE Transactions on Network Science and Engineering and IEEE Transactions on Vehicular Technology.