IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 3 January 2024, accepted 15 January 2024, date of publication 30 January 2024, date of current version 6 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3359708

RESEARCH ARTICLE

Adaptive Foveated Rendering and Offloading in an Edge-Assisted Virtual Reality System

BARAKA WILLIAM NYAMTIGA[1], DEREK KWAKU POBI ASIEDU[2], (Member, IEEE), AIRLANGGA ADI HERMAWAN[1], YAKUB FAHIM LUCKYARNO[1], AND JI-HOON YUN[3,4], (Senior Member, IEEE)

[1]Department of Electrical and Information Engineering, Seoul National University of Science and Technology, Seoul 01811, South Korea
[2]Department of Mathematics and Electrical Engineering, Institut Mines-Télécom (IMT) Atlantique, 29238 Brest, France
[3]Department of Electrical and Information Engineering, Seoul National University of Science and Technology, Seoul 01811, South Korea
[4]Research Center for Electrical and Information Technology, Seoul National University of Science and Technology, Seoul 01811, South Korea

Corresponding author: Ji-Hoon Yun (jhyun@seoultech.ac.kr)

This work was supported in part by the Institute of Information Communications Technology Planning and Evaluation (IITP) grant funded by the Korean Government [Ministry of Science and ICT (MSIT)] under Grant 2017-0-00650, and in part by the National Research Foundation of Korea (NRF) funded by the Korean Government (MSIT) under Grant 2022R1F1A1076387.

ABSTRACT Foveated rendering (FR)—in which the central foveal layer (the area around the eye gaze) of a virtual reality (VR) image is rendered at the highest resolution and the peripheral layers are rendered at progressively lower resolutions—is an advanced VR technique that controls the balance between computational load and visual quality by adjusting the foveal layer sizes. We consider an edge computing-assisted VR computation offloading system incorporating FR and develop a deep reinforcement learning (DRL)-based solution to maximize a unified objective combining the visual quality and the overheads of energy consumption and delay for multiple VR users by optimizing the foveal layer size determination, offloading decisions and radio resource allocation of wireless links with non-orthogonal multiple access (NOMA). To formulate the unified objective, the user-perceived visual quality of a VR image rendered via FR is modeled with the sizes of the foveal layers as variables. The proposed solution consists of three main modules: per-user FR and offloading modules, which determine the sizes of the foveal layers and make offloading decisions, including subband allocation, and a central transmit-power allocation module. The actions of these three modules determine the reward that is in turn fed back to each module. Evaluation results reveal that the proposed framework can better adapt the operations of FR, offloading and wireless transmission to the environmental conditions than other FR and offloading benchmarks in terms of overall reward, visual quality, energy consumption and delay.

INDEX TERMS Virtual reality, foveated rendering, edge computing, offloading, deep reinforcement learning.

I. INTRODUCTION

Immersive virtual reality (VR) is a rapidly growing media paradigm that offers users omnidirectional experiences in a three-dimensional (3D) environment through specialized headset hardware equipment. Such a headset captures the user's head pose and input triggers through sensory devices, allowing real-time generation of synthetic scenes [1].

To create an interactive VR environment that fully immerses users, it is essential to provide a high-quality visual experience and seamless navigation within the virtual world [2]. Conventional VR systems achieve these requirements by utilizing high-end local computers with dedicated graphics cards that boast abundant GPU cores for rendering graphics. However, the high costs associated with computing and graphics processing hardware for each end user have presented significant obstacles to the widespread adoption of VR [3]. Moreover, conventional VR systems often require

The associate editor coordinating the review of this manuscript and approving it for publication was Hosam El-Ocla.

cumbersome wire harnesses connecting the headset to a local computer, thereby restricting user mobility. Additionally, the need for content download and installation time can impede immediate playability, further limiting the user experience.

Offloading heavy VR computations to a network edge computing entity [3], [4], [5], [6], [7] possessing sufficient computational power, which we call *edge VR* in this paper, is rapidly emerging as a promising solution. The abundant processing and storage resources found at the edge can make high-quality and pay-as-you-go VR services available anytime and anywhere across the globe to a large group of online VR users with affordable thin client VR headsets wirelessly connected to the Internet [8], [9], [10]. With the edge providing the necessary resources to execute and render scenes, the client headsets are left only with the task of displaying the content streamed over wireless networks. Higher user mobility is facilitated, and the headsets' battery life is also expected to be prolonged due to their light operations [3].

To reduce the rendering load, which is the most significant load among the computational operations for VR, *foveated rendering (FR)* has recently gained much attention. With increasing eccentricity (angular distance) from the foveation point, the density of the receptors and ganglion cells in the retina decreases, causing visual acuity to fall off radically toward the periphery [11]. Therefore, instead of uniformly rendering a full-resolution image over the whole display, only the foveal layer (the area around the eye gaze) at is rendered at the highest resolution (high quality) in FR, while the peripheral layers are rendered at progressively lower resolutions (i.e., with lower quality). As illustrated in Fig. 1, for example, inner, middle and outer progressive layers with decreasing preset resolutions can be constructed, where the inner layer has the smallest angular diameter and the remaining layers cover increasingly larger diameters. Consequently, FR will result in fewer pixels to be processed without significantly compromising the visual quality perceived by the user. Since identifying the eye gaze would require the VR headset to be equipped with an eye tracking hardware device, which is not available in most current headsets, fixed FR assuming that the eye gaze lies at the center of the rendered image is widely adopted [12].

Applying FR in edge VR is still effective in reducing the collective computational load of the edge, thus enabling the edge to support VR users at a large scale. To minimize the computational load, the edge should minimize the foveal layer while expanding the peripheral layers, but this would come at the expense of a decreased perceived quality of the user experience. In contrast, expanding the foveal layer would increase the frame size to be streamed to a user, thus increasing the energy consumption of the user for processing this larger frame and also increasing the latency. Allocating more wireless resources or switching to local computing will reduce/eliminate the communication latency, while the energy consumption can be reduced by reducing the foveal layer (at the cost of decreasing the perceived quality). In summary, FR operations, wireless resource management, and offloading decisions have a mutual relationship determining the costs of edge VR (in terms of perceived quality, energy consumption and delay), thus necessitating that they be optimized in a unified and balanced manner.

Several recent studies have focused on the optimization of offloading decisions and radio resource allocation for VR offloading. The problem of allocating resources to balance the trade-off among communication, computing, and caching has been studied both under a fixed delay constraint [13] and with the objective of delay minimization [2], [14]. Lin et al. [15] solved a similar problem of resource allocation optimization for minimization of the energy consumption of VR headsets. Reinforcement learning (RL) has demonstrated its effectiveness in solving mixed-integer nonlinear problems, across diverse applications and dynamic environments, including systems involving multiple agents [16], [17]. Guo et al. [18] solved the problem using a distributed RL approach consisting of an offline training phase and an online running phase. In a similar context, Trinh and Muntean [19] developed an RL-based offloading scheme tailored for extended reality (XR) devices, considering energy consumption and execution delay. Other studies have investigated an edge-assisted 360-degree VR video streaming problem for optimization of the quality of experience (QoE) and resource allocation [14], [20], [21]. However, no existing works have considered the optimization of FR in combination with offloading and radio resource allocation.

In this paper, we formulate and develop a solution for the problem of jointly optimizing the foveal layer sizes in FR, offloading decisions, and radio resource allocation for multicarrier non-orthogonal multiple access (MC-NOMA) in an edge VR offloading system serving multiple VR users. The proposed solution minimizes a combined objective function of user-perceived visual quality, energy consumption and delay to provide an acceptable visual experience in VR at low latency and a minimal expenditure of energy. To formulate the unified objective, the user-perceived visual quality of a VR image rendered via FR is modeled with the sizes of the foveal layers as variables. The per-user deep RL (DRL) agent of the proposed solution is composed of FR and offloading modules that determine the sizes of the foveal layers and perform offloading decision-making and subband allocation, respectively. The action determined by the FR module is passed to the offloading module to obtain the payload information, and the action determined by the offloading module is passed to a central transmit-power allocation module. The actions of these three modules determine the reward that is in turn fed back to each module. Each per-user DRL agent makes actions based on the local observations of the corresponding user, without complete knowledge of the system dynamics and parametric details of other users, and reports only its determined actions to the central transmit-power allocation module. Therefore, the agent processes can

be distributed among local computing entities or user devices to achieve decentralized and load-distributed operation. For the specific algorithm of the DRL agent, we provide a model design incorporating double and dueling deep Q-network (DQN) techniques tailored for the FR, offloading, and power allocation modules of the agent.

Evaluation results reveal that the proposed solution better adapts the operations of FR, offloading and wireless transmission to the environmental conditions than other FR and offloading benchmarks in terms of overall reward, visual quality, energy consumption and delay.

The main contributions of our work are summarized as follows:

- Formulation of a unified optimization problem for the foveal layer sizes, offloading decisions, and radio resource (subband and transmit power) allocation for MC-NOMA in an edge VR offloading system serving multiple VR users.
- Modeling the user-perceived visual quality of a VR image rendered via FR, considering the characteristics of the human visual system (HVS).
- Design of an optimization framework consisting of per-user DRL agents consisting of FR and offloading modules, which determine the sizes of the foveal layers, offloading decisions and subband allocation, and a centralized power allocation module.
- Design of DRL agent algorithms incorporating double and dueling DQN techniques tailored for the FR, offloading, and power allocation modules of the agent.
- Comprehensive simulations illustrating the performance gain of the proposed solution over other options under different network and VR content conditions.

The organization of the rest of this paper is as follows. First, recent studies related to edge-assisted offloading for VR and other applications are presented in Section II. The system models and problem formulation are then described in Section III. In Section IV, the visual quality of a VR image rendered via FR is derived. In Section V, the proposed solution is detailed as the transformation of the problem into an RL task, and the associated algorithms are described. In-depth evaluations and discussions of the results are provided in Section VI. Finally, Section VII concludes the paper with potential future directions of study.

Notations: The subscripts m and j of a variable indicate that it belongs to user m and subband j, respectively. $x(t)$ denotes the variable x for a specific VR frame index t.

II. RELATED WORKS

In this section, we review and discuss related works. In particular, a comparative summary of related research on edge VR, including our work, is presented in Table 1. Despite the abundance of studies covering various aspects of edge resource management, no existing works have explored the adaptation of FR in conjunction with the management of other resources for edge VR. Thus, this study represents the first attempt to develop a holistic DRL-based solution for optimizing the foveal layer sizes, offloading decisions, and NOMA radio resource allocation in the context of edge VR.

There have been several recent studies on optimizing offloading decisions and radio resource allocation for edge VR. Mehrabi et al. [2] aimed to strike a balance between video quality and latency by formulating the problem as a mixed-integer nonlinear problem. Yang et al. [13] leveraged computation and caching resources at the VR headset to minimize communication resource consumption. They employed task modularization, in which a VR device constructs a task using a combination of components cached in its memory and others delivered from the edge. Chen et al. [22] proposed an edge VR framework that supports player interaction. They optimized the allocation of computing resources at the edge, wireless bandwidth allocation, and the postprocessing decision policy to minimize the average interplayer delay. Guo et al. [18] addressed the VR offloading problem by employing a distributed learning approach based on RL. Their approach consisted of an offline training phase and an online running phase.

Some studies have specifically focused on the edge-assisted delivery of 360-degree VR video. Gupta et al. [20] investigated a multiuser 360-degree VR video streaming system that leverages multilayer video tiling, edge computing, and wireless multiconnectivity, incorporating sub-6 GHz and millimeter wave links. Dang and Peng [14] presented a VR delivery framework that precaches portions of VR videos. They formulated a joint decision problem for resource allocation at both mobile VR devices and access points, aiming to maximize the average delay tolerance. Zhang and Chakareski [21] proposed an edge computing network assisted by unmanned aerial vehicles (UAVs) to enable high-quality mobile 360-degree VR video applications. They formulated a joint problem involving UAV placement, computation and radio resource allocation as well as 360-degree video content layer assignment with the objective of maximizing the delivered QoE.

Employing machine learning (ML) to address optimization in edge-assisted systems has gained tremendous popularity [5], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32]. An intelligent offloading framework was formulated in [25] that jointly utilizes licensed and unlicensed spectra for vehicular networks. An offloading algorithm enabling vehicles to use RL and multiarmed bandit theory to learn from their neighbors how to minimize delays was studied by Sun et al. [24]. A distributed cognitive network was investigated in [23] that utilizes ML to balance resource allocation and optimize spectrum utilization. ML can also be used by energy-harvesting Internet of Things (IoT) devices [31] to select edge nodes and learn optimal offloading policies, in VR to guide user movements by influencing their behaviors based on past movements [28], and to jointly optimize computation and network resource allocation to minimize service time in dynamic edge environments [30]. It has been further exploited in blockchain-based edge environments to address

TABLE 1. Summary of related works on edge VR.

Reference	Objective	VR content	Method	FR optimization	Offloading optimization	Radio resource allocation	Multiple access scheme
Yang *et al.* [13]	Transmission data minimization under a delay constraint	VR video	Lyapunov theory	No	Yes	No	OMA
Mehrabi *et al.* [2]	Maximization of video quality and minimization of latency by optimizing the viewport size	Interactive VR	Heuristic	No	Yes	No	OMA
Dang *et al.* [14]	Delay tolerance maximization by optimizing radio communication, caching and computing decision	VR video	Lagrangian dual decomposition	No	Yes	Yes	OMA
Chen *et al.* [22]	Interplayer delay minimization by optimizing the computing resource allocation at the edge, wireless bandwidth allocation, and postprocessing decision	Interactive VR	NESTT-G	No	Yes	Yes	OMA
Gupta *et al.* [20]	Immersion distortion minimization by optimizing transmitted tiles	VR video	Geometric programming	No	No	No	OMA
Zhang *et al.* [21]	QoE maximization by optimizing UAV placement, computation and radio resource allocation, 360° video content layer assignment	VR video	Heuristic	No	No	Yes	OMA
Lin *et al.* [15]	Energy consumption minimization by optimizing viewport rendering offloading, computing, and spectrum resource allocation	Interactive VR	RL	No	Yes	Yes	OMA
Guo *et al.* [18]	QoE maximization by optimizing user association, offloading mode selection, and caching policy	Interactive VR	RL	No	Yes	No	OMA
Trinh *et al.* [19]	Energy consumption and delay minimization by optimizing offloading decision	Interactive VR	RL	No	Yes	No	OMA
Ours	Maximization of QoE and minimization of power consumption and latency by optimizing FR layer sizes, offloading, subband and transmit power allocation	Interactive VR	RL	Yes	Yes	Yes	NOMA

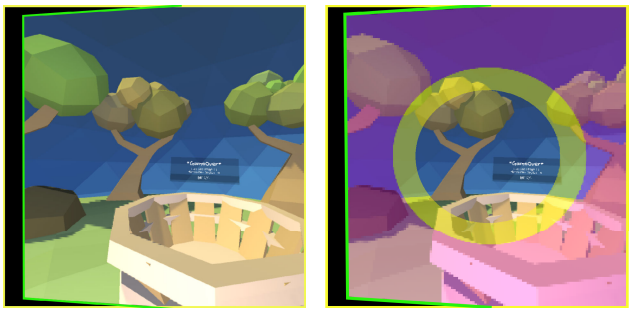
complexity in offloading the mining and data processing tasks [5], [33]. Other variations of learning algorithms using echo state networks were discussed in [26] and [34] in relation to enhancing resource management. Multistage DRL approaches have also been explored by [16] and [35] to maximize the computation rate of an edge server and regulate inverter-based energy resources, respectively. The dueling DQN DRL architecture [36] has been shown to improve the estimation of long-term rewards and to support fast training and reliable predictions with less complexity. This architecture has been deployed in [37] for task offloading, in [38] to optimize market making, and in [39] by UAVs providing uninterrupted acquisition of sensing data to IoT terminals. Qiong et al. [40] developed a DQN-based algorithm that predicts the optimal minimum contention window for channel access in vehicle-to-infrastructure communication networks to improve data freshness.

Many research works on edge offloading have considered NOMA as the multiple access scheme for users' wireless links. One study [41] addressed the joint optimization of task offloading decisions, NOMA transmission, and resource allocation for computation-intensive and delay-sensitive services in the industrial IoT. The objective was to minimize energy consumption while meeting specified latency constraints

TABLE 2. Summary of notations.

Symbol	Definition
$\mathcal{M}$, M	Set and number of UEs
$\mathcal{S}$, S	Set and number of subbands
K	Number of foveal layers
R_k	Radius of the kth foveal layer
r_k	Resolution of the kth foveal layer
n_k	Resolution factor of the kth foveal layer
N_{px}	Number of pixels of a VR frame at full (native) resolution, with N_W in width and N_H in height
PixelDepth	Number of bits per pixel
T_e (T_l)	End-to-end latency of a VR frame when processed by an edge server (UE)
T^{ps}	Pose data transmission time
T_e^{rn} (T_l^{rn})	Rendering time at the edge (UE)
T^{ec} (T^{dc})	Decoding time
T^{tx}	Frame transmission time
E_e (E_l)	Energy consumption to process a VR frame at the edge (UE)
V_e^{en} (V_l^{en})	Video encoding speed (bps) of the edge (UE)
V_e^{de} (V_l^{de})	Video decoding speed (bps) of the edge (UE)
C_e (C_l)	Processing capability (cycles/sec) of the edge (UE)
Z_s (Z_l)	Data volume computed per CPU cycle (bits/cycle) at the edge (UE)
ζ	Compression efficiency
α_m	Offloading indicator for UE m
B_s	Bandwidth of a subband
P	Total power budget at the BS
$g_{m,j}$	Channel gain of UE m in subband j
$p_{m,j}$	Power allocated to UE m in subband j
$\rho_{m,j}$	Fractional power allocated to UE m in subband j
N_0	Noise power spectral density
$\mathcal{N}_j$	Set of UEs assigned to subband j
Γ	User-perceived quality
Γ_r	Resolution-dependent quality factor
Γ_p	Position-dependent quality factor
f_c	Cutoff frequency
ω, ϕ	Balancing coefficients between reward terms
θ	Offloading neural network's parameters
γ	Discount factor
ϵ	Exploration probability

using an online DRL-based algorithm. In another work [42], the focus was on minimizing the energy consumption of IoT devices in wireless-powered backscatter communication networks. The authors formulated a joint offloading and radio resource allocation problem and optimized user association, task computation offloading coefficients, reflection coefficients of backscatter devices, transmit power, and transmission time of wireless devices. Ding et al. [43] investigated NOMA in mobile offloading scenarios to reduce latency and energy consumption. The use of NOMA to conserve energy for edge users has also been explored in other studies [44], [45], [46]. The authors of [46] investigated the weighted sum-energy consumption minimization problem in an eavesdropping scenario, subject to constraints regarding the secrecy offloading rate, computation latency and secrecy outage probability. Nduwayezu et al. [47] proposed the use of NOMA to maximize the computation rate of an edge server by optimizing computation offloading and subband allocation. NOMA has also been studied in relation to the Internet of Video Things [48], coupled with multiple edge servers to improve throughput and enhance devices' computation capabilities.



(a) No FR (b) Three-layer FR

FIGURE 1. Illustration of VR images without and with FR.

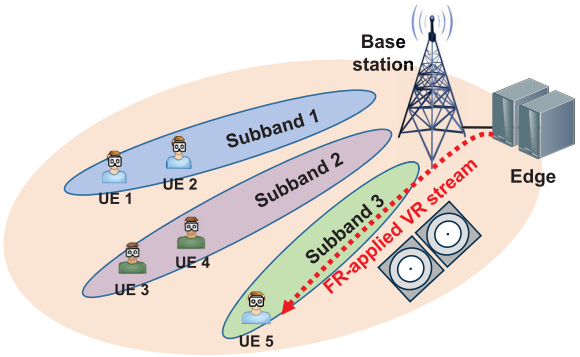


FIGURE 2. Edge-assisted VR offloading system.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. SYSTEM DESCRIPTION

The system model is illustrated in Fig. 2. We consider a typical radio access network (RAN) scenario composed of a base station (BS) and a set $\mathcal{M}$ of VR user headsets associated with an edge of the RAN. Each VR user headset, which we alternatively refer to as the *user equipment (UE)*, tracks the user's head pose and displays the streamed content to the user. Each VR headset operates in either local or offloading mode. In the local mode, viewport rendering is performed by the headset itself. If the headset is operating in the offloading mode, it periodically sends head pose information to the edge over a wireless link, and viewport rendering is performed by the edge based on the received pose information. Afterward, the rendered image data are streamed through the RAN, which has a spectral bandwidth W consisting of a set of subbands $\mathcal{S} = \{1, 2, \ldots, S\}$ that can be non-orthogonally shared by multiple UEs in accordance with the principle of NOMA. The notation $\alpha \in \{0, 1\}$ is used to define the selected offloading strategy in a particular decision step, where $\alpha(t) = 0$ means that rendering is done locally for VR frame $t \in \mathcal{T}$; otherwise, the rendering task will be offloaded to the edge. Table 2 lists the notations that are frequently used in this paper.

B. FR MODEL

In the considered system model, we assume that a VR image is rendered using FR with K circular layers based on individual layer configurations for different users. The

circular form of each foveal layer is derived from the circular symmetry of the HVS [11]. Specifically, we define the kth foveal layer ($k = 1, 2, \ldots, K-1$) as the area outside the $(k-1)$th layer and bounded by its radius R_k from the foveal point, where $R_{k-1} < R_k$. The outermost (Kth) layer represents the remaining area outside the $(K-1)$th layer. This design enables us to allocate more resources (i.e., higher resolutions) to regions closer to the foveal point, where human vision is most acute, while gradually reducing resource allocation in areas farther from the foveal point, where visual acuity decreases. Referring to the variable rate shading (VRS) feature of NVIDIA [49], [50], the resolution of the kth layer ($k = 1, \ldots, K$), which we denote by r_k, is determined by a factor of $\frac{1}{n_k^2}$ of the native resolution (r_1), i.e.,

$$r_k = \frac{r_1}{n_k^2}, \tag{1}$$

where n_k is an even number. Accordingly, an $n_k \times n_k$ pixel region is represented by one rendered pixel in the kth layer. For example, if $n_k = 4$, a 4×4-pixel region will be represented by one pixel of information, resulting in a resolution that is $\frac{1}{16}$ of the native resolution. Consequently, as the value of n_k increases, the resolution of the kth layer progressively decreases. We assume that the innermost layer ($k = 1$) is rendered at full resolution, i.e., every pixel in this region is shaded individually, denoted by $n_k = 1$, and the successive layers ($k = 2, 3, \ldots$) will have one-step decrements in resolution. Hence, the resolution-determining factors $\{n_k | k = 1, \ldots, K\}$ for the respective foveal layers are given as $\{1, \frac{1}{4}, \frac{1}{16}, \ldots\}$. With this configuration, the optimization problem for FR is now reduced to finding the optimal radii of the foveal layers, denoted by R_k, $k = 1, \ldots, K$, for a given K.

We define N_{px} as the total number of pixels in a per-eye frame at full (native) resolution, with N_W pixels in width and N_H pixels in height. Let a_k be the fractional area covered by layer k relative to the entire viewport area ($0 \leq a_k \leq 1$, $\sum_{k=1}^{K} a_k = 1$). We assume that the number of bits for a pixel is given as PixelDepth across all pixels in the frame. Then, a frame has the following number of bits:

$$D_{fov} = N_{px} \cdot \text{PixelDepth} \sum_{k=1}^{K} \frac{a_k}{n_k^2}. \tag{2}$$

C. LATENCY AND ENERGY MODELS

1) OFFLOADING MODE

The process of rendering a VR image at the edge involves a chain of multiple subtasks, with each subtask contributing to the overall latency. The execution time for each subtask is as follows:

a) *Pose data transmission* (T^{ps}): The process begins with the transmission of pose data, of size d, from the VR headset to the edge. This step requires $T^{ps} = \frac{d}{R^u}$ seconds, where R^u represents the uplink (UL) data rate.

b) *Rendering* (T_e^{rn}): The VR image rendering process depends on the computing capability C_e assigned by the edge. Under the assumption that a data size of Z_e can be processed per computing cycle, the time to render a VR image is given by $T_e^{rn} = \frac{D_{fov}}{C_e \cdot Z_e}$.

c) *Compression* (T^{ec}): To reduce network bandwidth consumption, the rendered VR image is compressed before transmission. Compression takes $T^{ec} = \frac{D_{fov}}{V_e^{ec}}$, where V_e^{ec} denotes the edge's video encoding speed. If we define ζ as the encoding efficiency, representing the ratio of the output data size of the compression codec to the input data size, the encoder outputs $D_{ec} = \zeta \cdot D_{fov}$.

d) *Frame transmission* (T^{tx}): The transmission of the compressed frame takes $T^{tx} = \frac{D_{ec}}{R^d}$, where R^d represents the downlink (DL) data rate.

e) *Decoding* (T^{dc}): Upon receiving the compressed VR frame, the headset decodes it to obtain the raw data. This decoding process takes $T^{dc} = \frac{D_{ec}}{V_l^{dc}}$, where V_l^{dc} represents the headset's video decoding speed.

Upon summing all the latency components, the end-to-end latency for edge-assisted rendering of a VR frame is given by

$$T_e = T^{ps} + T_e^{rn} + T^{ec} + T^{tx} + T^{dc}, \tag{3}$$

and the corresponding energy consumption of the headset during this process is computed as

$$E_e = T^{ps} P_{snd} + (T_e^{rn} + T^{ec}) P_{idle} + T^{tx} P_{rcv} + T^{dc} P_{dc}, \tag{4}$$

where P_{snd}, P_{idle}, P_{rcv} and P_{dc} represent the power consumed by the headset during UL transmission, in the idle state, during DL reception and during decoding, respectively.

2) LOCAL MODE

In the local mode, the headset encounters latency and consumes energy only during rendering. Thus, the total latency of the local mode for a VR frame is

$$T_l = T_l^{rn} = \frac{D_{fov}}{C_l \cdot Z_l}, \tag{5}$$

where C_l and Z_l are the computing capability and the data size processed per computing cycle, respectively, of the headset. The corresponding energy consumption of the headset is

$$E_l = T_l^{rn} P_{rn}, \tag{6}$$

where P_{rn} is the power consumed by the headset for rendering.

D. WIRELESS TRANSMISSION MODEL

In the offloading mode, users' pose data are transmitted to the edge via UL communication, while the rendered VR frames are transmitted to the UEs via DL communication. Since the size of the pose data (d) is expected to be significantly smaller than that of the VR frame data, our primary focus in this section is on DL communication. Following the principle of NOMA, the BS superimposes the DL data waveforms of

multiple UEs with an appropriate power allocation, allocating higher transmit power levels to UEs with weaker channel gains [51].

Upon reception, each UE decodes the strongest signal first and then subtracts it from the received signal. During the decoding process for a particular UE's target signal, signals for other UEs are treated as interference. The same decoding process continues for the remaining signals until the UE with the weakest received power has its signal decoded. Let $\mathcal{N}_j$ be the set of UEs allocated to subband j, and let $\mathcal{N}_{j,<m}$ be the subset of $\mathcal{N}_j$ consisting of the UEs with lower received powers than UE m. $g_{m,j}$ denotes the channel gain of UE m in subband j. The signal-to-interference-plus-noise ratio (SINR) for UE m in subband j is given by

$$\text{SINR}_{m,j} = \frac{g_{m,j}p_{m,j}}{N_0 B_s + \sum_{i\in\mathcal{N}_{j,<m}} g_{i,j}p_{i,j}}, \tag{7}$$

where N_0 represents the noise power spectral density and B_s is the transmission bandwidth in a subband. We use $\beta_{m,j} = 1$ to indicate that UE m is assigned to subband j, i.e., $\forall m \in \mathcal{N}_j$, and $\beta_{m,j} = 0$ otherwise. Accordingly, the DL throughput of UE m is calculated as follows:

$$R_m^d = \sum_{j\in\mathcal{S}} \beta_{m,j} B_s \log_2(1 + \text{SINR}_{m,j}). \tag{8}$$

It is known that the multiplexing gain of NOMA significantly decreases, even under ideal conditions, when more than two UEs are allocated to a subband [52]. Therefore, we limit the number of UEs allocated to a subband to two.

E. PROBLEM FORMULATION

With the models described above, our objective is to provide a VR experience that strikes a balance between the quality of user perception, denoted by Γ and modeled in the next section, and the energy and delay costs for the UEs. To achieve this, we formulate a target problem that involves the joint optimization of FR, offloading decisions, and radio resource allocation across the frames $t \in \mathcal{T}$ of a VR content application. The target problem, which we denote by $\mathcal{P}$, is expressed as follows:

$$\begin{aligned}
\mathcal{P}: \max_{\mathbf{R},\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{p}} & \sum_{t\in\mathcal{T}}\sum_{m\in\mathcal{M}} [\omega(1-\alpha_m(t))(\phi T_{l,m}(t) + E_{l,m}(t)) \\
& + \omega\alpha_m(t)(\phi T_{e,m}(t) + E_{e,m}(t)) + \Gamma_m(t)], \\
\text{subject to} \quad & C1: 0 \le R_1(t) \le \cdots \le R_{K-1}(t) \\
& \le \min\{N_W, N_H\},\ t \in \mathcal{T}, \\
& C2: \alpha_m(t) \in \{0,1\};\ m \in \mathcal{M},\ t \in \mathcal{T}, \\
& C3: \beta_{m,j}(t) \in \{0,1\};\ m \in \mathcal{M}, j \in \mathcal{S},\ t \in \mathcal{T}, \\
& C4: \sum_{j\in\mathcal{S}}\sum_{m\in\mathcal{N}_j} p_{m,j}(t) \le P;\ t \in \mathcal{T}, \\
& C5: p_{m,j}(t) = 0 \text{ if } \beta_{m,j}(t) = 0;\ m \in \mathcal{M}, \\
& j \in \mathcal{S},\ t \in \mathcal{T},
\end{aligned} \tag{9}$$

where $\mathbf{R}$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\mathbf{p}$ are the vectors of $R_k(t)$, $\alpha_m(t)$, $\beta_{m,j}(t)$, and $p_{m,j}(t)$, respectively, with $k = 1, 2, \ldots, K-1$, $m \in \mathcal{M}$, $j \in \mathcal{S}$ and $t \in \mathcal{T}$. Because of the different scales of the different reward terms, we introduce adjustable coefficients ω and ϕ to control the relative weights between the overhead and quality terms and between the energy and delay terms, respectively, in the overall reward. Constraint $C1$ limits the radius of the inner layer to be the smallest and that of the outer layer to be the largest. $C2$ dictates that only one of two offloading policies can be carried out by a particular UE in each time frame: either local rendering or remote rendering. $C3$ confines the allocation of a UE to a specific subband to be binary. Constraint $C4$ indicates that the total transmit power allocated to the UEs cannot exceed the power budget P of the BS. Constraint $C5$ guarantees that a UE is assigned zero transmit power in a subband unless the UE is allocated to that subband.

To simplify problem $\mathcal{P}$, we impose a limitation that each UE can be allocated to only one subband. With this constraint, we can extend the definition of the offloading indicator α_m to also indicate the index of the subband to which UE m is allocated when $\alpha_m > 0$. If $\alpha_m = 0$, this means that UE m operates in the local mode. Consequently, we can eliminate the problem variable $\boldsymbol{\beta}$. Additionally, we assume that the power budget P is evenly distributed among the subbands, i.e., the power available in each subband is P/S, while optimizing the power allocation within each subband. The modified problem, denoted by $\mathcal{P}'$, is expressed as follows:

$$\begin{aligned}
\mathcal{P}': \max_{\mathbf{R},\boldsymbol{\alpha},\mathbf{p}} & \sum_{t\in\mathcal{T}}\sum_{m\in\mathcal{M}} [\omega(1-\alpha_m(t))(\phi T_{l,m}(t) + E_{l,m}(t)) \\
& + \omega\alpha_m(t)(\phi T_{e,m}(t) + E_{e,m}(t)) + \Gamma_m(t)], \\
\text{subject to } & C1, C4, \\
& C2': \alpha_m(t) \in \{0,1,2,\ldots,S\};\ m \in \mathcal{M}, \\
& C4': \sum_{m\in\mathcal{N}_j} p_{m,j}(t) \le P/S, j \in \mathcal{S}, t \in \mathcal{T}, \\
& C5': p_{m,j}(t) = 0 \text{ if } \alpha_m(t) \ne j;\ m \in \mathcal{M}, \\
& j \in \mathcal{S}, t \in \mathcal{T},
\end{aligned} \tag{10}$$

where $C2'$, $C4'$ and $C5'$ are the modified versions of constraints $C2$, $C4$ and $C5$, respectively.

IV. VISUAL QUALITY MODEL FOR FR

We model the user-perceived image quality by jointly considering the resolution of the image and the nature of the HVS. Let us consider a point in a VR image located at coordinates (a, b). Let the normalized perceived quality be denoted by Γ, let the resolution-dependent factor be Γ_r, and let the factor that depends on the position in the visual system be Γ_p; then, it follows that

$$\Gamma(a,b) = \Gamma_r(a,b) \times \Gamma_p(a,b). \tag{11}$$

The total perceived quality of a VR image is then given by

$$\Gamma = \sum_{a=1}^{N_W}\sum_{b=1}^{N_H} \Gamma_r(a,b) \times \Gamma_p(a,b). \tag{12}$$

The distance x (measured in pixels) from the foveation point (a_f, b_f) to the point (a, b) is computed as $x = \sqrt{(a-a_f)^2 + (b-b_f)^2}$. As stated in Section III.B, due to the assumed circular symmetry of the foveal layers, the resolution r at this specified point is a function of x, and thus, Γ_r also varies as a function of x. Research in [53] has shown that the perceived quality of an image follows a logarithmic relationship with the image resolution. Based on this finding, we model Γ_r for a point at a distance x from the foveation point as

$$\Gamma_r(x) = \frac{\log r(x) + a_0}{\log r_1 + a_0}, \tag{13}$$

where a_0 is an adjustable constant.

For Γ_p, we use the critical/cutoff frequency f_c beyond which any higher frequency is imperceptible for a certain eccentricity e (in degrees). Higher perceived quality is thus attained with higher f_c, as the visual sensitivity is at its maximum at the foveation point and rapidly declines at positions farther from the foveation point. In [11], f_c is formulated for a given eccentricity e by fitting experimental data as follows:

$$f_c(e) = \frac{e_2 \ln(\frac{1}{CT_0})}{\kappa(e+e_2)} \left(\frac{\text{cycles}}{\text{degree}}\right), \tag{14}$$

which can be simplified to $f_c(e) = \frac{1}{a_1 e + a_2}$ by adopting the notations $a_1 = \kappa/(e_2 \ln(1/CT_0))$ and $a_2 = a_1 e_2$. Here, κ denotes the spatial frequency decay constant, CT_0 is the minimal contrast threshold, and e_2 is the half-resolution eccentricity constant. In [54], the best-fit parameters are given values of $CT_0 = 1/64$, $\kappa = 0.106$ and $e_2 = 2.3$.

To adopt this approach into our model, we assume that the line from the fovea to the foveation point in the image is perpendicular to the image plane. v is the viewing distance from the eye to the image plane, and D is the display size; both are fixed for a particular headset. For our simulations, we set $v = 5$ cm and $D = 3.5$ inches, which are common parameters for many VR headsets. We then calculate e for a pixel at position x that is viewed from a distance v as

$$e(v, x) = \tan^{-1}\left(\frac{Dx}{N_W v}\right) \approx \frac{\pi}{4} b_1 x, \tag{15}$$

for which we employ the arctangent methods provided in [55] to obtain the approximation $\tan^{-1}(b_1 x) \approx \frac{\pi}{4} b_1 x$, where $b_1 = \frac{D}{N_W v}$. At distance x, Γ_p is therefore computed as the normalized cutoff frequency, the maximum value of which is always assumed to be one, at $e = 0$. Γ_p now becomes

$$\begin{aligned}\Gamma_p(x) &= \frac{f_c(e)}{f_c(0)} = \left(\frac{1}{a_1 e + a_2}\right) / \left(\frac{1}{a_1(0) + a_2}\right) \\ &= \frac{a_2}{a_1 e + a_2} = \frac{a_2}{(a_2/e_2)e + a_2} \\ &= \frac{1}{(1/e_2)(\frac{\pi}{4} b_1 x) + 1} = \frac{1}{b_2 x + 1},\end{aligned} \tag{16}$$

where $b_2 = \frac{\pi b 1}{4 e_2}$ and a_1 is substituted with a_2/e_2 as stated earlier.

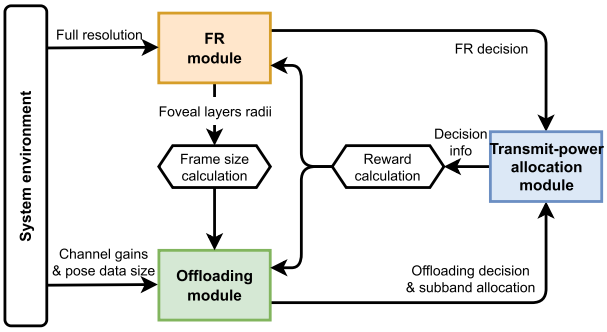


FIGURE 3. DRL-based solution framework.

The user-perceived VR quality Γ can now be obtained as the sum of the foveation-based cutoff frequencies over the image. For simplicity of exposition, we assume that the outermost layer, i.e., the Kth layer, is approximated as circular with a radius $R_K = \max\{N_W, N_H\}/2$. If we treat the discrete pixels as continuous and replace the sum with an integral, Eq. (12) becomes

$$\Gamma = \sum_{k=1}^{K} \int_{R_{k-1}}^{R_k} 2\pi x \Gamma_r(x) \Gamma_p(x)\, dx, \tag{17}$$

whose solution can be derived as

$$\begin{aligned}&= \frac{2\pi}{b_2} \sum_{k=1}^{K} \frac{\log r_k + a_0}{\log r_1 + a_0} ((R_k - R_{k-1}) \\ &\qquad - (\log(b_2 R_k + 1) - \log(b_2 R_{k-1} + 1))),\end{aligned}$$

from which it is observed that Γ is mainly a function of the resolutions r_k and radii R_k of the foveal layers. a_k in Eq. (2) is expressed in terms of R_k as

$$a_k = \frac{\pi(R_k^2 - R_{k-1}^2)}{N_{px}}. \tag{18}$$

V. EDGE VR OFFLOADING SOLUTION

To tackle the mixed-integer nonlinear optimization problem (MINLP) presented in Eq. (10), which involves the interaction between multiple tasks (FR adaptation, offloading decision, subband allocation and transmit power allocation), we propose a DRL-based VR offloading solution that integrates multiple cooperative RL agents.

A. OVERALL ARCHITECTURE

The proposed solution framework is depicted in Fig. 3. Problem (10) is decomposed into per-UE RL processes, with the foveal layer sizes and offloading decisions (along with subband allocation) as the user actions to be determined by the individual processes. The proposed solution framework uses two iterative per-UE modules running in sequence: an FR module and an offloading module. The FR module's role is to learn the optimal sizes of the foveal layers, while the offloading module is responsible for deciding the optimal offloading strategy along with subband allocation.

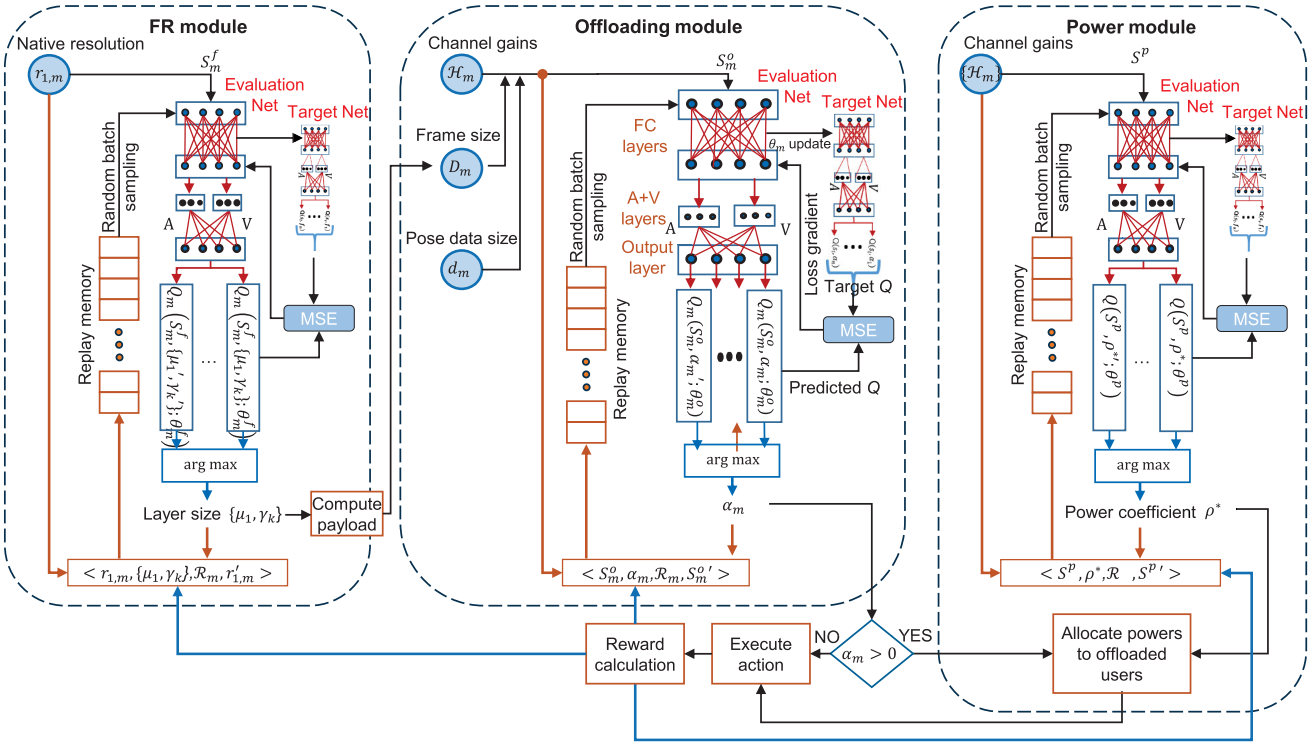


FIGURE 4. Network structures and data flow for training of the proposed solution framework using the double and dueling DQN approaches.

TABLE 3. Summary of the state space, action space and reward of the proposed solution.

Module	State space vector	Action space vector	Reward function
FR	$r_{1,m}$	$\{\mu_1, \gamma_k \mid k = 1, \ldots, K-1\}$	$\mathcal{R}_m(t)$ defined in Eq. (19)
Offloading	$\{\mathcal{H}_m, D_m, d_m\}$	$\alpha_m \in \{0, 1, 2, 3, ..., S\}$	
Power	$\{\mathcal{H}_m\}$	$\{\rho_j^*\} \in (0, 1]$	$\mathcal{R}(t)$ defined in Eq. (20)

The purpose of using two separate modules to determine these actions rather than using a single module to determine all of them is to reduce the action space size handled by each module. The FR module runs first and determines the foveal layer sizes as its output. Its results are subsequently used to calculate the payload size of a VR frame, which is combined with other input features to form the input state provided to the offloading module. The offloading decisions and subband allocation decisions made for all UEs by their individual offloading modules are then collected by a central transmit-power allocation module (referred to as the power module) to perform power allocation for the offloading UEs. Afterward, the reward is calculated and distributed to the UEs. The solution is obtained after the three modules are iteratively run until the system reaches convergence. In what follows, we describe the details of each module.

The state provided to the FR module of UE m comprises only the resolution of the first foveal layer in the VR stream, i.e., $r_{1,m}$. The action of the FR module constitutes a set of normalized radii $\mu_k = R_k / \min\{N_W, N_H\}$ for each foveal layer k, $k = 1, \ldots, K-1$. The remaining area outside of the $(K-1)$th layer is the Kth layer; thus, μ_K is excluded from the action. According to the definition of the foveal layers, the action of the FR module must satisfy $\mu_1 < \cdots < \mu_{K-1} \leq 1$. To ensure that the DRL agent outputs an action that satisfies these inequalities between the μ_k without additional processing, we introduce new variables γ_k ($0 \leq \gamma_k \leq 1$) such that $\mu_{k+1} = \gamma_k(1 - \mu_k)$. Consequently, the action associated with FR for UE m is given by $\{\mu_1, \gamma_k \mid k = 1, \ldots, K-1\}$.

On the other hand, the state information provided to the offloading module of UE m includes the payload data size D_m (in bits), the motion data size d_m (in bits) and a set of channel gains defined as $\mathcal{H}_m = \{g_{m,j} | j \in \mathcal{S}\}$, where $g_{m,j}$ is the channel gain of UE m in subband j. Thus, the state is given as $\{\mathcal{H}_m, D_m, d_m\}$. The action taken for each decision step in the offloading module is $\alpha_m \in \{0, 1, 2, 3, \ldots, S\}$, where S represents the number of subbands; $\alpha_m = 0$ means that viewport rendering for UE m is performed locally, while $\alpha_m > 0$ means that the viewport rendering task of UE m is offloaded to the edge using subband α_m. The set of subband allocations for all UEs, $\{\alpha_m \mid m \in \mathcal{M}\}$, is used for the power allocation process described below.

Once the UEs whose rendering tasks are to be offloaded have been determined along with their subband allocations,

this information is reported to the power module. The power module then utilizes this information to determine the NOMA power allocation using Algorithm 1. The state information provided to the power module includes a set of channel gains $\mathcal{H}_m$. At each decision step within the power module, the action taken is denoted as $\rho^* \in (0, 1]$, which represents the base power allocation coefficient (PwC) and is used in the subsequent power allocation process described below. We adopt the power allocation process of [56]. Let n_j represent the number of UEs allocated to subband j. Without loss of generality, we assume that for $\mathcal{N}_j = 1, 2, \ldots, n_j$, UE 1 has the weakest channel gain, while UE n_j has the strongest channel gain, i.e., $g_{1,j} \leq g_{2,j} \leq \cdots \leq g_{n_j,j}$. We introduce the variables $\rho_{1,j}, \rho_{2,j}, \ldots, \rho_{n_j,j}$ as the power allocation coefficients for the respective UEs. In this setup, UE m is assigned power in subband j in accordance with the formula $\rho_{m,j} \times P/S$. The power allocation must satisfy the following conditions [51]:

- $\rho_{1,j} \geq \rho_{2,j} \geq \cdots \geq \rho_{n_j,j}$,
- $\rho_{i,j} \geq \rho_{i+1,j} + \rho_{i+2,j} + \cdots + \rho_{n_j,j}, \quad i = \{1, 2, \ldots, n_j - 1\}$,
- $\rho_{1,j} + \rho_{2,j} + \cdots + \rho_{n_j,j} = 1$.

As outlined in Algorithm 1, we initiate the power allocation process by selecting the UE with the weakest channel gain and assigning it the base power coefficient ρ^* determined by the DRL agent of the power module, i.e., $\rho_{1,j} = \rho^*$. Subsequently, we generate a power allocation scheme that satisfies all the conditions mentioned above.

Algorithm 1 Power Allocation Module Process

```
p_{m,j}: Power assigned to UE m in subband j
ρ_{m,j}: Fractional power for UE m in subband j
for subband j ∈ S do
    The DRL agent deterimnes the base power coefficient ρ*
    Sort the UEs in N_j in ascending order of their channel gains
    Set ρ_{1,j} ← ρ*
    for UE m ∈ N_j do
        if UE m is the last then
            Assign ρ_{m,j} ← 1 − (ρ_{1,j} + ρ_{2,j} ... + ρ_{m−1,j})
        else
            Assign ρ_{m,j} = ρ_{1,j} × (1 − (ρ_{1,j} + ρ_{2,j} + ... + ρ_{m−1,j}))
        end if
    end for
    Set p_{m,j} ← ρ_{m,j} × P/S
end for
```

As illustrated in Fig. 3, the outcomes of the FR, offloading, and power allocation modules are integrated to evaluate the respective rewards for their actions, which are subsequently shared among the UE agents. To minimize the objective function in problem (10), we use the per-UE objective value to construct a reward expression for each UE agent within the FR and offloading modules. Meanwhile, we employ the total objective value to construct a reward expression for the agent operating within the power module. For UE m, the reward expression is given below:

$$\begin{aligned}\mathcal{R}_m(t) = -[&\omega(1-\alpha_m(t))(\phi T_{l,m}(t) + E_{l,m}(t)) \\ &+ \omega\alpha_m(t)(\phi T_{e,m}(t) + E_{e,m}(t)) + \Gamma_m(t)],\end{aligned} \tag{19}$$

and the total reward sum is given as

$$\mathcal{R}(t) = \sum_{m \in \mathcal{M}} \mathcal{R}_m(t). \tag{20}$$

With this reward structure, each agent is encouraged to find a good balance between overhead and visual quality. The agents are continuously trained to take actions that will maximize their future expected rewards as they utilize computation and spectrum resources more efficiently.

B. NEURAL NETWORK STRUCTURE AND TRAINING PROCEDURE

We present the overall network structures, employing a combination of the double DQN [57] and dueling DQN [36] approaches, as illustrated for UE m in Fig. 4. The associated training steps are also depicted in this figure. In this subsection, we primarily discuss the structure of the FR module, which shares an identical network structure with the offloading module, differing only in their input states and output actions. For brevity, we omit a detailed description of the offloading module here. We differentiate the components and parameters of the FR and offloading modules by using superscripts f and o, respectively.

Each module employs the double DQN approach, in which an agent maintains a pair of neural networks with similar structures— for the FR module of UE m specifically, an *evaluation network (vNet)* with parameters θ_m^f and a *target network (gNet)* with parameters ϑ_m^f. The agent's objective in the proposed solution framework is to determine the mapping from each state to a set of Q-values for all possible actions. The vNet approximates this mapping, based on which the agent chooses an action that produces a maximum Q-value for a given state in order to maximize its corresponding expected long-term reward. On the other hand, the gNet characterizes the desired Q-value, which is used to backpropagate through and train the vNet. The parameters of the vNet are updated to minimize the difference between the Q-values from the two networks. The FR module of UE m selects action $\{\mu_1, \gamma_k\}$ using vNet_m^f, the offloading module selects action α_m using vNet_m^o, and the power module selects action ρ^* using vNet^p, all based on an ϵ-*greedy* strategy whereby the action decision is made so as to obtain the largest Q-value of the corresponding vNet for a given state. The determined action is then reported to the central power allocation module.

Once an action decision is made by the vNet of each module and executed along with power allocation, the agent receives a reward and the next state, which serves as the input for the next decision step. Tuples of the forms $(r_{1,m}(t), \{\mu_1(t), \gamma_k(t)\}, \mathcal{R}_m(t), r_{1,m}(t+1))$, $(\{\mathcal{H}_m(t), D_m(t), d_m(t)\}, \alpha_m(t), \mathcal{R}_m(t), \{\mathcal{H}_m(t+1), D_m(t+1), d_m(t+1)\})$, and $(\{\mathcal{H}_m(t)\}, \rho^*(t), \mathcal{R}(t), \{\mathcal{H}_m(t+1)\})$ for

VR frame t are stored in the replay memories η_m^f, η_m^o, and η^p of the FR, offloading, and power modules, respectively. During the training episodes, batches of random experiences $\mathcal{E}_m^f$, $\mathcal{E}_m^o$, $\mathcal{E}^p$ are sampled from η_m^f for vNet$_m^f$, η_m^o for vNet$_m^o$, and η^p for vNet^p, respectively, and used to compute the corresponding target Q-values $Q_{m,n}^g$ using gNet$_m^f$, gNet$_m^o$, and gNet^p for each experience $n \in \mathcal{E}_m^f, \mathcal{E}_m^o, \mathcal{E}^p$. In each training step, the aim is to minimize the gap between the Q-values of the vNet and the target Q-values of the gNet. Thus, the loss function $L^f(\theta_m^f)$ of the FR module is defined as the difference between Q_m—the Q-value from the vNet—and $Q_{m,n}^g$—the target Q-value from the gNet—for the sampled batch of experiences:

$$L^f(\theta_m^f) = \frac{1}{|\mathcal{E}_m^f|} \sum_{n \in \mathcal{E}_m^f} \left(Q_m(r_{1,m}^{(n)}, \{\mu_1, \gamma_k\}^{(n)}; \theta_m^f) - Q_{m,n}^g \right)^2, \tag{21}$$

where $(r_{1,m}^{(n)}, \{\mu_1, \gamma_k\}^{(n)})$ is the state–action pair corresponding to experience n. $Q_{m,n}^g$ is obtained as

$$Q_{m,n}^g = \mathcal{R}_m^{(n)} + \gamma Q_m \left(r_{1,m}^{(n)+}, \{\mu_1, \gamma_k\}^{(n)+\star}; \vartheta_m^f \right), \tag{22}$$

where $\mathcal{R}_m^{(n)}$ is the per-UE reward for experience n, $r_{1,m}^{(n)+}$ is the next state, and $\{\mu_1, \gamma_k\}^{(n)+\star}$ corresponds to an action bearing the maximum Q-value for the next state under vNet$_m^f$:

$$\{\mu_1, \gamma_k\}^{(n)+\star} = \underset{\{\mu_1, \gamma_k\} \in \mathcal{A}^f}{\arg\max}\ Q_m(r_{1,m}^{(n)+}, \{\mu_1, \gamma_k\}; \theta_m^f), \tag{23}$$

where $\mathcal{A}^f$ is the action space of the FR module. We perform backpropagation on vNet$_m^f$ using gradient descent to minimize $L^f(\theta_m^f)$ and update θ_m^f. After training for a specified number of iterations, the parameters ϑ_m^f of the gNet are updated by replacing them with θ_m^f. This serves the purpose of keeping the parameters ϑ_m^f synchronized with those of vNet$_m^f$ to better approximate the expected reward when computing the targets. The loss functions for both the offloading and power modules are defined in a similar manner.

The structure of both the vNet and gNet is a dueling DQN consisting of serially linked fully connected (FC), advantage and value (A+V), and output layers. All the FR, offloading, and power modules share an identical network structure, and we describe the details based on the FR module case in the following. The input layer of each network provides an interface through which the state information is fed into the network to be passed to the first FC layer. The FC layers learn the mapping from the input states to the corresponding Q-values of the possible actions. Then, the Q-value from the FC layers is first decomposed into two parts:

$$Q_m(r_{1,m}, \{\mu_1, \gamma_k\}) = V_m(r_{1,m}) + A_m(r_{1,m}, \{\mu_1, \gamma_k\}), \tag{24}$$

where $V_m(r_{1,m})$ and $A_m(r_{1,m}, \{\mu_1, \gamma_k\})$ are the state and action-advantage values, respectively, for the FR module, and they are learned separately. This decoupling technique of dueling DQNs facilitates better approximation of the Q-values by allowing the expected returns resulting from a state and an action to be individually evaluated. The value stream passes through an FC network to estimate $V_m(r_{1,m})$, which allows the agent to discover the Q-value of state $r_{1,m}$ and the states for which it is not worthwhile to explore each action. The action-advantage stream passes through another FC network to estimate $A_m(r_{1,m}, \{\mu_1, \gamma_k\})$, through which the agent learns the advantage of executing an action $\{\mu_1, \gamma_k\}$ for FR in state $r_{1,m}$. With θ_m^f assumed to be the network parameter set for both streams, the action-advantage value for UE m in VR frame t is obtained as $A_m(r_{1,m}(t), \{\mu_1(t), \gamma_k(t)\}; \theta_m^f)$ for action $\{\mu_1(t), \gamma_k(t)\}$, and the corresponding state value is obtained as $V_m(r_{1,m}(t); \theta_m^f)$. Finally, the output layer aggregates the two streams into Q_m, which is computed relative to the average action-advantage value over all actions for the state under consideration:

$$\begin{aligned} &Q_m(r_{1,m}(t), \{\mu_1(t), \gamma_k(t)\}; \theta_m^f) \\ &= V_m(r_{1,m}(t); \theta_m^f) + [A_m(r_{1,m}(t), \{\mu_1(t), \gamma_k(t)\}; \theta_m^f) \\ &\quad - \frac{1}{|\mathcal{A}^f|} \sum\nolimits_{\{\mu_1, \gamma_k\} \in \mathcal{A}^f} A_m(r_{1,m}(t), \{\mu_1, \gamma_k\}; \theta_m^f)]. \end{aligned} \tag{25}$$

TABLE 4. Simulation parameters (default values are underlined).

Parameter	Value(s)
Number of UEs ($\lvert\mathcal{M}\rvert$)	[2, 4, <u>5</u>, 6, 10]
Number of subbands (S)	[1, 2, <u>3</u>, 4, 6]
Number of foveal layers (K)	2, 3
Native resolution (r_1)	$\{2160p, 1080p, 720p, 480p\}$
PixelDepth	8 bits
Edge computation speed (C_e)	3 Gcycles/sec
UE computation speed (C_l)	0.1, 0.2, <u>0.25</u>, 0.5, 1.0 M cycles/sec
Edge computation volume (Z_e)	2 kbits/cycle
UE computation volume (Z_l)	0.2 kbits/cycle
Edge encoding speed (V_e^{en})	3 Gbps
UE decoding speed (V_l^{de})	3 Gbps
Compression efficiency (ζ)	0.1
System DL bandwidth	8 MHz
System UL bandwidth	2 MHz
BS transmit power budget (P)	20 dBm
Noise power spectral density (N_0)	-174 dBm/Hz
Power allocation coefficient of the weakest-channel UE ($\rho_{1,j}$)	0.6
Communication power consumption (P_{snd}, P_{rcv})	1.0 W
Idle power consumption (P_{idle})	1 mW
Decoding power consumption (P_{dc})	0.1 W
Rendering power consumption (P_{rn})	0.5 W
Balance coefficient between overhead and quality terms (ω)	0.05, 0.1, <u>0.15</u>, 0.2, 0.5
Balance coefficient between energy and delay terms (ϕ)	1
Discount factor (γ)	0.9

VI. PERFORMANCE EVALUATION

In this section, we evaluate the reward, user-perceived quality, energy consumption and system delay performance of the proposed solution for varying system factors in comparison with other benchmarks and provide in-depth analyses.

TABLE 5. Specification of each module of the schemes evaluated in Section VI.

Figure	Scheme	FR	Offloading	Tx power
5	All	DRL	DRL	DRL
6, 7, 8, 9, 11	No FR	N/A	DRL	DRL
	DRL FR 3/2	DRL	DRL	DRL
	DRL FR-Local	DRL	non-DRL	N/A
	DRL FR-Edge	DRL	non-DRL	DRL
	Random FR	non-DRL	DRL	DRL
	Fixed FR-1/2/3	non-DRL	DRL	DRL
10	PwC = 0.5/0.75/0.9	DRL	DRL	non-DRL
	DRL Allocation	DRL	DRL	DRL
12	Exhaustive-NOMA/OMA	non-DRL	non-DRL	non-DRL
	DRL-NOMA/OMA	DRL	DRL	DRL

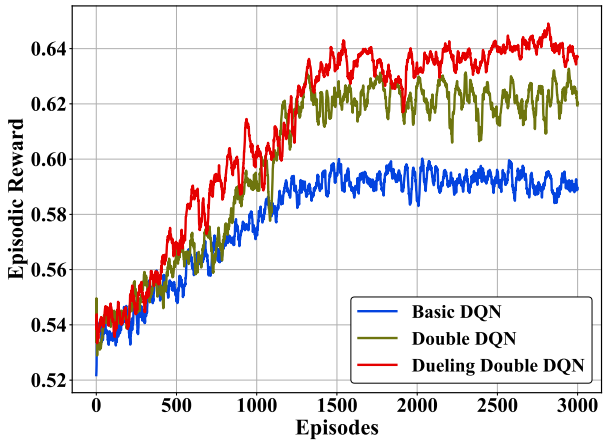


FIGURE 5. Comparison of reward evolution across episodes for various RL algorithms.

A. SIMULATION PARAMETERS AND ENVIRONMENT

The simulation parameters are listed in Table 4. In this table, the default value of a parameter is underlined if a set of values is considered for it.

We consider three foveal layers ($K = 3$), as suggested in [58]. Additionally, we investigate a two-layer scenario ($K = 2$) for the proposed solution to assess the impact of the number of layers on performance. We denote the three-layer and two-layer cases of the proposed solution by "DRL FR 3" and "DRL FR 2,", respectively, in the following result graphs. We set $0.05 \leq \mu_1 \leq 0.95$. For simplicity of simulation, discrete action spaces are assumed for μ_1 and a_1, with an increment of 0.01 between one action value and another. The full resolution (r_1) of each frame is randomly selected from the set of considered values. The scenario considered in most simulations, unless otherwise stated, consists of five UEs and three subbands. To configure the computation and power-related parameters, we refer to the works of [18], [37], [59], and [60].

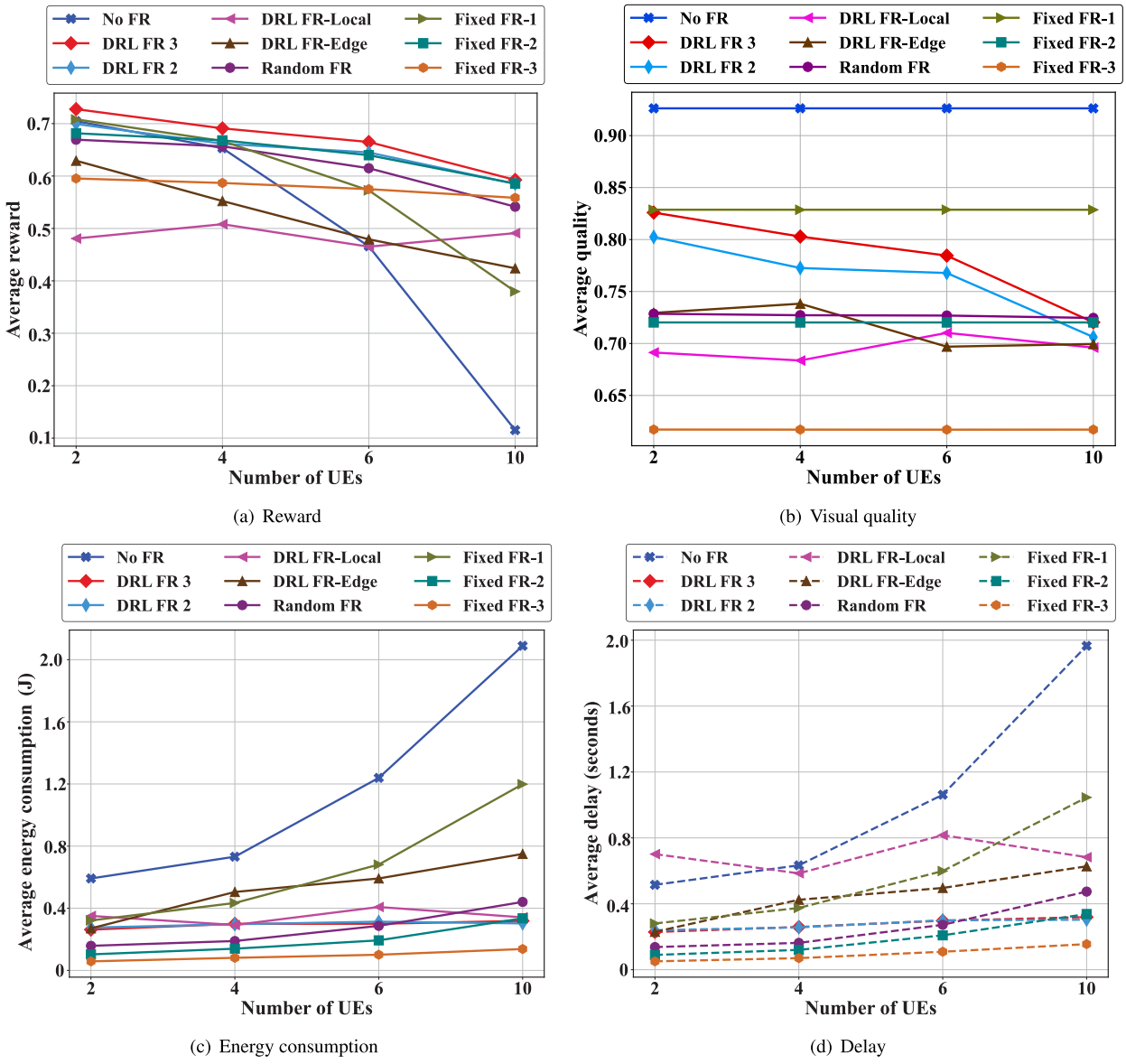
The average channel gain of each UE is given by the path loss model: $\overline{g}_{m,j} = A_d(\frac{3\times10^8}{4\pi F_c d_m})^{d_e}$, $\forall j \in \mathcal{S}$, with path loss exponent $d_e = 3$. $A_d = 3$ and $F_c = 915$ MHz denote the antenna gain and the carrier frequency, respectively, while d_m is the distance measured in meters between the BS and UE m. We distribute the UEs with $d_m = 120 + (m-1)i$, where i is an adjustable parameter determining the spacing between adjacent UEs, which is set to 15 meters in these simulations. The time-varying channel gain is generated from the Rician fading model with a factor of 0.3 and is assumed to be i.i.d. across time slots and subbands. The system bandwidths for DL and UL are 8 and 2 MHz, respectively. The subband bandwidth B_s is given as $\frac{W}{S}$, where W is the system bandwidth and S denotes the number of subbands. The noise power for a subband is $N_0 B_s$, where $N_0 = -174$ dBm/Hz is the noise power spectral density.

The proposed solution is implemented using the TensorFlow 1.14.0 platform with Python version 3.7.11. The hyperparameters of the neural networks are set following rigorous simulations performed with a range of values. The learning rate is set to 0.01, the batch size is 128, the memory size is 1024, and the discount factor is $\gamma = 0.9$. The epsilon value for exploration gradually decays in probability from 100% to 1%. Moreover, RMSProp is used as the optimizer, and the rectified linear unit (ReLU) function is used as the activation function for the two FC layers, which contain 20 and 15 neurons. The total number of episodes is 2000, and training is performed for the first 80% of the episodes. Every data point in the plots is an average of values from the last 20% of the episodes.

B. BENCHMARKS

Because no previous works in the literature have explored the offloading avenue in the context of FR-enabled VR, we evaluate the performance of our proposed DRL FR solution in comparison with the following schemes as benchmarks (the offloading and power allocation modules of the proposed solution are still used with the following benchmarks):

- *Fixed FR*: Constant values are maintained for the FR factors μ_1 and μ_2. Three pairs of (μ_1, μ_2) are considered: Fixed FR-1 with (0.49, 0.51), representing the case in which the sizes of the inner and middle layers are nearly the same; Fixed FR-2 with (0.25, 0.75), representing a moderately small inner layer and a moderately large middle layer; and Fixed FR-3 with (0.05, 0.95), representing a combination of a very small inner layer and a very large middle layer.



(a) Reward

(b) Visual quality

(c) Energy consumption

(d) Delay

FIGURE 6. Performance comparison of various schemes for FR adaptation and offloading with varying numbers of UEs.

- *Random FR*: Random FR factors are generated for the inner and middle layer sizes for each decision step in an episode.
- *No FR*: As a baseline, we also implement the No FR strategy, in which a traditional rendering technique is used to render the VR image uniformly at full resolution across all layers.

The action space constraints described earlier for the FR module are also applied to the FR factors in the benchmark schemes.

To evaluate the gains of adaptive offloading, we introduce the following non-DRL offloading strategies to serve as benchmarks for comparison (the FR module of the proposed solution is still used with the following benchmarks):

- *Edge rendering*: This approach involves offloading the VR computation tasks of $2S$ UEs in ascending order of their channel gains, i.e., $\alpha_m(t) > 0$ for these UEs, while the remaining UEs operate in local mode. When this strategy is integrated with the DRL-based FR module, we refer to it as "DRL FR-Edge."
- *Local rendering*: In this strategy, all UEs operate in local mode, meaning that $\alpha_m(t) = 0$ for all $m \in \mathcal{M}$ and $t \in \mathcal{T}$. Each UE acts independently and uses its own local resources to execute its VR computation tasks. When combined with the DRL-based FR module, this strategy is referred to as "DRL FR-Local."

Fig. 5 illustrates the training progress of our proposed solution framework under various DRL methods, including basic DQN and double DQN. The results demonstrate that all DRL methods achieve good convergence under similar parameter settings. Particularly, the dueling double DQN, as depicted in its network structure in Fig. 4, consistently

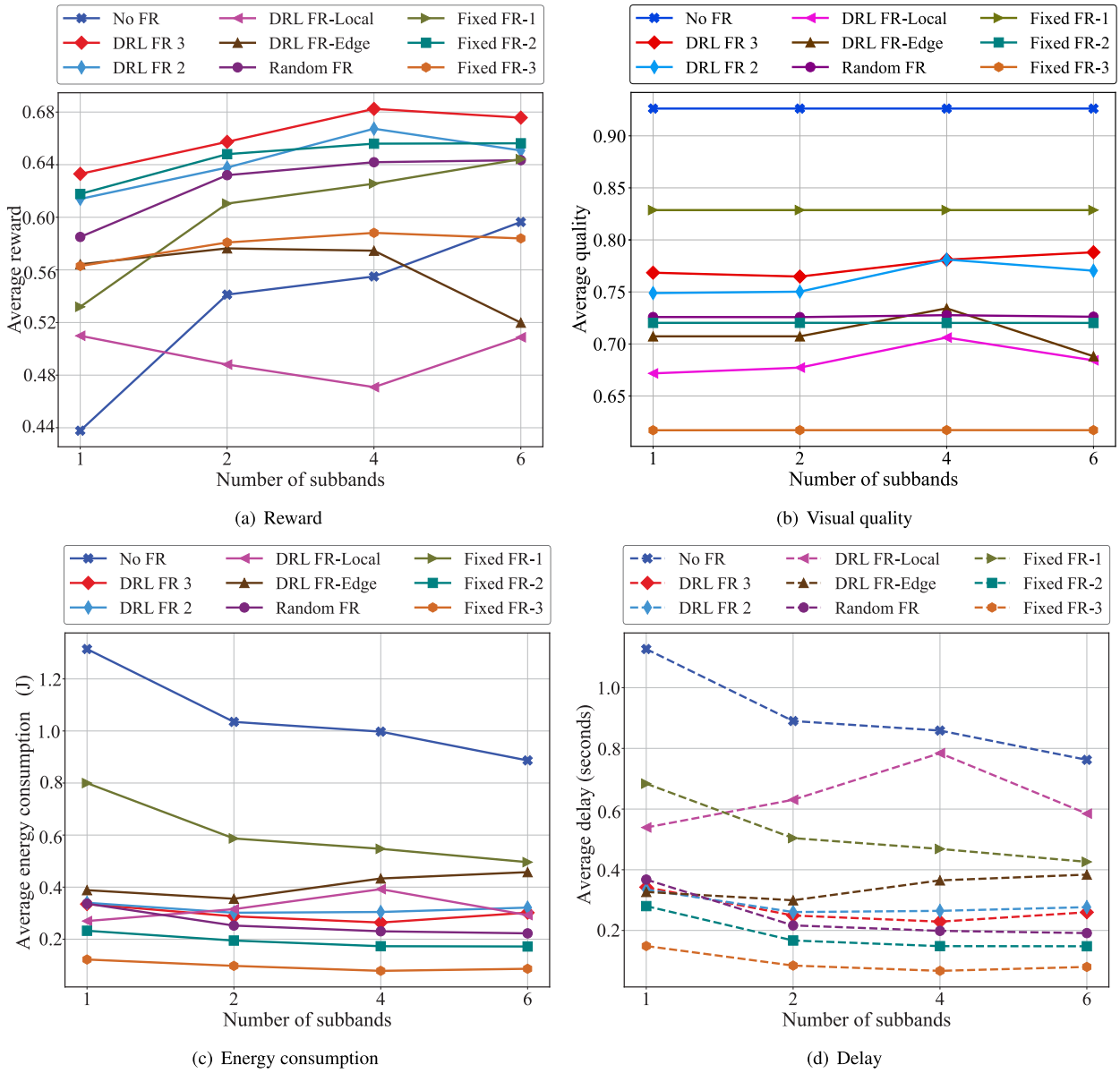


FIGURE 7. Performance comparison of various schemes for FR adaptation and offloading with varying numbers of subbands.

outperforms the other two methods throughout the training episodes, while the double DQN also shows significant improvement compared to the basic DQN. Thus, we employ the dueling double DQN for the proposed solution in the subsequent simulation.

C. IMPACT OF THE NUMBERS OF UEs AND SUBBANDS

The performance results for varying numbers of UEs are illustrated in Fig. 6. The DRL FR approach consistently demonstrates the best performance in terms of the total reward (objective). In Fig. 6(a), the FR schemes exhibit a notable advantage over the No FR scheme. However, the reward diminishes as the number of UEs increases because each UE receives a progressively smaller share of the subband resources on average, leading to higher energy consumption and delays. The gap between the DRL FR and No FR schemes is smallest when there are few UEs (e.g., 2 UEs) and becomes more pronounced with an increasing number of UEs. The rapidly decreasing reward of No FR with an increasing number of UEs is due to the high data volume it produces, as shown in Fig. 6(b). DRL FR consistently outperforms the other FR schemes across all numbers of UEs. While No FR may perform better than some fixed schemes when the number of UEs is small, all FR schemes achieve superior performance as the number of UEs increases. DRL FR 3 achieves a slightly higher reward than DRL FR 2, primarily due to the higher flexibility of its FR configuration. The gap between DRL FR and DRL FR-Edge widens as the number of UEs increases since fewer wireless resources (less bandwidth and power) are allocated

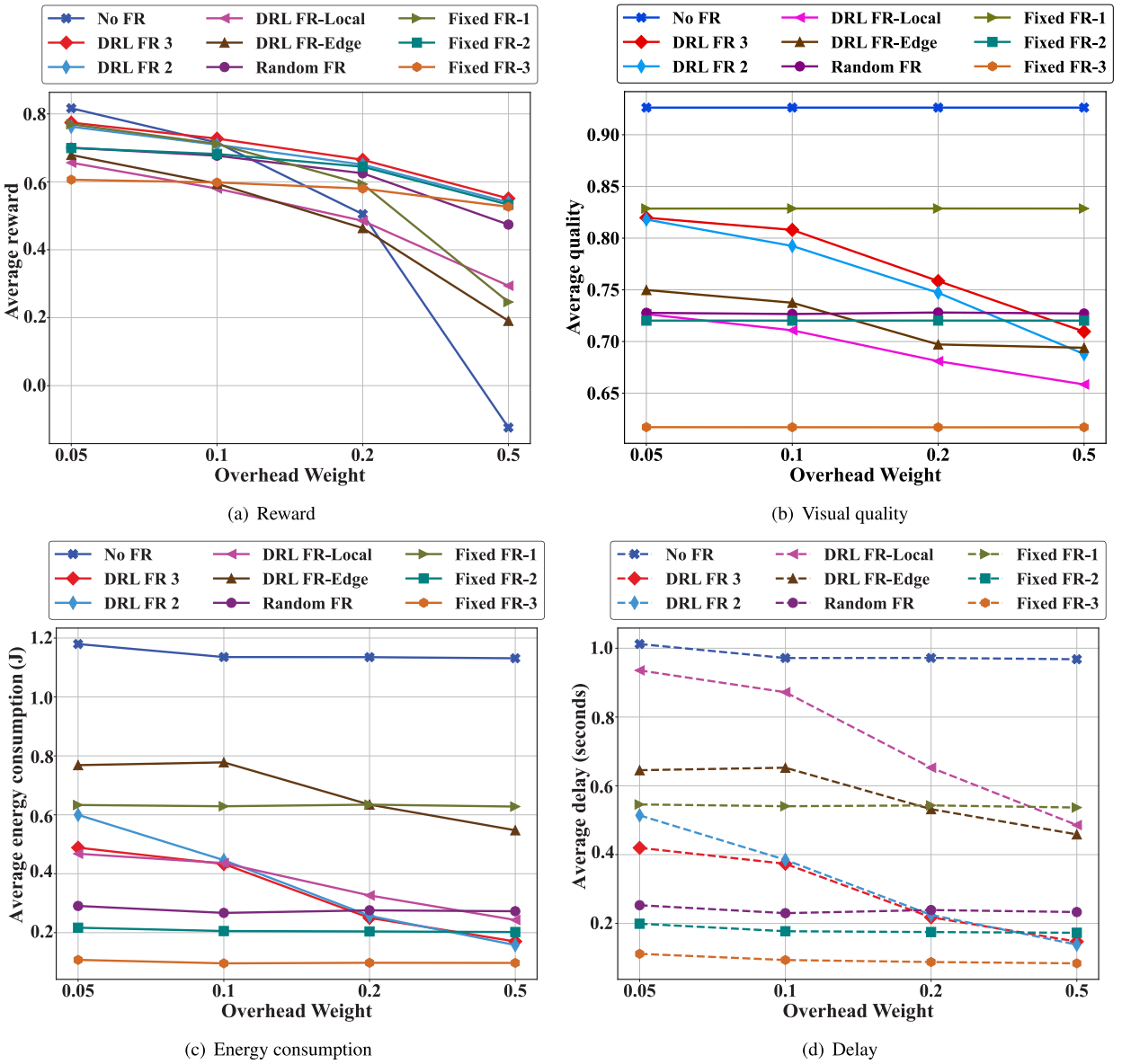
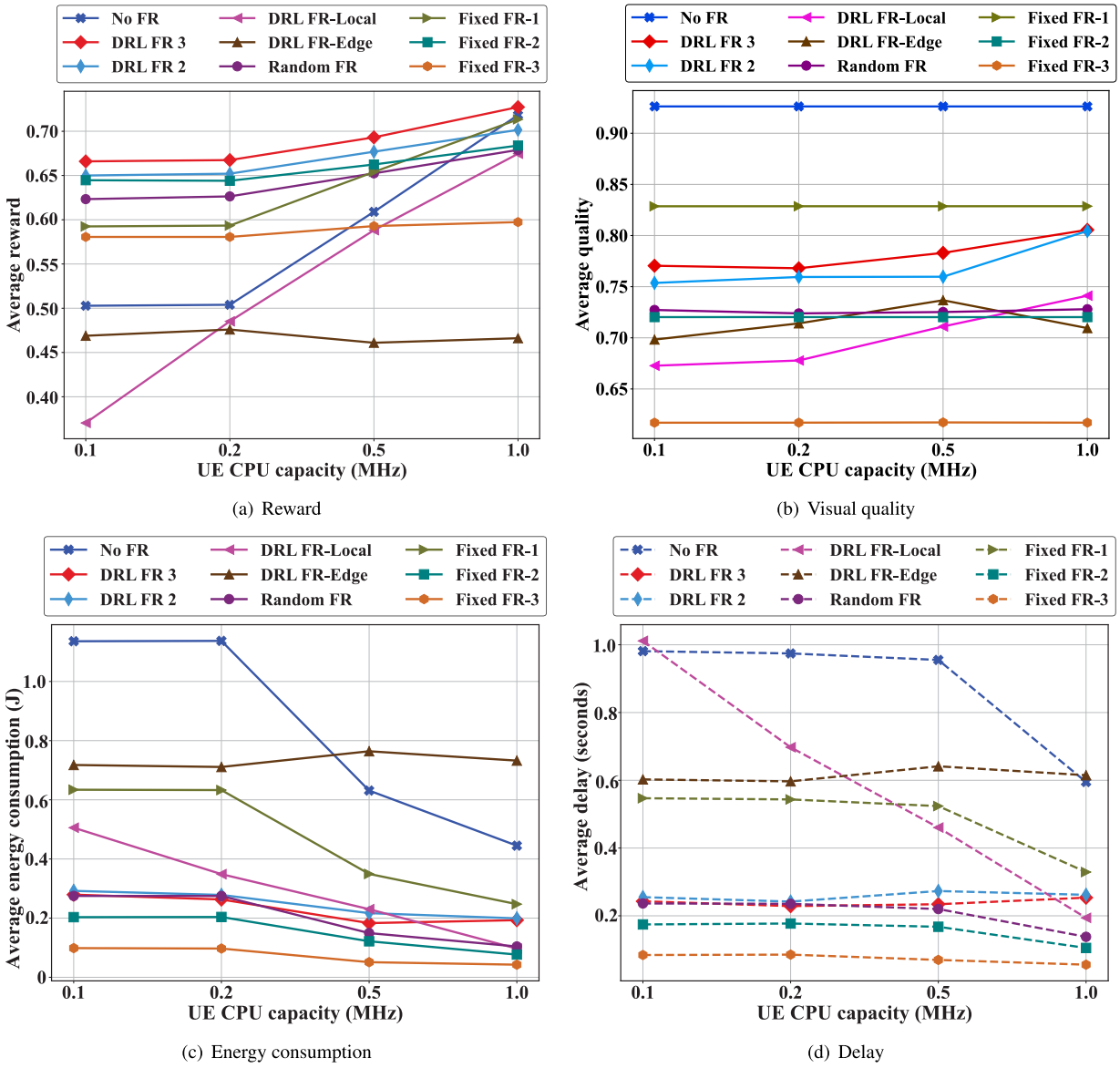


FIGURE 8. Performance comparison of various schemes for FR adaptation and offloading with varying balance coefficients ω.

per UE, resulting in rapidly increasing delays and energy consumption. Fig. 6(c) and (d) display the corresponding patterns of average energy consumption and latency, respectively, per VR frame. The energy consumption and delay under DRL FR remain relatively constant compared to those under most other schemes. This is because DRL FR strives to maintain both metrics at the expense of experiencing decreasing visual quality under the current setting of the balance weight between the overhead and quality terms. The DRL FR approach thus achieves an effective trade-off for the VR system with a varying number of UEs, enabling it to achieve the highest overall reward.

In Fig. 7, we present the overall performance of all schemes with an increasing number of subbands. The system reward for the FR schemes shows a consistent upward trend with a growing number of subbands, allowing more UEs to offload their VR computation tasks simultaneously. Notably, the performance gap between the best FR scheme (DRL FR) and the worst one (Fixed FR-3) ranges from 12.8% (with one subband) to 15.8% (with six subbands). However, this increase in the number of subbands introduces a trade-off. Because the spectrum is evenly distributed among all subbands, UEs that choose to offload experience a reduction in the available bandwidth, along with a reduction in the available power. Consequently, more UEs may opt for local operation, especially when the costs associated with offloading exceed the benefits. This trend is most apparent in DRL FR-Edge. As the number of subbands increases from four to six, the reward of DRL FR-Edge significantly decreases, becoming nearly equivalent to that



(a) Reward

(b) Visual quality

(c) Energy consumption

(d) Delay

FIGURE 9. Performance comparison of various schemes for FR adaptation and offloading under varying local CPU speeds.

of DRL FR-Local. This observation implies that offloading the tasks of all UEs may not always be beneficial in practical scenarios where wireless resources are limited.

D. PERFORMANCE VS. OVERHEAD COEFFICIENT

Fig. 8 shows the performance results for a range of values of the balance coefficient ω between the overhead and quality terms of the reward function. In Fig. 8(a), we observe that No FR achieves a slightly higher reward than DRL FR at $\omega = 0.05$. This can be attributed to the fact that for small ω values, visual quality takes precedence over the overhead terms, and the performance of DRL FR is constrained by the upper bound of the first layer in our simulation setup. As shown in Fig. 8(b), No FR always utilizes the highest quality setting, prioritizing visual quality above all else. However, as ω increases, DRL FR consistently outperforms the other schemes, and the performance of No FR progressively deteriorates. Fig. 8(b) shows the gradual quality adjustment of the DRL-based FR schemes, achieving a balance between visual quality and energy/delay overheads. When ω assigns a higher weight to quality ($\omega = 0.05$), these DRL-based schemes maximize visual quality. As the dominance of quality decreases (with increasing ω), they gradually sacrifice visual quality to minimize the overheads. This trade-off between visual quality and overhead is reflected in the energy consumption and delays, as demonstrated in Fig. 8(c) and (d). The DRL-based FR schemes exhibit high overheads at $\omega = 0.05$, while the overheads reach their lowest levels at $\omega = 0.5$. Consequently, the adaptive FR configuration outperforms fixed configurations, including No FR, for which the quality remains the same across all values of ω.

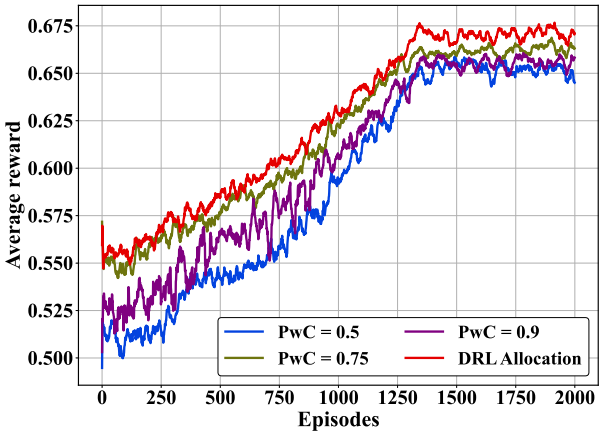


FIGURE 10. Comparison of reward evolution across episodes for various power allocation schemes.

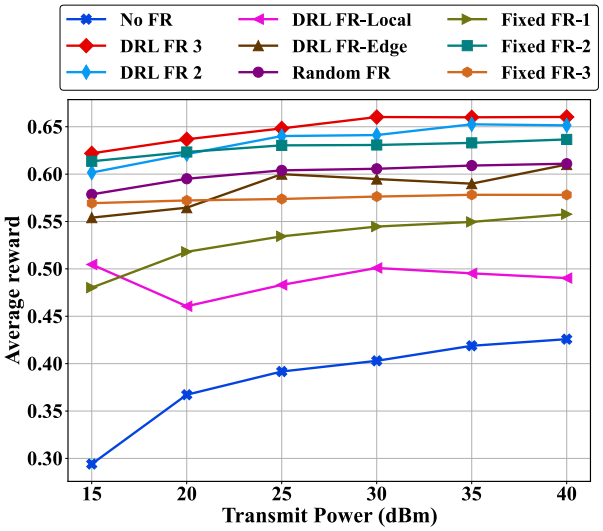


FIGURE 11. Comparison of average rewards for various schemes with varying BS transmit power budget.

E. PERFORMANCE VS. LOCAL CPU SPEED

In Fig. 9, we present the performance results for varying local CPU speeds of the UEs. As the UE CPU speed increases, the local rendering time (T_l^{rn}) decreases, as evident from Eq. (5). We assume that the power consumption for local rendering remains constant, which leads to a reduction in local energy consumption with increasing UE CPU speed, as shown in Eq. (6). Fig. 9(a) shows that DRL FR consistently achieves the highest reward among all the compared schemes. The DRL-based offloading schemes, including DRL FR, are generally not heavily affected by the changing UE CPU speed, as they optimally exploit offloading opportunities. However, the performance improvement of two schemes—No FR and DRL FR-Local—is notable. As shown in Fig. 9(c) and (d), they experience significant energy and delay overheads due to their dominant usage of the local mode. With increasing UE CPU speed, the energy and delay costs of the local mode significantly decrease, making local operation more attractive under such settings. Consequently, the rewards of both No FR and DRL FR-Local increase rapidly as the UE CPU speed rises. This indicates that higher UE CPU speeds, or, more accurately, higher UE CPU power efficiency, can enhance the benefits of local processing, making it an affordable option in certain scenarios.

F. PERFORMANCE IMPACT OF POWER ALLOCATION

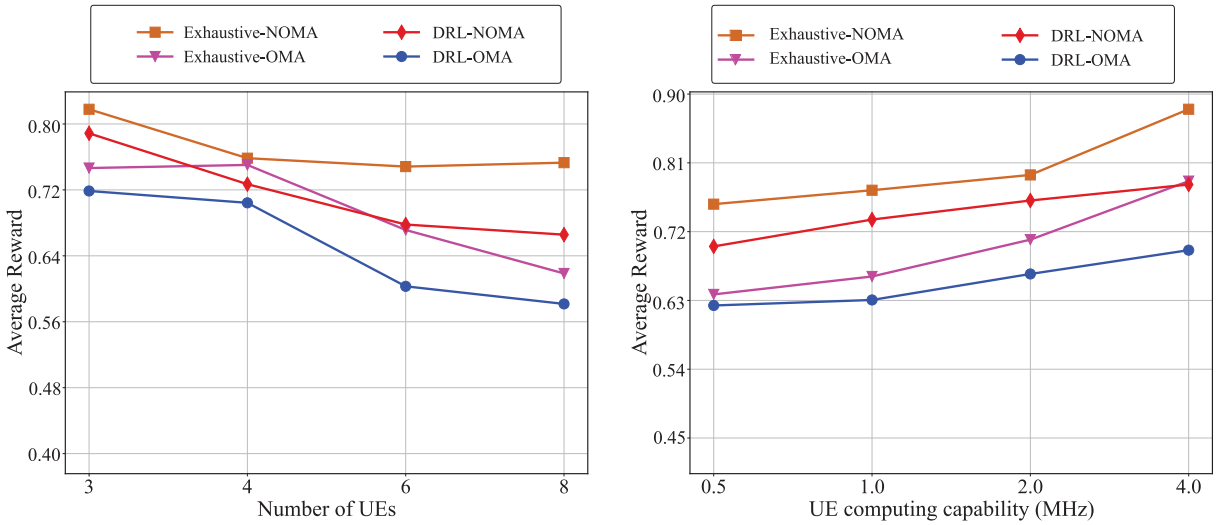
Fig. 10 presents a comparison between fixed power allocation under various fixed base power coefficients (PwC) and DRL-based power allocation (referred to as DRL Allocation). It is evident that low power coefficients (0.5), representing a fraction of the power budget allocated to the farthest UE and subsequent UEs, result in lower rewards. Conversely, excessively large coefficients (0.9), indicating an over-allocation of power to the farthest UE, are also less desirable. Intermediate coefficients (0.75) yield comparatively higher rewards. However, the DRL-determined power coefficient consistently achieves the best reward performance across training episodes, implying the importance of using the optimal base power coefficient.

Fig. 11 shows the performance of the schemes under varying transmit power budgets at the BS. In general, as more power budget is allocated to transmission, the average reward increases for all schemes, as it leads to an expansion of wireless link capacity. However, DRL FR-Local does not benefit from the increased power budget. Among all the schemes, DRL FR 3 consistently achieves the highest reward, with DRL FR 2 following closely across different power budgets.

G. PERFORMANCE IMPACT OF NOMA AND OMA

In Fig. 12, we present the performance comparison of different multiple access schemes, namely DRL-NOMA and DRL-OMA, alongside the exhaustive search-based schemes, Exhaustive-NOMA and Exhaustive-OMA. The Exhaustive scheme explores all possible subband allocations to UEs. At each decision step, the scheme computes the reward that each UE would receive when offloading via a particular channel or computing locally. After evaluating rewards for all subband (and local) allocations, each UE is assigned to the subband with the highest reward. Both Exhaustive-NOMA and Exhaustive-OMA provide optimal policies, serving as benchmarks to illustrate the upper bounds of achievable performance for their respective multi-access schemes.

In Fig. 12(a), the reward results are presented for an increasing number of UEs. This figure consistently demonstrates that DRL-NOMA outperforms DRL-OMA, highlighting the significance of NOMA in expanding wireless capacity and thereby reducing the constraints on offloading decisions imposed by capacity limitations. Fig. 12(b) also confirms this trend, with DRL-NOMA consistently outperforming DRL-OMA. DRL-NOMA even surpasses Exhaustive-OMA at several points, underscoring the importance of NOMA.



(a) number of UEs (b) varying capabilities computing

FIGURE 12. Comparison of average rewards for NOMA and OMA versus exhaustive search for various settings of UEs and computational speeds.

Furthermore, the figures indicate that DRL-NOMA achieves performance close to that of Exhaustive-NOMA, with over 88% of Exhaustive-NOMA's performance, thus efficiently optimizing edge VR operation. However, this result also implies that there is still room for further optimization.

VII. CONCLUSION

We have developed a DRL-based solution for an edge-assisted VR offloading system, addressing adaptive FR, offloading decisions, and wireless resource management. The main objective of our approach is to enhance the user-perceived visual quality of VR images while minimizing energy consumption and delay overheads. To achieve this, we derived a model of the user-perceived visual quality of a VR image rendered via FR and formulated an objective function by combining visual quality, energy consumption, and delay overhead terms. Our solution optimizes the foveal layer sizes, offloading decisions, and subband and power allocation for wireless links. Our solution framework consists of two per-UE modules, each dedicated to learning the optimal foveal layer sizes and offloading decisions along with subband allocation, complemented by a central power allocation module. Through rigorous evaluation, we demonstrated that our proposed DRL-based solution can effectively adapt to various environmental conditions, such as different numbers of UEs, different numbers of subbands, different values of the balance coefficient between the visual quality and overhead terms, and different UE processing capabilities. Notably, our solution consistently outperformed traditional rendering schemes and other established benchmarks.

For further research, we can consider including additional parameters related to FR optimization, such as the number of foveal layers and the per-layer resolutions. Additionally, the integration of eye gaze tracking, overfill rendering, and other VR content creation techniques can also be considered. Another important area for potential improvement involves designing a new DRL structure to enhance performance while reducing the computational load. Furthermore, investigating the impacts of diverse network systems and components on edge-assisted VR services and designing integrated VR systems are also worthwhile topics to explore. These topics encompass new multiple access schemes such as rate-splitting multiple access (RSMA), new spectrum resources such as the millimeter wave (mmWave) bands, and new nodes such as intelligent reflecting surfaces (IRSs).

REFERENCES

[1] S. M. LaValle, *Virtual Reality*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[2] A. Mehrabi, M. Siekkinen, T. Kämäräinen, and Y.-J. Antti, "Multi-tier CloudVR: Leveraging edge computing in remote rendered virtual reality," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 17, no. 2, pp. 1–24, May 2021.

[3] B. W. Nyamtiga, A. A. Hermawan, Y. F. Luckyarno, T.-W. Kim, D.-Y. Jung, J. S. Kwak, and J.-H. Yun, "Edge-computing-assisted virtual reality computation offloading: An empirical study," *IEEE Access*, 10, pp. 95892–95907, 2022.

[4] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing-A key technology towards 5G," *ETSI White Paper*, vol. 11, pp. 1–16, Dec. 2015.

[5] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, "Integrated blockchain and edge computing systems: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1508–1532, 2nd Quart., 2019.

[6] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Computation offloading and resource allocation in wireless cellular networks with mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 4924–4938, Aug. 2017.

[7] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular edge computing and networking: A survey," *Mobile Netw. Appl.*, vol. 26, no. 3, pp. 1145–1168, Jun. 2021.

[8] *Preparing for a Cloud AR/VR Future (White Paper)*, Huawei, Shenzhen, China, 2017.

[9] D. T. Tan, S. Kim, and J.-H. Yun, "Enhancement of motion feedback latency for wireless virtual reality in IEEE 802.11 WLANs," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[10] S. Kim and J.-H. Yun, "Motion-aware interplay between WiGig and WiFi for wireless virtual reality," *Sensors*, vol. 20, no. 23, p. 6782, Nov. 2020.

[11] Z. Wang, L. Lu, and A. C. Bovik, "Foveation scalable video coding with automatic fixation selection," *IEEE Trans. Image Process.*, vol. 12, no. 2, pp. 243–254, Feb. 2003.

[12] *Fixed Foveated Rendering (FFR)*. Oculus Documentation for Developers. Accessed: Feb. 1, 2024. [Online]. Available: https://developer.oculus.com/documentation/native/android/mobile-ffr/

[13] X. Yang, "Communication-constrained mobile edge computing systems for wireless virtual reality: Scheduling and tradeoff," *IEEE Access*, vol. 6, pp. 16665–16677, 2018.

[14] 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.

[15] P. Lin, Q. Song, D. Wang, F. R. Yu, L. Guo, and V. C. M. Leung, "Resource management for pervasive-edge-computing-assisted wireless VR streaming in industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7607–7617, Nov. 2021.

[16] M. Nduwayezu and J.-H. Yun, "Latency and energy aware rate maximization in MC-NOMA-based multi-access edge computing: A two-stage deep reinforcement learning approach," *Comput. Netw.*, vol. 207, Apr. 2022, Art. no. 108834.

[17] S. Wang, T. Lv, W. Ni, N. C. Beaulieu, and Y. J. Guo, "Joint resource management for MC-NOMA: A deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 20, no. 9, pp. 5672–5688, Sep. 2021.

[18] F. Guo, F. R. Yu, H. Zhang, H. Ji, V. C. M. Leung, and X. Li, "An adaptive wireless virtual reality framework in future wireless networks: A distributed learning approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8514–8528, Aug. 2020.

[19] B. Trinh and G.-M. Muntean, "A deep reinforcement learning-based offloading scheme for multi-access edge computing-supported eXtended reality systems," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 1254–1264, Jan. 2023.

[20] S. Gupta, J. Chakareski, and P. Popovski, "mmWave networking and edge computing for scalable 360° video multi-user virtual reality," *IEEE Trans. Image Process.*, vol. 32, pp. 377–391, 2023.

[21] L. Zhang and J. Chakareski, "UAV-assisted edge computing and streaming for wireless virtual reality: Analysis, algorithm design, and performance guarantees," *IEEE Trans. Veh. Technol.*, vol. 71, no. 3, pp. 3267–3275, Mar. 2022.

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

[23] D. Wang, B. Song, D. Chen, and X. Du, "Intelligent cognitive radio in 5G: AI-based hierarchical cognitive cellular networks," *IEEE Wireless Commun.*, vol. 26, no. 3, pp. 54–61, Jun. 2019.

[24] Y. Sun, "Adaptive learning-based task offloading for vehicular edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3061–3074, Apr. 2019.

[25] Z. Ning, P. Dong, X. Wang, M. S. Obaidat, X. Hu, L. Guo, Y. Guo, J. Huang, B. Hu, and Y. Li, "When deep reinforcement learning meets 5G-enabled vehicular networks: A distributed offloading framework for traffic big data," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1352–1361, Feb. 2020.

[26] M. Chen, W. Saad, and C. Yin, "Virtual reality over wireless networks: Quality-of-service model and learning-based resource management," *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5621–5635, Nov. 2018.

[27] Y. Wei, Z. Wang, D. Guo, and F. Richard Yu, "Deep Q-Learning based computation offloading strategy for mobile edge computing," *Comput., Mater. Continua*, vol. 59, no. 1, pp. 89–104, 2019.

[28] A. Rovira and M. Slater, "Reinforcement learning as a tool to make people move to a specific location in immersive virtual reality," *Int. J. Hum.-Comput. Stud.*, vol. 98, pp. 89–94, Feb. 2017.

[29] P. Lin, Q. Song, J. Song, A. Jamalipour, and F. R. Yu, "Cooperative caching and transmission in comp-integrated cellular networks using reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5508–5520, May 2020.

[30] J. Wang, L. Zhao, J. Liu, and N. Kato, "Smart resource allocation for mobile edge computing: A deep reinforcement learning approach," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 3, pp. 1529–1541, Jul. 2021.

[31] M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu, and W. Zhuang, "Learning-based computation offloading for IoT devices with energy harvesting," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1930–1941, Feb. 2019.

[32] N. Cheng, "Space/aerial-assisted computing offloading for IoT applications: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1117–1129, May 2019.

[33] X. Qiu, L. Liu, W. Chen, Z. Hong, and Z. Zheng, "Online deep reinforcement learning for computation offloading in blockchain-empowered mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8050–8062, Aug. 2019.

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

[35] H. Liu and W. Wu, "Two-stage deep reinforcement learning for inverter-based volt-VAR control in active distribution networks," *IEEE Trans. Smart Grid*, vol. 12, no. 3, pp. 2037–2047, May 2021.

[36] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1995–2003.

[37] M. Tang and V. W. S. Wong, "Deep reinforcement learning for task offloading in mobile edge computing systems," *IEEE Trans. Mobile Comput.*, vol. 21, no. 6, pp. 1985–1997, Jun. 2022.

[38] Y. Patel, "Optimizing market making using multi-agent reinforcement learning," 2018, *arXiv:1812.10252*.

[39] W. Liu, P. Si, E. Sun, M. Li, C. Fang, and Y. Zhang, "Green mobility management in UAV-assisted IoT based on dueling DQN," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2019, pp. 1–6.

[40] Q. Wu, S. Shi, Z. Wan, Q. Fan, P. Fan, and C. Zhang, "Towards V2I age-aware fairness access: A DQN based intelligent vehicular node training and test method," *Chin. J. Electron.*, vol. 32, no. 6, pp. 1230–1244, 2023.

[41] L. Qian, Y. Wu, F. Jiang, N. Yu, W. Lu, and B. Lin, "NOMA assisted multi-task multi-access mobile edge computing via deep reinforcement learning for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5688–5698, Aug. 2021.

[42] Y. Xu, B. Gu, R. Q. Hu, D. Li, and H. Zhang, "Joint computation offloading and radio resource allocation in MEC-based wireless-powered backscatter communication networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 6200–6205, Jun. 2021.

[43] Z. Ding, P. Fan, and H. V. Poor, "Impact of non-orthogonal multiple access on the offloading of mobile edge computing," *IEEE Trans. Commun.*, vol. 67, no. 1, pp. 375–390, Jan. 2019.

[44] A. Kiani and N. Ansari, "Edge computing aware NOMA for 5G networks," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1299–1306, Apr. 2018.

[45] Z. Yang, J. Hou, and M. Shikh-Bahaei, "Energy efficient resource allocation for mobile-edge computation networks with NOMA," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2018, pp. 1–7.

[46] W. Wu, F. Zhou, R. Q. Hu, and B. Wang, "Energy-efficient resource allocation for secure NOMA-enabled mobile edge computing networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 493–505, Jan. 2020.

[47] M. Nduwayezu, Q.-V. Pham, and W.-J. Hwang, "Online computation offloading in NOMA-based multi-access edge computing: A deep reinforcement learning approach," *IEEE Access*, vol. 8, pp. 99098–99109, 2020.

[48] F. Guo, H. Lu, B. Li, D. Li, and C. W. Chen, "NOMA-assisted multi-MEC offloading for IoVT networks," *IEEE Wireless Commun.*, vol. 28, no. 4, pp. 26–33, Aug. 2021.

[49] NVIDIA-Corporation. *VRWorks-Variable Rate Shading*. Accessed: Feb. 1, 2024. [Online]. Available: https://developer.nvidia.com/vrworks/graphics/variablerateshading

[50] J. Carrera Iseland and L. Grolleman, "Evaluation of performance on variable rate shading," M.S. thesis, Dept. Comput. Sci., Blekinge Inst. Technol., Karlskrona, Sweden, 2021.

[51] R. C. Kizilirmak and H. K. Bizaki, "Non-orthogonal multiple access (NOMA) for 5G networks," *Towards 5G Wireless Networks-A Phys. Layer Perspective*, vol. 83, pp. 83–98, 2016.

[52] A. Haddad, D. Slimani, A. Nafkha, and F. Bader, "Users' power multiplexing limitations in NOMA system over Gaussian channel," in *Proc. 8th Int. Conf. Wireless Netw. Mobile Commun. (WINCOM)*, Oct. 2020, pp. 1–7.

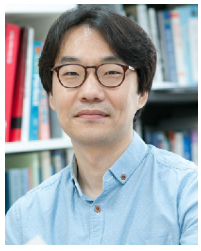
[53] W. Zou, J. Song, and F. Yang, "Perceived image quality on mobile phones with different screen resolution," *Mobile Inf. Syst.*, vol. 2016, pp. 1–17, Mar. 2016.
[54] W. S. Geisler and J. S. Perry, "Real-time foveated multiresolution system for low-bandwidth video communication," *Proc. SPIE*, vol. 3299, pp. 294–305, Jul. 1998.
[55] S. Rajan, S. Wang, R. Inkol, and A. Joyal, "Efficient approximations for the arctangent function," *IEEE Signal Process. Mag.*, vol. 23, no. 3, pp. 108–111, May 2006.
[56] Z. Yang, Z. Ding, P. Fan, and N. Al-Dhahir, "The impact of power allocation on cooperative non-orthogonal multiple access networks with SWIPT," *IEEE Trans. Wireless Commun.*, vol. 16, no. 7, pp. 4332–4343, Jul. 2017.
[57] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. AAAI Conf. on Artif. Intell.*, vol. 30, no. 1, 2016, pp. 2094–2100.
[58] B. Guenter, M. Finch, S. Drucker, D. Tan, and J. Snyder, "Foveated 3D graphics," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 1–10, Nov. 2012.
[59] J. Du, F. R. Yu, G. Lu, J. Wang, J. Jiang, and X. Chu, "MEC-assisted immersive VR video streaming over terahertz wireless networks: A deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9517–9529, Oct. 2020.
[60] 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.

BARAKA WILLIAM NYAMTIGA received the B.S. degree in computer science from the University of Dar es Salaam, Tanzania, and the M.S. degree in information and communication science and engineering from The Nelson Mandela African Institution of Science and Technology, Arusha, Tanzania. He is currently pursuing the Ph.D. degree in electrical and information engineering with the Seoul National University of Science and Technology, Seoul, South Korea. His current research interest includes virtual reality offloading.

DEREK KWAKU POBI ASIEDU (Member, IEEE) received the B.S. degree in biomedical engineering from the University of Ghana, Accra, Ghana, in 2011, and the M.S. and Ph.D. degrees in electronic engineering from Hanbat National University, Daejeon, South Korea. He was a Postdoctoral Fellow with the Wireless Intelligent Systems Laboratory, Hanbat National University, from September 2019 to December 2020. Afterwards, he was a Research Assistant Professor with the Department of Electrical and Information Engineering, Seoul National University of Science and Technology, Seoul, South Korea, from May 2021 to January 2023. He is currently a Research and Development Engineer with the Department of Mathematics and Electrical Engineering, Institut Mines-Télécom (IMT) Atlantique, Brest, France. His research interests include machine learning-based and model-based applications in next-generation wireless communication, radio frequency wireless information and power transfer communication, and symbiotic cognitive radio in the Internet of Things networks.

AIRLANGGA ADI HERMAWAN received the B.S. degree in electrical engineering from Gadjah Mada University, Yogyakarta, Indonesia, and the M.S. degree in computer science and engineering from the Eindhoven University of Technology, Eindhoven, The Netherlands. He is currently pursuing the Ph.D. degree in electrical and information engineering with the Seoul National University of Science and Technology, Seoul, South Korea. His current research interest includes edge computing-assisted virtual reality systems.

YAKUB FAHIM LUCKYARNO received the B.S. degree in engineering physics from Gadjah Mada University, Yogyakarta, Indonesia, and the M.S. degree in computer engineering with the King Mongkut's Institute of Technology Ladkrabang, Bangkok. He is currently pursuing the Ph.D. degree in electrical and information engineering with the Seoul National University of Science and Technology, Seoul, South Korea. His current research interest includes edge computing-assisted virtual reality systems.



JI-HOON YUN (Senior Member, IEEE) received the B.S. degree in electrical engineering and the M.S. and Ph.D. degrees in electrical engineering and computer science from Seoul National University (SNU), Seoul, South Korea, in 2000, 2002, and 2007, respectively.

He was a Senior Engineer with the Telecommunication Systems Division, Samsung Electronics, Suwon, South Korea, from 2007 to 2009, and a Postdoctoral Researcher with the Real-Time Computing Laboratory, University of Michigan, Ann Arbor, MI, USA, in 2010. He is currently a Professor with the Department of Electrical and Information Engineering, Seoul National University of Science and Technology (SeoulTech), Seoul. Before joining SeoulTech, in 2012, he was working with the Department of Computer Software Engineering, Kumoh National Institute of Technology, as an Assistant Professor. His current research interests include wireless multimedia systems and mobile applications.