IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 January 2024, accepted 4 March 2024, date of publication 7 March 2024, date of current version 13 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3375113

RESEARCH ARTICLE

Joint Content Caching, Recommendation, and Transmission for Layered Scalable Videos Over Dynamic Cellular Networks: A Dueling Deep Q-Learning Approach

JUNFENG XIE[1], (Member, IEEE), QINGMIN JIA[2], XINHANG MU[1], AND FENGLIANG LU[1]
[1]School of Information and Communication Engineering, North University of China, Taiyuan 030051, China
[2]Purple Mountain Laboratories, Nanjing 211111, China

Corresponding author: Junfeng Xie (xiejunfeng@nuc.edu.cn)

This work was supported in part by the Research Project Supported by Shanxi Scholarship Council of China under Grant 2022-147, in part by the Research Project Supported by Fundamental Research Program of Shanxi Province under Grant 202203021212151 and Grant 202203021221117, and in part by the Research Project Supported by the National Natural Science Foundation of China under Grant 92367104 and Grant 92267301.

ABSTRACT Scalable Video Coding (SVC) and edge caching are two techniques that hold the potential to improve user-perceived video viewing experience. Moreover, video recommendation can further enhance the caching gain by reshaping users' video preferences. In this paper, we investigate the video caching, recommendation and transmission for layered SVC streaming in cache-enabled cellular networks. Considering the dynamic characteristics of video popularity distribution and wireless network environment, to improve energy efficiency by minimizing system energy consumption and ensure the average user preference deviation tolerance, we begin by formulating a long-term optimization problem that focuses on video caching, recommendation and user association (UA). The problem is then transformed into a Markov decision process (MDP), which is solved by designing a dueling deep Q-learning network (DDQN)-based algorithm. Using this algorithm, we can obtain the optimal video caching, recommendation and UA solutions. Since the action space of the MDP is huge, to cope with the ''curse of dimensionality'', linear approximation is integrated into the designed algorithm. Finally, the proposed algorithm's convergence and effectiveness in reducing long-term system energy consumption are demonstrated through extensive simulations.

INDEX TERMS Scalable video coding, edge caching, recommendation, user association, energy efficiency, dueling deep Q-learning.

I. INTRODUCTION

In recent years, driven by mobile network technologies' rapid advance and smart devices' popularization, the mobile data traffic is experiencing an explosive growth. According to the report from Ericsson [1], the total global mobile data traffic will reach 325EB every month in 2028, which is nearly four times of 2022. Meanwhile, the unprecedented increase of multimedia applications results in video service being one of the most popular services. It is estimated that video traffic will account for 80 percent of all mobile data traffic in 2028 [1]. Thus, enhancing user-perceived video viewing experience in cellular networks becomes very important.

Edge caching [2], [3], [4] and Scalable Video Coding (SVC) are two techniques that hold the potential to improve user-perceived video viewing experience. Edge caching brings videos much closer to users by caching a part of high-popular videos at the mobile network edge (e.g., base station (BS)) in advance during off-peak hours. Once the video requested by a mobile user has already been cached, it will be transmitted to the user directly, thereby reducing the end-to-end content delivery latency, mitigating duplicate

The associate editor coordinating the review of this manuscript and approving it for publication was Majed Haddad.

content transmissions, saving backhaul resources, alleviating network congestions, as well as improving mobile users' quality of experience (QoE).

Considering the heterogeneity of mobile users' devices and the time-varying wireless channel conditions, SVC [5], [6], [7] has been proposed. The main idea of SVC is to encode a video into one base layer (BL) and multiple enhancement layers (ELs). The BL can realize the basic quality version, the BL combined with ELs can realize high quality versions. The more ELs, the higher quality version is realized. Different quality versions of a video have different bitrates and resolutions. In this way, layered SVC streaming is able to adaptively adjust the bitrates of videos transmitted to mobile users according to their preferences and wireless channel conditions. For example, when a mobile user's device is highly capable and the wireless channel condition is good, he can receive high quality videos, whereas the mobile user can receive low quality videos when the wireless channel condition is poor.

Based on the above analysis, combining edge caching and SVC can improve user-perceived video viewing experience. However, it is particularly challenging due to the following two major reasons. First, with SVC, not only different videos, but also different layers of the same video will compete for edge nodes' limited caching capacity. Thus, the layer-based caching decision should be considered. Second, when a quality version of a video is requested by a mobile user, all layers of the video related to the requested quality version need to be transmitted to the mobile user. Thus, the relationship between different layers should be considered.

In general, video delivery in cache-enabled cellular networks consists of video caching and video transmission. For video caching, BSs or other mobile edge nodes prefetch high-popular videos and cache them in advance. Considering the conflict between mobile edge nodes' limited caching capacity and the massive number of videos, mobile edge nodes can only cache a part of videos. Therefore, in order to satisfy as many users' requests as possible, it is a critical problem to make caching policy decision to select the proper videos for each mobile edge node to cache. For video transmission, considering the densification trend of radio access networks (RANs), it is high-probability for mobile users to be covered by multiple edge nodes [8], [9]. Therefore, it is a critical problem to make user association (UA) strategy decision to obtain the appropriate association relationship between mobile users and edge nodes. The UA strategy in cache-enabled cellular networks should consider not only the wireless channel conditions, but also the mobile users' requirements and edge nodes' cache status. For example, to reduce the content delivery latency, the mobile user may prefer to associate with the edge node that has cached the requested video instead of the edge node with the best wireless channel condition. Apparently, caching policy and UA strategy are naturally coupled.

In order to further enhance the caching gain, video recommendation [10], [11], [12] is considered as an effective approach by reshaping users' video preferences. In general, different users' video popularity distribution (VPD), i.e., probability distribution of requesting video contents, is different. To weaken the impact of the heterogeneity of users' video preferences and make users' VPD less heterogeneous, recommending carefully selected video contents to users is proposed, which is the basic idea of video recommendation. Apparently, caching policy and recommendation mechanism are naturally coupled. Recommendation mechanism has a direct influence on users' VPD, which further affects caching policy.

In cache-enabled cellular networks, caching policy and recommendation mechanism strongly depend on the VPD, while UA strategy is largely affected by the wireless channel state information (WCSI). Considering the dynamic time-varying characteristics of VPD and WCSI, as well as the strong coupled relationship between the video caching, recommendation and transmission, the caching policy, recommendation mechanism and UA strategy should be optimized jointly. As a result, it is of great significance to design efficient caching policy, recommendation mechanism and UA strategy for layered SVC streaming in dynamic cache-enabled cellular networks.

A. RELATED WORKS

Due to the potential to enhance video delivery efficiency and mobile users' QoE, layered SVC streaming has been widely investigated. Literature [13] gives a detailed survey on the LCEVC technology. In [14], the authors propose an adaptive policy iteration algorithm to improve the QoS in wireless scalable video multicast. The authors in [15] optimize the bit allocation for images/videos' scalable codec. In [16], a low-complexity SICO scheme is proposed for AVS3. The authors in [17] propose a SS-CVS framework with hierarchical subspace learning to improve video transmission in heterogeneous networks. Literature [18] optimizes the devices' downloading and sharing activities to enhance users' QoE. However, these works rarely took into account the video caching for layered SVC streaming.

There have been studies focusing on cache-enabled layered SVC streaming. Considering the relationship between different layers, literature [19] proposes a heuristic caching placement solution for layered SVC streaming aiming at minimizing the average download time. In [20], the authors propose a FPTA per-video-layer caching algorithm to minimize the aggregate video delivery delay and design an approximation algorithm based on a cache-partition technique to solve the cooperative caching problem among multiple network operators. The authors in [21] optimize the random caching strategy in a wireless video broadcasting system by using a gradient-based iterative algorithm to maximize the successful transmission probability. In [22], to improve the system capacity, transmission scheduling and rate allocation are optimized jointly. The authors in [23] propose a Lagrangian dual pricing algorithm to optimize the caching placement, video quality decision and UA

jointly. Literature [24] investigates the secure edge caching problem and exploits the distributed alternating direction method of multipliers (ADMM) to achieve the optimal edge caching strategy. The caching placement policy and the video transmission scheme are optimized in [25]. However, these works only focused on utilizing edge nodes' limited caching capacity to cache different layers of videos, whereas the video recommendation was largely ignored.

Some research efforts have been conducted to consider both the video caching and recommendation. In order to maximize the total cache hit ratio, the authors in [26] present a heuristic scheme to optimize the UA, content caching and recommendation jointly. In [27], the authors propose ε-greedy algorithm to learn a user-specific threshold, which is used to control the impact of recommendation. Then content caching and recommendation are optimized jointly to improve the successful offloading probability. Literature [28] proposes a heuristic algorithm to maximize the cache hit ratio under the user preference distortion constraint. In order to balance QoE and delivery rate, a heuristic algorithm called GPA is presented in [29] to group users and files, based on which the set of recommended files is optimized. Another notable work is [30], which proposes a scheme called GRACE to balance the average hit ratio and the peak rate by optimizing user grouping and content recommendation jointly. In [31], the content transmission latency in Fog-RANs is minimized by optimizing caching, recommendation and beamforming jointly. The authors in [32] focus on the cache-enabled mobile social networks and investigate the optimal caching placement based on three recommendation operations aiming at maximizing the traffic offloading ratio. However, these works studied in static scenarios, whereas the dynamic characteristics of the system states in practical scenarios were largely ignored. Considering the time-varying VPD and WCSI, the optimal system performance at a certain time slot cannot guarantee the optimal system performance over a long time period.

There have been studies that specifically focus on video caching and transmission in dynamic scenarios. In [33], the cache hit rate is maximized by utilizing a DQN-based content caching algorithm. Literature [34] proposes a learning-based algorithm to predict the future content popularity and optimize the edge caching policy. The authors in [35] focus on the two time-scale caching placement and UA problem, and propose a BP-based UA algorithm and DDPG-based caching placement algorithm. Literature [36] uses Q-learning algorithm to optimize caching placement and resource allocation. Literature [37] adopts the Stackelberg game to optimize UA, power allocation of non-orthogonal multiple access (NOMA), unmanned aerial vehicle (UAV) deployment and caching placement to minimize the content delivery delay. In [38], the authors improve the content caching and sharing of D2D networks by a CAQL-based caching placement algorithm. Taking into account Coordinated MultiPoint (CoMP) joint transmission technique, a reinforcement learning (RL)-based algorithm is presented in [39] to maximize the delay reduction. However, these works just considered multiple single videos, whereas the video contents with multiple layers and the relationship between different layers were not taken into account.

Recently, as artificial intelligence (AI) and machine learning algorithms continue to advance rapidly, many researchers have studied on the integration of intelligence technology and wireless communication system optimization [40]. RL-based algorithms, as one critical category of AI algorithms, have been widely used in many domains, such as blockchain, edge caching, computation offloading and resource allocation. Considering the immersive VR video services, literature [41] provides an asynchronous advantage actor-critic (A3C)-based algorithm to minimize the long-term energy consumption of Terahertz wireless networks. In [42], the authors propose an A3C-based algorithm to maximize the computation rate and the transaction throughput of blockchain-enabled Mobile Edge Computing (MEC) systems. Another notable work is [43], which proposes a RL-based energy-aware resource management scheme for wireless VR streaming in industrial Internet of Things (IIoTs). In [44], quantum collective learning and many-to-many matching game are adopted to solve the spectrum resource allocation problem and the distributed vehicles selection problem respectively. The authors in [45] aim at obtaining the optimal intelligence sharing policy by a collective deep reinforcement learning algorithm. Literature [46] improves the scalability of a service-oriented blockchain system by considering consensus protocols selection, block producers selection and network bandwidth allocation jointly.

B. MOTIVATION AND CONTRIBUTION

As discussed above, most of the existing works that focused on the video caching, recommendation and transmission optimization problem rarely took into account the time-varying VPD and WCSI. Some research contributions considered the dynamic scenarios, but they just considered multiple single videos and ignored the video contents with multiple layers. In addition, they rarely considered the video recommendation. To fulfill this gap, this article focuses on optimizing video caching, recommendation and UA for layered SVC streaming in dynamic cache-enabled cellular networks with time-varying VPD and WCSI. Due to the surging energy cost of information industry, developing green communication becomes very urgent and important, which makes the energy efficiency a key performance indicator in current 5G and future 6G networks [47], [48], [49]. Thus, we adopt energy efficiency as the performance metric and aim at minimizing the long-term system energy consumption while ensuring the average user preference deviation tolerance. Dueling deep Q-learning network (DDQN) is proposed to solve the problem. More specifically, the main contributions of this article are summarized as follows:

- We focus on the content caching, recommendation and transmission of layered SVC streaming in cache-enabled cellular networks. Considering the time-varying VPD and

WCSI, a joint video caching, recommendation and UA optimization problem is formulated to enhance the energy efficiency by minimizing the long-term system energy consumption while ensuring the average user preference deviation tolerance. The system energy consumption is composed of video transmission energy consumption and caching energy consumption.

- The formulated problem is then transformed into a discrete Markov decision process (MDP), which is solved by designing a DDQN-based algorithm. Using this algorithm, we can obtain the optimal video caching, recommendation and UA solutions. Considering the large state space and action space of the MDP, to cope with the "curse of dimensionality", linear approximation is integrated into the designed algorithm.
- Finally, during the simulations, the proposed algorithm's convergence is evaluated and its effectiveness is verified. The results show that the proposed algorithm outperforms benchmark algorithms in terms of energy efficiency and system energy consumption reduction over a long period.

C. ORGANIZATION

The remainder of this paper is organized as follows. Section II introduces the system model and formulates the optimization problem for video caching, recommendation and UA. In Section III, we propose a DDQN-based algorithm to solve the problem. The simulation settings, results, analysis and discussions are presented in Section IV. Finally, we conclude our work in Section V.

II. SYSTEM DESCRIPTION

In this section, we first present the system model, including network model, video request and caching model, recommendation model, transmission model and energy consumption model. Then the joint video caching, recommendation and UA optimization problem for layered SVC streaming is formulated.

A. SYSTEM MODEL

1) NETWORK MODEL

As illustrated in Figure 1, we are interested in delivering video content in a cache-enabled cellular network that consists of M ground BSs and N users. Let $\mathcal{M} = \{1, 2, \ldots, m, \ldots M\}$ and $\mathcal{N} = \{1, 2, \ldots, n, \ldots N\}$ denote the set of M ground BSs and the set of N users, respectively. Each ground BS is equipped with a MEC server with limited caching resources, which is denoted as C_m^{cache}. The ground BSs are connected to the core network using wired backhaul links, which have a limited capacity. Meanwhile, users communicate with the ground BSs through radio access links, the bandwidth of which is assumed to be B. This bandwidth is shared among all the ground BSs.

Suppose that there are total F videos in the network, the set of which is denoted as $\mathcal{F} = \{1, 2, \ldots, f, \ldots F\}$. Each video has K different quality versions with different bitrates and resolutions. Let $\mathcal{K} = \{1, 2, \ldots, k, \ldots K\}$ denote the set of K quality versions. We assume that quality version 1 has the lowest bitrate and resolution, while quality version K has the highest bitrate and resolution. Let $v_{f,k}$ denote the video f with quality version k. In addition, we assume that each video is encoded into K layers, including one BL and $K-1$ ELs. We also use $\mathcal{K} = \{1, 2, \ldots, k, \ldots K\}$ to denote the set of K layers. The BL can realize the quality version 1, the BL combined with the first EL can realize the quality version 2, and so on. Let $\widetilde{v_{f,k}}$ denote the kth layer of video f. The size of $\widetilde{v_{f,k}}$ is denoted as $\widetilde{s_{f,k}}$, which generally decreases with the increase of k, i.e., $\widetilde{s_{f,1}} > \widetilde{s_{f,2}} > \cdots > \widetilde{s_{f,K}}$.

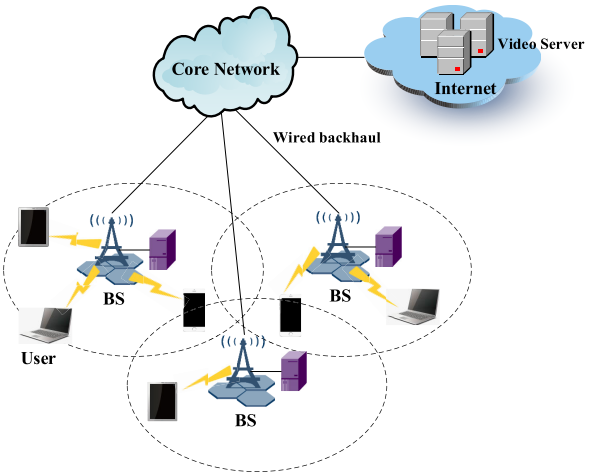


FIGURE 1. The cache-enabled cellular network architecture.

In general, caching policy and recommendation mechanism in cache-enabled cellular networks strongly depend on the VPD, while UA strategy is largely affected by the WCSI. It is assumed that the VPD and WCSI vary in different time slots. Suppose that the equal-sized time series is represented as $\mathcal{T} = \{1, 2, \ldots, t, \ldots\}$, where t denotes a time slot. Therefore, at the beginning of each time slot, the caching policy decision, recommendation mechanism decision and UA strategy decision are made and updated.

2) VIDEO REQUEST AND CACHING MODEL

Note that the VPD is based on users' video interest and preference. In this article, the VPD, i.e., the request probabilities of different videos are modeled as a Zipf-like distribution [50], [51]. Let $p_{n,f}^{(t)}$ be the probability that user n requests video f at time slot t and sort all F videos in a descending order based on their corresponding request probabilities, i.e., $p_{n,1}^{(t)} > p_{n,2}^{(t)} > \cdots > p_{n,f}^{(t)} > \cdots > p_{n,F}^{(t)}$. Thus, $p_{n,f}^{(t)}$ can be expressed as

$$p_{n,f}^{(t)} = \frac{f^{-\eta_n^{(t)}}}{\sum_{i=1}^{F} i^{-\eta_n^{(t)}}} \tag{1}$$

where $\eta_n^{(t)}$ is the skewness factor of user n at time slot t.

For each video, the request probabilities of different quality versions are modeled as a normal distribution. The mean of this distribution is denoted by ϑ, which represents the dominant quality version [52]. Thus, the request probability

of quality version k can be expressed as

$$p_k = e^{-(k-\vartheta)^2/2\sigma^2} \Big/ \sqrt{2\pi}\sigma \tag{2}$$

where σ^2 is the variance of the request probabilities of quality versions. A smaller σ leads to more concentrated requests of the dominant quality version ϑ. Based on $p_{n,f}^{(t)}$ and p_k, the probability of user n requesting $v_{f,k}$ at time slot t can be obtained by

$$p_{n,f,k}^{(t)} = p_{n,f}^{(t)} \cdot p_k \tag{3}$$

Denote $p_n^{(t)} = \left\{p_{n,1,1}^{(t)}, \ldots, p_{n,1,K}^{(t)}; \ldots; p_{n,F,1}^{(t)}, \ldots, p_{n,F,K}^{(t)}\right\}$ as user n's preference for all videos with different quality versions at time slot t. The time-varying VPD is modeled as a finite state Markov chain (FSMC). The corresponding state set is represented as $\mathcal{P} = \{p(\eta_1), p(\eta_2), \ldots, p(\eta_H)\}$, where $p(\eta_H)$ is the H-th state of VPD with skewness factor η_H. The total number of states is H and each state is a probability distribution of users requesting videos. The VPD transfers over time slots and the transition probability from one state to another state is denoted as $\mathbb{P}\left(p_n^{(t+1)} \middle| p_n^{(t)}\right)$, where $p_n^{(t)} \in \mathcal{P}, p_n^{(t+1)} \in \mathcal{P}$.

As for the video caching model, we denote the caching policy decision at time slot t as $\mathcal{X}^{(t)} = \left\{x_{m,f,k}^{(t)} | m \in \mathcal{M}, f \in \mathcal{F}, k \in \mathcal{K}\right\}$, where $x_{m,f,k}^{(t)} \in \{0,1\}$. If BS m caches $\widetilde{v_{f,k}}$ at time slot t, $x_{m,f,k}^{(t)} = 1$; otherwise, $x_{m,f,k}^{(t)} = 0$. Due to the limited caching resources, each BS can only store a part of video layers. For simplicity, we assume that BS m can store at most C_m^{cache} video layers and $C_m^{cache} < F \cdot K$. Under the caching capacity constraint, $x_{m,f,k}^{(t)}$ satisfies

$$\sum_{f=1}^{F}\sum_{k=1}^{K} x_{m,f,k}^{(t)} \leq C_m^{cache}, \quad \forall m \in \mathcal{M} \tag{4}$$

Given the caching policy decision, the caching energy consumption at time slot t can be represented as

$$E_{cache}^{(t)} = \sum_{m=1}^{M}\sum_{f=1}^{F}\sum_{k=1}^{K} w_{cache} x_{m,f,k}^{(t)} \widetilde{sf_{,k}} \tag{5}$$

where w_{cache} denotes the energy of caching one bit data in MEC servers (in J/bit).

3) VIDEO RECOMMENDATION MODEL

We denote the recommendation mechanism decision at time slot t as $\mathcal{Y}^{(t)} = \left\{y_{m,f,k}^{(t)} | m \in \mathcal{M}, f \in \mathcal{F}, k \in \mathcal{K}\right\}$, where $y_{m,f,k}^{(t)} \in \{0,1\}$. If BS m recommends $v_{f,k}$ to its associated users at time slot t, $y_{m,f,k}^{(t)} = 1$; otherwise, $y_{m,f,k}^{(t)} = 0$. The recommendation list, i.e., the videos that are recommended by BS m, is denoted as $\mathbb{R}_m^{(t)} = \left\{v_{f,k} \middle| y_{m,f,k}^{(t)} = 1\right\}$. We assume that BS m can only recommend C_m^{rec} videos to its associated users at each time slot, where $C_m^{rec} \leq C_m^{cache}$ because users generally would not like to read a long recommendation list. Therefore, $y_{m,f,k}^{(t)}$ satisfies

$$\sum_{f=1}^{F}\sum_{k=1}^{K} y_{m,f,k}^{(t)} = C_m^{rec}, \quad \forall m \in \mathcal{M} \tag{6}$$

Video recommendation generally boosts the request probabilities of the recommended videos and proportionately decreases the request probabilities of the non-recommended videos. Thus, if user n is associated with BS m at time slot t, user n's video preference after recommendation is given by

$$\widetilde{p_{n,f,k}^{(t)}} = \begin{cases} \alpha_n p_{m,f,k}^{(t),rec} + (1-\alpha_n) p_{n,f,k}^{(t)}, & v_{f,k} \in \mathbb{R}_m^{(t)} \\ (1-\alpha_n) p_{n,f,k}^{(t)}, & v_{f,k} \notin \mathbb{R}_m^{(t)} \end{cases} \tag{7}$$

where $\left\{p_{n,f,k}^{(t)} | f \in \mathcal{F}, k \in \mathcal{K}\right\}$ represents user n's inherent video preference, $\left\{\widetilde{p_{n,f,k}^{(t)}} | f \in \mathcal{F}, k \in \mathcal{K}\right\}$ represents user n's video preference after recommendation. $p_{m,f,k}^{(t),rec}$ represents the recommendation gain. According to [53], the recommendation gain of videos in the recommendation list is assumed to be equal, i.e., $p_{m,f,k}^{(t),rec} = \frac{1}{C_m^{rec}}$. $\alpha_n \in [0,1]$ represents the probability that user n is influenced by the video recommendation. $\alpha_n = 1$ means that user n accepts the recommended videos sufficiently and only requests videos in the recommendation list. On the contrary, $\alpha_n = 0$ means that video recommendation has no impact on user n's video preference, i.e., $\widetilde{p_{n,f,k}^{(t)}} = p_{n,f,k}^{(t)}$.

Although video recommendation can further enhance the caching gain, unfortunately, excessive recommendation which means that users are recommended with video contents that they are not interested in, has negative impact on users' QoE. To quantify the impact of video recommendation on users' preferences, we introduce the average user preference deviation, which is a non-negative function to measure the deviation of $\left\{p_{n,f,k}^{(t)} | f \in \mathcal{F}, k \in \mathcal{K}\right\}$ and $\left\{\widetilde{p_{n,f,k}^{(t)}} | f \in \mathcal{F}, k \in \mathcal{K}\right\}$, and is defined as

$$D^{(t)} = \frac{\sum_{n=1}^{N}\sum_{f=1}^{F}\sum_{k=1}^{K}\left(\widetilde{p_{n,f,k}^{(t)}} - p_{n,f,k}^{(t)}\right)^2}{N} \tag{8}$$

To alleviate the negative effects of excessive recommendation, we assume that the average user preference deviation should meet specific criteria, specifically, it should not exceed the maximum deviation tolerance $D^{\max}$. Therefore, $D^{(t)}$ satisfies

$$D^{(t)} \leq D^{\max} \tag{9}$$

4) TRANSMISSION MODEL

We denote the UA strategy decision at time slot t as $\mathcal{Z}^{(t)} = \left\{z_{m,n}^{(t)} | m \in \mathcal{M}, n \in \mathcal{N}\right\}$, where $z_{m,n}^{(t)} \in \{0,1\}$. If user n is associated with BS m at time slot t, $z_{m,n}^{(t)} = 1$; otherwise,

$z_{m,n}^{(t)} = 0$. We assume that one user can only be associated with one BS at each time slot, therefore, $z_{m,n}^{(t)}$ satisfies

$$\sum_{m=1}^{M} z_{m,n}^{(t)} = 1, \quad \forall n \in \mathcal{N} \tag{10}$$

If user n is associated with BS m at time slot t, the downlink transmission rate from BS m to user n can be represented as

$$R_{m,n}^{(t)} = \frac{B}{\sum_{n=1}^{N} z_{m,n}^{(t)}} \log\left(1 + \gamma_{m,n}^{(t)}\right) \tag{11}$$

where $\gamma_{m,n}^{(t)}$ denotes the received signal-to-interference-plus-noise-ratio (SINR) of user n from BS m at time slot t, $\sum_{n=1}^{N} z_{m,n}^{(t)}$ is the number of users associated with BS m. Here, B is equally allocated to all associated users [34]. The time-varying WCSI, i.e., the SINR $\gamma_{m,n}^{(t)}$, is modeled as a FSMC [54], [55]. In this model, the value range of $\gamma_{m,n}^{(t)}$ is quantized into L discrete levels: if $\gamma_0^* \le \gamma_{m,n}^{(t)} < \gamma_1^*$, γ_1; if $\gamma_1^* \le \gamma_{m,n}^{(t)} < \gamma_2^*$, γ_2; ...; if $\gamma_{m,n}^{(t)} \ge \gamma_{L-1}^*$, γ_L. Each level is a state of the FSMC and the corresponding state set is represented as $\Re = \{\gamma_1, \gamma_2, \ldots, \gamma_L\}$. The SINR transfers over time slots and the transition probability from one state to another state is denoted as $\mathbb{P}\left(\gamma_{m,n}^{(t+1)} \middle| \gamma_{m,n}^{(t)}\right)$, where $\gamma_{m,n}^{(t)} \in \Re$, $\gamma_{m,n}^{(t+1)} \in \Re$.

5) CONTENT DELIVERY ENERGY CONSUMPTION MODEL

Let $E_{m,n}^{(t)}$ denote the content delivery energy consumption from BS m to user n at time slot t. According to the basic principle of SVC, if a mobile user requests $v_{f,k}$, all layers of video f from the BL up to the $k-1$ EL (i.e., $\widetilde{v_{f,k'}}$, $k' = 1, 2, \ldots, k$) need to be transmitted to the mobile user. According to whether BSs cache the required layer of a video, there are two cases to handle requests from users. In the following, the content delivery energy consumption of these two cases will be discussed.

Case 1: BS m has cached the required $\widetilde{v_{f,k}}$ at time slot t (i.e., $x_{m,f,k}^{(t)} = 1$), and $\widetilde{v_{f,k}}$ can be delivered to user n directly. In this case, the content delivery energy consumption of the required $\widetilde{v_{f,k}}$ only contains the energy consumption for downlink radio transmission from BS m to user n denoted as $E_{m,n,f,k}^{(t),trans}$, which can be calculated by

$$E_{m,n,f,k}^{(t),trans} = P_m \frac{\widetilde{s_{f,k}}}{R_{m,n}^{(t)}} \tag{12}$$

where P_m is the transmission power of BS m.

Case 2: BS m does not cache the required $\widetilde{v_{f,k}}$ at time slot t (i.e., $x_{m,f,k}^{(t)} = 0$). In this case, handling users' requests is divided into two steps: backhaul link transmission and downlink radio transmission. Here, we assume that different layers of all videos are available in the core network. Thus, the content delivery energy consumption of the required $\widetilde{v_{f,k}}$ contains $E_{m,n,f,k}^{(t),trans}$ and the energy consumption for backhaul link transmission from the core network to BS m denoted as $E_{m,n,f,k}^{(t),BH}$. $E_{m,n,f,k}^{(t),BH}$ can be calculated by

$$E_{m,n,f,k}^{(t),BH} = w_{BH} \widetilde{s_{f,k}} \frac{\widetilde{s_{f,k}}}{R_{m,n}^{(t),BH}} \tag{13}$$

where w_{BH} represents the power of backhaul links to transmit one bit data (in Watt/bit), $R_{m,n}^{(t),BH}$ represents the backhaul link transmission rate allocated to user n by BS m at time slot t. Here, we assume that the backhaul link transmission rate of BSs is equally allocated to all associated users. Thus, $R_{m,n}^{(t),BH}$ can be expressed as $R_{m,n}^{(t),BH} = \frac{R_m^{BH}}{\sum_{n=1}^{N} z_{m,n}^{(t)}}$.

Based on the above analysis of the content delivery energy consumption of these two cases, $E_{m,n}^{(t)}$ is given by

$$E_{m,n}^{(t)} = \sum_{f=1}^{F} \sum_{k=1}^{K} \widetilde{p_{n,f,k}^{(t)}} \sum_{k'=1}^{k} \left(E_{m,n,f,k'}^{(t),trans} + \left(1 - x_{m,f,k'}^{(t)}\right) E_{m,n,f,k'}^{(t),BH}\right) \tag{14}$$

Therefore, the total content delivery energy consumption at time slot t can be represented as

$$E_{delivery}^{(t)} = \sum_{m=1}^{M} \sum_{n=1}^{N} z_{m,n}^{(t)} E_{m,n}^{(t)} \tag{15}$$

B. PROBLEM FORMULATION

Given the above models, the total energy consumption at each time slot consists of the caching energy consumption and the content delivery energy consumption, which can be expressed as

$$E_{total}^{(t)} = E_{cache}^{(t)} + E_{delivery}^{(t)} \tag{16}$$

In this article, our goal is to minimize the long-term system energy consumption by jointly optimizing video caching, recommendation and UA with the given WCSI $\gamma_{m,n}^{(t)}, \forall m \in \mathcal{M}, \forall n \in \mathcal{N}$ and VPD $p_n^{(t)}, \forall n \in \mathcal{N}$ in dynamic networks. To achieve this goal, according to (16), we formulate the long-term optimization problem as follows:

$$\min_{\mathcal{X}^{(t)}, \mathcal{Y}^{(t)}, \mathcal{Z}^{(t)}} \sum_{t \in \mathcal{T}} E_{total}^{(t)} \tag{17}$$

$$\text{s.t.} \quad \sum_{f=1}^{F} \sum_{k=1}^{K} x_{m,f,k}^{(t)} \le C_m^{cache}, \ \forall m \in \mathcal{M} \tag{17a}$$

$$\sum_{f=1}^{F} \sum_{k=1}^{K} y_{m,f,k}^{(t)} = C_m^{rec}, \ \forall m \in \mathcal{M} \tag{17b}$$

$$D^{(t)} \le D^{\max} \tag{17c}$$

$$\sum_{m=1}^{M} z_{m,n}^{(t)} = 1, \ \forall n \in \mathcal{N} \tag{17d}$$

$$x_{m,f,k}^{(t)} \in \{0, 1\}, \ \forall m \in \mathcal{M}, \ \forall f \in \mathcal{F}, \ \forall k \in \mathcal{K} \tag{17e}$$

$$y_{m,f,k}^{(t)} \in \{0, 1\}, \ \forall m \in \mathcal{M}, \ \forall f \in \mathcal{F}, \ \forall k \in \mathcal{K} \tag{17f}$$

$$z_{m,n}^{(t)} \in \{0, 1\}, \ \forall m \in \mathcal{M}, \ \forall n \in \mathcal{N} \tag{17g}$$

where (17a)-(17g) show the constraints of the optimization problem. Constraint (17a) indicates the caching capacity

limitation of each BS. Constraint (17b) indicates that BS m can only recommend C_m^{rec} videos to its associated users at each time slot. Constraint (17c) indicates the average user preference deviation does not exceed the maximum deviation tolerance $D^{\max}$. Constraint (17d) indicates that one user can only be associated with one BS at each time slot. Constraints (17e), (17f) and (17g) indicate that $x_{m,f,k}^{(t)}$, $y_{m,f,k}^{(t)}$ and $z_{m,n}^{(t)}$ are all binary variables, i.e., the values of them are either 0 or 1.

So far, the joint video caching, recommendation and UA optimization problem has been formulated. In Section III, we will show how to solve the problem (17) and present a solution to it.

III. SOLUTION TO JOINT VIDEO CACHING, RECOMMENDATION AND UA OPTIMIZATION PROBLEM

In this section, aiming at minimizing the long-term total energy consumption in dynamic networks, we first transform the optimization problem into a MDP. Then, for each time slot, given VPD $p_n^{(t)}, \forall n \in \mathcal{N}$ and WCSI $\gamma_{m,n}^{(t)}, \forall m \in \mathcal{M}, \forall n \in \mathcal{N}$, a DDQN-based algorithm is designed, using which the optimal video caching, recommendation and UA solutions can be obtained.

A. MARKOV DECISION PROCESS MODEL

In general, a MDP problem is defined by a tuple $\{\mathbb{S}, \mathbb{A}, \mathbb{P}, r\}$, where $\mathbb{S}$ represents state space, $\mathbb{A}$ represents action space, $\mathbb{P}$ represents state transition probability, r represents the immediate reward. Specifically, according to the optimization problem (17), the key elements of MDP are defined as follows.

1) STATE SPACE

$\mathbb{S}$ contains all possible states in dynamic networks, so $S^{(t)} \in \mathbb{S}$, where $S^{(t)}$ represents the network state at time slot t. $S^{(t)}$ is composed of VPD and WCSI at time slot t. Therefore, $S^{(t)}$ is defined as

$$S^{(t)} = \left\{p^{(t)}, \gamma^{(t)}\right\} \tag{18}$$

where $p^{(t)} = \left\{p_n^{(t)} \,|n \in \mathcal{N}\right\}$, $\gamma^{(t)} = \left\{\gamma_{m,n}^{(t)} \,|m \in \mathcal{M}, n \in \mathcal{N}\right\}$.

2) ACTION SPACE

$\mathbb{A}$ is the set of feasible actions in dynamic networks, so $A^{(t)} \in \mathbb{A}$, where $A^{(t)}$ represents the action at time slot t. $A^{(t)}$ is composed of video caching policy decision $\mathcal{X}^{(t)} = \left\{x_{m,f,k}^{(t)} \,|m \in \mathcal{M}, f \in \mathcal{F}, k \in \mathcal{K}\right\}$, recommendation mechanism decision $\mathcal{Y}^{(t)} = \left\{y_{m,f,k}^{(t)} \,|m \in \mathcal{M}, f \in \mathcal{F}, k \in \mathcal{K}\right\}$ and UA strategy decision at time slot t $\mathcal{Z}^{(t)} = \left\{z_{m,n}^{(t)} \,|m \in \mathcal{M}, n \in \mathcal{N}\right\}$. Therefore, $A^{(t)}$ is defined as

$$A^{(t)} = \left\{\mathcal{X}^{(t)}, \mathcal{Y}^{(t)}, \mathcal{Z}^{(t)}\right\} \tag{19}$$

3) TRANSITION PROBABILITY

The state transition probability is defined as $\mathbb{P}\left(S^{(t+1)} \left| S^{(t)}, A^{(t)}\right.\right)$.

4) REWARD FUNCTION

At time slot t, an agent first observes and senses the state of the dynamic network environment $S^{(t)}$. Then, according to a certain policy function π, the agent performs an action $A^{(t)}$. After the action is taken, the agent can obtain an immediate reward. Since the goal of the optimization problem (17) is to minimize the energy consumption, to achieve this goal, the energy consumption is set as the main reward. Therefore, the immediate reward is defined as

$$r\left(S^{(t)}, A^{(t)}\right) = \begin{cases} -E_{total}^{(t)}, & \text{if (17c) is satisfied} \\ -E_{total}^{(t)} - \varphi\left(D^{(t)} - D^{\max}\right), & \text{otherwise} \end{cases} \tag{20}$$

where φ represents the penalty reward factor.

The MDP problem can be solved by determining the optimal policy π^* that maximizes the long-term system reward. Here, $\pi : \mathbb{S} \to \mathbb{A}$ is a policy function that maps a state $S \in \mathbb{S}$ to an action $A \in \mathbb{A}$. There are two popular methods to assess the long-term system reward, namely state value function and state-action value function. Given a policy π, the state value function in state S is defined as

$$V^{\pi}(S) = \mathbb{E}^{\pi}\left[\psi^{(t)} \left| S^{(t)} = S\right.\right] \tag{21}$$

where $\mathbb{E}^{\pi}[\cdot]$ represents the mathematical expectation, $\psi^{(t)} = \sum_{\tau=t}^{\infty} \beta^{\tau-t} r\left(S^{(\tau)}, A^{(\tau)}\right)$. $\beta \in (0, 1]$ is the discount factor to determine the importance of immediate reward and future rewards. Similarly, the state-action value function in state S and action A is defined as

$$Q^{\pi}(S, A) = \mathbb{E}^{\pi}\left[\psi^{(t)} \left| S^{(t)} = S, A^{(t)} = A\right.\right] \tag{22}$$

RL algorithms as a branch of machine learning algorithms are generally used to solve the MDP problem. Q-learning algorithm [56] is a classical RL algorithm, which aims to train an agent to learn π^*. Since the state-action value function is used to assess the long-term system reward, learning π^* is equivalent to learning the optimal state-action value function $Q^*(S, A)$. π^* can be determined by $Q^*(S, A)$, i.e., $\pi^*(S) = \arg\max_{A \in \mathbb{A}} Q^*(S, A), \forall S \in \mathbb{S}$. In order to learn $Q^*(S, A)$, the agent needs to interact with the dynamic network environment repeatedly. Specifically, during each interaction step, the agent observes the environment's state, chooses an action, executes it, and receives an immediate reward. As a result of executing the action, the state S is transferred to the next state S'. Then, the state-action value function is updated by

$$Q(S, A) \leftarrow (1-\zeta) Q(S, A) + \zeta\left[r(S, A) + \beta \max_{A' \in \mathbb{A}} Q\left(S', A'\right)\right] \tag{23}$$

where $\zeta \in (0, 1]$ represents the learning rate. After several interaction steps, the agent can eventually learn π^* and $Q^*(S, A)$.

During the learning process, there are two methods for the agent to select an action, namely ''exploitation'' and ''exploration''. ''Exploitation'' means that the agent selects the action with the highest state-action value. ''Exploration'' means that the agent randomly selects an action except for the action with the highest state-action value. The ''exploitation'' process can maximize the long-term system reward, while the ''exploration'' process can avoid the Q-learning algorithm converging into a local optimum. Therefore, to learn π^*, the ''exploitation'' and ''exploration'' need to be balanced when selecting an action under a given state.

B. THE DDQN-BASED CACHING, RECOMMENDATION AND UA ALGORITHM

In the Q-learning algorithm, the state-action values of all state-action pairs are stored in a Q-table. However, with the increase of state space and action space, the Q-learning algorithm faces the challenge of ''curse of dimensionality'', which means that when the number of state-action pairs is huge, storing and searching the Q-table will take lots of time and space, leading to a slow learning speed and influencing the convergence efficiency. Motivated by deep learning, DDQN [57] has been proposed to overcome the above challenges.

Based on deep neural networks (DNNs)' nonlinear nature, deep learning can utilize DNNs to approximate almost any function by finding the low-dimensional features of high-dimensional data. The core idea of DDQN is to combine Q-learning algorithm with deep learning. DNNs are utilized to approximate the state-action value function, i.e., $Q(S, A; \theta, \varsigma, \varpi) \approx Q(S, A)$, where θ, ς and ϖ represent the set of weights and biases in DNNs. Three outstanding innovations are applied to make DDQN more efficient and robust:

1) DECOMPOSITION OF THE STATE-ACTION VALUE FUNCTION

The DDQN decomposes the state-action value function $Q(S, A; \theta, \varsigma, \varpi)$ into the state value function $\mathcal{V}(S; \theta, \varsigma)$ and the action advantage function $\mathcal{A}(S, A; \theta, \varpi)$, i.e., $Q(S, A; \theta, \varsigma, \varpi) = \mathcal{V}(S; \theta, \varsigma) + \mathcal{A}(S, A; \theta, \varpi)$. $\mathcal{V}(S; \theta, \varsigma)$ is a scalar and helps improve the capability of estimating the state value. $\mathcal{A}(S, A; \theta, \varpi)$ is an $|\mathbb{A}|$-dimensional vector and represents all actions' relative advantages.

2) EXPERIENCE REPLAY

Experience replay utilizes a finite-sized replay memory to store the agent's past learning experience, i.e., $\left(S^{(t)}, A^{(t)}, r^{(t)}, S^{(t+1)}\right)$. By this way, the DDQN can break the temporal correlations among past learning experiences and make the DNNs updating more efficient.

3) FIXED TARGET DNN

There are two DNNs in DDQN, i.e., the evaluated DNN and the target DNN, which have the same architecture. At each training step, the weights and biases in the evaluated DNN are updated. However, the weights and biases in the target DNN are kept fixed for a period of time and are only updated with the evaluated DNN periodically. Here, we assume that the weights and biases in the target DNN are updated every G training steps, i.e., $\theta^{-(t)} = \theta^{(t-G)}$, $\varsigma^{-(t)} = \varsigma^{(t-G)}$, $\varpi^{-(t)} = \varpi^{(t-G)}$, where θ, ς, ϖ and θ^-, ς^-, ϖ^- are the weights and biases in the evaluated DNN and the target DNN respectively. This innovation can stabilize and smooth the learning process.

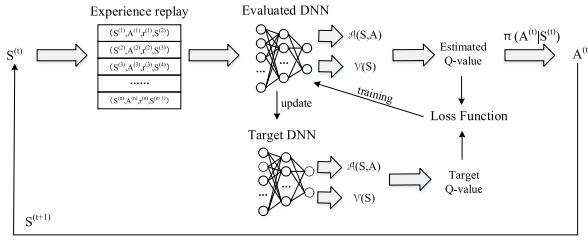


FIGURE 2. The workflows of DDQN.

Both the evaluated DNN and the target DNN take the state $S \in \mathbb{S}$ as input and all actions' state-action values under the state S as output. At each training step, the agent randomly selects a mini-batch of samples from the replay memory and updates the weights and biases in the evaluated DNN to minimize loss function $Loss(\theta, \varsigma, \varpi)$, which is defined as the mean-squared deviation between the target state-action value $Q_{target} = r(S, A) + \beta \max_{A' \in \mathbb{A}} Q\left(S', A'; \theta^-, \varsigma^-, \varpi^-\right)$ and the estimated state-action value $Q(S, A; \theta, \varsigma, \varpi)$, i.e., $Loss(\theta, \varsigma, \varpi) = \mathbb{E}\left[\left(Q_{target} - Q(S, A; \theta, \varsigma, \varpi)\right)^2\right]$. The workflows of DDQN are presented in Figure 2.

In the standard DDQN algorithm, the output of the evaluated DNN and the target DNN is all actions' state-action values, resulting that the output layer's dimension in the evaluated DNN and the target DNN is related to the size of the action space, which is $|\mathbb{A}| = \binom{FK}{C_m^{cache}}^M \binom{FK}{C_m^{rec}}^M \left(2^{MN}\right)$ in the formulated MDP problem. $\binom{FK}{C_m^{cache}}^M$ is the number of all possible caching policies, $\binom{FK}{C_m^{rec}}^M$ is the number of all possible recommendation mechanisms and 2^{MN} is the number of all possible UA strategies. It is obvious that as the number of videos F, the number of ground BSs M and the number of users N increasing, the size of the action space increases exponentially, resulting in the challenge of ''curse of dimensionality''. To overcome the challenge, we integrate linear approximation into the DDQN algorithm, which reduces the action space from exponential size $\binom{FK}{C_m^{cache}}^M \binom{FK}{C_m^{rec}}^M \left(2^{MN}\right)$ to linear size $2MFK + MN$. In linear approximation-integrated DDQN algorithm, the first MFK outputs of the evaluated DNN are used to select the caching policy $\mathcal{X}^{(t)}$, the next MFK outputs of the evaluated DNN are used to select the recommendation mechanism $\mathcal{Y}^{(t)}$ and the last MN outputs of the evaluated

DNN are used to select the UA strategy $\mathcal{Z}^{(t)}$, both of which are combined as the action $A^{(t)} = \left\{\mathcal{X}^{(t)}, \mathcal{Y}^{(t)}, \mathcal{Z}^{(t)}\right\}$.

Specifically, we denote the first MFK outputs of the evaluated DNN under a state as $\{Q_1, Q_2, \ldots, Q_{MFK}\}$. Then, the elements of BS 1's caching policy that correspond with the largest C_m^{cache} values in $\{Q_1, Q_2, \ldots, Q_{FK}\}$ are set to 1 and the other elements are set to 0. Similarly, the elements of BS 2's caching policy that correspond with the largest C_m^{cache} values in $\{Q_{FK+1}, Q_{FK+2}, \ldots, Q_{2FK}\}$ are set to 1 and the other elements are set to 0. By this way, all M BSs' caching policies $\mathcal{X}^{(t)}$ can be obtained. Moreover, we denote the next MFK outputs of the evaluated DNN under a state as $\{Q_{MFK+1}, Q_{MFK+2}, \ldots, Q_{2MFK}\}$. Then, the elements of BS 1's recommendation mechanism that correspond with the largest C_m^{rec} values in $\{Q_{MFK+1}, Q_{MFK+2}, \ldots, Q_{MFK+FK}\}$ are set to 1 and the other elements are set to 0. Similarly, the elements of BS 2's recommendation mechanism that correspond with the largest C_m^{rec} values in $\{Q_{MFK+FK+1}, Q_{MFK+FK+2}, \ldots, Q_{MFK+2FK}\}$ are set to 1 and the other elements are set to 0. By this way, all M BSs' recommendation mechanisms $\mathcal{Y}^{(t)}$ can be obtained. In addition, we denote the last MN outputs of the evaluated DNN under a state as $\{Q_{2MFK+1}, Q_{2MFK+2}, \ldots, Q_{2MFK+MN}\}$. Then, the element of user 1's UA strategy that corresponds with the largest value in $\{Q_{2MFK+1}, Q_{2MFK+2}, \ldots, Q_{2MFK+M}\}$ is set to 1 and the other elements are set to 0. Similarly, the element of user 2's UA strategy that corresponds with the largest value in $\{Q_{2MFK+M+1}, Q_{2MFK+M+2}, \ldots, Q_{2MFK+2M}\}$ is set to 1 and the other elements are set to 0. By this way, all N users' UA strategies $\mathcal{Z}^{(t)}$ can be obtained. Thus, we can ensure that the obtained $\mathcal{X}^{(t)}$ satisfies the constraint (17a), $\mathcal{Y}^{(t)}$ satisfies the constraint (17b) and $\mathcal{Z}^{(t)}$ satisfies the constraint (17d).

According to the basic principle of DDQN and linear approximation, a DDQN-based caching, recommendation and UA algorithm, i.e., Algorithm 1 is presented to solve the optimization problem (17). ε-greedy policy (lines 10-15) is used to select an action under the observed state aiming at balancing the "exploitation" and "exploration".

Algorithm 1 The DDQN-based Caching, Recommendation and UA Algorithm

1: Initialization:
2: Initialize the maximum number of training episodes $\Xi_{\max}$ and the maximum number of steps in each episode $g_{\max}$.
3: Initialize the experience replay memory and the mini-batch size.
4: Initialize the discount factor β, the learning rate ζ and the exploration probability ε.
5: Initialize the weights and biases in the evaluated DNN with $\theta, \varsigma, \varpi$.
6: Initialize the weights and biases in the target DNN with $\theta^- = \theta, \varsigma^- = \varsigma, \varpi^- = \varpi$.
7: **for** *episode* $= 1, 2, \ldots, \Xi_{\max}$ **do**
8: Reset the environment with the initial state S_{ini}, i.e., $S^{(t)} = S_{ini}$.
9: **for** $t = 1, 2, \ldots, g_{\max}$ **do**
10: Choose a random probability p.
11: **if** $p > \varepsilon$ **then**
12: Select an action $A^{(t)}$ with linear approximation.
13: **else**
14: Randomly select an action $A^{(t)}$.
15: **end if**
16: Execute the selected action, obtain the immediate reward $r^{(t)}$ and observe the next state $S^{(t+1)}$.
17: Store $\left(S^{(t)}, A^{(t)}, r^{(t)}, S^{(t+1)}\right)$ into the experience replay memory.
18: Randomly select a mini-batch of samples from the experience replay memory.
19: Calculate $\mathcal{V}(S; \theta, \varsigma)$ and $\mathcal{A}(S, A; \theta, \varpi)$, and combine them as the estimated state-action value $Q(S, A; \theta, \varsigma, \varpi)$.
20: Calculate the target state-action value Q_{target} by $Q_{target} = r(S, A) + \beta \max_{A' \in \mathbb{A}} Q\left(S', A'; \theta^-, \varsigma^-, \varpi^-\right)$.
21: Train the evaluated DNN to minimize the loss function $Loss(\theta, \varsigma, \varpi) = \mathbb{E}\left[\left(Q_{target} - Q(S, A; \theta, \varsigma, \varpi)\right)^2\right]$.
22: Update $\theta^-, \varsigma^-, \varpi^-$ every G training steps.
23: Set $S^{(t)} \leftarrow S^{(t+1)}$.
24: **end for**
25: **end for**

IV. SIMULATION RESULTS AND DISCUSSIONS

In this section, we employ computational simulation method to evaluate the effectiveness of the proposed DDQN-based caching, recommendation and UA algorithm. For the sake of simplicity in our simulations, the algorithm is referred to as "DDQN-based CA, UA and REC". All simulations are conducted on a X64-based laptop, which is equipped with 2.8GHz Intel Core i7, 32GB LPDDR3 and 512GB memory. The proposed algorithm is implemented in PyTorch 1.12.1 with Python 3.9. We consider a cellular network with $M = 4$ BSs and $N = 10$ users. We set $F = 30$, $K = 3$, $C_m^{cache} = 10$, $P_m = 46dBm$ [37], $B = 20MHz$, $\vartheta = 2$, $R_m^{BH} = 1.5Mbps$ [38], $w_{cache} = 8 \times 10^{-8} J/bit$ [58], $w_{BH} = 8 \times 10^{-6} Watt/bit$ [59], $C_m^{rec} = 5$, $\alpha_n = 0.7$, $D^{\max} = 0.08$. Besides, the sizes of three layers of each video are set as 4, 2 and 1Mbit. Furthermore, the VPD is set as a three-state FSMC with three different skewness factors $\{\eta_1, \eta_2, \eta_3\} = \{0.2, 0.5, 0.8\}$. Their transition probability matrix is assumed as

$$P^{\eta} = \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.1 & 0.6 & 0.3 \\ 0.3 & 0.1 & 0.6 \end{bmatrix} \tag{24}$$

Similarly, the WCSI is set as a three-state FSMC with three different spectrum efficiency parameters, i.e., 10, 2 and 0.2, which means that the state of wireless channels between BSs and users are good, medium and bad respectively. Their

transition probability matrix is assumed as

$$P^{SE} = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.1 & 0.7 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{bmatrix} \quad (25)$$

For comparison, the following four benchmark algorithms are considered:

"DDQN-Based CA and UA Only": In this algorithm, the learning agent only tries to learn the optimal caching policy and UA strategy, but makes recommendation mechanism decision randomly. That is to say, each BS recommends videos with different quality versions to its associated users randomly. Compared with "DDQN-based CA, UA and REC", the potential benefits of optimizing recommendation mechanism can be indicated.

"DDQN-Based CA and REC Only": In this algorithm, the learning agent only tries to learn the optimal caching policy and recommendation mechanism, but makes UA strategy decision randomly. That is to say, users are associated with BSs randomly. Compared with "DDQN-based CA, UA and REC", the potential benefits of optimizing UA strategy can be indicated.

"DDQN-Based UA and REC Only": In this algorithm, the learning agent only tries to learn the optimal UA strategy and recommendation mechanism, but makes caching policy decision randomly. That is to say, each BS caches videos' different layers randomly until its caching capacity is filled up. Compared with "DDQN-based CA, UA and REC", the potential benefits of optimizing caching policy can be indicated.

"Random CA, UA and REC": In this algorithm, the learning agent makes caching policy, UA strategy and recommendation mechanism decisions randomly. Compared with "DDQN-based CA, UA and REC", the potential benefits of jointly optimizing caching policy, UA strategy and recommendation mechanism can be indicated.

We first investigate the convergence performance of all the algorithms. Here, we set $\zeta = 0.01$, $\beta = 0.9$, $\varepsilon = 0.1$. In Figure 3, the abscissa denotes the number of episodes (each episode contains 40 time slots) and the ordinate represents the values of reward per episode, i.e., the system energy consumption. It can be seen that all DDQN-based algorithms can gradually converge to a stable value with the number of episodes increasing. Specifically, "DDQN-based CA, UA and REC", "DDQN-based CA and UA only", "DDQN-based CA and REC only" and "DDQN-based UA and REC only" reach stability after about 600 episodes, 700 episodes, 500 episodes and 550 episodes respectively. Besides, it shows that the converged stable value of "DDQN-based CA, UA and REC" is the smallest, the corresponding value of "Random CA, UA and REC" is the largest, while the corresponding values of other algorithms are medium. This demonstrates the potential benefits of jointly optimizing caching policy, recommendation mechanism and UA strategy.

Figure 4 illustrates the system energy consumption of all the algorithms under different values of the caching

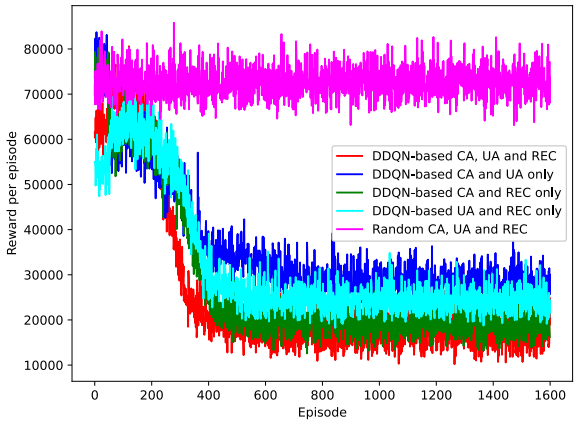


FIGURE 3. Convergence of the proposed algorithm.

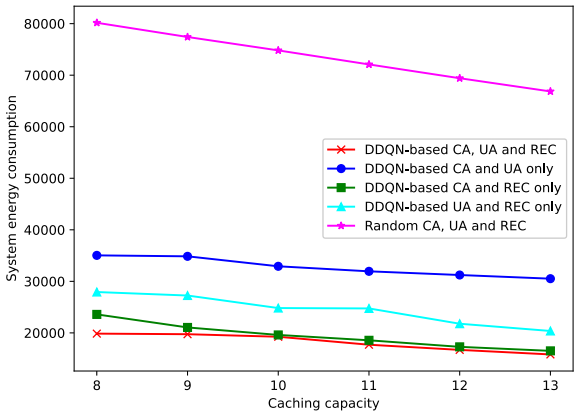


FIGURE 4. Total energy consumption under different values of the caching capacity of each BS.

capacity of each BS C_m^{cache}. We observe that the system energy consumption of all the algorithms decreases with the increase of C_m^{cache}. This is intuitive, when C_m^{cache} is larger, each BS can cache more video layers, and more users' requests can be satisfied locally without incurring backhaul link transmission. As a result, the backhaul link transmission energy consumption is decreased, which leads to the decrease of the total energy consumption. This figure also shows that the total energy consumption achieved by the proposed algorithm is smaller than the other benchmark algorithms.

Figure 5 reveals the relationship between the performance of all the algorithms and the number of users N. As we can see, the larger the number of users, the larger the system energy consumption of all the algorithms, which is in line with the intuition. Larger N leads to the increase of the number of video requests. In this case, given the VPD and caching capacity, more video requests consume more radio transmission energy consumption and backhaul link transmission energy consumption, resulting in the increase of the total energy consumption.

How the number of videos F affects the performance of all the algorithms is illustrated in Figure 6. As we can see, the system energy consumption of all the algorithms gradually increases with the increase of F. The reason is that when F becomes larger, users' requests scatter more widely, which

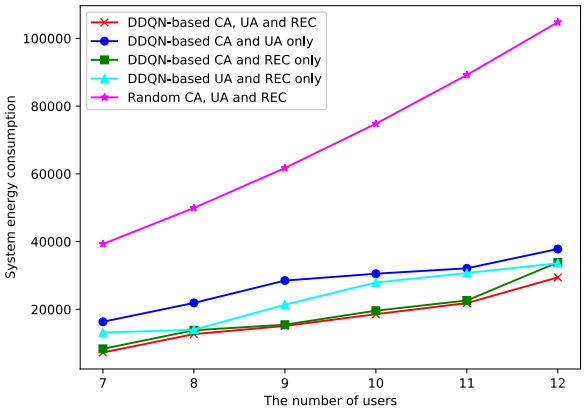


FIGURE 5. Total energy consumption under different values of the number of users.

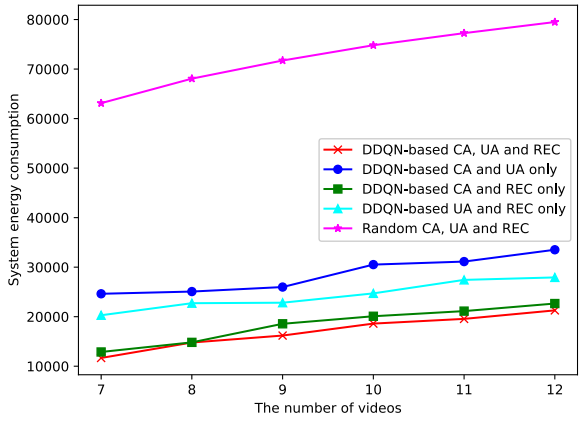


FIGURE 6. Total energy consumption under different values of the number of videos.

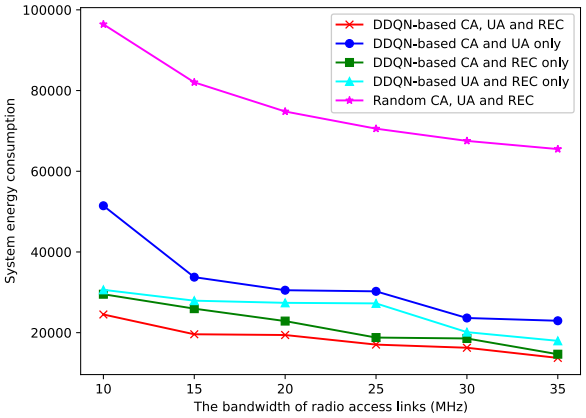


FIGURE 7. Total energy consumption under different values of the bandwidth of radio access links.

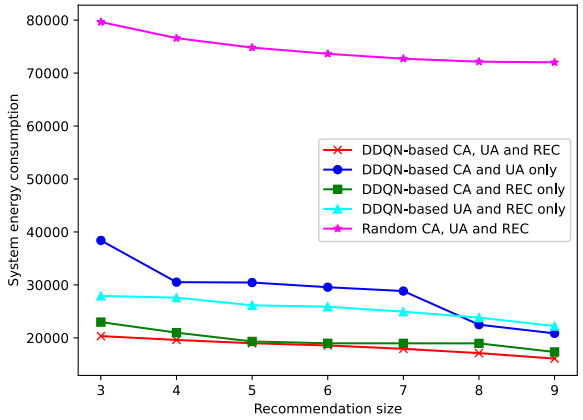


FIGURE 8. Total energy consumption under different values of the recommendation size.

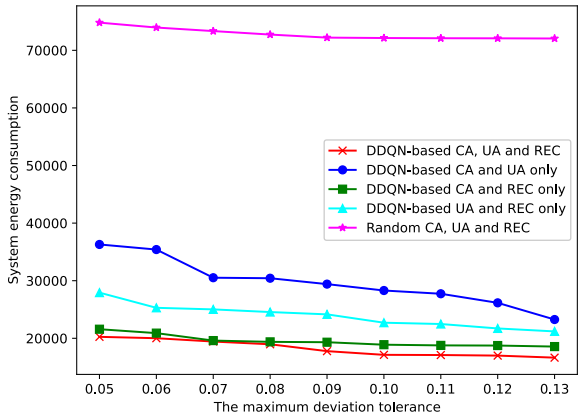


FIGURE 9. Total energy consumption under different values of the maximum deviation tolerance.

leads to higher cache miss rate and definitely consumes larger content delivery energy consumption, which leads to the increase of the total energy consumption.

Figure 7 reveals the impact of the bandwidth of radio access links B on the performance of all the algorithms. As expected, with the increase of B, the system energy consumption of all the algorithms gradually decreases. The reason is that when B becomes larger, $R_{m,n}^{(t)}$ increases which leads to the decrease of radio transmission delay and energy consumption. As a result, the total energy consumption is reduced.

The relationship between the performance of all the algorithms and the recommendation size C_m^{rec} is shown in Figure 8. We observe that the larger the recommendation size C_m^{rec}, the smaller the system energy consumption of all the algorithms, which is in line with the intuition. In the case that C_m^{rec} is less than C_m^{cache}, larger C_m^{rec} makes users' video preferences after recommendation more flat, which increases the effect of recommendation mechanism. Hence, the total energy consumption is decreased.

Next, the impact of the maximum deviation tolerance $D^{\max}$ on the performance of all the algorithms is revealed in Figure 9. As expected, when $D^{\max}$ becomes larger, the system energy consumption of all the algorithms decreases. The reason is that larger $D^{\max}$ means more actions taken by the agent can satisfy (17c). As a result, the obtained immediate reward is increased, which leads to the decrease of the total energy consumption.

V. CONCLUSION

This article has investigated the content caching, recommendation and transmission for layered SVC streaming in cache-enabled cellular networks. Taking into account the dynamic characteristics of video popularity distribution and wireless channels, we focused on minimizing the

long-term system energy consumption while ensuring the average user preference deviation tolerance. To achieve this, we formulated the problem of optimizing video caching, recommendation and UA as a MDP. We utilized both DDQN and linear approximation techniques to tackle the MDP problem and determine the optimal video caching, recommendation and UA decisions. Simulation results have demonstrated that both video caching, recommendation and UA decisions have effect on the system energy consumption, and the proposed algorithm yields significant performance gains in enhancing the energy efficiency compared with benchmark algorithms. In our future works, UAV-assisted cellular networks will be considered, and UAV deployment, UA, content caching, recommendation and resource allocation will be jointly optimized to improve users' QoE for layered SVC streaming.

REFERENCES

[1] ERICSSON. (2022). *ERICSSON Mobility Report*. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/mobility-report

[2] Z. Piao, M. Peng, Y. Liu, and M. Daneshmand, "Recent advances of edge cache in radio access networks for Internet of Things: Techniques, performances, and challenges," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 1010–1028, Feb. 2019.

[3] J. Yao, T. Han, and N. Ansari, "On mobile edge caching," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2525–2553, 3rd Quart., 2019.

[4] B. Jedari, G. Premsankar, G. Illahi, M. D. Francesco, A. Mehrabi, and A. Ylä-Jääski, "Video caching, analytics, and delivery at the wireless edge: A survey and future directions," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 431–471, 1st Quart., 2021.

[5] Y. Huo, C. Hellge, T. Wiegand, and L. Hanzo, "A tutorial and review on inter-layer FEC coded layered video streaming," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 1166–1207, 2nd Quart., 2015.

[6] F. Afsana, M. Paul, M. Murshed, and D. Taubman, "Efficient scalable UHD/360-video coding by exploiting common information with cuboid-based partitioning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 6, pp. 3961–3977, Jun. 2022.

[7] A. Heindel, B. Prestele, A. Gehlert, and A. Kaup, "Enhancement layer coding for chroma sub-sampled screen content video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 2, pp. 788–801, Feb. 2022.

[8] M. Kamel, W. Hamouda, and A. Youssef, "Ultra-dense networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2522–2545, 4th Quart., 2016.

[9] Y. Teng, M. Liu, F. R. Yu, V. C. M. Leung, M. Song, and Y. Zhang, "Resource allocation for ultra-dense networks: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2134–2168, 3rd Quart., 2019.

[10] L. Xu, C. Jiang, N. He, Y. Qian, Y. Ren, and J. Li, "Check in or not? A stochastic game for privacy preserving in point-of-interest recommendation system," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 4178–4190, Oct. 2018.

[11] J. Liu, L. Fu, X. Wang, F. Tang, and G. Chen, "Joint recommendations in multilayer mobile social networks," *IEEE Trans. Mobile Comput.*, vol. 19, no. 10, pp. 2358–2373, Oct. 2020.

[12] P. Sermpezis, T. Giannakas, T. Spyropoulos, and L. Vigneri, "Soft cache hits: Improving performance through recommendation and delivery of related content," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1300–1313, Jun. 2018.

[13] S. Battista, G. Meardi, S. Ferrara, L. Ciccarelli, F. Maurer, M. Conti, and S. Orcioni, "Overview of the low complexity enhancement video coding (LCEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 11, pp. 7983–7995, Nov. 2022.

[14] Q. Jiang, V. C. M. Leung, and H. Tang, "QoS-guaranteed adaptive modulation and coding for wireless scalable video multicast," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 1696–1700, Mar. 2022.

[15] W.-L. Hwang, C.-C. Lee, and G.-J. Peng, "Multi-objective optimization and characterization of Pareto points for scalable coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 7, pp. 2096–2111, Jul. 2019.

[16] J. Zhang, M. Wang, C. Jia, S. Wang, S. Ma, and W. Gao, "Scalable intra coding optimization for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 10, pp. 7092–7106, Oct. 2022.

[17] Y. Li, W. Dai, J. Zou, H. Xiong, and Y. F. Zheng, "Scalable structured compressive video sampling with hierarchical subspace learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 10, pp. 3528–3543, Oct. 2020.

[18] X. Zhang, X. Hu, L. Zhong, S. Shirmohammadi, and L. Zhang, "Cooperative tile-based 360° panoramic streaming in heterogeneous networks using scalable video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 1, pp. 217–231, Jan. 2020.

[19] C. Zhan and Z. Wen, "Content cache placement for scalable video in heterogeneous wireless network," *IEEE Commun. Lett.*, vol. 21, no. 12, pp. 2714–2717, Dec. 2017.

[20] K. Poularakis, G. Iosifidis, A. Argyriou, I. Koutsopoulos, and L. Tassiulas, "Distributed caching algorithms in the realm of layered video streaming," *IEEE Trans. Mobile Comput.*, vol. 18, no. 4, pp. 757–770, Apr. 2019.

[21] Y. Han, R. Wang, and J. Wu, "Random caching optimization in large-scale cache-enabled Internet of Things networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 385–397, Jan. 2020.

[22] C. Zhan and G. Yao, "SVC video delivery in cache-enabled wireless HetNet," *IEEE Syst. J.*, vol. 12, no. 4, pp. 3885–3888, Dec. 2018.

[23] W.-Y. Chen, P.-Y. Chou, C.-Y. Wang, R.-H. Hwang, and W.-T. Chen, "Dual pricing optimization for live video streaming in mobile edge computing with joint user association and resource management," *IEEE Trans. Mobile Comput.*, vol. 22, no. 2, pp. 858–873, Feb. 2023.

[24] Q. Xu, Z. Su, and J. Ni, "Incentivizing secure edge caching for scalable coded videos in heterogeneous networks," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 2480–2492, 2023.

[25] J. Ma, L. Liu, B. Shang, S. Jere, and P. Fan, "Performance analysis and optimization for layer-based scalable video caching in 6G networks," *IEEE/ACM Trans. Netw.*, vol. 31, no. 4, pp. 1494–1506, Aug. 2023.

[26] L. E. Chatzieleftheriou, G. Darzanos, M. Karaliopoulos, and I. Koutsopoulos, "Joint user association, content caching and recommendations in wireless edge networks," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 46, no. 3, pp. 12–17, Jan. 2019.

[27] D. Liu and C. Yang, "A learning-based approach to joint content caching and recommendation at base stations," in *Proc. IEEE GLOBECOM*, Abu Dhabi, United Arab Emirates, Jul. 2018, pp. 1–7.

[28] L. E. Chatzieleftheriou, M. Karaliopoulos, and I. Koutsopoulos, "Jointly optimizing content caching and recommendations in small cell networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 1, pp. 125–138, Jan. 2019.

[29] B. Zhu and W. Chen, "Coded caching with moderate recommendation: Balancing delivery rate and quality of experience," *IEEE Wireless Commun. Lett.*, vol. 8, no. 5, pp. 1456–1459, Oct. 2019.

[30] B. Zhu and W. Chen, "Coded caching with joint content recommendation and user grouping," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–6.

[31] X. Yang, Y. Fu, W. Wen, T. Q. S. Quek, and Z. Fei, "Mixed-timescale caching and beamforming in content recommendation aware fog-RAN: A latency perspective," *IEEE Trans. Commun.*, vol. 69, no. 4, pp. 2427–2440, Apr. 2021.

[32] Y. Wang, M. Ding, Z. Chen, and L. Luo, "Caching placement with recommendation systems for cache-enabled mobile social networks," *IEEE Commun. Lett.*, vol. 21, no. 10, pp. 2266–2269, Oct. 2017.

[33] F. Jiang, Z. Yuan, C. Sun, and J. Wang, "Deep Q-learning-based content caching with update strategy for fog radio access networks," *IEEE Access*, vol. 7, pp. 97505–97514, 2019.

[34] Y. Jiang, M. Ma, M. Bennis, F.-C. Zheng, and X. You, "User preference learning-based edge caching for fog radio access network," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1268–1283, Feb. 2019.

[35] T. Zhang, Y. Wang, W. Yi, Y. Liu, C. Feng, and A. Nallanathan, "Two time-scale caching placement and user association in dynamic cellular networks," *IEEE Trans. Commun.*, vol. 70, no. 4, pp. 2561–2574, Apr. 2022.

[36] T. Zhang, Z. Wang, Y. Liu, W. Xu, and A. Nallanathan, "Caching placement and resource allocation for cache-enabling UAV NOMA networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12897–12911, Nov. 2020.

[37] T. Zhang, Z. Wang, Y. Liu, W. Xu, and A. Nallanathan, "Joint resource, deployment, and caching optimization for AR applications in dynamic UAV NOMA networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 3409–3422, May 2022.

[38] T. Zhang, X. Fang, Z. Wang, Y. Liu, and A. Nallanathan, "Stochastic game based cooperative alternating Q-learning caching in dynamic D2D networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 12, pp. 13255–13269, Dec. 2021.

[39] 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.

[40] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, C. Wang, and Y. Liu, "A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 393–430, 1st Quart., 2019.

[41] 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.

[42] J. Feng, F. Richard Yu, Q. Pei, X. Chu, J. Du, and L. Zhu, "Cooperative computation offloading and resource allocation for blockchain-enabled mobile-edge computing: A deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6214–6228, Jul. 2020.

[43] 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.

[44] Y. Ren, R. Xie, F. R. Yu, T. Huang, and Y. Liu, "Quantum collective learning and many-to-many matching game in the metaverse for connected and autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 71, no. 11, pp. 12128–12139, Nov. 2022.

[45] Q. Tang, R. Xie, F. R. Yu, T. Chen, R. Zhang, T. Huang, and Y. Liu, "Collective deep reinforcement learning for intelligence sharing in the Internet of Intelligence-empowered edge computing," *IEEE Trans. Mobile Comput.*, pp. 1–16, 2022.

[46] C. Qiu, H. Yao, F. R. Yu, C. Jiang, and S. Guo, "A service-oriented permissioned blockchain for the Internet of Things," *IEEE Trans. Services Comput.*, vol. 13, no. 2, pp. 203–215, Mar. 2020.

[47] A. H. Sodhro, S. Pirbhulal, Z. Luo, K. Muhammad, and N. Z. Zahid, "Toward 6G architecture for energy-efficient communication in IoT-enabled smart automation systems," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5141–5148, Apr. 2021.

[48] B. Mao, F. Tang, Y. Kawamoto, and N. Kato, "AI models for green communications towards 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 210–247, 1st Quart., 2022.

[49] U. M. Malik, M. A. Javed, S. Zeadally, and S. U. Islam, "Energy-efficient fog computing for 6G-enabled massive IoT: Recent trends and future opportunities," *IEEE Internet Things J.*, vol. 9, no. 16, pp. 14572–14594, Aug. 2022.

[50] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. IEEE INFOCOM*, Jun. 1999, pp. 1–15.

[51] L. Cherkasova and M. Gupta, "Analysis of enterprise media server workloads: Access patterns, locality, content evolution, and rates of change," *IEEE/ACM Trans. Netw.*, vol. 12, no. 5, pp. 781–794, Oct. 2004.

[52] B. Shen, S.-J. Lee, and S. Basu, "Caching strategies in transcoding-enabled proxy systems for streaming media distribution networks," *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 375–386, Apr. 2004.

[53] L. E. Chatzieleftheriou, M. Karaliopoulos, and I. Koutsopoulos, "Caching-aware recommendations: Nudging user preferences towards better caching performance," in *Proc. EEE Conf. Comput. Commun.*, Atlanta, GA, USA, May 2017, pp. 1–9.

[54] Y. He, Z. Zhang, F. R. Yu, N. Zhao, H. Yin, V. C. M. Leung, and Y. Zhang, "Deep-reinforcement-learning-based optimization for cache-enabled opportunistic interference alignment wireless networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 10433–10445, Nov. 2017.

[55] F. Guo, F. R. Yu, H. Zhang, H. Ji, M. Liu, and V. C. M. Leung, "Adaptive resource allocation in future wireless networks with blockchain and mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1689–1703, Mar. 2020.

[56] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.

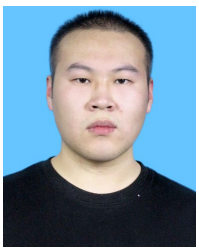
[57] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1995–2003.

[58] J. Hachem, N. Karamchandani, and S. Diggavi, "Content caching and delivery over heterogeneous wireless networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 756–764.

[59] R. Xie, Z. Li, J. Wu, Q. Jia, and T. Huang, "Energy-efficient joint caching and transcoding for HTTP adaptive streaming in 5G networks with mobile edge computing," *China Commun.*, vol. 16, no. 7, pp. 229–244, Jul. 2019.

JUNFENG XIE (Member, IEEE) received the B.S. degree in communication engineering from the University of Science and Technology Beijing, in 2013, and the Ph.D. degree from the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, in 2019. From September 2017 to September 2018, he visited Carleton University, Ottawa, ON, Canada, as a Visiting Ph.D. Student. He is currently an Assistant Professor with the North University of China. His research interests include machine learning, content delivery networks, resource management, and wireless networks.

QINGMIN JIA received the B.S. degree in communication engineering from Qingdao University of Technology, in 2014, and the Ph.D. degree in information and communication engineering from Beijing University of Posts and Telecommunications, in 2019. From July 2019 to May 2020, he was with China Mobile Hangzhou Research and Development Center. He is currently a Researcher with the Future Network Research Center, Purple Mountain Laboratories. His current research interests include edge intelligence, computing and network convergence, and the Industrial Internet of Things.



XINHANG MU received the B.S. degree in electronic information science and technology from the North University of China, in 2021, where he is currently pursuing the M.S. degree with the School of Information and Communication Engineering. His research interests include machine learning, signal processing, and resource management.

FENGLIANG LU received the B.S. degree in communication engineering from the North University of China, in 2023, where he is currently pursuing the M.S. degree with the School of Information and Communication Engineering. His research interests include machine learning, information processing and reconstruction, resource management, and wireless networks.