Received 12 April 2024; accepted 2 May 2024. Date of publication 9 May 2024; date of current version 21 May 2024.

Digital Object Identifier 10.1109/OJCOMS.2024.3397054

Learning-Based Joint User Association and Cache Replacement for Cache-Enabled Cloud RAN

SANG-EUN JEON[1], JAE-WOOK JUNG[2], KISONG LEE[3] (Member, IEEE), AND JUN-PYO HONG[2] (Member, IEEE)

[1]Department of Cloud Operations and Innovation (CO+I), Microsoft, Seoul 5673, South Korea

[2]Department of Smart Robot Convergence and Application Engineering, Pukyong National University, Busan 48513, South Korea

[3]Department of Information and Communication Engineering, Dongguk University, Seoul 04620, South Korea

CORRESPONDING AUTHOR: J.-P. HONG (e-mail: jp_hong@pknu.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) under Grant 2022R1F1A1062930.

ABSTRACT Mobile edge caching is regarded as a promising technology for reducing network latency and alleviating network congestion by efficiently offloading data traffic and computations to cache-enabled edge nodes. To fully leverage the benefits of edge caching, it is essential to jointly optimize caching and communication strategies, accounting for dynamic content request pattern and unstable nature of wireless mobile networks. Motivated by this, we study a joint cache replacement and user association strategy for minimizing the content delivery latency in cache-enabled cloud radio access network (C-RAN) where remote radio heads (RRHs) cache some contents for serving the content request without downloading the requested content from centralized baseband unit (BBU) via fronthaul. Unlike traditional cache placement strategies, our cache replacement facilitates gradual and timely updates while serving user content requests, without imposing additional network overhead. Specifically, whenever a user requests a content, BBU makes decisions on selecting a RRH for serving user request and on replacing the cached data of the selected RRH by taking into account the user location, cache status of RRHs, and impact on subsequent content deliveries. We optimize the selection of RRH to serve user request and the replacement of cached data by formulating a latency minimization problem using Markov Decision Process (MDP). This formulation considers the tradeoff between cache hit ratio and communication reliability. To develop an effective strategy for solving the MDP, we employ a deep reinforcement learning (DRL) algorithm and design a novel neural network structure and input feature map, specifically tailored to our problem domain. Simulation results show that the proposed approach learns effective strategy appropriate to a given environment, thereby outperforming not only the traditional rule-based strategies but also a typical DRL algorithm in terms of average latency. The proposed approach is shown to be relatively robust to time-variant content popularity by quickly adapting to new popularity distribution.

INDEX TERMS Cache replacement, C-RAN, DRL, mobile edge caching, user association.

I. INTRODUCTION

NETWORK densification is considered as an effective solution to cope with the rapid growth of mobile traffics, driven primarily by the proliferation of mobile devices and the increasing demand for high data-rate services [1]. The dense deployment of base stations (BSs) significantly improve the area throughput by increasing spatial spectrum reuse with the reduced communication distance [2], [3]. However, the dense BS deployment entails a high inter-cell interference and a high total cost of ownership (TCO) including capital expenditure (CAPEX) and operating expenditure (OPEX), and these are considered as main obstacles for deploying more BSs.

As an effective architecture to deal with aforementioned challenges, cloud radio access network (C-RAN) has been considered in [4], [5], [6], [7]. In C-RAN, the baseband processing for multiple distributed access points, called remote radio heads (RRHs), is conducted at a centralized

baseband unit (BBU). In virtue of the centralized processing, C-RAN architecture enhances the inter-cell interference management as well as improves the resource efficiency with resource pooling. Furthermore, wireless fronthaul between RRHs and centralized BBU provides ease of network deployment, flexibility, and cost-effectiveness. One of the main challenges of C-RAN is to deal with the performance bottleneck caused by the limited wireless fronthaul capacity and stability [8]. To improve the network throughput by avoiding bottleneck in wireless fronthaul, the communication resource allocation and transmission techniques have been extensively investigated in various scenarios [4], [5], [6], [7].

With the paradigm shift from connection-centric communications to content-centric communications, such as video streaming services, caching capability at edge nodes has attracted great attention as a powerful tool to mitigate the problems caused by the limited fronthaul capacity [9], [10], [11]. In cache-enabled wireless networks, the content cached at the edge nodes can be exploited to reduce not only the traffic burden on the fronthaul network but also the communication latency from a remote server by offloading traffic to relative cheap memory resources. Since such gains are generally proportional to the cache hit ratio, edge caching are becoming more prominent with the facts that a few popular contents account for most of mobile traffic and the price per unit of the memory capacity required for caching declines consistently [12], [13], [14]. In a cache-enabled network with a single BS, the BS is required to cache some contents with highest request probabilities to maximize the cache hit ratio and the network performance. When there is no prior information on the content popularity, the processes of learning content popularity and caching popular content have been tackled with algorithms for the multi-armed bandit (MAB) problem [15]. In small-cell networks with multiple BSs, the content placements at BSs should additionally take account of content-centric user association to improve cache hit ratio [16], [17]. In other words, since the content-centric user association makes it possible to utilize the cached contents at multiple nearby BSs to deal with a user request, caching contents with highest request probabilities at all BSs are not always the optimal content placement strategy for maximizing the cache hit ratio. Accordingly, for given user and BS deployments, the content placement for BSs has been investigated to improve the network performance in a centralized manner [18], [19], [20]. In situations where the content popularity information is initially unknown, an algorithm for MAB problem has been adopted for the decentralized content placement and sharing among BSs [21]. In addition to the content placement, the user association also should be carefully decided by taking account of BS-user link conditions and cache status at BSs jointly to strike a balance between the cache hit ratio and communication reliability [22], [23]. In this context, the joint optimization of content placement and user association have been tackled with iterative algorithms [24], [25], [26], [27], [28], [28], [29], latent factor model (LFM) [30], and deep reinforcement learning (DRL) [31], [32].

Generally, the content placement strategy periodically updates the cached contents in a proactive manner. Such proactive and periodic cache update makes the content placement strategy unsuitable for dynamic environments with high user mobility and non-stationary content popularity distribution since it does not change the cached content in-between the update periods. Especially in scenarios where content popularity must be inferred from user requests, changes in content popularity information tends to be predominantly observed during peak hours. However, traditional cache placement strategies, which typically schedule periodic updates during off-peak hours to avoid network congestion, often struggle to effectively adapt to popularity changes observed during peak hours. For continuous and prompt adaptation to dynamic environments, the cached content can be managed in a reactive manner with cache replacement strategies. Although the typical strategies used in content delivery network (CDN), such as least recently used (LRU) and frequently used (LFU), can be utilized for online cache update in cache-enabled edge networks, their performance gain is limited by their independence on network topology and link condition. The optimization of cache replacement strategy can be considered as sequential decision problem, where a content is sequentially replaced to enhance the long-term network performance under uncertainty on its consequences. Based on the sequential structure, the cache replacement has been formulated as Markov decision process (MDP) and solved with DRL approaches in [33], [34]. Multi-agent Q-learning-based cache replacement has been investigated to improve cache hit ratio in a cache-enabled dense network with simplified wireless channel model [33]. In the cellular network where devices are capable of device-to-device communication, effective cache replacement strategy for minimizing transmission cost has been investigated in [34]. These works have shown that the cache replacement strategies derived from the reinforcement learning (RL) outperforms LRU and LFU based strategies by taking additional account of network topology and cache status. Although the results in [33], [34] have shown the effectiveness of RL on solving the cache replacement problems, it is unable to apply their methods to practical scenarios with high-dimensional state spaces due to limitations of lookup table-based learning algorithms. To deal with such limitations of the classical RL algorithms, DRL algorithms have been utilized for deriving practical cache replacement strategies [35], [36], [37], [38], [39]. The DRL-based cache replacement for a single BS has been investigated to cope with dynamic content popularity in [35]. In cellular networks with multiple cache-enabled BSs, DRL has been utilized to derive centralized cache replacement strategy that takes advantage of cooperative edge caching to improve network performances [36]. To improve scalability, the decentralized cache replacement strategies have been developed with multi-agent DRL algorithms in cellular

networks [37], [38]. A federated learning-based cooperative caching framework was developed in [39] to address the challenge of reducing high computation and communication costs for distributed cache optimization in mobile edge networks.

Although the joint optimization with user association has potential to boost the edge caching gain by expanding available caches for serving a user, similarly as in the case of content placement [24], [25], [26], [27], [28], [31], [32], the conventional work on the cache replacement have not considered the cache-aware flexible user association. Specifically, most existing studies on cache replacement have considered fixed cellular regions and corresponding user associations, so that their cache replacement strategies have concentrated on cooperative edge caching for given user association [34], [36], [37], [38]. Furthermore, existing studies on cache replacement have neglected the physical layer issues, such as wireless channel fading and transmission errors, in the problem formulation.

Motivated by these, we study the joint cache replacement and user association problem for minimizing the expected content delivery latency in cache-enabled C-RAN where the cache status and user association of multiple RRHs are coordinated by the centralized BBU. We focus on reducing the latency caused by re-transmissions with wireless transmission failure. In other words, bad channel conditions with deep fading may lead to the delivery latency with re-transmissions in wireless fronthaul and access links. Moreover, since the latency depends on not only the number of re-transmissions but also the availability of the associated RRH's cache, there is a tradeoff between the communication reliability and the cache hit ratio in the joint optimization of user association and cache replacement. By taking into account this tradeoff relationship and the nature of the cache replacement, the problem is formulated by MDP and is tackled with the DRL algorithm to derive the strategy adaptable to time-variant wireless channel and content popularity. Moreover, to facilitate learning, we design a novel structure of deep neural network (DNN) and corresponding input feature map on the basis of the problem characteristics, such as the consecutive chunk delivery for a content request and the interrelationships among caches of RRHs. Simulation results show that the proposed strategy is able to achieve lower average latency than location-based and content-based user associations with traditional rule-based cache replacement. This indicates that the proposed learning approach is able to learn an effective cache replacement and user association strategy to minimize the expected latency by balancing communication reliability and cache hit ratio. Furthermore, it is important to note that the proposed learning approach outperforms a typical DRL algorithm in terms of not only the average latency but also the convergence rate of the training process. This indicates that the DNN structure and input feature design based on the problem characteristics are useful to facilitate appropriate training by limiting unnecessary propagation in the model update. Based on such benefits, the proposed approach is shown to rapidly adapt to environmental changes in non-stationary scenarios with time-variant content popularity.

TABLE 1. List of abbreviations.

Abbreviation	Description
BBU	Baseband unit
BS	Base station
C-RAN	Cloud radio access network
CNN	Convolutional neural network
CSI	Channel state information
CUA	Cache-based user association
DDPG	Deep deterministic policy gradient
DDQN	Double deep Q-network
DNN	Deep neural network
DQN	Deep Q-network
DRL	Deep reinforcement learning
DUA	Distance-based user association
FC	Fully connected
LFU	Least frequently used
LRU	Least recently used
MAB	Multi-armed bandit
MDP	Markov decision process
RAT	Radio access technology
ReLU	Rectified linear unit
RL	Reinforcement learning
RRH	Remote radio head
SNR	Signal-to-noise ratio
TTI	Transmission time interval

The main contributions of our work can be summarized as follows:

1) To the best of our knowledge, this is an initial work that formulates the joint optimization of user association and cache replacement in cache-enabled C-RAN with considering the tradeoff relationship between communication reliability and cache hit ratio.
2) Based on the characteristics of the problem, we design novel structures of DNN and input feature map that are specialized for solving the formulated MDP with DRL algorithm. The proposed learning approach not only improves the convergence speed of model training but also enables the converged strategy to achieve lower average latency than the typical DRL algorithm. Furthermore, such benefits make it possible for the strategy to adapt to the content popularity changes rapidly.
3) The simulation results show that the proposed approach not only outperforms the traditional rule-based strategies in all simulation environments but also achieves the performance comparable to the optimal strategy in some special cases. Furthermore, the behavior of the converged strategy provides valuable insights into the way of user association and cache replacement to minimize the expected latency in various situations.

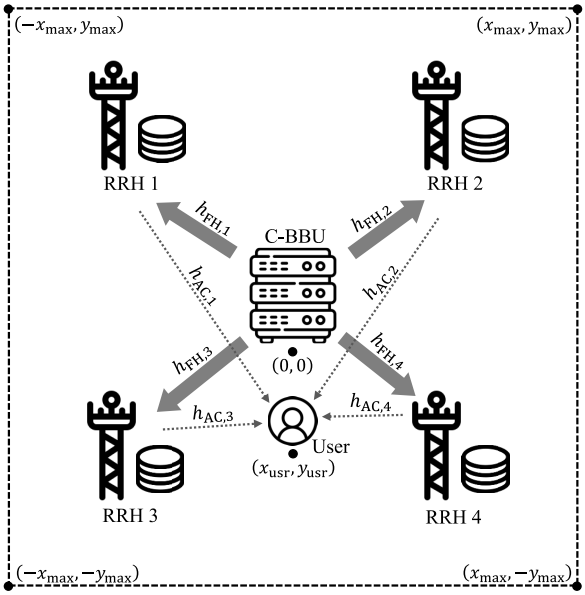

FIGURE 1. Cache-enabled C-RAN with $K = 4$ RRHs.

The remainder of the paper is organized as follows. Section II introduces the system model. In Section III, joint user association and cache replacement problem is formulated in the form of MDP, and the DRL algorithm for solving MDP is presented. Section IV provides the novel structure of DNN and corresponding input feature map for facilitating the training process of the DRL algorithm, and the performance of the proposed approach is evaluated through intensive simulations in Section V. Finally, we conclude this paper in Section VI.

II. SYSTEM MODEL

As illustrated in Fig. 1, we consider C-RAN with a single centralized BBU and K cache-enabled RRHs. The centralized BBU manages the entire network to efficiently deal with the content request of user located within the coverage area, and the one of RRHs delivers the requested content after association with the content requesting user. Note that a user is able to associate with any RRH within entire coverage area. It is assumed that there is no interference in wireless fronthaul and access links with the centralized resource coordination at the BBU. We consider 2-dimensional coordinate to represent the location of nodes, and the coverage is defined as the set of coordinates $\mathcal{C} = \{(x, y) : |x| \leq x_{\max}, |y| \leq y_{\max}\}$. The centralized BBU is assumed to be located in the center of the coverage, and its coordinate is denoted by the origin $(0, 0)$. A content requesting user is assumed to be located in a random coordinate $(x_{\mathrm{usr},i}, y_{\mathrm{usr},i}) \in \mathcal{C}$. Fig. 1 shows an example of network deployment with $K = 4$. Each content is partitioned into N_{p} equal-sized chunks and is transferred and cached in the unit of chunk. The centralized BBU is assumed to be able to download all L contents from data servers via the wired-backhaul network, but RRH $k \in \{1, 2, \ldots, K\}$ is able to cache a set of chunks $\mathcal{M}_k$ whose cardinality cannot exceed the memory capacity, $|\mathcal{M}_k| \leq M$. If a chunk of the requested content is not cached at the associated RRH, it needs to retrieve the chunk from the centralized BBU via wireless fronthaul. The centralized BBU is assumed to know the locations of RRHs, cache status of each RRH, and location of content requesting user. Based on the location information, the centralized BBU knows the long-term channel state information (CSI) of all fronthaul and access links. Based on such information, the centralized BBU makes decisions on the user association by selecting RRH to deliver a chunk of the requested content and the cache replacement of the associated RRH by selecting a cached chunk to be replaced with new chunk in case that the requested chunk is not cached at the associated RRH. In other words, for each chunk delivery of the requested content, the centralized BBU controls the user association and cache replacement to reduce the long-term average latency for dealing with a series of content requests. Note that user re-associations are facilitated within the centralized BBU, significantly reducing the associated overhead and delay [40]. Hence, the frequent user re-associations are assumed not to be problematic in our framework, unlike conventional cellular networks.

Suppose RRH k is selected for delivering a chunk $\mu_{i,n}$, which is chunk n of the i-th content request, to the user. If the selected RRH does not cache the chunk $\mu_{i,n} \notin \mathcal{M}_k$, it has to retrieve the chunk from BBU through wireless fronthaul link and replace one of cached chunks with the new one. Considering channel fading in wireless fronthaul, the probability of transmission failure can be represented as

$$p_{\mathrm{FH},k} = \Pr\left[B_{\mathrm{FH}} \log_2\left(1 + |h_{\mathrm{FH},k}|^2 \rho_{\mathrm{FH}}\right) \leq R\right] = 1 - \exp\left[-\frac{2^{R_{\mathrm{FH}}} - 1}{\rho_{\mathrm{FH}} d_{\mathrm{FH},k}^{-\alpha_{\mathrm{FH}}}}\right], \quad (1)$$

where ρ_{FH} denotes a transmit signal-to-noise-ratio (SNR) in the fronthaul transmission, B_{FH} denotes a transmission bandwidth of fronthaul link, R denotes a transmission rate, $R_{\mathrm{FH}} = \frac{R}{B_{\mathrm{FH}}}$, and $h_{\mathrm{FH},k} \sim \mathcal{CN}(0, d_{\mathrm{FH},k}^{-\alpha_{\mathrm{FH}}})$ denotes the fading channel gain of link from the centralized BBU to RRH k. Herein, $d_{\mathrm{FH},k}$ and α_{FH} denote the link distance and path loss exponent, respectively. For reliable content delivery, the centralized BBU re-transmits the chunk until RRH successfully receives it. Hence, the number of transmissions for successful chunk delivery $n_{\mathrm{FH},k}$ can be considered as a random variable that follows geometric distribution with success probability $1 - p_{\mathrm{FH},k}$. After the successful chunk reception at RRH k, the set $\mathcal{M}_k$ is updated by replacing one of cached chunks with the newly received one according to the decision of the centralized BBU. Then, the RRH k is able to deliver the chunk $\mu_{i,n}$ to the requesting user via access link. Similar to the wireless fronthaul link, the chunk transmission in the access link can fail with the probability

$$p_{\mathrm{AC},k} = 1 - \exp\left[-\frac{2^{R_{\mathrm{AC}}} - 1}{\rho_{\mathrm{AC}} d_{\mathrm{AC},k}^{-\alpha_{\mathrm{AC}}}}\right], \quad (2)$$

where $d_{\mathrm{AC},k}$ denotes distance of the access link from RRH k to user, $R_{\mathrm{AC}} = \frac{R}{B_{\mathrm{AC}}}$ denotes a transmission rate normalized by the access link bandwidth B_{AC}, and ρ_{AC} and α_{AC} denote SNR and path loss exponent in access link. The number of transmissions $n_{\mathrm{AC},k}$ for successful chunk delivery follows geometric distribution with success probability $1 - p_{\mathrm{AC},k}$. Note that the failure probability $p_{\mathrm{AC},k}$ depends on the location of the content requesting user, while the failure probability $p_{\mathrm{FH},k}$ is constant since the location of RRH does not change.

On the other hand, if the selected RRH k already caches the requested chunk $\mu_{i,n} \in \mathcal{M}_k$, the RRH is able to transmit the chunk $\mu_{i,n}$ to the user via access link without fronthaul use. Hence, there is no update on $\mathcal{M}_k$.

Suppose that transmissions over fronthaul link and access link take times of τ_{FH} and τ_{AC}, respectively. The values of τ_{FH} and τ_{AC} are dependant on processing time of receiving node and transmission time interval (TTI) of radio access technology (RAT). Then, the communication latency for delivering the chunk $\mu_{i,n}$ to the user can be represented as

$$\tau_{i,n} = \begin{cases} \tau_{\mathrm{BH}} + n_{\mathrm{FH},k}\tau_{\mathrm{FH}} + n_{\mathrm{AC},k}\tau_{\mathrm{AC}}, & \mu_{i,n} \notin \mathcal{M}_k \\ n_{\mathrm{AC},k}\tau_{\mathrm{AC}}, & \mu_{i,n} \in \mathcal{M}_k \end{cases}, \tag{3}$$

where τ_{BH} denotes the time for BBU to download the requested chunk from a remote data server. Note that it is able to avoid the latency occurred in backhaul and fronthaul links if the requested chunk is already cached at the associated RRH, $\mu_{i,n} \in \mathcal{M}_k$. Eventually, as a performance measure, we consider the long-term latency for dealing with T consecutive content requests from users as follows.

$$\tau_{\mathrm{sum}} = \sum_{i=1}^{T}\sum_{n=1}^{N_{\mathrm{p}}} \tau_{i,n}. \tag{4}$$

As shown in (1), (2), and (3), the transmission reliability and the cache availability are two main factors affecting the latency τ_{sum}. There is a tradeoff between the two factors in the network latency reduction. For example, if the centralized BBU adopts the distance-based user association to minimize the transmission failure, the available caches are limited to the chunks cached at the nearest RRH to the user. If the centralized BBU adopts the content-based user association with non-overlapped chunk caching to maximize the cache hit ratio, the access link transmission fails more frequently with a long access link distance. Accordingly, the joint optimization of the user association and cache replacement is important to minimize the long-term latency τ_{sum}.

III. LEARNING BASED JOINT USER ASSOCIATION AND CACHE REPLACEMENT

It is hard to derive an explicit solution of the joint user association and cache replacement problem with the offline optimization techniques due to the following reasons: i) Our problem is required to make a series of decisions on user association and cache replacement without the future information, such as the user locations, the requested contents, and the cached chunks of the RRHs. ii) The history of decision makings determines the cache status of RRHs, which has impact on subsequent decision makings. Consequently, each decision making has to be concerned with not only the efficient chunk delivery in present situation but also its influence on the future chunk delivery.

A. PROBLEM FORMULATION WITH MDP

We formulate the problem with MDP, which provides a mathematical framework for modeling decision making in situations where outcomes are partly random and partly under the control of a decision maker. For the sake of notational simplicity, we define $t = N_{\mathrm{p}}(i-1) + n$ as the decision making step for delivering chunk n of the i-th requested content. Hence, the subscripts i and n of symbols can be replaced with t, e.g., $\mu_{i,n} \to \mu_t$. The MDP of our problem consists of the following four components:

- *State:* A state at time step t is defined as a tuple

$$s_t = \left(\mu_t, \left(x_{\mathrm{usr},t}, y_{\mathrm{usr},t}\right), \mathcal{M}_{\mathrm{all},t}\right), \tag{5}$$

  where $(x_{\mathrm{usr},t}, y_{\mathrm{usr},t})$ denotes a coordinate of the content requesting user at step t and $\mathcal{M}_{\mathrm{all},t} = (\mathcal{M}_{1,t}, \mathcal{M}_{2,t}, \ldots, \mathcal{M}_{K,t})$ denotes a tuple of the caching sets at step t. Note that the coordinate $(x_{\mathrm{usr},t}, y_{\mathrm{usr},t})$ is assumed to be fixed at least for N_{p} consecutive steps, $N_{\mathrm{p}}(i-1)+1 \le t \le N_{\mathrm{p}}i$, to complete the content delivery to the i-th content requesting user. Define $\mathcal{S}$ as the set of all possible states.
- *Action*: An action at step t is defined as a tuple

$$a_t = (k, \breve{\mu}_k), \tag{6}$$

  where k denotes the index of RRH which serves the content requesting user, and $\breve{\mu}_k \in \mathcal{M}_{k,t}$ denotes a chunk that is replaced with the new one μ_t. Note that user association persists when the value of k remains constant across successive time intervals. Additionally, cache replacement is disregarded if the replacement indicator $\breve{\mu}_k$ aligns with the requested chunk μ_k. Define $\mathcal{A}$ as the set of all possible actions.
- *Transition probability:* If $t \bmod N_{\mathrm{p}} \neq 0$, the delivery of a requested content is not completed, so that the next chunk μ_{t+1} will be transmitted to the same user. Hence, if $t \bmod N_{\mathrm{p}} \neq 0$, the action $a_t = (k, \breve{\mu}_k)$ leads to the state transition from s_t to s_{t+1} composed of

$$\begin{aligned} &\mu_{t+1} = \mu_{i,n+1}, \\ &\left(x_{\mathrm{usr},t+1}, y_{\mathrm{usr},t+1}\right) = \left(x_{\mathrm{usr},t}, y_{\mathrm{usr},t}\right), \\ &\mathcal{M}_{k',t+1} = \begin{cases} \mathcal{M}_{k',t} & \text{if } k' \neq k \\ \mathcal{M}_{k',t} \backslash \{\breve{\mu}_{k'}\} \cup \{\mu_t\} & \text{if } k' = k \end{cases}. \end{aligned} \tag{7}$$

  On the other hand, if $t \bmod N_{\mathrm{p}} = 0$ and t is not terminal step, the system starts to deliver a content requested by a new user. Since the requesting user and its requesting content change independently with the history, the requesting user location $(x_{\mathrm{usr},t+1}, y_{\mathrm{usr},t+1})$ and the chunk μ_{t+1} are determined independently with

the previous action a_t and state s_t. Specifically, the action $a_t = (k, \breve{\mu}_k)$ leads to the state transition from s_t to s_{t+1} composed of

$$\begin{aligned} &\mu_{t+1} = \mu_{i+1,1}, \\ &\left(x_{\mathrm{usr},t+1}, y_{\mathrm{usr},t+1}\right) \in \mathcal{C}, \\ &\mathcal{M}_{k',t+1} = \begin{cases} \mathcal{M}_{k',t} & \text{if } k' \neq k \\ \mathcal{M}_{k',t} \backslash \{\breve{\mu}_{k'}\} \cup \{\mu_t\} & \text{if } k' = k \end{cases}. \end{aligned} \tag{8}$$

The next chunk μ_{t+1} becomes the initial chunk of the new requesting content $\mu_{i+1,1}$, which is randomly determined with the content popularity distribution. The user location $(x_{\mathrm{usr},t+1}, y_{\mathrm{usr},t+1})$ is an independent random coordinate belonging to $\mathcal{C}$. The caching sets are updated by the action a_t in the same way as (7).

- *Reward:* After taking action a_t, it is able to obtain the reward

$$r_{t+1} = -\tau_t. \tag{9}$$

From (3), it is evident that transmission reliability, influenced by the associated RRH k, and cache hit ratio, influenced by cache status $\mathcal{M}_k$, are primary factors affecting reward enhancement. However, as discussed in Section II, it is unable to maximize the gains of both factors simultaneously due to conflicting strategies for optimizing cache hit ratio and transmission reliability in terms of reward. Therefore, the centralized BBU should carefully balance these factors when taking action.

With a discount factor $\gamma \in [0, 1]$, the return at time step $t \in \{1, 2, \ldots, N_{\mathrm{p}}T\}$ is defined as

$$\begin{aligned} G_t &= \sum_{t'=t}^{TN_{\mathrm{p}}} \gamma^{t'-t} r_{t'+1} \\ &= -\sum_{t'=t}^{TN_{\mathrm{p}}} \gamma^{t'-t} \tau_{t'}. \end{aligned} \tag{10}$$

Eventually, under the MDP framework, the objective is to derive the optimal way of selecting actions, called a policy, for maximizing the expected value of return (10) for $t \in \{1, 2, \ldots, N_{\mathrm{p}}T\}$. Note that the maximization of the expected return is equivalent to the minimization of the expected latency for dealing with T consecutive content requests.

B. DEEP REINFORCEMENT LEARNING

Numerous advanced DRL algorithms are applicable for deriving a joint user association and cache replacement strategy that effectively minimizes the latency under the formulated MDP. In this study, we employ deep Q-network (DQN) as a fundamental and representative DRL algorithm and enhance its training process by developing a feature map and DNN structure. Importantly, these developed components can be reused for evaluating the state in other advanced DRL algorithms, such as double DQN (DDQN) [41] and deep deterministic policy gradient (DDPG) [31], [42], [43].

Algorithm 1 Deep Q-Network [44]

1: Initialize replay memory $\mathcal{B}$
2: Initialize parameters θ randomly
3: Initialize target parameters $\bar{\theta} \leftarrow \theta$
4: **for** each episode **do**
5: **for** $t = 1, N_{\mathrm{p}}T$ **do**
6: Observe state s_t
7: Select action $a_t = \arg\max_{a \in \mathcal{A}} Q(s_t, a; \theta) + w_a$
8: Observe reward r_t and next state s_{t+1}
9: Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{B}$
10: Sample mini-batch of $(s_{t'}, a_{t'}, r_{t'}, s_{t'+1})$ from $\mathcal{B}$
11: $g_{t'} \leftarrow \begin{cases} r_{t'} & , t'+1 = N_{\mathrm{p}}T \\ r_{t'} + \gamma \max_{a' \in \mathcal{A}} Q(s_{t'+1}, a'; \bar{\theta}) & , t'+1 \neq N_{\mathrm{p}}T \end{cases}$
12: Perform a stochastic gradient descent step on $\left(g_{t'} - Q(s_{t'}, a_{t'}; \theta)\right)^2$ with respect to θ
13: Reset target parameters $\bar{\theta} \leftarrow \theta$
14: **end for**
15: **end for**

At every time step t, the average return, called Q-value, is computed to evaluate state-action pairs and strategy is updated on the basis of the Q-value. The Q-value for action $a \in \mathcal{A}$ and state $s \in \mathcal{S}$ is defined as

$$Q(s, a) \ = \mathbb{E}[G_t | s_t = s, a_t = a]. \tag{11}$$

In our MDP structure, it is hard to compute and store Q-values for all possible state-action pairs since there are infinitely many states with $\binom{LN_{\mathrm{p}}}{M}^K$ possible cache status and continuous user coordinate. To deal with infinitely many states, we adopt DQN algorithm that utilizes DNN to approximate the relationship between state-action pair and its corresponding expected return [44]. In other words, instead of computing Q-values for all state-action pairs, DQN constructs a DNN-based action-value function that well approximates the actual Q-values for all state s and action a pairs as follows

$$Q(s, a) \approx Q(s, a; \theta), \tag{12}$$

where $\theta \in \mathbb{R}^D$ denotes parameters for DNN model.

Specifically, the numerical representation of state s, called input feature $\mathbf{X}(s)$, is defined to facilitate processing with DNN model. For an input feature $\mathbf{X}(s)$, the DNN model $\zeta(\mathbf{X}(s); \theta)$ is trained to return the approximated Q-values of all $|\mathcal{A}|$ actions at the state s. In other words, a Q-value $Q(s, a; \theta)$ is one entry of the output vector of the DNN model $\zeta(\mathbf{X}(s); \theta)$. For the sake of notational simplicity, we omit writing the explicit dependence of the input feature on the state s in the rest of paper.

The DQN algorithm for training the parameter θ is presented in Algorithm 1. In the line 7 of Algorithm 1, $w_a \sim \mathcal{N}(0, \sigma_w^2)$ denotes the artificial Gaussian noise for exploration of under-exploited actions. With the trained parameters θ from Algorithm 1, the strategy is determined by selecting

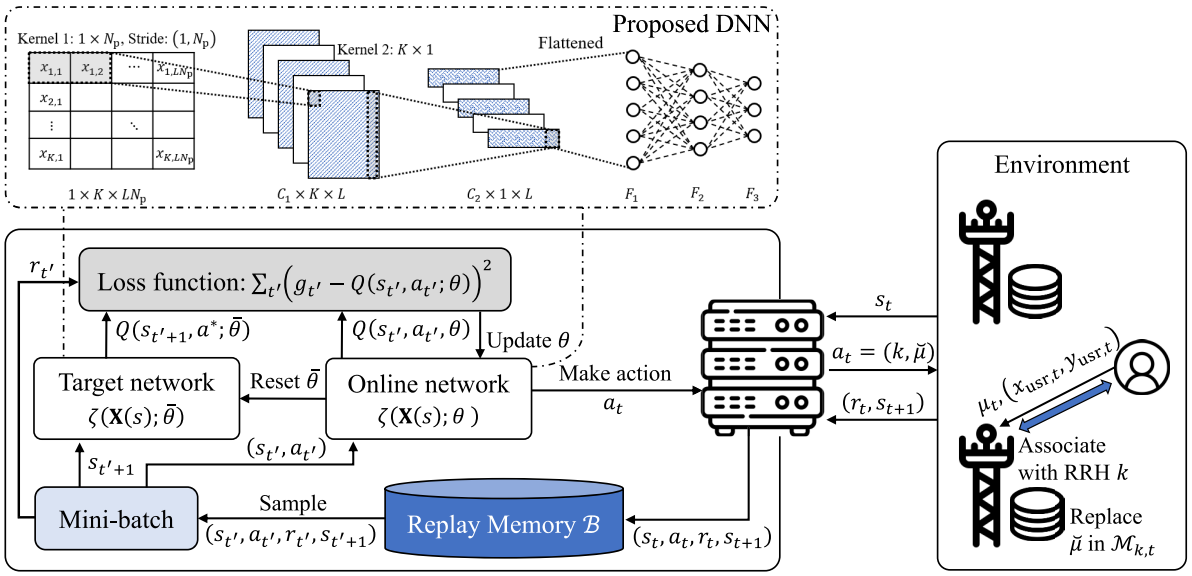

FIGURE 2. Process of the proposed framework.

an action $a \in \mathcal{A}$ that maximizes the approximated Q-value of current state $s \in \mathcal{S}$,

$$\pi(a|s) = \begin{cases} 1 \text{ if } a = \arg\max\limits_{a' \in \mathcal{A}} Q(s, a'; \theta) \\ 0 \text{ otherwise.} \end{cases} \tag{13}$$

Eventually, the training process for optimized joint user association and cache replacement, including its interaction with the environment, is depicted in Fig. 2. This illustration offers a comprehensive overview of our system's entire process.

IV. NEURAL NETWORK DESIGN

Although universal approximation theorem states that DNN can well-approximate any continuous function [45], it is not straightforward to train the parameters θ for the accurate Q-value approximation. Therefore, it is required to carefully design the input features $\mathbf{X}$ and DNN model $\zeta(\mathbf{X}; \theta)$ by taking into account the problem characteristics to facilitate the desirable parameter training.

In the proposed design, we construct an input feature $\mathbf{X}$ as a matrix whose entries are functions of the state information $(\mu_t, (x_{\mathrm{usr},t}, y_{\mathrm{usr},t}), \mathcal{M}_{\mathrm{all},t})$. Specifically, the chunks of all contents are indexed by $j \in \{1, 2, \ldots, LN_p\}$, and the content $l \in \{1, 2, \ldots, L\}$ consists of the chunks indexed by $j \in \{(l-1)N_p + 1, (l-1)N_p + 2, \ldots, lN_p\}$. Based on this indexing, each chunk j is represented as an one-hot encoded vector $\mathbf{q}_j^T \in \{0, 1\}^{1 \times LN_p}$. In other words, the requested chunk j is denoted by a vector $\mathbf{q}_j$ with all entries 0 except 1 in the j-th entry. To represent the cache statuses of K RRHs, we define a matrix $\mathbf{M} \in \{0, 1\}^{K \times LN_p}$ whose entry $m_{k,j}$ in the k-th row and j-th column is 1 if the chunk j is cached in RRH k and 0 otherwise. Due to the memory capacity of RRHs, the entries in k-th row of the matrix $\mathbf{M}$ satisfy $\sum_{j=1}^{LN_p} m_{k,j} \leq M$ for $k \in \{1, 2, \ldots, K\}$. The transmission failure probabilities of K access links are denoted by a vector $\mathbf{p}_{\mathrm{AC}} = [p_{\mathrm{AC},1}, p_{\mathrm{AC},2}, \ldots, p_{\mathrm{AC},K}]^T$.

Based on the requested chunk $\mathbf{q}_j$, cache status $\mathbf{M}$, and transmission failure probability $\mathbf{p}_{\mathrm{AC}}$, we define the feature map $\mathbf{X} \in \mathbb{R}^{K \times LN_p}$ whose entry in the k-th row and j-th column is

$$x_{k,j} = \theta_1^{(0)}\left(1 - p_{\mathrm{AC},k}\right) + \theta_2^{(0)} q_j + m_{k,j}, \tag{14}$$

where $\theta_1^{(0)}, \theta_2^{(0)} \in \mathbb{R}$ denote trainable parameters for balancing the importance of transmission reliability $1 - \mathbf{p}_{\mathrm{AC}}$, requested chunk $\mathbf{q}_j$, and cache status $\mathbf{M}$ in a decision making.

As illustrated in Fig. 2, the proposed DNN consists of two 1D-convolution layers and three fully connected (FC) layers. In the proposed DNN structure, the convolution layers are designed on the basis of two important characteristics of the problem. First, in the process of a content delivery, N_p consecutive chunks of the requested content should be sequentially delivered to the user. In other words, the chunk $j + 1$ is delivered after successful delivery of the chunk j if j mod $N_p \neq 0$. For this reason, the cache status on remaining undelivered chunks of the requested content is important to avoid replacing them before delivery. On the other hand, if j mod $N_p = 0$, the delivery of chunk $j + 1$ is independent with the delivery of chunk j. Based on such correlation between N_p chunks of a content, we introduce a convolution layer with C_1 filters tailored to the content size as a first layer of the proposed DNN. Specifically, each filter $c_1 \in \{1, 2, \ldots, C_1\}$ is a row vector consisting of N_p trainable parameters, $\theta_{c_1}^{(1)} \in \mathbb{R}^{1 \times N_p}$, and the strides of the convolution operation with the feature map $\mathbf{X}$ along column and row are N_p and 1, respectively. Second, in selection of the cached chunk to be replaced, the duplicated chunk caching at multiple RRHs is an important information for managing the cache hit ratio of the network. The replacement of a

TABLE 2. Specification of the proposed DNN.

	Conv1	Conv2	FC1	FC2	FC3
Input shape	$K \times LN_{\mathrm{p}}$	$C_1 \times K \times L$	$C_2 L$	N_{L1}	N_{L2}
Filter shape	$1 \times N_{\mathrm{p}}$	$K \times 1$	-	-	-
Strides along row, column	$1, N_{\mathrm{p}}$	$K, 1$	-	-	-
No. of filters	C_1	C_2	-	-	-
Activation func.	Leaky ReLU				
Output shape	$C_1 \times K \times L$	$C_2 \times L$	F_1	F_2	KLN_{p}

chunk with high cache duplication can increase the cache hit ratio. To facilitate the extraction of the feature on the cache duplication from the state information, we introduce the convolution layer with C_2 filters as a second layer of the proposed DNN. Specifically, each filter $c_2 \in \{1, 2, \ldots, C_2\}$ is a column vector consisting of K trainable parameters, $\theta_{c_2}^{(2)} \in \mathbb{R}^{K \times 1}$, and the stride of the convolution operation with the output of the first layer is 1. Note that there are no pooling layers after the convolution layers, contrary to the general convolutional neural network (CNN).

Following the two convolution layers, three FC layers produce the approximated Q-values for all actions in current state on the basis of the extracted features. The three FC layers involve F_1, F_2, and $F_3 = KLN_{\mathrm{p}}$ units. The leaky rectified linear unit (ReLU), which returns $\max\{x, 0.01x\}$ for an input x, is adopted as the activation function for all convolution and FC layers. Eventually, the details of the proposed DNN structure are summarized in Table 2.

V. SIMULATION RESULT

In this section, the performance of the learning-based joint user association and cache replacement strategy is evaluated via computer simulations. In all simulation environments, each content request is generated according to Zipf's law with the exponent s, and K RRHs are regularly deployed within the coverage areas. Unless otherwise stated, we use the system parameters described in Table 3 in all simulation results. For performance comparisons, we consider some baseline strategies on cache replacement and user association as follows:

- No-cache: All K RRHs do not have the caching capability. The user associates with the nearest RRH, and the associated RRH should receive all chunks of the requested content from the centralized BBU to forward them to the user.
- Distance-based user association and least frequently used replacement (DUA-LFU): The user associates with the nearest RRH, and the associated RRH removes the least frequently used chunk whenever its memory is overflowed with the new chunk received from the centralized BBU.
- Cache-based user association and least frequently used replacement (CUA-LFU): The user associates with the nearest RRH that caches the requested chunk, and the RRH removes the least frequently used chunk whenever its memory is overflowed. If the requested chunk is not cached at any RRHs, it operates in the same manner as DUA-LFU.

TABLE 3. Simulation environment.

Parameter	Value
Number of contents, L	20 [contents]
Chunks per content, N_p	4 [chunks/content]
Number of RRHs , K	4 [RRHs]
Memory capacity of each RRH, M	16 [chunks]
Path loss exponents [46], $\alpha_{\mathrm{FH}}, \alpha_{\mathrm{AC}}$	4.5 , 3.6
Exponent of Zipf's popularity dist., s	0.8
Coverage, $(x_{\max}, y_{\max})$	(250, 250) [m]
Transmit SNR, ρ	90 [dB]
Transmission rate, R_{FH}, R_{AC}	0.1, 1 [bits/s/Hz]
Latency in fronthaul/access link [47], $\tau_{\mathrm{FH}} = \tau_{\mathrm{AC}}$	1 [ms]
Latency in backhaul link, τ_{BH}	4 [ms]
Content requests in an episode, T	1,000 [requests]

Note that most conventional edge caching strategies are not applicable to our problem as they were not designed for making sequential decisions on user association and cache replacement. In addition, to see the effectiveness of the proposed DRL approach, we compare it with several learning-based benchmark methods.

- DDQN: This method utilizes DDQN algorithm [41] for training the model and making decisions. It leverages the input feature map and DNN structure proposed in our work.
- DDPG: This method utilizes DDPG algorithm [42] for training the model and making decisions. The proposed input feature map is applied to both the actor and critic networks. The critic network adopts the proposed DNN structure, while the actor network consists of three FC layers.
- Typical DQN/DDQN/DDPG: These methods train models and make decisions according to their respective DQN/DDQN/DDPG algorithms. They employ a standard DNN architecture comprising five FC layers and a basic input feature vector as follows.

$$\mathbf{x}_{\mathrm{FCN}} = \left[\mathbf{m}^T, \mathbf{q}_j^T, \mathbf{p}_{\mathrm{AC}}^T\right]^T, \tag{15}$$

where $\mathbf{m} \in \{0, 1\}^{KLN_{\mathrm{p}} \times 1}$ denotes a binary cache status vector that is concatenation of the columns of $\mathbf{M}$.

The parameter updates in DRL-based methods are performed using the adaptive moment estimation (ADAM) optimizer, with the hyperparameters specified in Table 4. In the proposed DNN structure, the numbers of filters are $C_1 = C_2 = 20$ and the numbers of units in three FC layers are $F_1 = 400$, $F_2 = 320$, and $F_3 = 320$. In the DNN structure of typical DQN, the numbers of units in five FC layers are $F_1 = 400$, $F_2 = 400$, $F_3 = 400$, $F_4 = 320$, and $F_5 = 320$.

Fig. 3 shows the long-term average latency, τ_{sum}/T, as the episode progresses. At the beginning of the initial episode, the memory of each RRH is initialized with random M

TABLE 4. Hyper-parameters for learning.

Parameter	Value
Learning rate, η	0.001
Discount factor, γ	0.99
Maximum replay memory, $\mathcal{B}$	1,000,000
Mini-batch size	4,096

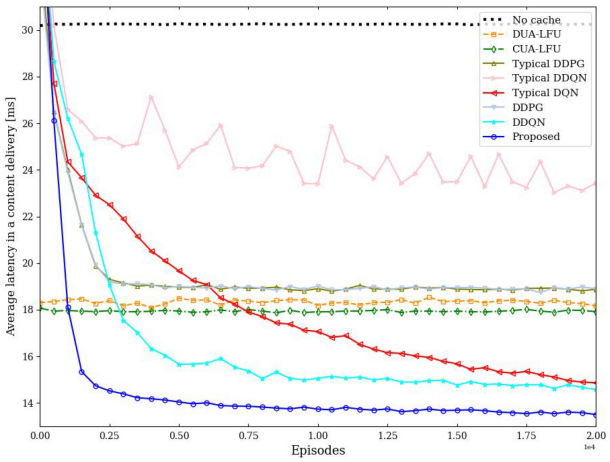

FIGURE 3. Average latency versus episodes.

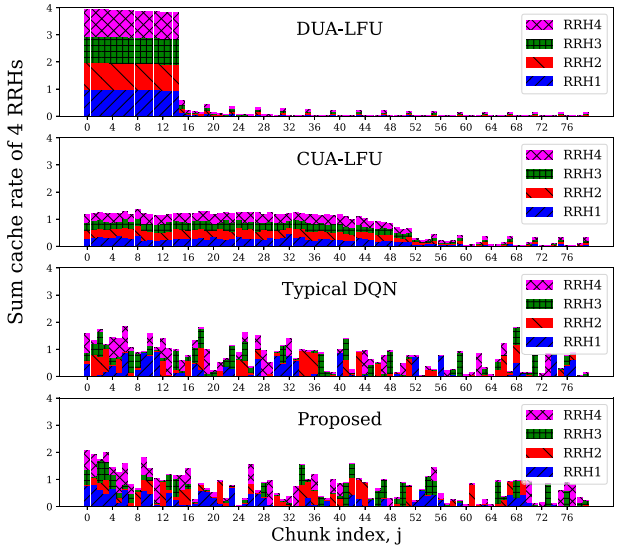

FIGURE 4. Normalized histogram of the chunk cache counts.

chunks in all methods since no prior information regarding content popularity is assumed. The large performance gap between No-cache and LFU-based baseline strategies, i.e., DUA-LFU and CUA-LFU, shows that the caching capability can provide significant performance gain in terms of the average latency. Unlike three baseline strategies that do not incorporate learning, learning-based approaches iteratively refine their strategies through experience gained from episodes. Eventually, learning-based approaches using DQN and DDQN are shown to outperform the two rule-based strategies after undergoing sufficient training episodes. Additionally, both our proposed method and DDQN exhibit enhanced performance over typical DQN and DDQN in terms of convergence rate and average latency, thereby validating the effectiveness of the proposed input feature map and DNN structure. However, methods based on DDPG fail to learn an effective strategy. This is likely because DDPG, primarily designed for continuous action spaces, is not optimally suited for the discrete action space of our problem [48]. Notably, in our problem, the DQN-based method is shown to outperform more advanced DRL algorithms, specifically DDQN and DDPG, in terms of convergence rate and training stability. While DDQN and DDPG have been developed to address certain limitations of the DQN algorithm, they introduce additional complexity, which can hinder convergence and lead to instability during training, especially in simpler environments [49], [50]. To focus on our main contributions and streamline the presentation, the results from DDQN and DDPG are omitted in the subsequent figures. This decision stems from DDQN's behavior being similar to that of DQN, albeit with a bit slower convergence rate, and DDPG's inability to learn an effective strategy.

Fig. 4 shows the normalized histogram of the chunk cache counts at $K = 4$ RRHs while handling 10,000 consecutive content requests. For example, if the chunk j is cached at all $K = 4$ RRHs all the time, the length of corresponding stacked bar becomes 4 with four equi-sized pieces. For DQN-based approaches, the simulation results are obtained by adopting the trained DNN after 20,000 training episodes. It is shown that the cache distributions of the LFU-based strategies are relatively more concentrated on some popular contents compared to the DQN-based approaches due to their popularity-based cache replacement. Furthermore, in the LFU-based strategies, the cache distributions of all RRHs are shown to be almost the same. This is because each RRH conducts the same popularity-based cache replacement independently with the cache status of the other RRHs. On the other hand, the cache distributions of DQN-based approaches are relatively spread out since they additionally consider the cooperative caching concept to improve the cache hit ratio by jointly optimizing the user association and cache replacement. Specifically, some popular contents are cached at multiple RRHs similarly to LFU-based strategies for improving communication reliability in access link, but unpopular contents are cached at a single or only a few RRHs for increasing the cache hit ratio. However, from the observation of typical DQN that some unpopular contents are cached at unnecessarily many RRHs, we can see that the strategy learnt from typical DQN does not fully exploit the cooperative caching gain. Consequently, the proposed approach facilitates learning cooperative caching concept for less popular content, which primarily contributes to its achievement of lower average latency compared to the typical DRL approaches, as demonstrated in Fig. 3.

Figs. 5, 6, 7 show the effect of system parameters on the average latency. In all these figures, the results obtained

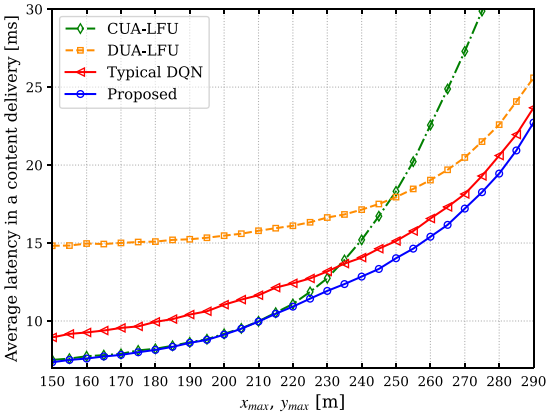

FIGURE 5. Average latency versus coverage area, x_{max} and y_{max}.

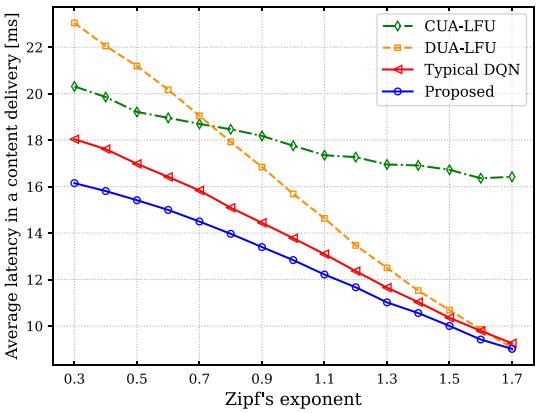

FIGURE 6. Average latency versus content popularity concentration, s.

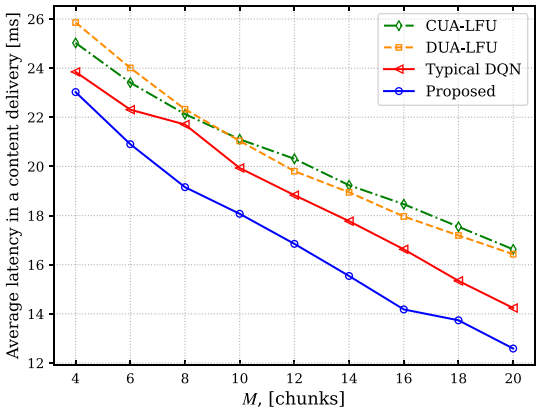

FIGURE 7. Average latency versus memory capacity of a RRH, M.

from learning-based approaches utilize DNNs pre-trained with 20,000 episodes. Results for the No-cache are omitted due to its significant performance gap compared to other methods.

Fig. 5 shows the long-term average latency against the system coverage, $x_{\max}$ and $y_{\max}$. Since the system coverage is directly related to the communication link distance, the wide system coverage leads to high transmission failure probability as shown in (1) and (2). In consequence, the average latency increases with the system coverage in all methods. However, the average latency increases differently across methods due to variations in the number of channel uses, influenced significantly by user association and cache status. In scenarios where the transmission failure probability is low with a limited coverage area, the influence of re-transmissions on the latency becomes negligibly small. When communication reliability is no longer the primary concern, the optimal strategy prioritizes maximizing the cache hit ratio, leading to behave like CUA-LFU. Note that the proposed method demonstrates an average latency that aligns with CUA-LFU, exhibiting optimal performance within a limited coverage area. On the other hand, when the transmission failure probability is high with a large coverage, the communication reliability becomes the dominant factor in the latency. Therefore, in such contexts, prioritizing meticulous management of communication reliability over cache hit ratio is crucial for user association and cache replacement strategies, resembling the behavior of DUA-LFU. The proposed method exhibits this optimal characteristics by showing the average latency of the proposed approach getting closer to that of DUA-LFU as the coverage increases. Based on these observations, it is evident that the proposed approach is able to learn an effective strategies tailored to specific cell coverage areas, facilitated by the designed input feature map and DNN.

Fig. 6 shows the long-term average latency against the content popularity concentration, s. The popularity concentration on a few contents grows with Zipf's exponent s, and the repeated requests of a few popular contents can reduce the latency by improving the cache hit ratio. Consequently, the average latency decreases with increasing s across all examined methods. When the popularity distribution is highly concentrated on a few contents with a high s, it becomes necessary to duplicate cache the most popular contents across multiple RRHs. This is essential for enhancing cache hit ratio while maintaining communication reliability. As a result, the optimal strategy tends to converge towards DUA-LFU for extremely high s. It's worth noting that the proposed method is demonstrated to achieve such optimal performance for very large s. On the other hand, when the popularity distribution is widely dispersed with a low s, the proposed method demonstrates significant performance improvements through cooperative caching. In this approach, RRHs strategically cache data chunks to minimize duplication.

Fig. 7 shows the long-term average latency against the memory capacity of a RRH, M. It is shown that the latency decreases with M in all strategies, and the proposed approach outperforms all the other baseline methods regardless of M. Observing that the performance gap between the proposed scheme and other baseline methods widens with increasing M, it becomes evident that joint optimization of user association and cache replacement is crucial, especially in scenarios requiring cooperative caching.

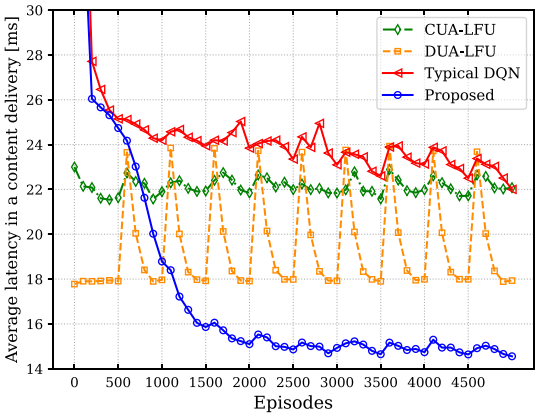

FIGURE 8. Average latency versus episode with time-variant content popularity.

In Fig. 8, we consider a non-stationary scenario where the content popularity changes over time. The content request probabilities are assumed to be circularly shifted by 5 ranks every 500 episodes. For example, if the request probability of content $l \in \{1, 2, \ldots, L\}$ is q_r, it changes to q_{r-5} if $r > 5$ or q_{L+r-5} if $r \leq 5$ after 500 episodes. The result shows that the change in content popularity increases the average latency instantly in all strategies, but it is reduced back down after a few episodes. DUA-LFU shows a relatively high latency rise when content popularity changes, compared to the other strategies. This phenomenon comes from the fact that the cache distribution of DUA-LFU is highly concentrated on a few popular contents as shown in Fig. 4. In other words, since all the RRHs cache almost the same chunks in DUA-LFU, the cache hit ratio is affected severely by the change in content popularity. Furthermore, it is interesting to note that although typical DQN fails to converge due to its slow learning convergence speed, the accelerated training process with the proposed approach makes it possible to successfully learn an effective strategy despite the content popularity changes. This is attributed to the design of our input feature map and DNN, facilitating the capture of relationships among content chunks and RRHs' cache status.

VI. CONCLUSION

In this paper, we have proposed a DRL based the joint user association and cache replacement strategy to minimize the content delivery latency in cache-enabled small-cell networks. We have formulated a sequential decision problem with MDP, and designed a novel DNN structure and input feature map for DQN algorithm by taking into account the problem characteristics to derive an effective strategy. Simulation results in various environments have shown that the proposed scheme outperforms not only the traditional user association and cache replacement strategies but also typical DQN algorithm in terms of the average latency. This implies that the proposed DNN and input feature designs facilitate to derive the improved strategy by exploiting the interrelation between network elements in the training process of DQN algorithm. From the observation that the performance gap between the proposed approach and the other baseline strategies grows with the memory capacity of RRHs, we can also see that the well optimized strategy is increasingly important with the rapid advances in memory technology. In addition, even if the content popularity changes in the middle of training process, the proposed scheme has been shown to quickly adapt to the new content popularity.

REFERENCES

[1] C. V. N. Index, "Cisco visual networking index: Forecast and methodology 2017-2022," Cisco Technol. Co., San Jose, CA, USA, White paper, Feb. 2015.

[2] J. G. Andrews, H. Claussen, M. Dohler, S. Rangan, and M. C. Reed, "Femtocells: Past, present, and future," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 3, pp. 497–508, Apr. 2012.

[3] N. Golrezaei, P. Mansourifard, A. F. Molisch, and A. G. Dimakis, "Base-station assisted device-to-device communications for high-throughput wireless video networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 7, pp. 3665–3676, Jul. 2014.

[4] L. Liu, S. Bi, and R. Zhang, "Joint power control and fronthaul rate allocation for throughput maximization in OFDMA-based cloud radio access network," *IEEE Trans. Commun.*, vol. 63, no. 11, pp. 4097–4110, Nov. 2015.

[5] R. G. Stephen and R. Zhang, "Joint millimeter-wave fronthaul and OFDMA resource allocation in ultra-dense CRAN," *IEEE Trans. Commun.*, vol. 65, no. 3, pp. 1411–1423, Mar. 2017.

[6] U. Siddique, H. Tabassum, E. Hossain, and D. I. Kim, "Wireless backhauling of 5G small cells: Challenges and solution approaches," *IEEE Wireless Commun.*, vol. 22, no. 5, pp. 22–31, Oct. 2015.

[7] J. Park, J.-P. Hong, and S. Beak, "Optimal beamforming with limited feedback for millimeter-wave in-band full-duplex mobile X-haul network," *IEEE Access*, vol. 6, pp. 51038–51048, 2018.

[8] X. Ge, H. Cheng, M. Guizani, and T. Han, "5G wireless backhaul networks: Challenges and research advances," *IEEE Netw.*, vol. 28, no. 6, pp. 6–11, Nov./Dec. 2014.

[9] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 82–89, Aug. 2014.

[10] J.-P. Hong and W. Choi, "User prefix caching for average playback delay reduction in wireless video streaming," *IEEE Trans. Wireless Commun.*, vol. 15, no. 1, pp. 377–388, Jan. 2016.

[11] Y. He, F. R. Yu, N. Zhao, V. C. M. Leung, and H. Yin, "Software-defined networks with mobile edge computing and caching for smart cities: A big data deep reinforcement learning approach," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 31–37, Dec. 2017.

[12] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "Analyzing the video popularity characteristics of large-scale user generated content systems," *IEEE/ACM Trans. Netw.*, vol. 17, no. 5, pp. 1357–1370, Oct. 2009.

[13] X. Cheng, J. Liu, and C. Dale, "Understanding the characteristics of Internet short video sharing: A YouTube-based measurement study," *IEEE Trans. Multimedia*, vol. 15, no. 5, pp. 1184–1194, Aug. 2013.

[14] G. Ma, M. Zhang, J. Ye, M. Chen, and W. Zhu, "Understanding performance of edge content caching for mobile video streaming," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1076–1089, May 2017.

[15] P. Blasco and D. Gündüz, "Learning-based optimization of cache content in a small cell base station," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2014, pp. 1897–1903.

[16] S. H. Chae and W. Choi, "Caching placement in stochastic wireless caching helper networks: Channel selection diversity via caching," *IEEE Trans. Wireless Commun.*, vol. 15, no. 10, pp. 6626–6637, Oct. 2016.

[17] S. H. Chae, T. Q. S. Quek, and W. Choi, "Content placement for wireless cooperative caching helpers: A tradeoff between cooperative gain and content diversity gain," *IEEE Trans. Wireless Commun.*, vol. 16, no. 10, pp. 6795–6807, Oct. 2017.

[18] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.
[19] W. Jiang, G. Feng, and S. Qin, "Optimal cooperative content caching and delivery policy for heterogeneous cellular networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 5, pp. 1382–1393, May 2017.
[20] J. Song, H. Song, and W. Choi, "Optimal content placement for wireless femto-caching network," *IEEE Trans. Wireless Commun.*, vol. 16, no. 7, pp. 4433–4444, Jul. 2017.
[21] J. Song, M. Sheng, T. Q. S. Quek, C. Xu, and X. Wang, "Learning-based content caching and sharing for wireless networks," *IEEE Trans. Commun.*, vol. 65, no. 10, pp. 4309–4324, Oct. 2017.
[22] T. Zhang, S. Biswas, and T. Ratnarajah, "On the performance of cache-enabled hybrid wireless networks," *IEEE Trans. Commun.*, vol. 69, no. 3, pp. 1818–1834, Mar. 2021.
[23] M. S. ElBamby, M. Bennis, W. Saad, and M. Latva-aho, "Content-aware user clustering and caching in wireless small cell networks," in *Proc. IEEE 11th Int. Symp. Wireless Commun. Syst. (ISWCS)*, 2014, pp. 945–949.
[24] B. Dai and W. Yu, "Joint user association and content placement for cache-enabled wireless access networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2016, pp. 3521–3525.
[25] W. Jing, X. Wen, Z. Lu, and H. Zhang, "User-centric delay-aware joint caching and user association optimization in cache-enabled wireless networks," *IEEE Access*, vol. 7, pp. 74961–74972, 2019.
[26] W. Teng, M. Sheng, K. Guo, and Z. Qiu, "Content placement and user association for delay minimization in small cell networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10201–10215, Oct. 2019.
[27] C. Chen, T. Zhang, Y. Liu, G. Y. Li, and Z. Zeng, "Joint user association and caching placement for cache-enabled UAVs in cellular networks," in *Proc. IEEE Int. Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2019, pp. 1–6.
[28] M. Karaliopoulos, L. Chatzieleftheriou, G. Darzanos, and I. Koutsopoulos, "On the joint content caching and user association problem in small cell networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC WKSHPS)*, 2020, pp. 1–6.
[29] X. Yang, Z. Fei, B. Li, J. Zheng, and J. Guo, "Joint user association and edge caching in multi-antenna small-cell networks," *IEEE Trans. Commun.*, vol. 70, no. 6, pp. 3774–3787, Jun. 2022.
[30] H. Li et al., "Intelligent content caching and user association in mobile edge computing networks for smart cities," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 1, pp. 994–1007, Jan./Feb. 2024.
[31] 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.
[32] Y.-C. Chuang, W.-Y. Chiu, R. Y. Chang, and Y.-C. Lai, "Deep reinforcement learning for energy efficiency maximization in cache-enabled cell-free massive MIMO networks: Single- and multi-agent approaches," *IEEE Trans. Veh. Technol.*, vol. 72, no. 8, pp. 10826–10839, Aug. 2023.
[33] J. Sung, K. Kim, J. Kim, and J.-K. Rhee, "Efficient content replacement in wireless content delivery network with cooperative caching," in *Proc. 15th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, 2016, pp. 547–552.
[34] W. Wang, R. Lan, J. Gu, A. Huang, H. Shan, and Z. Zhang, "Edge caching at base stations with device-to-device offloading," *IEEE Access*, vol. 5, pp. 6399–6410, 2017.
[35] P. Wu, J. Li, L. Shi, M. Ding, K. Cai, and F. Yang, "Dynamic content update for wireless edge caching via deep reinforcement learning," *IEEE Trans. Commun.*, vol. 23, no. 10, pp. 1773–1777, Oct. 2019.
[36] D. Li et al., "Deep reinforcement learning for cooperative edge caching in future mobile networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2019, pp. 1–6.
[37] C. Zhong, M. C. Gursoy, and S. Velipasalar, "Deep reinforcement learning-based edge caching in wireless networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 1, pp. 48–61, Mar. 2020.
[38] F. Wang, F. Wang, J. Liu, R. Shea, and L. Sun, "Intelligent video caching at network edge: A multi-agent deep reinforcement learning approach," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, 2020, pp. 1897–1903.
[39] A. Tian et al., "Efficient federated learning DRL-based cooperative caching for mobile edge networks," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 1, pp. 246–260, Mar. 2023.
[40] A. Checko et al., "Cloud RAN for mobile networks—A technology overview," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 405–426, 1st Quart., 2015.
[41] H. V. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 2094–2100.
[42] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," 2019, *arXiv:1509.02971*.
[43] M. Yan, R. Xiong, Y. Wang, and C. Li, "Edge computing task offloading optimization for a UAV-assisted Internet of vehicles via deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 73, no. 4, pp. 5647–5658, Apr. 2024.
[44] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.
[45] G. Cybenko, "Approximations by superpositions of a sigmoidal function," *Math. Control Signals, Syst.*, vol. 2, pp. 303–314, Dec. 1989.
[46] G. R. MacCartney Jr., J. Zhang, S. Nie, and T. S. Rappaport, "Path loss models for 5G millimeter wave propagation channels in urban microcells," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2013, pp. 3948–3953.
[47] F. Rinaldi, A. Raschella, and S. Pizzi, "5G NR system design: A concise survey of key features and capabilities," *Wireless Netw.*, vol. 27, no. 8, pp. 5173–5188, 2021.
[48] G. Matheron, N. Perrin, and O. Sigaud, "The problem with DDPG: Understanding failures in deterministic environments with sparse rewards," 2019, *arXiv:1911.11679*.
[49] S. Kumar, "Balancing a CartPole system with reinforcement learning–a tutorial," 2020, *arXiv:2006.04938*.
[50] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Merger, "Deep reinforcement learning that matters," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 3207–3214.