IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 29, 2020, accepted February 10, 2021, date of publication February 18, 2021, date of current version March 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3060323

Deep Reinforcement Learning for Trustworthy and Time-Varying Connection Scheduling in a Coupled UAV-Based Femtocaching Architecture

ZOHREH HAJIAKHONDI-MEYBODI[1], (Graduate Student Member, IEEE), ARASH MOHAMMADI[2], (Senior Member, IEEE), AND JAMSHID ABOUEI[3], (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, Concordia University, Montreal, QC H3G 2W1, Canada
[2]Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC H3G 2W1, Canada
[3]Department of Electrical Engineering, Yazd University, Yazd 89196-741, Iran

Corresponding author: Arash Mohammadi (arash.mohammadi@concordia.ca)

This work was supported in part by the Department of National Defence's Innovation for Defence Excellence and Security (IDEaS) Program, Canada.

ABSTRACT The paper is motivated by the urgent need, imposed by the COVID-19 pandemic, for trustworthy access to secure communication systems with the highest achievable availability and minimum latency. In this regard, we focus on an ultra-dense wireless network consisting of Femto Access Points (FAPs) and Unmanned Aerial Vehicles (UAVs), known as caching nodes, where there are more than one possible caching node to handle user's request. To efficiently cope with the dynamic topology of wireless networks and time-varying behavior of ground users, our focus is to develop an efficient connection scheduling framework, where ground users are autonomously trained to determine the optimal caching node, i.e., UAV or FAP. Our aim is to minimize users' access delay by maintaining a trade-off between the energy consumption of UAVs and the occurrence of handovers. To achieve these objectives, we formulate a multi-objective optimization problem and propose the Convolutional Neural Network (CNN) and Q-Network-based Connection Scheduling (CQN-CS) framework. More specifically, to solve the constructed multi-objective connection scheduling problem, a deep Q-Network model is developed as an efficient Reinforcement Learning (RL) approach to train ground users to handle their requests in an optimal and trustworthy fashion within the coupled UAV-based femtocaching network. The effectiveness of the proposed CQN-CS framework is evaluated in terms of the cache-hit ratio, user's access delay, energy consumption of UAVs, handover, lifetime of the network, and cumulative rewards. Based on the simulation results, the proposed CQN-CS framework illustrates significant performance improvements in companion to Q-learning and Deep Q-Network (DQN) schemes across all the aforementioned aspects.

INDEX TERMS Caching, cache-hit-ratio, connection scheduling, femtocaching, femto access point (FAP), reinforcement learning, unmanned aerial vehicle (UAV).

I. INTRODUCTION

Living in the COVID-19 pandemic area, we observe and face, first hand, abrupt and sudden changes in our daily life. In particular, the COVID-19 pandemic resulted in significant dependence on the Internet and Communication Networks [1] as almost everything went online. Of particular interest to this work, is the urgent need for trustworthy access to secure communication systems with the highest achievable availability and minimum latency for timely transfer of large multimedia contents especially to the ones living in remote and rural areas [2], [3]. For instance, it is of paramount importance under these pandemic situations to provide trustworthy services to students leaving in remote areas, and to communities without immediate access to the latest medical expertise, the ones who might be too ill to travel or may be unable to afford the cost of the trip. Consequently, there is an unmet need for increasing the Quality of Service (QoS) of communication systems to adapt to the new online reality during and post-pandemic times, while mitigating the burden of the

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

network's traffic over the backhaul channels. To meet the dynamic nature of Heterogeneous Networks (HetNets) such as unknown/varying number of active users, and adverse environmental conditions, the focus of recent research works [4], [5] have been shifted to the Reinforcement Learning (RL) approaches instead of deriving fixed mathematical models.

In this paper, we aim to address the aforementioned issues by incorporation of Unmanned Aerial Vehicles (UAVs), as mobile caching nodes, and Femto Access Points (FAPs) to increase the network's coverage. In a coupled UAV-based femtocaching scheme, however, it is critical to develop an efficient connection scheduling scheme to assign an appropriate caching node to ground users. Generally speaking, the overall objective of connection scheduling in HetNets is to improve the QoS of ground users, UAVs, and FAPs. In this regard, there are several conflicting objectives such as users' access delay, energy consumption of UAVs, and handover phenomena occurred between FAPs. As will be discussed in the literature review section of the manuscript, recent HetNet scheduling research works, however, mainly focused on one of these issues. The main motivation behind this paper is to address this lack of prior research studies on HetNet scheduling to simultaneously satisfy concerns of ground users, UAVs, and FAPs. The Convolutional Neural Network (CNN) with Q-learning Connection Scheduling (CQN-CS) framework is, therefore, proposed to address this gap. More specifically, the first objective of the proposed CQN-CS framework is allowing ground users to be autonomously trained to select an optimal caching node, i.e., UAV and/or FAP. Moreover, in contrary to existing research works [6], [7], where only one criterion is optimized to improve the QoS of the network, the proposed CQN-CS framework simultaneously considers optimization of three metrics, which are the energy consumed by UAVs, the probability of FAPs' handover, and the users' access delay.

A. LITERATURE REVIEW

Recently, several promising approaches have been developed to store the most popular multimedia contents in local caching nodes, including femtocaching architectures [8], Device-to-Device (D2D) communications [9], [10], UAV-based frameworks [11]–[13], and a combination of above schemes [14], [15]. In this regard, using UAVs over wireless networks helps FAPs to offload traffic via wireless backhaul, improve the network's coverage, and support a highly reliable and low-latency transmission [2], [16]. With the emphasis on the features of high mobility and low-cost manufacturing, the satisfactory rate of clients will increase through the utilization of UAVs as additional caching nodes for providing services to ground users residing in rural areas [17], [18]. Consequently, there has been recent widespread attention to UAVs due to their impressive potentials in supporting a wide range of commercial and industrial applications [19]. UAVs provide an adaptive platform, that can be altered by time-varying states of the environment and the need of ground users. Regarding the fact that UAVs are aircraft without human pilots, there are several approaches to manage and control multiple UAVs to lead them in a specific trajectory without any collisions, including manual and automatic modes [20]. For example, in a manual mode, an operator who is aware of the environment is required. The interaction between operator and UAVs should be managed in an optimum fashion based on an Intelligent Adaptive Interface (IAI) design to maximize the network performance [21]–[24]. In this paper, our focus is on UAV-based femtocaching schemes, where a ground user's access is updated autonomously with any change of the environment. In other words, we consider scenarios that users' access to requested contents can be managed autonomously by learning how to handle their requests in a trustworthy fashion (via Reinforcement Learning (RL) techniques) within a coupled UAV-based femtocaching network.

Recent developments on stand-alone UAV-based networks have brought several benefits, including but not limited to wide coverage and low-cost services. Therefore, several UAV-based small cell networks were introduced, such as the UAV clustering scheme [25], where UAVs play the role of caching nodes to serve all terrestrial users in the network. Reference [26], for instance, presented a wireless network architecture that employed cache-enabled UAVs with the goal of achieving considerable improvements in the users' Quality of Experience (QoE). A UAV-based machine learning algorithm was employed in [27] to predict the distribution of video content requested based on Echo State Networks. A critical drawback of using stand-alone UAVs in comparison to coupled UAV-based and femtocaching schemes, is the limited battery life of UAVs, particularly in situations where numerous active users in the network request multimedia contents. To overcome this issue, we consider a combination of UAVs and FAPs in this paper to mitigate the traffic load on either of the two. In this case, the overload on the backhaul link and the energy consumption of UAVs reduce significantly.

One of the most remarkable aspects of wireless networks is to enhance the QoS of ground users, in terms of the user's access delay. This has been investigated by several researchers in both femtocaching (e.g., [8], [28]), and UAV infrastructure networks (e.g., [29], [30]). Along this line of research, authors in [8] introduced a fairness scheduling framework to determine the most popular content in a femtocaching network to minimize the user's access delay. In this regard, the requests of ground users are ranked according to their previous experienced delay. Reference [31] introduced a double Q-learning UAV-based framework to optimize flying trajectories of UAVs and increase the number of satisfied users in terms of the experienced latency. In this case, the latency is defined as the waiting time, i.e., the time that a user must be in a queue to be served by a UAV, the flying time, and the transmission delay. To the best of our knowledge, despite all the research conducted in this field, there is no coupled UAV-based and femtocaching framework that presents a connection scheduling based on the user's access delay. Capitalizing on the significance of this unmet quest, the paper proposes the novel CQN-CS framework to address this gap.

Although potential benefits come by coupling UAVs with FAPs for the development of advanced femtocaching strategies, several critical challenges arise, especially keeping a trade-off between the energy consumption of UAVs and handover, occurring between FAPs. Existing solutions [15], [29], [32]–[34] focused on the energy consumption of UAVs (delays are not considered), for instance, in [15] a neural-blockchain-based UAV-caching approach is designed to provide a high reliability communication in terms of improvement in the energy consumption of UAVs, the maximum failure rate, the probability of connectivity, and survivability. Authors in [32] proposed a closed-form model for the energy consumption of rotary-wing UAVs with the aim of optimizing UAVs' trajectories and required communication times allocated to handle user requests. In [35], a fixed altitude for UAV's fly is assumed and an Air-to-Ground (A2G) communication scheduling scheme is proposed to optimize the trajectory, transmit power, and speed of UAVs, with the focus on decreasing the UAV propulsion energy consumption. In summary, the focus of these research works is on optimizing the energy consumption of UAVs, however, without addressing challenges associated with additionally introduced delays in the system that users experience when their requests are served through UAVs.

Finally, handover is known as one of the inevitable phenomena that needs to be mitigated in dynamic femtocaching networks. Particularly, consecutive handovers trigger during a request, if the ground user moves through the coverage area of femtocells [36]. Since handover is one of the key factors affecting the overall user's access delay, we introduced a Mobility-Aware Femtocaching algorithm based on Handover (MAFH) in our previous work, in which the best caching node is selected according to the Received Signal Strength Indicator (RSSI) value and the velocity of ground users as decision criteria [37]. In comparison to our proposed CQN-CS framework, where candidate caching nodes consist of UAVs and FAPs, the wireless network of [37] only consisted of FAPs and mobile users, supported by Device-to-Device (D2D) communications. One of the main goals of our proposed UAV-based CQN-CS framework is to serve users' requests to achieve the maximum QoS, in terms of having a considerable reduction in users' access delay. It should be noted that the wide transmission coverage of UAVs in comparison to FAPs, has brought several benefits, including the ability to handle the majority of users' requests. However, UAVs suffer from limited battery life, particularly in situations that numerous active users in the network request contents. On the other hand, by considering a dynamic femtocaching network, where users move consistently in the limited coverage area of FAPs, serving users' requests through FAPs, leads to triggering frequent handovers. However, to the best of our knowledge, there is no framework concerning this problem of connection scheduling between FAPs and UAVs. The paper addresses this gap.

B. CONTRIBUTIONS

The main novelty of the paper is the design of an autonomous and decentralized HetNet scheduling approach via simultaneous incorporation of three key objectives, i.e., users' access delay, energy consumption of UAVs, and handover phenomena. In other words, the proposed CQN-CS framework allows ground users to autonomously determine (via the RL-based component) an optimal caching node to handle their requests without reliance on any central processing unit. In summary, the paper makes the following key contributions:

- A practical UAV-based ultra-dense wireless network is considered consisting of UAVs and unlimited-energy FAPs, equipped with extended storage. In this model, the femtocell infrastructure network is partitioned according to the K-means clustering algorithm, where normally distributed ground users located in each cluster are served by its corresponding UAV. To incorporate a real wireless network, the proposed scheme uses the Difference Correlated Random Walk (DCRW) model as the mobility pattern, where the initial location of ground users is determined based on the Angle of Arrival (AoA) localization approach.
- Despite the surging interest in the coupled UAV-based femtocaching networks, there is no framework concerning the problem of connection scheduling over FAPs and UAVs. The proposed CQN-CS framework addresses this gap via its multi-objective optimization design with the goal of minimizing the user's access delay, energy consumption of UAVs, and handover phenomena.
- To solve the constructed multi-objective connection scheduling problem, we propose an efficient decentralized and self-organizing framework to maximize the QoS of the network by minimizing three objective functions related to user's access delay, energy consumption of UAVs, and handover phenomena. Due to the dynamic nature and unforeseen behavior of ground users, we apply a Deep Q-Network (DQN) architecture with CNN as an efficient RL approach to train our coupled UAV-based femtocaching network to respond to users' requests in an optimal fashion.

The effectiveness of the proposed CQN-CS framework is evaluated through comprehensive simulation studies in terms of the cache-hit ratio, user's access delay, energy consumption of UAVs, handover, lifetime of the network, and cumulative rewards. Simulation results illustrate that the efficiency of the proposed CQN-CS scheme is considerably superior to the Q-learning and DQN schemes over all the aforementioned aspects.

The rest of this paper is organized as follows: In Section II, the network's model is described and the main assumptions required for our algorithm are introduced. Section III deals with introducing our multi-objectives model. In Section IV, the proposed CQN-CS framework is introduced. Simulation results are presented in Section V. Finally, in Section VI, an overview of the results, and concluding remarks are presented.

II. SYSTEM MODEL AND PROBLEM DESCRIPTION

In this paper, we consider a UAV-based femtocaching network consisting of a set of ground mobile users, denoted by GU_j, for ($1 \leq j \leq N_g$), N_f number of FAPs, denoted by f_i, for ($1 \leq i \leq N_f$), N_u number of UAVs, as the flying caching nodes, denoted by u_k, for ($1 \leq k \leq N_u$), and a cloud server. As depicted in Fig. 1, the transmission range of FAPs, denoted by R_f, is much more limited than that of UAVs, denoted by R_u, where a UAV covers N_c number of FAPs in a neighbourhood. Without loss of generality, it is assumed that similar to FAPs that have the same storage of size S_f, UAVs are equipped with the cache storage with an equal size of S_u.

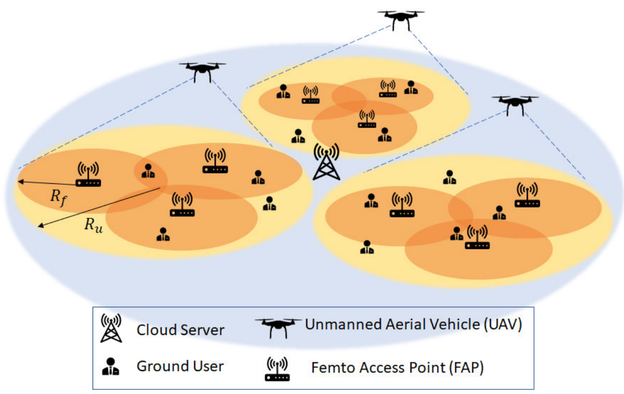


FIGURE 1. A typical structure of the proposed UAV-based femtocaching model.

In this model, ground users request an equal-sized video file c_l, for ($1 \leq l \leq \mathcal{C}$), with the probability of p_l, where p_l is calculated based on the Zipf-like distribution as follows [38]

$$p_l = \frac{l^{-\gamma}}{\sum_{r=1}^{\mathcal{C}} r^{-\gamma}}, \tag{1}$$

where $\mathcal{C}$ denotes the total number of video contents in the network, stored at a cloud-based content server, known as the main server in the network, and γ represents the popularity skewness. In a UAV-based femtocaching network, if a requested content exists in the caching nodes including FAPs and UAVs, the cache-hit occurs, otherwise, it is known as a hit-miss and the requested content is served by the main server. To decrease the video traffic load on the main server, UAV-based femtocaching strategies increase the number of requests directly served by FAPs and/or UAVs [39]. However, due to the large size of video files, it is not feasible to store all contents in the storage of caching nodes. Additionally, taking into account the varying popularity of video contents over time and the constraint of cache space in both UAVs and FAPs, caching nodes' contents are periodically updated in the replacement phase according to the Fairness Scheduling algorithm with an Adaptive Time Window (FS-ATW) presented in [8].

In this paper, by focusing on the delivery phase, the main goal is to propose a multi-objective UAV-based femtocaching strategy based on a real dense network to determine how to select caching nodes by ground users in order to enhance the QoS, in terms of user's access delay, minimize the UAVs' energy consumption and handover phenomena. In this section, we briefly introduce some concepts required to develop the proposed CQN-CS framework. In this paper, it is assumed that all ground users randomly move in all directions with a specific velocity, that will be introduced in subsection "User Mobility Pattern." Then, we will fully describe four alternative ways to support users' requests in subsection "User Access Pattern." A summary of the notations used hereinafter is provided in Table 1.

A. USER MOBILITY PATTERN

In this work, the Difference Correlated Random Walk (DCRW) is considered to model the ground users' movement pattern, where the current position of GU_j at time slot t, denoted by $L_j(t) = [x_j(t), y_j(t)]^T$, depends on the movement's velocity. According to bivariate Stochastic Differential Equation (SDE), a ground user's velocity at time slot t, denoted by $\boldsymbol{\upsilon}_j(t) = [\upsilon_j^{(x)}(t), \upsilon_j^{(y)}(t)]^T$, is calculated as [40]

$$d\boldsymbol{\upsilon}_j(t) = -\begin{pmatrix} -\log \varsigma_1 & \theta \\ -\theta & -\log \varsigma_2 \end{pmatrix}(\boldsymbol{\upsilon}_j(t) - \boldsymbol{\mu})dt + JdB_t, \tag{2}$$

where ς_1 and ς_2 denote auto-correlation parameters in (x,y) coordinates, respectively, J denotes a (2×2) lower triangular matrix with positive diagonal components to determine the covariance of velocity shifts, and B_t represents a (2×1) vector to illustrate the standard Brownian motion at time slot t. In addition, $\boldsymbol{\mu}$ and θ signify the mean velocity vector and the DCRW model's mean turning angle, respectively. Accordingly, the location of the ground user GU_j at time slot t is obtained as [40]

$$L_j(t) = L_j(t-1) + \boldsymbol{\upsilon}_j(t-1)\Delta t, \tag{3}$$

where Δt represents the time interval between two consecutive estimated locations. To determine the initial location of ground users, we utilize the Angle of Arrival (AoA) scheme [41], [42], which has been recognized as an efficient and high accurate triangulation localization method among all the available schemes (e.g., see [43], [44]). By assuming the known position of FAPs, the initial location of ground user GU_j at time $t = 0$, denoted by $L_j(0) = [x_j(0), y_j(0)]^T$, is obtained as follows

$$x_j(0) = \frac{d_{n,i} \tan\theta_{i,j}}{\tan\theta_{i,j} - \tan\theta_{n,j}}, \tag{4}$$

$$y_j(0) = \frac{d_{n,i} \tan\theta_{n,j} \tan\theta_{i,j}}{\tan\theta_{i,j} - \tan\theta_{n,j}}, \tag{5}$$

where $\theta_{i,j}$ and $\theta_{n,j}$ represent the angle between x-axis and the line between ground user GU_j and FAPs f_i and f_n, respectively [41]. Moreover, $d_{n,i}$ denotes the distance between i^{th} and n^{th} FAPs.

TABLE 1. List of Notations.

Notation	Description	Notation	Description
N_g	Number of ground users	$\mathbf{L}_j(t)$	Location of ground user GU_j at time slot t
N_f	Number of FAPs	$\boldsymbol{v}_j(t)$	Velocity of ground user GU_j at time slot t
N_u	Number of UAVs	h_k	Altitude of UAV u_k
N_c	Number of FAPs covered by a UAV	$d_{c,k}(t)$	Euclidean distance between the cloud server and UAV u_k
$\mathcal{C}$	Number of contents	$d_{k,j}(t)$	Euclidean distance between UAV u_k and ground user GU_j
R_f	Transmission range of FAPs	$\mathcal{L}_0$	Path loss in reference distance d_0
R_u	Transmission range of UAVs	$\mathcal{L}_{k,j}^{(LoS)}(t)$	LoS path loss from UAV u_k to ground user GU_j
S_f	Storage capacity of FAPs	$\bar{\mathcal{L}}_{k,j}(t)$	Average path loss from UAV u_k to ground user GU_j
S_u	Storage capacity of UAVs	α, β	Path loss exponent of LoS and NLoS links between cloud server and UAV u_k
L_c	Size of file c_l	$\eta^{(LoS)}$	Path loss exponent of LoS
γ	Popularity skewness c_l	$\chi_\sigma^{(LoS)}$	Shadowing effect of LoS link
p_l	Probability of requesting content c_l	τ_f, τ_p	Flyby and the pause time of UAV u_k
$p_k^{(h)}(t)$	Probability of cache-hit at time slot t	$\mathcal{U}_j^{(f)}$	Set of accessible FAPs for ground user GU_j
$p_k^{(m)}(t)$	Probability of cache-miss at time slot t	$\mathcal{U}_j$	Set of FAPs f_i and UAV u_k, associated with ground user GU_j
$p_{k,j}^{(LoS)}(t)$	Probability of LoS link u_k and GU_j	$E_k^{(LoS)}(t)$	Energy consumption of UAV u_k over LoS link
$\mathcal{D}_u(t)$	User's access delay through UAVs	$P_T(t)$, $P_R(t)$	Transmission and reception powers of 1 Mb file sharing
$\mathcal{D}_f(t)$	User's access delay through FAPs	$P_j(t)$	Received power of UAV u_k at ground user GU_j
P_k	Constant transmit power of UAV u_k	P_{th}	Minimum strength signal detected by ground users
N_0	Noise power	$RSSI_{i,j}(t)$	Received signal strength by ground user GU_j from FAP f_i
$I_k(t, \mathbf{u}_{-k})$	Interference of UAV-user links for u_k	$\mathcal{HO}_i(t)$	Handover parameter

B. USERS' ACCESS PATTERN

Given the location of users, all available FAPs and UAVs in the vicinity of ground users are determined. Therefore, by requesting content by ground user GU_j, this request must be served by one of the FAPs or UAVs in the neighborhood. In some cases, however, the requested content cannot be found in the storage of either available FAPs or UAVs, due to the limited embedded storage of caching nodes. Consequently, they need to provide the corresponding content from the cloud server. To serve users' requests, there are four alternative approaches, i.e., (*i*) FAP-ground user; (*ii*) UAV-ground user; (*iii*) Cloud-FAP-ground user, and; (*iv*) Cloud-UAV-ground user links. It should be noted that the last two approaches would happen when the requested content does not exist in the storage of either FAPs or UAVs.

This completes the description of the network's model and the main assumptions required for the development of the proposed CQN-CS framework. Next, we construct a multi-objective optimization problem over which the CQN-CS framework is designed.

III. MULTI-OBJECTIVE OPTIMIZATION PROBLEM

In this section, we present users' access delay, energy consumption of UAVs, and handover of FAPs in order to formulate a multi-objective optimization problem. The constructed optimization problem is associated with the selection of an appropriate UAV and/or FAP among all candidate caching nodes to serve users' requests.

A. USERS' ACCESS DELAY

By considering the common consideration in the literature that all video files in the network are of similar size [18], [28], [39], users' access delay exclusively depends on the connection type, i.e., the user's request is served by the FAP or the UAV. Note that this assumption is only used for notational convenience as larger contents in caching-based networks could be broken into the same length packets by packetization. Additionally, another key factor that has a great impact on the user's access delay is the availability of the requested content in the storage of FAPs and/or UAVs. If the requested content can be found in the cache of FAP and/or UAV, this request is directly handled by the caching node and the cache-hit occurs, otherwise, the corresponding file must be provided by the cloud server for FAP and/or UAV to manage the request, and in this case, the ground user will experience much more delay. As will be described shortly, the user's access delay is expressed as a function of the distance between the ground user and the target caching node, and the popularity of the requested file.

1) USERS' ACCESS DELAY THROUGH UAV CONNECTION

To provide content via a UAV, an air-to-ground connection from the UAV to the ground user must be formed. In such a case that the requested content is not accessible in the UAV cache, in addition to the air-to-ground connection, a ground-to-air connection requires to be established as well, which is the wireless fronthaul link between the cloud server and the UAV. To determine the user's access delay through UAVs, due to the unavoidable obstacles throughout the network, we consider both Line-of-Sight (LoS) and Non-Line-of-Sight (NLoS) connections over the Cloud-to-UAV and the UAV-to-Ground user links as follows:

- *UAV-to-Ground User Link:* The LoS and NLoS path loss models from UAV u_k to ground user GU_j at time slot t are expressed as follows [11]

$$\mathcal{L}_{k,j}^{(LoS)}(t) = \mathcal{L}_0 + 10\eta^{(LoS)} \log(d_{k,j}(t)) + \chi_\sigma^{(LoS)}, \quad (6)$$

$$\mathcal{L}_{k,j}^{(NLoS)}(t) = \mathcal{L}_0 + 10\eta^{(NLoS)} \log(d_{k,j}(t)) + \chi_\sigma^{(NLoS)}, \quad (7)$$

where $\mathcal{L}_0 = 20\log\left(\frac{4\pi f_c d_0}{c}\right)$ represents the path loss in reference distance d_0, f_c is the carrier frequency, and c denotes the light speed. In addition, $\eta^{(LoS)}$ and $\eta^{(NLoS)}$ indicate the path loss exponents of LoS and NLoS, respectively, and $d_{k,j}(t)$ is the Euclidean distance between UAV u_k and ground user GU_j at time slot t. Moreover, $\chi_\sigma^{(LoS)}$ and $\chi_\sigma^{(NLoS)}$, as the shadowing effects, denote zero-mean Gaussian-distributed random variables with the standard deviation σ. Considering the fact that the probability of LoS connection relies on the environment, the probability of LoS link, denoted by $p_{k,j}^{(LoS)}(t)$, is expressed as [39]

$$p_{k,j}^{(LoS)}(t) = \left(1 + \psi \exp\left(-\zeta[\phi_{k,j}(t) - \psi]\right)\right)^{-1}, \quad (8)$$

where ψ and ζ are environmental constant parameters, and $\phi_{k,j}(t) = \sin^{-1}\left(\frac{h_k}{d_{k,j}(t)}\right)$ represents the elevation angle, where h_k is the altitude of the UAV u_k. In our optimization problem, we assume that each UAV u_k flies in a fixed altitude h_k over the hovering time. As a result, the average path loss between UAV u_k and ground user GU_j is calculated as

$$\overline{\mathcal{L}}_{k,j}(t) = p_{k,j}^{(LoS)}(t)\mathcal{L}_{k,j}^{(LoS)}(t) + (1 - p_{k,j}^{(LoS)}(t))\mathcal{L}_{k,j}^{(NLoS)}(t). \quad (9)$$

- *Cloud-to-UAV Link:* Similarly, since the terrain knowledge is not available, the link between UAV and the cloud cannot deterministically be identified as LoS or NLoS. Therefore, we use probabilistic mean path loss with two possible events (i.e., LoS or NLoS). The average path loss over the LoS and NLoS conditions is then computed by considering $p_{c,k}^{(LoS)}(t)$ denoting the probability of having a LoS, and $1 - p_{c,k}^{(LoS)}(t)$ representing the probability of having a NLoS link as follows

$$\overline{\mathcal{L}}_{c,k}(t) = p_{c,k}^{(LoS)}(t)\mathcal{L}_{c,k}^{(LoS)}(t) + (1 - p_{c,k}^{(LoS)}(t))\mathcal{L}_{c,k}^{(NLoS)}(t), \quad (10)$$

where $\mathcal{L}_{c,k}^{(LoS)}(t) = d_{c,k}^{-\alpha}(t)$ and $\mathcal{L}_{c,k}^{(NLoS)}(t) = \beta\mathcal{L}_{c,k}^{(LoS)}(t)$, in which $d_{c,k}(t)$ denotes the distance between the cloud server and UAV u_k, and α and β represent the path loss exponent and the additional path loss of the NLoS connection, respectively [30].

To express the user's access delay through UAVs, first we calculate the cache-hit probability, i.e., the probability of serving a request by UAV u_k at time slot t, denoted by $p_k^{(h)}(t)$, and cache-miss probability, denoted by $p_k^{(m)}(t)$, as follows

$$p_k^{(h)}(t) = \sum_{l \in \mathcal{C}_k} p_l(t) \leq 1, \quad (11)$$

$$p_k^{(m)}(t) = 1 - p_k^{(h)}(t), \quad (12)$$

where $\mathcal{C}_k$ denotes a set of contents stored in the cache of UAV u_k. We use the fact that the popularity of contents $p_l(t)$ in Eq. (11) is changing over time. Following a similar concept used to form the average path loss in Eq. (10), there are two possible events to define the user's access delay through UAV, i.e., the cache-hit event (which can happen with probability of $p_k^{(h)}(t)$) and the cache-miss event (which occurs with probability of $p_k^{(m)}(t)$). Accordingly, we have

$$\mathcal{D}_u(t) = p_k^{(h)}(t)\mathcal{D}_u^{(h)}(t) + p_k^{(m)}(t)\mathcal{D}_u^{(m)}(t), \quad (13)$$

where $\mathcal{D}_u^{(h)}(t)$, known as the cache-hit delay through UAV, is expressed as

$$\mathcal{D}_u^{(h)}(t) = \frac{L_c}{R_{k,j}} = L_c \log^{-1}\left(1 + \frac{P_k 10^{\overline{\mathcal{L}}_{k,j}(t)/10}}{I_k(t, u_{-k}) + N_0}\right), \quad (14)$$

with L_c, $R_{k,j}$, P_k, and N_0 denoting the size of file c_l, the data rate of the transmission from UAV u_k to GU_j, the constant transmit power of UAV u_k, and the noise power, respectively. Moreover, $I_k(t, u_{-k})$ indicates the interference from other UAV-to-Ground user links except for the corresponding u_k link. Similarly, the cache-miss delay, denoted by $\mathcal{D}_u^{(m)}(t)$, is calculated as

$$\mathcal{D}_u^{(m)}(t) = L_c \log^{-1}\left(1 + \frac{P_k 10^{\overline{\mathcal{L}}_{c,k}(t)/10}}{I_k(t, u_{-k}) + N_0}\right) + L_c \log^{-1}\left(1 + \frac{P_k 10^{\overline{\mathcal{L}}_{k,j}(t)/10}}{I_k(t, u_{-k}) + N_0}\right), \quad (15)$$

where the first term illustrates the user's access delay associated with the Cloud-to-UAV link, while the second term on the Right Hand Side (RHS) of Eq. (15) is related to the UAV-to-Ground user link.

2) USER's ACCESS DELAY THROUGH FAP CONNECTION

Another promising approach to provide the desired content for ground users is serving the request by FAPs, which leads to a ground-to-ground connection type between FAPs and users. Similarly, if the requested content cannot be found in the cache of neighboring FAPs, the corresponding content is transmitted by the cloud server to the FAP, and then it is sent to the ground user. Therefore, the user's access delay through the FAP connection is defined as follows

$$\mathcal{D}_f(t) = p_k^{(h)}(t)\mathcal{D}_f^{(h)}(t) + p_k^{(m)}(t)\mathcal{D}_f^{(m)}(t), \quad (16)$$

where $\mathcal{D}_f^{(h)}(t)$, as the cache-hit delay through FAPs, is expressed as

$$\mathcal{D}_f^{(h)}(t) = L_c \log^{-1}\left(1 + \frac{P_i|\tilde{\mathcal{H}}_{i,j}(t)|^2}{I_i(t, f_{-i}) + N_0}\right), \quad (17)$$

where $\tilde{\mathcal{H}}_{i,j}(t) = \frac{h_{i,j}(t)}{\sqrt{\mathcal{L}_{i,j}(t)}}$ illustrates the fading channel effect with path loss between FAP f_i and ground user GU_j at time slot t, where $g_{i,j}(t) = |h_{i,j}(t)|^2$ represents the power gain of the short-term fading channel coefficient $h_{i,j}(t)$, which is a complex zero-mean Gaussian random variable with the standard deviation equals to one, and $\mathcal{L}_{i,j}(t)$ denotes the corresponding path loss. In addition, $I_k(t, f_{-i})$ represents the interference from other FAP-user links excluding the corresponding f_i link.

B. ENERGY CONSUMPTION OF UAVs

The energy consumption of UAV u_k, due to transmitting file c_l with the size of L_c to ground user GU_j, is calculated as [15]

$$E_{u_k}^{(LoS)}(t) = L_c P_T(t)\tau_p + L_c P_R(t)\tau_p + P_j^{(LoS)}(t)(\tau_f - \tau_p), \tag{18}$$

$$E_{u_k}^{(NLoS)}(t) = L_c P_T(t)\tau_p + L_c P_R(t)\tau_p + P_j^{(NLoS)}(t)(\tau_f - \tau_p), \tag{19}$$

where τ_f and τ_p represent the flyby and the pause times of UAV u_k, respectively. Moreover, $P_T(t)$, $P_R(t)$, and $P_j(t)$ denote the transmission and reception powers of 1 Mb file sharing, and the received power at ground user GU_j, respectively, where $P_j^{(LoS)}(t)$ and $P_j^{(NLoS)}(t)$ are calculated as

$$P_j^{(LoS)}(t) = P_0^{(LoS)} - 10\eta^{(LoS)} \log\left(\frac{d_{k,j}(t)}{d_0}\right) + \chi_\sigma^{(LoS)}, \tag{20}$$

$$P_j^{(NLoS)}(t) = P_0^{(NLoS)} - 10\eta^{(NLoS)} \log\left(\frac{d_{k,j}(t)}{d_0}\right) + \chi_\sigma^{(NLoS)}, \tag{21}$$

where $P_0^{(LoS)}$ and $P_0^{(NLoS)}$ represent the received power at distance d_0 in LoS and NLoS models, respectively. Consequently, the average energy consumption of UAV u_k is

$$\overline{E}_u(t) = p_{k,j}^{(LoS)}(t)E_{u_k}^{(LoS)}(t) + (1 - p_{k,j}^{(LoS)}(t))E_{u_k}^{(NLoS)}(t). \tag{22}$$

C. HANDOVER OF FAPs

Dynamic UAV-based femtocaching networks, consisting of massively dense FAPs with small transmission ranges, are exposed to triggering frequent handovers during users' movements. By considering $RSSI_{i,j}(t)$ as the received signal strength by ground user GU_j from FAP f_i, we have

$$RSSI_{i,j}(t) = RSSI_0 + 10\eta \log_{10}\left(\frac{d_{i,j}(t)}{d_0}\right) + \chi_\sigma, \tag{23}$$

where $d_{i,j}(t)$ and d_0 denote the distance between FAP f_i and GU_j, and the reference distance equal to 1 m, respectively. Moreover, η represents the path loss exponent, which is 10 dB or 20 dB, and χ_σ is a zero-mean Gaussian with the standard deviation σ that represents the effect of shadowing in our femtocaching scheme [45].

During movement of the ground user GU_j, once $RSSI_{i,j}(t)$ drops below the threshold level P_{th}, defined as the minimum signal strength that can be detected by ground users, handover triggers, and GU_j connects to another neighboring FAP with the strongest signal. Since the received signal strength depends on the distance between the ground user GU_j and FAP f_i, the low value of $RSSI_{i,j}(t)$ indicates that GU_j is far from FAP f_i, leading to triggering handover within the shortest possible time. Moreover, taking into account a dynamic femtocaching network, where ground users move consistently in the coverage area of FAPs, it is essential to consider that the ground user GU_j is becoming close to the corresponding FAP or moving farther away during its movement. Hence, we define $\Delta_{i,j}(t)$ as follows

$$\Delta_{i,j}(t) = RSSI_{i,j}(t) - RSSI_{i,j}(t-1), \quad i = 1, \ldots, \mathcal{U}_j^{(f)}, \tag{24}$$

where $\mathcal{U}_j^{(f)}$ denotes a set of accessible FAPs for the ground user GU_j. In this case, $\Delta_{i,j}(t) = 0$, $(1 \le i \le \mathcal{U}_j^{(f)})$, indicates that the ground user GU_j is a stationary user, therefore, it should be connected to the FAP with the highest value of $RSSI$. $\Delta_{i,j}(t) > 0$ shows that the ground user GU_j is becoming close to FAP f_i, while $\Delta_{i,j}(t) < 0$, which is the worst case scenario, shows that GU_j is moving far away from FAP f_i. In order to decrease the number of handovers, after requesting a content by the ground user GU_j, the RSSI value of all candidate FAPs in the vicinity of GU_j is measured and the target FAP f_i, with the highest value of handover $\mathcal{HO}_{i,j}(t)$ is selected to serve the user's request, where $\mathcal{HO}_{i,j}(t)$ is obtained as follows

$$\mathcal{HO}_{i,j}(t) = RSSI_{i,j}(t) + \Delta_{i,j}(t). \tag{25}$$

D. PROBLEM FORMULATION

Based on the above derivations and considering the system model described in Section II, the goal here is to develop an efficient scheduling connection to assign an appropriate caching node (i.e., UAV or FAP) to ground users. To construct the optimization problem, we consider the following three objectives: (i) User's access delay; (ii) Energy consumption of UAVs, and; (iii) Handover phenomena. On the other hand, to evaluate the efficiency of the proposed CQN-CS framework, we use the following six performance metrics: (1) Cache-hit ratio; (2) User's access delay; (3) Energy consumption of UAVs; (4) Handover; (5) Lifetime of the network, and; (6) Cumulative rewards. Toward this goal, we formulate a multi-objective optimization problem with the aim of minimizing the user's access delay depending on whether the user is served through FAPs or UAVs. Access delay experienced by the j^{th} ground user, energy consumption of UAVs, and the occurrence of handovers are denoted by $\mathcal{D}_{l,j}(t) \in \{\mathcal{D}_{f_i,j}(t), \mathcal{D}_{u,j}(t)\}$, $\overline{E}_{l,j}(t)$, and $\mathcal{HO}_{l,j}(t)$, respectively. Note that subscript $l \in \mathcal{U}_j = \{u_k, f_1, \ldots, f_{\mathcal{U}_j^{(f)}}\}$ denotes all caching nodes in the vicinity of the ground user GU_j. By considering the system model in Fig. 1, our proposed wireless network consists of some clusters, where each cluster is supported by one UAV and several overlapped FAPs. Consequently, the ground user GU_j (for $1 \le j \le N_g$) at time slot t can access one UAV u_k, (for $1 \le k \le N_u$), and several FAPs f_i, $i = 1, \ldots, \mathcal{U}_j^{(f)}$. Therefore, the cardinality of all caching nodes (UAV and FAPs) supporting the ground user GU_j at time slot t is given by $|\mathcal{U}_j| = |\mathcal{U}_j^{(f)}| + 1$. One of the most efficient methods to scalarize a set of objectives into a single objective, is the Weighted Sum (WS) method, in which normalized objectives are pre-multiplied by weights ω_q, and are combined as follows

$$\mathcal{F}_j(\mathbf{x}) = \sum_{l\in\mathcal{U}_j} \Big(\omega_1\, x_l \mathcal{D}_{l,j}(t) + \omega_2\, x_l \overline{E}_{l,j}(t) - \omega_3\, x_l \mathcal{HO}_{l,j}(t)\Big), \quad (26)$$

where $\mathcal{F}_j(\mathbf{x})$ denotes the cost function associated with the ground user GU_j, and $\mathbf{x} = [x_1, \ldots, x_{\mathcal{U}_j}]$ is an indicator vector, where x_l would be 1 if l^{th} caching node serves the request of the ground user GU_j, otherwise it equals to 0. Note that x_1 illustrates the connection between GU_j and UAV u_k. Moreover, the weight coefficient ω_q should be valued in such a way that the higher value of ω_q indicates the superiority of that parameter, where $\sum_{q=1}^{3} \omega_q = 1$. Since the dimension of these three parameters is not the same, we assume normalized values, where the delay associated with UAVs and FAPs are normalized by the maximum tolerable delay of UAVs ($\mathcal{D}_u^{(max)}$) and that of FAPs ($\mathcal{D}_f^{(max)}$), respectively. Similarly, the energy consumption of UAVs and the handover parameter are normalized by E_{max} and P_{th}, respectively. More precisely, by considering the fact that ground users will experience maximum latency in the boarder of transmission range of both UAVs and FAPs, $\mathcal{D}_u^{(max)}$ and $\mathcal{D}_f^{(max)}$ can be determined according to Eqs. (13) and (16), where $d_{k,j}(t) = R_u$ and $d_{i,j}(t) = R_f$, respectively. Similarly, E_{max} is obtained according to Eq. (22), where $d_{k,j}(t) = R_u$ in Eqs. (20) and (21).

Our aim is to minimize the objective function $\mathcal{F}_j(\mathbf{x})$. Note that the minus sign for the handover parameter in Eq. (26) is due to the fact that the RSSI value linearly depends on the $\mathcal{HO}_{l,j}(t)$ in Eq. (25). Therefore, connecting to the FAP with the highest value of $\mathcal{HO}_{l,j}(t)$ leads to decreasing the number of handovers. In this case, we expand $\mathcal{D}_{l,j}(t)$ in Eq. (26) as

$$\mathcal{D}_{l,j}(t) = \mathcal{D}_{u,j}(t) + \sum_{l\in\mathcal{U}_j^{(f)}} \mathcal{D}_{f_l,j}(t). \quad (27)$$

Then, we can expand the objective function, including two terms; UAV connection, and FAP connection, as follows

$$\mathcal{F}_j(\mathbf{x}) = \underbrace{\omega_1\, x_1 \mathcal{D}_{u,j}(t) + \omega_2\, x_1 \overline{E}_{u,j}(t)}_{\triangleq \mathbf{C_{U-G}}} + \underbrace{\sum_{l\in\mathcal{U}_j^{(f)}} \omega_1\, x_l \mathcal{D}_{f_l,j}(t) - \omega_3\, x_l \mathcal{HO}_{l,j}(t)}_{\triangleq \mathbf{C_{F-G}}}, \quad (28)$$

where $\mathbf{C_{U-G}}$ denotes the cost function associated with UAVs and the ground users' connections, and $\mathbf{C_{F-G}}$ is the connection link between FAPs and the ground users. Consequently, $\mathbf{C_{U-G}}$ would be zero, if the ground user GU_j should be served by FAPs. Similarly, $\mathbf{C_{F-G}}$ would be zero, when GU_j is served by UAV u_k. To determine the time varying connection scheduling of ground users, by considering the fact that there are N_g ground users in the network, the proposed multi-objective optimization problem is expressed as follows

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{j\in N_g} \mathcal{F}_j(\mathbf{x}) \\ \text{s.t.}\ & C1.\ x_l \in \{0,1\}, \\ & C2.\ \sum_{l=1}^{\mathcal{U}_j} x_l = 1, \\ & C3.\ 0 \le \mathcal{D}_{u,j}(t) \le 1, \\ & C4.\ 0 \le \mathcal{D}_{f_l,j}(t) \le 1, \\ & C5.\ 0 \le \overline{E}_{u,j}(t) \le 1, \\ & C6.\ RSSI_{i,j}(t) \ge P_{th}. \end{aligned} \quad (29)$$

For the above optimization problem, x_l in constraint **C1** is the indicator variable. Constraint **C2** represents that the request of each ground user is served by one caching node. Constraints **C3** $\sim$ **C5** are utilized to illustrate all possible caching nodes in the vicinity of the ground user GU_j, where the normalized latency associated with candidate UAVs and FAPs, and the energy consumption constraints should be positive and less than or equal to one. Moreover, constraint **C6** represents that the RSSI value should be equal to or greater than the threshold level of RSSI (P_{th}), otherwise handover occurs. Following a similar argument as in [28], our minimization problem in Eq. (29) can be expressed as a maximization problem, i.e.,

$$\begin{aligned} \max_{\mathbf{x}} \quad & \sum_{j\in N_g} \mathcal{F}'_j(\mathbf{x}) \\ \text{s.t.}\ & C1 \sim C6, \end{aligned} \quad (30)$$

where $\mathcal{F}'_j(\mathbf{x})$ is defined as

$$\mathcal{F}'_j(\mathbf{x}) = \omega_1\, x_1\Big(1 - \mathcal{D}_{u,j}(t)\Big) + \omega_2\, x_1\Big(1 - \overline{E}_{u,j}(t)\Big) + \sum_{l\in\mathcal{U}_j^{(f)}} \omega_1\, x_l\Big(1 - \mathcal{D}_{f_l,j}(t)\Big) + \omega_3\, x_l \mathcal{HO}_{l,j}(t). \quad (31)$$

This completes our formulation of a multi-objective optimization problem to present the user's access delay, energy consumption of UAVs, and the handover of FAPs. Next, we develop the proposed CQN-CS scheduling architecture based on the developed multi-objective optimization formulation.

IV. THE CQN-CS SCHEDULING FRAMEWORK

In this section, we present an optimum framework, the CQN-CS, to identify how users access UAVs and/or FAPs based on the RL model, in order to solve the optimization problem of Eq. (30). To be specific, we first briefly introduce the required background on RL, then we present the proposed CQN-CS, which is an efficient DQN model with an embedded CNN connection scheduling architecture developed for a UAV-based femtocaching network.

A. REINFORCEMENT LEARNING

Generally speaking, RL algorithms consist of an agent, interacting with an environment based on a set of given actions.

The agent receives feedback, as a reward or punishment, from the environment after each interaction, and updates its states accordingly. Markov Decision Process (MDP) provides the rigorous mathematical foundation for RL algorithms, and includes a set of $\mathcal{A}$ actions, a set of states $\mathcal{S}$, a transition function $\mathcal{T}$, and a reward function, denoted by $\mathcal{R}$. Each action $\boldsymbol{a}_t \in \mathcal{A}$ at time slot t in any state $\boldsymbol{s}_t \in \mathcal{S}$ results in a new state $\boldsymbol{s}_{t+1} \in \mathcal{S}$ at time slot $t+1$ based on the transition function $\mathcal{T}(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})$ and a reward $r_t = \mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t)$. The aim of MDP is to find the optimum policy π^* to achieve the maximum accumulated rewards obtained over an infinite number of interactions [46], where π^* is expressed as follows

$$\pi^* = \arg\max_{\pi} \mathbb{E}_\pi \Big\{ \sum_{t=0}^{H-1} \gamma^t r_{t+1} | \boldsymbol{s}_0 = \boldsymbol{s} \Big\}, \tag{32}$$

where H denotes the number of finite episodes in MDP and $\gamma \in [0, 1]$ is the discount factor. The low value of γ leads to maximizing short-term rewards, while a high value of γ increases rewards over a longer period of time.

The Q-Network framework, as one of the most commonly used value-based and model-free RL algorithms, can be considered as a function approximator, where the value of $Q(\boldsymbol{s}_t, \boldsymbol{a}_t)$ relies on action $\boldsymbol{a}_t$ and the state $\boldsymbol{s}_t$ of the agent at time slot t, expressed as follows [46]

$$Q(\boldsymbol{s}_t, \boldsymbol{a}_t) = \mathbb{E}_\pi \Big\{ \sum_{t=0}^{H-1} \gamma^t r_{t+1} | \boldsymbol{s}_0 = \boldsymbol{s}, \boldsymbol{a}_0 = \boldsymbol{a}, \boldsymbol{a}_t = \pi(\boldsymbol{s}_t) \Big\}. \tag{33}$$

In this regard, the value of $Q(\boldsymbol{s}_t, \boldsymbol{a}_t)$ in each time slot is updated by the agent as follows

$$Q(\boldsymbol{s}_t, \boldsymbol{a}_t) \leftarrow (1-\lambda) Q(\boldsymbol{s}_t, \boldsymbol{a}_t) + \lambda (r_t + \gamma \max Q(\boldsymbol{s}_{t+1}, \boldsymbol{a}_{t+1})), \tag{34}$$

where $\lambda \in [0, 1]$ is the learning rate. If the number of states is finite, the Q-learning approach performs efficiently to update the state-action value function in each state. In scenarios, where the number of states is infinite, however, it is not feasible to visit all the states, therefore, deep learning methods can contribute to approximate the state-action value function. In deep Q-learning approaches, a deep model is used for prediction and training, instead of building a Q-table to look up and update values. In this paper, we apply CNN, as one of the most efficient deep learning methods, to estimate Q-values.

B. THE CQN-CS ARCHITECTURE

Due to the dynamic behavior of the UAV-based femtocaching network, which is a result of the mobility of ground users in the environment, we train our CQN-CS model using QoS requirements, including users' access delay, and the QoE from UAV and FAP perspectives. Fig. 2 illustrates the block diagram of our proposed CQN-CS framework. By considering a slotted structure, each ground user learns how to access the UAV and/or FAP autonomously, where each time slot t consists of the following four steps:

1) STEP 1 (LOCALIZATION)

To determine the best caching node for serving a request, we need to know all possible caching nodes in the vicinity of ground users. Toward this goal and in the first stage, the location of ground users must be estimated. According to the AoA localization technique, the location of each user is calculated based on Eqs. (4) and (5).

2) STEP 2 (CACHING NODE IDENTIFICATION)

In this phase and according to the constraint **C2** in Eq. (29), we need to determine all FAPs and UAVs in the vicinity of the ground user GU_j to build $\mathcal{U}_j = \{u_k, f_1, \ldots, f_{\mathcal{U}_j^{(f)}}\}$. Accordingly, the distance between FAP f_i and the ground user GU_j is calculated as follows

$$\sqrt{(x_j - x_{f_i})^2 + (y_j - y_{f_i})^2} \leq R_f, \tag{35}$$

where (x_j, y_j) and (x_{f_i}, y_{f_i}) represent spatial coordinates of GU_j and FAP f_i, respectively. Therefore, all ground users that are positioned in the transmission range of FAP f_j can be supported by the corresponding FAP. Similarly, to calculate the distance between UAV u_k and the ground user GU_j, we have

$$\sqrt{(x_j - x_{u_k})^2 + (y_j - y_{u_k})^2 + h_{u_k}^2} \leq R_u, \tag{36}$$

where $(x_{u_k}, y_{u_k}, h_{u_k})$ is the spatial coordinate of UAV u_k. Since the height of ground users is much lower than the altitude of flying UAVs, the height of ground users is negligible. Note that the above statements are equivalent to constraints **C3** $\sim$ **C5**. According to Eqs. (35) and (36), the set of available candidate caching nodes for the ground user GU_j, denoted by $\mathcal{U}_j$, is built.

3) STEP 3 (QoE BROADCASTING)

Given $\mathcal{U}_j$, constructed from Step 2 above, all FAPs and UAVs in the vicinity of the ground user GU_j broadcast the energy consumption and the probability of handover, calculated based on Eqs. (22) and (25). Considering these decision criteria, consequently, results in the minimization of the user's access delay in the next time slots, obtained according to Eqs. (13) and (16). Then, the cost of selecting FAPs and UAVs is calculated by ground users based on the QoE, and air-to-air and air-to-ground channel path loss models. Consequently, all parameters in Eq. (30) are known. In the next stage, ground users will select the target caching node, either UAVs or FAPs, in order to maximize Eq. (30). Accordingly, to satisfy constraints **C1** and **C2**, only x_l associated with the target caching node would be 1.

4) STEP 4 (RESPONDING A REQUEST)

To solve Eq. (30), we propose a CNN-based Q-learning approach in the context of the UAV-based femtocaching network. Our CQN-CS framework, represented as a tuple $\{\boldsymbol{s}_t, \boldsymbol{a}_t, r_t\}$, has the following components:

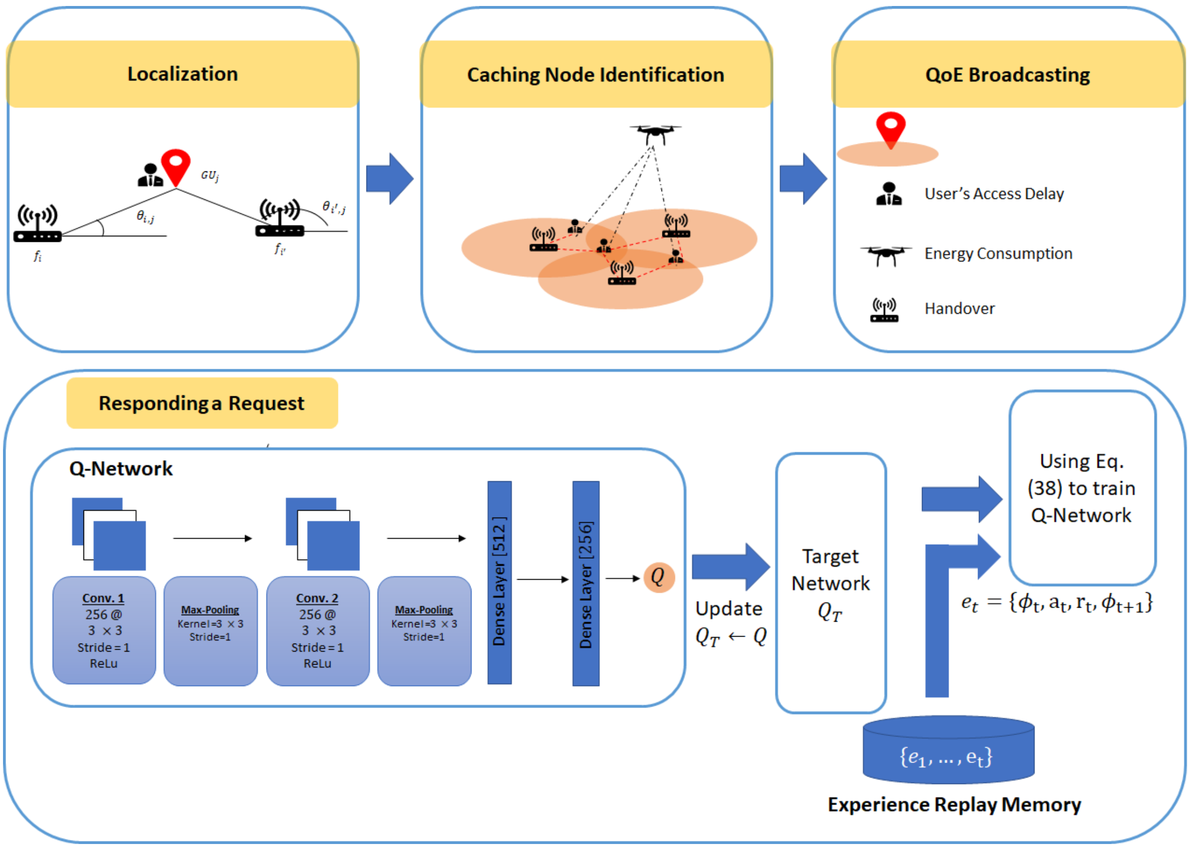


FIGURE 2. The block diagram of our proposed CQN-CS framework.

a: CQN-CS AGENTS

In our proposed CQN-CS framework, ground users, as the intelligent members of our problem, act as agents.

b: CQN-CS ACTION SPACE

The CQN-CS architecture has an action generation engine to find a globally optimized policy based on the previous states to learn the most suitable action. Here, the action refers to the selection of a suitable caching node to support ground users' requests in various circumstances in terms of the location of ground users, the battery life of UAVs, channels' condition, and the probability of handover. After requesting a content by user GU_j, this request should be served by one of the caching nodes in the vicinity of GU_j, denoted by $l \in \mathcal{U}_j = \{u_k^{(j)}, f_1^{(j)}, \dots, f_{\mathcal{U}_j^{(f)}}^{(j)}\}$, where superscript (j), $j = 1, \dots N_g$, indicates all ground users in the network. Therefore, all UAVs and FAPs in this set can be an action for GU_j, i.e., $\boldsymbol{a}_t = [u_k^{(1)}, f_1^{(1)}, \dots, f_{\mathcal{U}_1^{(f)}}^{(1)}, \dots, u_k^{(N_g)}, f_1^{(N_g)}, \dots, f_{\mathcal{U}_{N_g}^{(f)}}^{(N_g)}]^T$, for $(1 \leq k \leq N_u)$, which is equivalent to $\mathbf{x} = [x_1^{(1)}, \dots, x_{\mathcal{U}_1}^{(1)}, \dots, x_1^{(N_g)}, \dots, x_{\mathcal{U}_{N_g}}^{(N_g)}]^T$ in Eq. (30). Therefore, by selecting an action, the corresponding value of $x_l^{(j)}$ would be 1, otherwise it would be 0. Note that selecting the optimal action leads to maximizing the reward of the network.

c: CQN-CS STATE

The action is selected based on the current system state $\boldsymbol{s}_t$ at time slot t. In each time slot t, the value of user's access delay via UAVs and FAPs, energy consumption of UAVs, and handover represent states in our proposed framework, i.e., $\boldsymbol{s}_t = [L_j(t), \mathcal{D}'_{u,j}(t), \overline{E}'_{u,j}(t), \mathcal{D}'_{f_l,j}(t), \mathcal{HO}'_{l,j}(t)]^T$. More specifically, $\boldsymbol{s}_t$ consists of the following five components:

- $L_j(t)$: The location of ground user GU_j at time slot t, which is determined according to Eq. (3).
- $\mathcal{D}'_{u,j}(t)$: The total delay that the ground user GU_j will experience if its request is served by UAV u_k until time slot t.
- $\overline{E}'_{u,j}(t)$: The total energy consumed by UAV u_k at time slot t for establishing a connection with the ground user GU_j.
- $\mathcal{D}'_{f_l,j}(t)$: The total user's access delay GU_j due to the connection with FAP f_l.
- $\mathcal{HO}'_{l,j}(t)$: The handover indicator of FAP f_l at the user side GU_j.

Therefore, each ground user should build a Q-table, where rows denote all possible states, and columns indicate actions. However, since the number of states is infinite, it is not possible to visit all the states. For this reason, we apply CNN on the Q-Network to estimate Q-values. It should be noted that by serving a request by FAPs, the energy consumption of UAVs has no change, therefore, the state value of the energy consumption would be constant. Moreover, since we assume that the handover phenomenon just occurs in FAPs and ground users' connection, there is no change in the state value of handover when a request is served by UAVs. In such a case that more than one ground user simultaneously selects a specific caching node, the corresponding caching node will randomly select one of them to support the request.

d: CQN-CS REWARD

According to the optimization problem in Eq. (30), any reduction in the involved metrics (i.e., user's access delay via UAVs and FAPs, energy consumption of UAVs, and handover) results in a higher reward. When it comes to the problem of finding optimal action policies within the Pareto optimal set in scenarios with multiple and conflicting objectives, different multi-objective RL-based models [47]–[49] were considered in the literature. In this paper, we employ the weighted-sum approach, where the underlying set of objectives is converted into a single function by assigning a pre-defined weight to each individual objective. More specifically, the reward function in our multi-objective RL-based framework, denoted by $\mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t)$, is extended to a vector reward function, denoted by $\mathbf{R}(\boldsymbol{s}_t, \boldsymbol{a}_t) = [\mathcal{R}_1(\boldsymbol{s}_t, \boldsymbol{a}_t), \mathcal{R}_2(\boldsymbol{s}_t, \boldsymbol{a}_t), \mathcal{R}_3(\boldsymbol{s}_t, \boldsymbol{a}_t)]$, where $\mathcal{R}_1(\boldsymbol{s}_t, \boldsymbol{a}_t)$, $\mathcal{R}_2(\boldsymbol{s}_t, \boldsymbol{a}_t)$, and $\mathcal{R}_3(\boldsymbol{s}_t, \boldsymbol{a}_t)$ represent the reward function associated with the users' access delay, energy consumption of UAVs, and FAPs' handover, respectively, calculated as follows:

$$\mathcal{R}_1(\boldsymbol{s}_t, \boldsymbol{a}_t) = \begin{cases} 1 - \mathcal{D}_{u,j}(t), & \text{UAV Link,} \\ 1 - \mathcal{D}_{f_l,j}(t), & \text{FAP Link.,} \end{cases} \tag{37}$$

$$\mathcal{R}_2(\boldsymbol{s}_t, \boldsymbol{a}_t) = 1 - \overline{E}_{u,j}(t), \tag{38}$$

$$\text{and} \quad \mathcal{R}_3(\boldsymbol{s}_t, \boldsymbol{a}_t) = \mathcal{HO}_{l,j}(t). \tag{39}$$

In this case, the weighted-sum form of the Q-value is calculated as

$$Q(\boldsymbol{s}_t, \boldsymbol{a}_t) = \sum_{q=1}^{N_o=3} \omega_q Q_q(\boldsymbol{s}_t, \boldsymbol{a}_t), \tag{40}$$

where N_o denotes the number of incorporated objectives. Finally, the action associated with the largest $Q(\boldsymbol{s}_t, \boldsymbol{a}_t)$ is selected by an agent [49]. Considering the Pareto optimality, the trade-off is parameterized by the weight coefficients $\omega_q \in [0, 1]$, where $\sum_{q=1}^{N_o} \omega_q = 1$ [49]. In this case, higher values of ω_q indicate the superiority of that objective. In this paper, we have three conflicting objectives, i.e., users' access delay, energy consumption of UAVs, and handover phenomena occurred between FAPs. To maintain a balance between the QoS of ground users and the QoE of FAPs and UAVs, the proposed connection scheduling framework aims to simultaneously satisfy concerns of ground users, UAVs, and FAPs. For this reason, we assign equal weights to the three underlying objectives. In this case, the policy is said to be Pareto optimal if the value of π^*, obtained according to Eq. (32), strictly dominates or is incomparable with the value functions of other policies [49].

After selecting a caching node with the largest $Q(\boldsymbol{s}_t, \boldsymbol{a}_t)$, the connection information associated with the corresponding action, the location of the ground user, the probability of handover, the energy consumption of UAVs, and the user's access delay are stored in the memory replay of the proposed CQN model. In our proposed framework, for each state-action pair, CQN approximates the Q-function by using CNN with tunable weight parameters, which is a non-linear approximator. The CNN model, however, needs to be retrained due to the mobility of ground users and the dynamic nature of UAVs (i.e., the battery life). Therefore, a replay memory is used for past experienced state-action pairs and the associated rewards. The weight of filters in each layer at time slot t is denoted by ξ_t. The observed state sequence, including β state-action pairs at time slot t, is denoted by $\boldsymbol{\phi}_t = [\boldsymbol{s}_{t-\beta}, \boldsymbol{a}_{t-\beta}, \ldots, \boldsymbol{a}_{t-1}, \boldsymbol{s}_t]$, which is the input of the CNN to estimate $Q(\boldsymbol{\phi}_t, \boldsymbol{a}_t|\xi_t)$. The experience memory pool is denoted by $D = \{e_1, \ldots, e_t\}$, where $e_t = (\boldsymbol{\phi}_t, \boldsymbol{a}_t, r_t, \boldsymbol{\phi}_{t+1})$. The state sequence in replay buffer e_m is selected randomly to update the weight parameter ξ_t according to the Stochastic Gradient Descent (SGD) method. By choosing ξ_t, our goal is to minimize the loss function, denoted by $\mathbb{L}(\xi_t)$, which is the mean-squared error of the target optimal Q-function with the minibatch updates, given by

$$\mathbb{L}(\xi_t) = \mathbb{E}_{\boldsymbol{\phi}_t, \boldsymbol{a}_t, r_t, \boldsymbol{\phi}_{t+1}} \Big[\big(Q_T - Q(\boldsymbol{\phi}_t, \boldsymbol{a}_t|\xi_{t+1}) \big)^2 \Big], \tag{41}$$

where Q_T is the target optimal Q-function, given by

$$Q_T = r_t + \gamma \max_{\boldsymbol{a}'_t} Q(\boldsymbol{\phi}_{t+1}, \boldsymbol{a}'_t|\xi_{t-1})). \tag{42}$$

Eventually, the action $\boldsymbol{a}_t$ is chosen for the state $\boldsymbol{s}_t$ based on the ϵ-greedy algorithm. With the probability of $(1-\epsilon)$, the best action $\boldsymbol{a}^*_t$ is chosen from the set of Q-functions as follows

$$\boldsymbol{a}^*_t = \arg\max_{\boldsymbol{a}'_t} Q(\boldsymbol{\phi}_t, \boldsymbol{a}'_t). \tag{43}$$

Then, the user's request should be served by the corresponding caching node with action $\boldsymbol{a}^*_t$. Accordingly, the reward r_t is calculated by agent, and the new experience $\{\boldsymbol{\phi}_t, \boldsymbol{a}_t, r_t, \boldsymbol{\phi}_{t+1}\}$ is stored in the replay memory by agent. The pseudo-code of our proposed CQN-CS framework is outlined in **Algorithm 1**. The rationale behind the design of **Algorithm 1** is described in more details based on the following steps:

- *Initialization:* In each epoch, all parameters are selected according to the values shown in Table 2. Moreover, we reset all the parameters related to the environment such as the replay buffer. These actions are equivalent to Lines 1-5 of Algorithm 1.

There are N_g number of ground users who are agents in the proposed network model. The following steps will be performed for all ground users in each episode, denoted by t.

- *Localization:* To construct the action space of each agent, the current location of ground users and the

Algorithm 1 The CQN-CS Framework

1: **Initialization:**
2: Set all parameters according to Table 2.
3: Initialize state-action pairs β.
4: Initialize batch size B.
5: Reset replay memory size.
6: **for** $j = 1, \ldots, N_g$ **do**
7: **for** $t = 1, 2, \ldots$ **do**
8: **Input:**
9: $s_0 = [L_j(0), \mathcal{D}'_{u,j}(0), \overline{E}'_{u,j}(0), \mathcal{D}'_{fl,j}(0), \mathcal{HO}'_{l,j}(0)]$
10: **Localization:** Determine the location of ground
11: user GU_j based on Eqs. (4) and (5).
12: **Identification:** Determine all possible caching
13: nodes in the vicinity of GU_j.
14: **QoE broadcasting:** Calculate and broadcast the
15: energy consumption, handover, and delay.
16: **Responding to a Request:** Select the best
17: caching node according to the following steps
18: **if** $t \leq \beta$ **then**
19: Choose an action randomly $\boldsymbol{a}_t \in \{1, \ldots, \mathcal{U}_j\}$
20: **else**
21: Obtain CNN output $Q(\boldsymbol{\phi}_t, \boldsymbol{a}_t|\xi_t)$ with input
22: $\boldsymbol{\phi}_t$ and weights ξ_t.
23: Choose $\boldsymbol{a}_t$ via ϵ-greedy algorithm
24: **end if**
25: Observe $L_j(t), \mathcal{D}'_{u,j}(t), \overline{E}'_{u,j}(t), \mathcal{D}'_{fl,j}(t), \mathcal{HO}'_{l,j}(t))$
26: Estimate the Reward r_t and obtain
27: $s_{t+1} = [L_j(t+1), \mathcal{D}'_{u,j}(t+1), \overline{E}'_{u,j}(t+1),$
28: $\mathcal{D}'_{fl,j}(t+1), \mathcal{HO}'_{l,j}(t+1))]$
29: Create state, action, and reward vector:
30: $\boldsymbol{\phi}_{t+1} = [\boldsymbol{s}_{t-\beta+1}, \boldsymbol{a}_{t-\beta+1}, \ldots, \boldsymbol{a}_{t+1}, \boldsymbol{s}_{t+1}]^T$
31: Add the new experience $\{\boldsymbol{\phi}_t, \boldsymbol{a}_t, r_t, \boldsymbol{\phi}_{t+1}\}$
32: to memory D
33: **for** $d = 1, \ldots, B$ **do**
34: Select randomly $(\phi_d, \boldsymbol{a}_d, r_d, \phi_{d+1})$ from D
35: Train CNN for N iterations
36: Calculate Q_T using Eq. (42).
37: **end for**
38: Update the weight parameter ξ_t using Eq. (41).
39: **end for**
40: **end for**
41: **Output:** Optimal UAV/FAP connection scheduling with maximum r_t.

possible caching nodes in their vicinity are gathered. After initializing the location of ground users in the environment, the current location of ground users in each episode is determined according to Lines 10 and 11 of Algorithm 1.

- *Caching Node Identification:* After identifying the location of agents, all available caching nodes in the vicinity of ground users should be determined. In this case, the action space is constructed for each agent, which is equivalent to Lines 12 and 13 of Algorithm 1.
- *QoE broadcasting:* To update the state space of ground users, all UAVs and FAPs in the vicinity of each ground user broadcast their energy consumption and experienced handover, which is equivalent to Lines 14 and 15 of Algorithm 1.
- *Responding to a Request:* Considering the current state space, each agent selects the best caching node in such a way that its Q-value is maximized. If the value of the current episode is less than a pre-specified threshold β, a random action with a probability of ϵ is selected (Lines 18 and 19); otherwise, an optimal action is selected based on the ϵ-greedy policy (Lines 21-24). After taking an action, the state-space, the action-space, and the reward vector are updated and stored in the memory of the CQN model (Lines 25-32). For each state-action pair, CQN approximates the Q-function using CNN with tunable weight parameters (Lines 33-37).
- *Termination:* At the end, weight parameters are selected in such a way that the loss function, expressed in Eq. (41), is minimized. Consequently, the optimal action, which leads to increasing the Q-value, is selected by each agent at episode t.

C. COMPUTATIONAL COMPLEXITY

As previously mentioned, due to the infinite number of states in the UAV-based femtocaching framework, the size of the action-state space observed by each ground user is relatively high. In such scenarios with a high number of state-action pairs, the computational cost of conventional Q-learning algorithms is significantly high. Therefore, Deep Q-Learning (DQN) models are typically used where instead of storing expected rewards associated with each state-action pair in a Q-table, a Deep Neural Network (DNN) model is used to select the actions according to the agent's current state [50]. In complex problems such as the one at hand, several information sources (such as the position of agents, their movement directions, and available caching nodes) are simultaneously required to perform the action selection task. CNN architecture is an attractive solution to extract the relevant features from this pool of information. CNN-based architecture uses convolutional kernels to compress the state-space and extract temporal correlations between the current state of a ground user and previous state-action pairs. Within the CNN architecture, weights are shared between the episodes, which leads to a considerable reduction in the computational complexity. To compute the computational complexity of the proposed learning method, we follow the approach introduced in Reference [51] as CNN constitutes the main component (computational wise) of the proposed CQN-CS framework. The computational complexity of CNN depends on the number of multiplications in each convolutional layer [51]. Generally speaking, CNN models consist of $\mathcal{N}_l$ number of convolutional layers, where each layer includes F_l filters with size $W_l^f \times L_l^f$, zero-padded by P_l number of padding layers, and with stride size of S_l. Moreover, there are $\mathcal{N}_{fc}$ number of fully connected layers, including $\mathcal{N}_r$ number of Rectified Linear Units (ReLUs) to estimate the Q-value associated with each possible action. By considering the fact that the pooling and fully connected layers take only $5-10\%$ of the computational time [51], the computational complexity of CNN can be expressed as follows

$$\mathcal{N} = \sum_{l=1}^{\mathcal{N}_l} F_{l-1} W_l^f L_l^f F_l W_l^o L_l^o, \tag{44}$$

where l is the index of the convolutional layer. Terms F_{l-1} and F_l denote the number of input channels and the number of filters of l^{th} layer, respectively. In addition, W_l^o and L_l^o represent the width and the length of the output, calculated as follows

$$W_l^o = \frac{W_{l-1}^o - W_l^f + 2P_l}{S_l} + 1, \tag{45}$$

$$\text{and } L_l^o = \frac{L_{l-1}^o - L_l^f + 2P_l}{S_l} + 1, \tag{46}$$

where S_l and P_l are the size of stride and padding layers corresponding to the l^{th} layer, respectively. To calculate the computational complexity of the learning process of the proposed CQN-CS according to Eqs. (44)-(46), the value of $F_0 \times W_0^o \times L_0^o$ is equal to $\beta l_s \mathcal{U}$. In this case, β is the temporal memory depth, l_s is the length of the state space that is equal to 5, and $\mathcal{U}$ represents the number of possible actions in each episode.

V. SIMULATION RESULTS

In this section, we evaluate the performance of our proposed CQN-CS UAV-based femtocaching framework in terms of the cache-hit ratio, user's access delay, energy consumption of UAVs, handover, cumulative rewards, and the lifetime of the network. For the scenarios under simulation, we investigate how the CQN-based connection scheduling scheme affects the aforementioned performance metrics.

Simulation Setup: We consider an ultra-dense network with the radius $R = 5000$ m, covered by a cloud server. In our proposed network, there are $N_f = 240$ FAPs with $R_f = 30$ m, uniformly distributed in the network, where each FAP overlaps with neighboring FAPs. Additionally, by considering the restrictions of the aviation regulations, we consider $N_u = 10$ UAVs flying horizontally at a specific altitude $h_k = 100$ m, for $(k = 1, \ldots, N_u)$ to cover a region of the network with $R_u = 500$ m. The area of interest is divided into K clusters according to the K-means algorithm, where $N_c \in \{20, 25, 30\}$ neighboring FAPs are covered by a UAV. Table 2 illustrates a list of parameters.

To consider the dynamic nature of the environment, we assume the Difference Correlated Random Walk (DCRW) model for the mobility pattern of ground users. By requesting a content by ground users, each UAV hovers at its location or flies a distance to manage the request, which both of them are energy consuming. However, since these concepts have been well studied in the literature of positioning management of UAVs, we only focus on the energy consumed by stationary UAVs to handle a request, depending on the distance between the UAV and the ground user, channel condition, and availability of the requested content in the cache of UAV. The proposed CQN-CS framework is performed for 100 epochs. By considering the fact that the battery life of UAVs is limited, we assume that each epoch is terminated if the total energy consumption of at least one UAV exceeds the UAV's battery life. Our CNN model consists of two 2-dimensional convolutional layers and two Fully Connected (FC) layers. Convolutional layers consist of 256 filters, each with the size of 3 and stride 1. We use max-pooling and ReLU as the activation function in each layer. The first FC layer consists of 512 ReLU units and the second FC has 256 ReLU units. To the best of our knowledge, there is no coupled UAV and femtocaching framework, studied from the connection scheduling perspective, for comparison purpose. Therefore, we introduce two baseline models for comparison:

TABLE 2. List of Parameters.

Notation	Value	Notation	Value
N_g	500	$\eta^{(LoS)}, \eta^{(NLoS)}$	2.5, 3 [15]
N_f	240	h_k	100 m [52]
N_u	10	α, β	2, 20 [39]
N_c	$\{20, 25, 30\}$	L_c	37.5 MB [8]
R_f	30 m [54]	τ_p	$0-5$ s [15]
R_u	500 m	P_k	15 dBm [39]
S_f	22.5 GB	$\chi_\sigma^{(LoS)}, \chi_\sigma^{(NLoS)}$	3.5, 3 [15]
S_u	22.5 GB	P_{th}	-67 dBm [36]
$\mathcal{C}$	184670	N_0	-94 dBm [39]
$P_T(t), P_R(t)$	0.5 , 0.25 W [15]	γ	0.6 [53]

- ***Q-Network Connection Scheduling (QN-CS) UAV-based Femtocaching Scheme:*** In this algorithm, the best caching node to handle users' requests is selected as the result of a Q-learning framework. All parameters of the RL approach, including actions, states, and rewards are kept similar to our proposed scheme for a fair comparison.
- ***Deep Q-Network Connection Scheduling (DQN-CS) UAV-based Femtocaching Scheme:*** Similar to the previous baseline, all the parameters are the same as our proposed framework, with the difference that in this baseline, we use Multilayer Perceptron (MLP) instead of CNN with two hidden layers, where each layer consists of 256 neurons.

Taking the above considerations into account, we illustrate the superiority of our proposed CQN-CS framework compared with conventional schemes from the aspect of the cache-hit ratio, user's access delay, energy consumption of UAVs, handover, cumulative rewards, and the lifetime of the network.

A. PERFORMANCE EVALUATION

In this subsection, we first evaluate the effectiveness of the AoA scheme as an efficient localization method to estimate the proximity of ground users, in order to determine possible caching nodes in their vicinity. Fig. 3 illustrates a typical 20 × 20 m^2 area, where ground users are randomly distributed. By Assuming (x_j, y_j) and $(\hat{x}_j, \hat{y}_j)$ as the real and the estimated coordinates of user GU_j, respectively, the Root Mean Square Error (RMSE) is defined as $R_{RMSE} = \sqrt{\frac{1}{N_g}\sum_{j=1}^{N_g}(\hat{x}_j - x_j)^2 + (\hat{y}_j - y_j)^2}$ to evaluate the accuracy of location estimation. Taking into account the multipath and path loss effects, the RMSE of the AoA method in our proposed network is about 0.4 m, which is acceptable in comparison to the transmission range of FAPs.

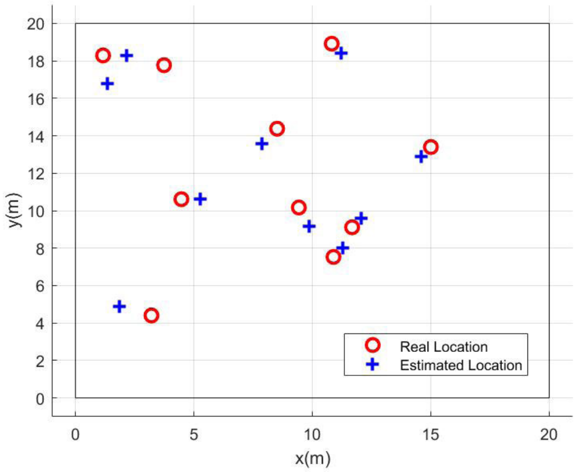


FIGURE 3. A typical location estimation result based on the AoA localization scheme.

We evaluate the convergence of the proposed CQN-CS framework in Fig. 4. More specifically, the convergence of the proposed CQN-CS framework is the crucial property to obtain a policy, which maps states to the optimal actions. According to Eq. (41), the main goal of the learning process is to minimize the loss function, which is the mean-squared error of the target optimal Q-function with the minibatch updates. Fig. 4 illustrates the convergence behavior of the proposed CQN-CS scheme. According to the result shown in Fig. 4, the CQN-CS framework converges after 80 epochs, which is an acceptable speed.

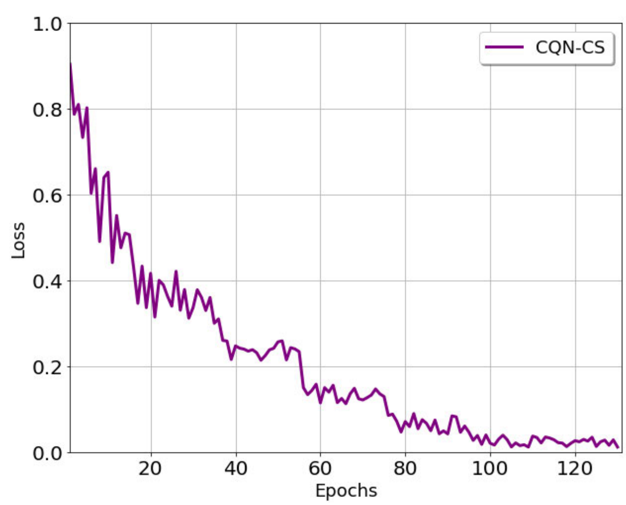


FIGURE 4. The convergence of the proposed CQN-CS framework.

Cache-hit ratio is a metric used to express the number of requests served by caching nodes, either FAPs or UAVs, in each episode. We assume a pre-specified threshold for the battery life of UAVs, where reaching the energy consumption of at least one UAV to the threshold level is known as the *game over* in our RL network. The normalized cache-hit-ratio, denoted by $\mathcal{CH}^{(n)}$, consists of two terms, i.e., the satisfied requests served by FAPs, denoted by $\mathcal{CH}_f$, and the satisfied requests managed by UAVs, denoted by $\mathcal{CH}_u$, given by

$$\mathcal{CH}^{(n)} = \frac{\mathcal{CH}_f + \mathcal{CH}_u}{\mathcal{CH}_{max}}, \tag{47}$$

where $\mathcal{CH}_{max}$ is the maximum value of $\mathcal{CH}$ in all episodes. Fig. 5 evaluates the performance of our proposed scheme and other two baselines mentioned above from the aspect of the cache-hit ratio in different epochs. According to the results in Fig. 5, we can see that the area below the curve increases as the number of epochs grows. This is due to the fact that by passing the time, our network learns how to manage requests to expand the lifetime of UAVs. Moreover, our proposed CQN-CS method experiences more cache-hit-ratio, which indicates its superiority.

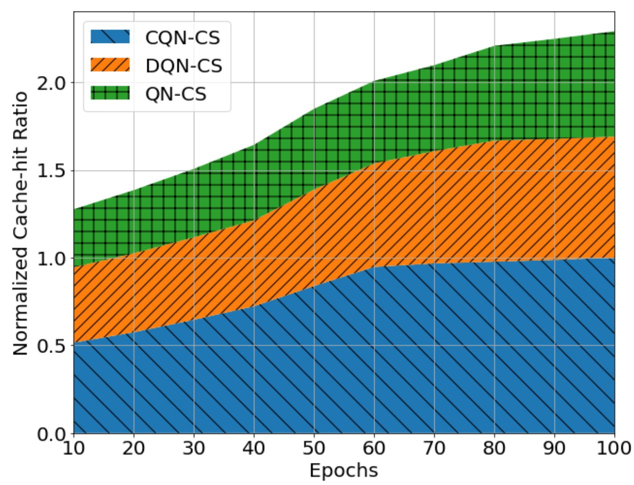


FIGURE 5. The normalized cache-hit ratio versus different epochs.

Fig. 6 illustrates the cumulative rewards of all caching nodes in each epoch before the energy consumption of at least one UAV reaches the battery life of the UAVs. According to the definition of rewards in our CQN-CS framework, connecting to the nearest FAP instead of the far one leads to a remarkable reduction in the user's access delay and handover, followed by increasing the reward value. UAV's connection would be efficient in such cases where there is no available FAPs, and/or when serving by UAVs leads to the lower experienced latency than by FAPs. Fig. 6 illustrates

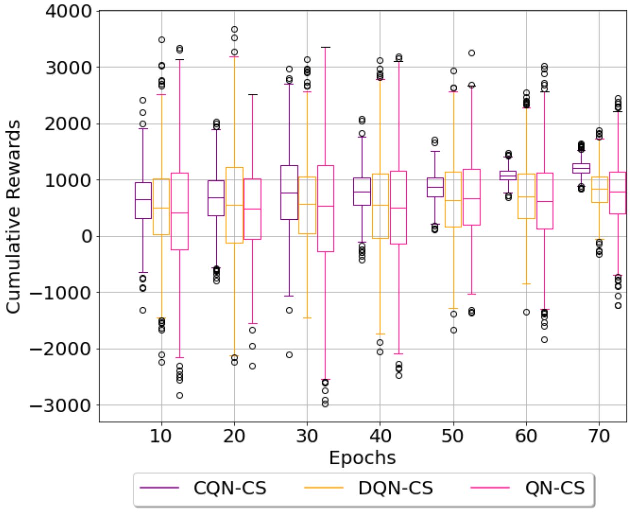


FIGURE 6. The variation of cumulative rewards versus different epochs.

the distribution of cumulative rewards for different clusters in each epoch, where the number of clusters is equal to the number of UAVs (i.e., 10). Then, we run the program for 10 iterations. As it can be seen from Fig. 6, the cumulative rewards variations of the proposed CQN-CS framework is much less than the DQN-CS and the QN-CS approaches, which means our proposed algorithm reaches the optimum connection scheduling immediately.

Fig. 7 evaluates the performance of the proposed CQN-CS algorithm with other schemes from the aspect of the network's lifetime, depending on the energy consumption of UAVs. Note that there are three parameters which have a great impact on the energy consumption of UAVs; (*i*) The number of requests they served; (*ii*) The distance between the requested ground user and the UAV, and; (*iii*) The probability of the existence of LoS link between them. By considering all these metrics in our proposed scheme, UAVs are involved in such communications that consume less energy as much as possible, which leads to expanding the network's lifetime.

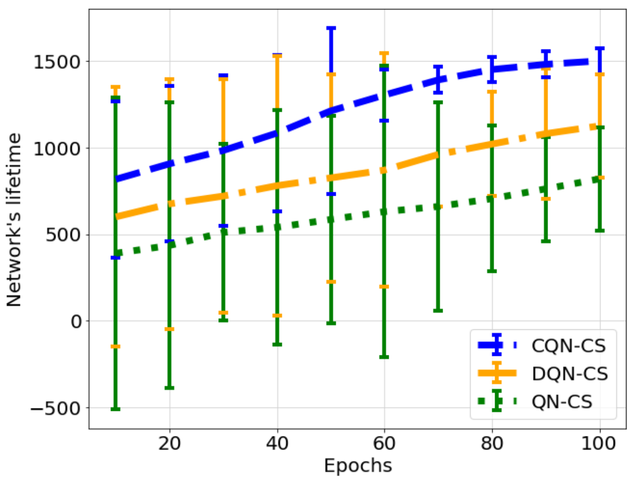


FIGURE 7. Normalized lifetime of the network in each epoch.

Fig. 8 compares the normalized average delay that all ground users in the network experience through the proposed CQN-CS, the DQN-CS, and the QN-CS frameworks. Note that the users' access delay depends on the availability of the requested content in the cache of the responsible FAP and/or UAV, otherwise, the corresponding content must be provided by the cloud server, leading to more delay. Taking into account that the location of the ground user has a great impact on the distance between the responsible caching node and the ground user, followed by the channel condition, different ground users may experience a wide range of latency. As it can be seen from Fig. 8, ground users during their movement and by considering the unforeseen conditions of the UAV-based femtocaching network, learn how to manage their requests by optimal caching nodes to experience lower latency.

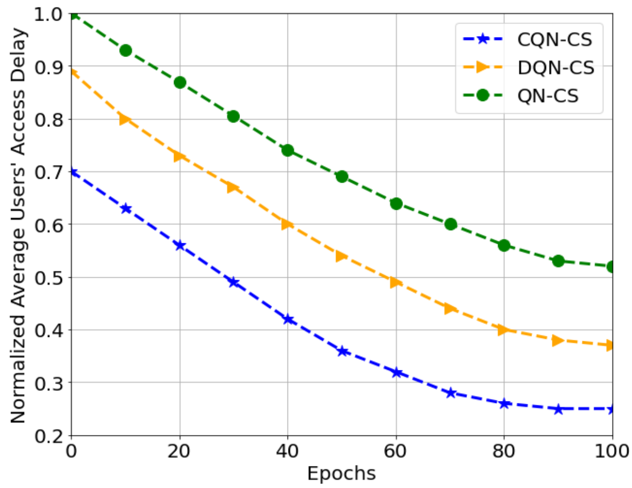


FIGURE 8. The normalized average users' access delay versus different epochs.

In Fig. 9, we evaluate the handover rate in the proposed CQN-CS framework along with other baselines versus the number of epochs. Handover rate, denoted by $\mathcal{HR}(t)$, indicates the probability of handover, occurring between FAP f_i and the ground user GU_j at time slot t, which is obtained as

$$\mathcal{HR}(t) = \frac{P_{th}}{\mathcal{HO}_{i,j}(t)}, \tag{48}$$

where the high value of $\mathcal{HO}_{i,j}(t)$ means that the ground user connects to the close FAP instead of the far one, leading to a decrease in the handover rate. In a dynamic femtocaching network, however, ground users may become close or farther away from the nearest FAP during their movements, which is considered in our proposed CQN-CS framework, denoted by $\Delta_{i,j}(t)$. To illustrate the handover rate improvement of our proposed CQN-CS framework, we compare the handover rate, with the case that $\Delta_{i,j}(t)$ is disregarded, which is named CQN-CS2 framework. Fig. 9 illustrates the superiority of our proposed CQN-CS framework in terms of the average handover rate.

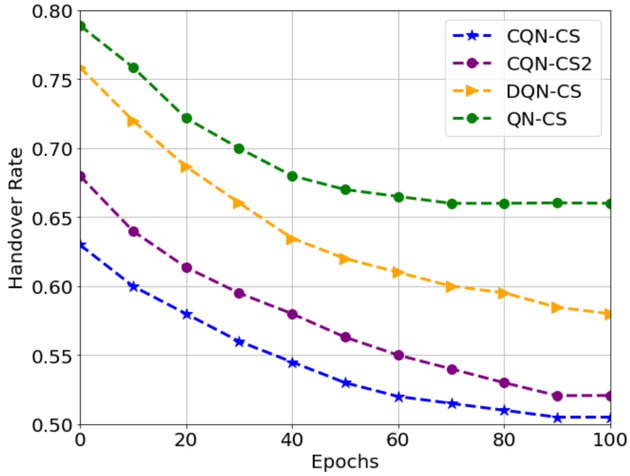


FIGURE 9. **Handover rates versus different epochs.**

Finally, we compare the performance of the proposed CQN-CS framework with the DQN-CS and QN-CS baselines from the aspect of the average energy consumption of UAVs in Fig. 10. Using the fact that the coverage areas of UAVs are much more expanded than FAPs, all users that have access FAPs can be managed by UAVs, as well. Despite FAPs, that are unlimited energy caching nodes, serving through UAVs inherently decreases the lifetime of UAVs' battery. Consequently, it is essential to manage ground users by FAPs, especially in such cases that ground users have access to at least one FAP to extend the lifetime of the network. As it can be seen from Fig. 10, the average normalized energy consumption of UAVs in our proposed CQN-CS framework is lower than the other schemes. In addition, the variation of the energy consumption of UAVs is negligible in comparison to the two baselines.

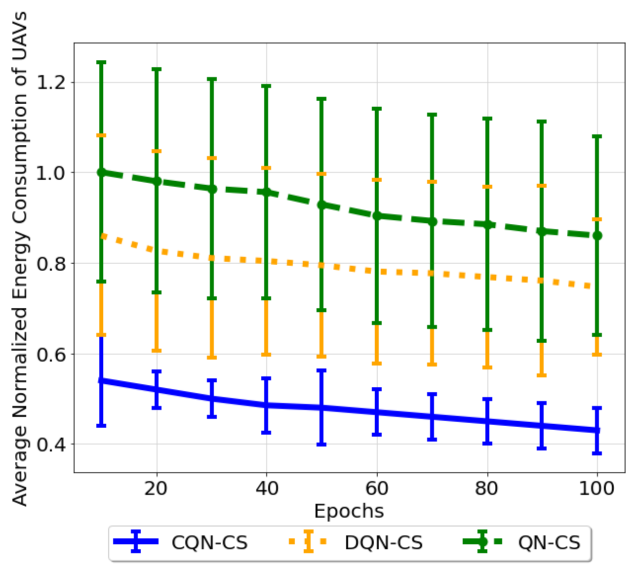


FIGURE 10. **The normalized average energy consumption of UAVs versus different epochs.**

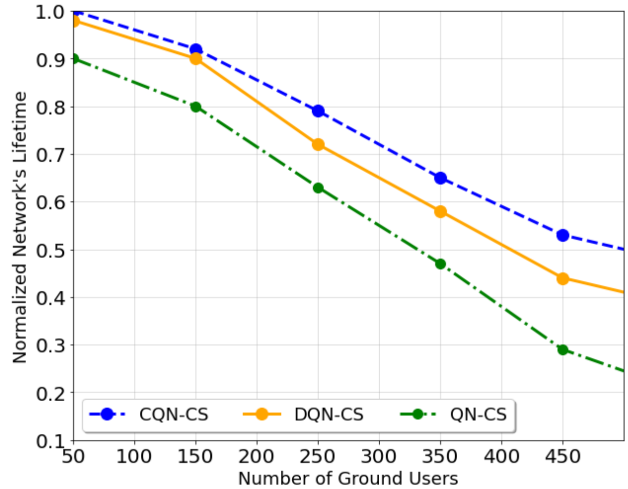


FIGURE 11. **Normalized network's lifetime versus number of ground users.**

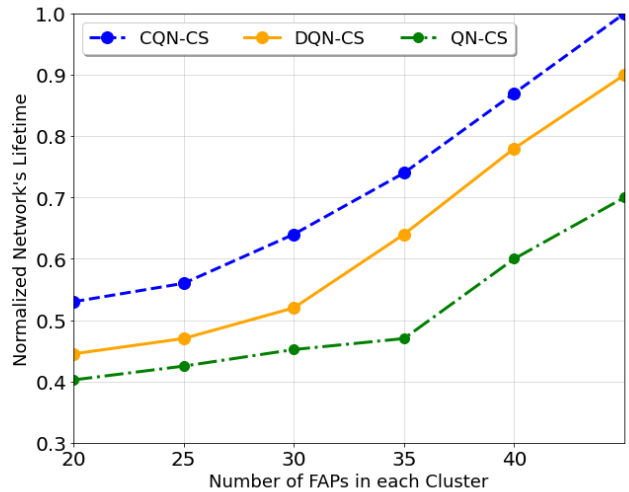


FIGURE 12. **Normalized network's lifetime versus the number of FAPs in each cluster.**

Moreover, we evaluate the effects of the number of ground users, the number of FAPs in each cluster, and the total number of content in the network on the performance of the UAV-based femtocaching network in Figs. 11-13. Prior research studies [28], [53] considered a small-scale wireless network with a limited number of ground users and multimedia content and illustrated the impact of the number of ground users and caching nodes on the network's performance. As it can be seen from [28], [53], considering a large number of ground users and multimedia content in wireless networks lead to a considerable increase in the number of distinct requests. In this case, providing high QoS and QoE communication links through the network is more challenging in comparison with a small-scale wireless network. Note that if a UAV-based femtocaching framework can perform effectively in an ultra-dense wireless network, it will definitely perform well in small-scale wireless networks. For this reason, we consider a sufficiently large number of ground users and multimedia content. According to the results shown in Fig. 11, increasing the number of ground users leads to an

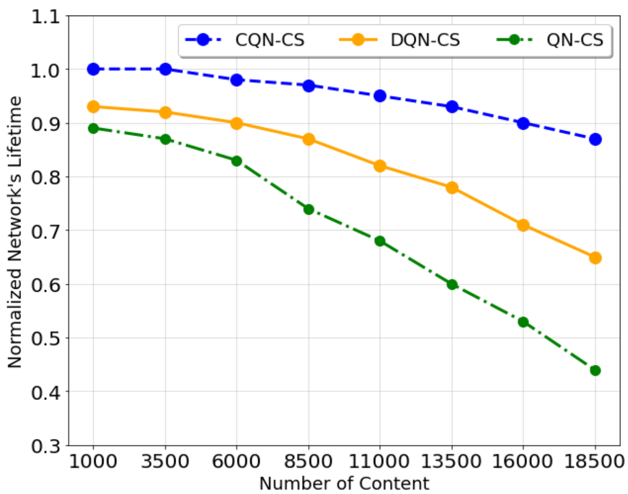


FIGURE 13. Normalized network's lifetime versus number of multimedia content.

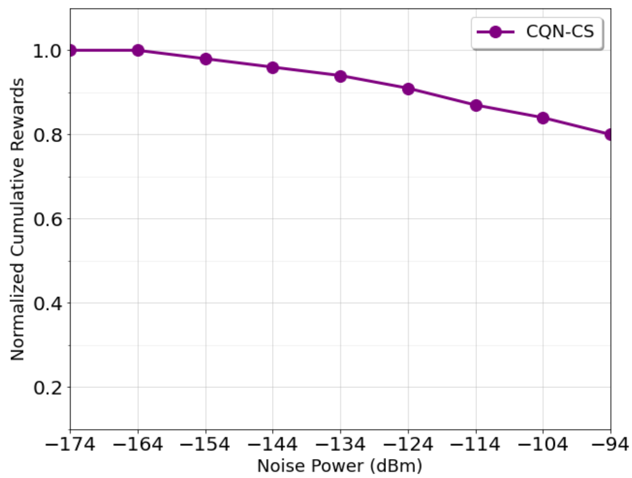


FIGURE 14. Normalized cumulative rewards versus different values of noise power (dBm).

increase in the number of requests in each time slot. Since the network's lifetime depends on the energy consumption of UAVs, managing more requests by UAVs considerably reduces the network's lifetime. Fig. 12 illustrates the impact of the number of FAPs in each cluster on the network's lifetime. Increasing the number of FAPs in each cluster decreases the distance between FAPs and ground users. Therefore, ground users experience less delay by connecting to FAPs instead of UAVs. Therefore, the majority of requests will be managed through FAPs, which increases the network's lifetime. Finally, Fig. 13 evaluates the normalized network's lifetime versus the total number of content in the network. Note that increasing the number of content increases the content diversity throughout the network. Consequently, the number of requests that can directly be served through caching nodes decreases. Therefore, the requested content should be provided by the cloud server, leading to consuming more energy by UAVs.

Fig. 14 illustrates the robustness of the proposed CQN-CS method against the Channel State Information (CSI) uncertainty. It should be noted that the state space of the proposed deep Q-Network, including users' access delay, handover, and UAV's energy consumption is prone to noise. Consequently, the CSI uncertainty and RSSI measurement errors have potential negative impacts on both the state and action spaces. To evaluate the noise robustness of the proposed CQN-CS, we have computed the cumulative reward (after a specific epoch where the proposed CQN-CS framework is well trained) versus different values of the noise power. By considering the fact that the common value of noise power in UAV-based femtocaching networks is in the range of $-174 \leq N_0 \leq -94$ dBm [39], [52], our proposed CQN-CS framework is robust against CSI uncertainty and RSSI measurement errors for a small value of noise (see Fig. 14). It is worth mentioning that in the existing literature on UAV-based femtocaching, commonly -174 dBm is used as the value of the noise power [15], [18], [52]. It can be observed from Fig. 14 that the proposed CQN-CS is robust in this vicinity. On the other hand, to evaluate the performance of the proposed approach, comparison studies are performed based on the worst-case scenario (-94 dBm) and as can be seen in Figs. 5-13, our proposed method outperforms its counterparts in the worst-case scenario. To further improve the robustness of the proposed CQN-CS, robust RL models [55] can be incorporated within the CQN-CS framework, which is the focus of our ongoing research.

VI. CONCLUSION

In this paper, we presented a Convolutional Neural Network (CNN) with Q-learning Connection Scheduling (CQN-CS) architecture in an ultra-dense UAV-based femtocaching network. To improve the network's coverage and support a highly reliable and low-latency transmission, we deployed UAVs beside FAPs over a heterogeneous wireless cellular network. In order to select the optimal caching node, i.e., UAVs and/or FAPs, we formulated a multi-objective connection scheduling problem, with the focus on minimizing the user's access delay by maintaining a reduction in the energy consumption of UAVs and handover phenomena. The mobility of ground users, however, leads to a time-varying topology of the wireless network. For this reason, we proposed the CQN-CS framework to train our coupled UAV-based femtocaching network to respond to users' requests in an optimal fashion. Simulation results showed that the proposed CQN-CS scheme improves the cache-hit ratio, user's access delay, energy consumption of UAVs, handover, cumulative rewards, and network's lifetime during each epoch when compared to Q-learning and Deep Q-Network (DQN) schemes. In this paper, our focus was on the outdoor environment, where both UAVs and FAPs operate efficiently. With the emphasis on the poor signal of UAVs in indoor areas, our future research involves the deployment of a cluster-centric and coded UAV-based femtocaching framework in a heterogeneous integrated network, covering both indoor and outdoor environments. In addition, our future direction is to investigate the security factor and the effect of other mobility patterns on the performance of the proposed CQN-CS framework.

REFERENCES

[1] V. Chamola, V. Hassija, V. Gupta, and M. Guizani, "A comprehensive review of the COVID-19 pandemic and the role of IoT, drones, AI, blockchain, and 5G in managing its impact," *IEEE Access*, vol. 8, pp. 90225–90265, 2020.

[2] B. Li, Z. Fei, and Y. Zhang, "UAV communications for 5G and beyond: Recent advances and future trends," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2241–2263, Apr. 2019.

[3] Y. Zeng, Q. Wu, and R. Zhang, "Accessing from the sky: A tutorial on UAV communications for 5G and beyond," *Proc. IEEE*, vol. 107, no. 12, pp. 2327–2375, Dec. 2019.

[4] X. Chen, Z. Zhao, C. Wu, M. Bennis, H. Liu, Y. Ji, and H. Zhang, "Multi-tenant cross-slice resource orchestration: A deep reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2377–2392, Oct. 2019.

[5] X. Chen, C. Wu, T. Chen, Z. Liu, H. Zhang, M. Bennis, H. Liu, and Y. Ji, "Information freshness-aware task offloading in air-ground integrated edge computing systems," 2020, *arXiv:2007.10129*. [Online]. Available: http://arxiv.org/abs/2007.10129

[6] D. Athukoralage, I. Guvenc, W. Saad, and M. Bennis, "Regret based learning for UAV assisted LTE-U/WiFi public safety networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, USA, Feb. 2017 pp. 1–7.

[7] S. Zhu, L. Gui, N. Cheng, F. Sun, and Q. Zhang, "Joint design of access point selection and path planning for UAV-assisted cellular networks," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 220–233, Jan. 2020.

[8] Z. Hajiakhondi-Meybodi, J. Abouei, and A. H. F. Raouf, "Cache replacement schemes based on adaptive time window for video on demand services in femtocell networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 7, pp. 1476–1487, Jul. 2019.

[9] A. Ahmed, H. Shan, and A. Huang, "Modeling the delivery of coded packets in D2D mobile caching networks," *IEEE Access*, vol. 7, pp. 20091–20105, 2019.

[10] F. Jameel, Z. Hamid, F. Jabeen, S. Zeadally, and M. A. Javed, "A survey of device-to-device communications: Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2133–2168, 3rd Quart., 2018.

[11] B. Jiang, J. Yang, H. Xu, H. Song, and G. Zheng, "Multimedia data throughput maximization in Internet-of-Things system based on optimization of cache-enabled UAV," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3525–3532, Apr. 2019.

[12] B. Ji, Y. Li, B. Zhou, C. Li, K. Song, and H. Wen, "Performance analysis of UAV relay assisted IoT communication network enhanced with energy harvesting," *IEEE Access*, vol. 7, pp. 38738–38747, 2019.

[13] F. Cheng, G. Gui, N. Zhao, Y. Chen, J. Tang, and H. Sari, "UAV-relaying-assisted secure transmission with caching," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3140–3153, May 2019.

[14] N. Zhao, F. Cheng, F. R. Yu, J. Tang, Y. Chen, G. Gui, and H. Sari, "Caching UAV assisted secure transmission in hyper-dense networks based on interference alignment," *IEEE Trans. Commun.*, vol. 66, no. 5, pp. 2281–2294, May 2018.

[15] V. Sharma, I. You, D. N. K. Jayakody, D. G. Reina, and K.-K.-R. Choo, "Neural-blockchain-based ultrareliable caching for edge-enabled UAV networks," *IEEE Trans. Ind. Informat.*, vol. 15, no. 10, pp. 5723–5736, Oct. 2019.

[16] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133–3174, 4th Quart., 2019.

[17] M. Samir, S. Sharafeddine, C. Assi, T. M. Nguyen, and A. Ghrayeb, "Trajectory planning and resource allocation of multiple UAVs for data delivery in vehicular networks," *IEEE Netw. Lett.*, vol. 1, no. 3, pp. 107–110, Sep. 2019.

[18] M. Chen, W. Saad, and C. Yin, "Echo-liquid state deep learning for 360° content transmission and caching in wireless VR networks with cellular-connected UAVs," *IEEE Trans. Commun.*, vol. 67, no. 9, pp. 6386–6400, Sep. 2019.

[19] F. Song, J. Li, M. Ding, L. Shi, F. Shu, M. Tao, W. Chen, and H. V. Poor, "Probabilistic caching for small-cell networks with terrestrial and aerial users," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 9162–9177, Sep. 2019.

[20] M. Hou, G. Ho, G. R. Arrabito, S. Young, and S. Yin, "Effects of display mode and input method for handheld control of micro aerial vehicles for a reconnaissance mission," *IEEE Trans. Human-Machine Syst.*, vol. 43, no. 2, pp. 149–160, Mar. 2013.

[21] M. Hou, H. Zhu, M. Zhou, and G. R. Arrabito, "Optimizing operator–agent interaction in intelligent adaptive interface design: A conceptual framework," *IEEE Trans. Syst., Man, Cybern., C, Appl. Rev.*, vol. 41, no. 2, pp. 161–178, Mar. 2011.

[22] M. Hou, R. D. Kobierski, and M. Brown, "Intelligent adaptive interfaces for the control of multiple UAVs," *J. Cognit. Eng. Decis. Making*, vol. 1, no. 3, pp. 327–362, Sep. 2007.

[23] J. J. Roessingh, A. Toubman, J. van Oijen, G. Poppinga, R. A. Lovlid, M. Hou, and L. Luotsinen, "Machine learning techniques for autonomous agents in military simulations–Multum in Parvo," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 3445–3450.

[24] M. Hou, S. Banbury, and C. Burns, *Intelligent Adaptive Systems: An Interaction-centered Design Perspective*. Boca Raton, FL, USA: CRC Press, Dec. 2014.

[25] H. Wu, X. Tao, N. Zhang, and X. Shen, "Cooperative UAV cluster-assisted terrestrial cellular networks for ubiquitous coverage," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2045–2058, Sep. 2018.

[26] L. Zhang, Z. Zhao, Q. Wu, H. Zhao, H. Xu, and X. Wu, "Energy-aware dynamic resource allocation in UAV assisted mobile edge computing over social Internet of vehicles," *IEEE Access*, vol. 6, pp. 56700–56715, 2018.

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

[28] 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.

[29] N. H. Motlagh, M. Bagaa, and T. Taleb, "Energy and delay aware task assignment mechanism for UAV-based IoT platform," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6523–6536, Aug. 2019.

[30] K. Xue, L. Li, F. Yang, H. Zhang, X. Li, and Z. Han, "Multi-UAV delay optimization in edge caching networks: A mean field game approach," in *Proc. 28th Wireless Opt. Commun. Conf. (WOCC)*, May 2019, pp. 1–5.

[31] X. Liu, M. Chen, and C. Yin, "Optimized trajectory design in UAV based cellular networks for 3D users: A double Q-learning approach," *J. Commun. Inf. Netw.*, vol. 4, no. 1, pp. 24–32, Mar. 2019.

[32] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, Apr. 2019.

[33] A. Rahmati, Y. Yapici, N. Rupasinghe, I. Guvenc, H. Dai, and A. Bhuyan, "Energy efficiency of RSMA and NOMA in cellular-connected mmWave UAV networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2019, pp. 1–6.

[34] Y. Cai, Z. Wei, R. Li, D. W. Kwan Ng, and J. Yuan, "Energy-efficient resource allocation for secure UAV communication systems," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Rabat, Morocco, Apr. 2019, pp. 1–8.

[35] S. Ahmed, M. Zaman Chowdhury, and Y. Min Jang, "Energy-efficient UAV-to-user scheduling to maximize throughput in wireless networks," *IEEE Access*, vol. 8, pp. 21215–21225, 2020.

[36] M. Rajabizadeh and J. Abouei, "An efficient femtocell-to-femtocell handover decision algorithm in LTE femtocell networks," in *Proc. 23rd Iranian Conf. Electr. Eng.*, May 2015, pp. 213–218.

[37] Z. HajiAkhondi-Meybodi, J. Abouei, M. Jassemuddin, and A. Mohammadi, "Mobility-aware femtocaching algorithm in D2D networks based on handover," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 10188–10201, Sep. 2020.

[38] L. Qiu and G. Cao, "Popularity-aware caching increases the capacity of wireless networks," *IEEE Trans. Mobile Comput.*, vol. 19, no. 1, pp. 173–187, Jan. 2020.

[39] M. Chen, W. Saad, and C. Yin, "Liquid state machine learning for resource and cache management in LTE-U unmanned aerial vehicle (UAV) networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1504–1517, Mar. 2019.

[40] C. M. Albertsen, "Generalizing the first-difference correlated random walk for marine animal movement data," *Sci. Rep.*, vol. 9, no. 1, pp. 4017–4031, Mar. 2019.

[41] Z. Hajiakhondi-Meybodi, M. Salimibeni, K. N. Plataniotis, and A. Mohammadi, "Bluetooth low energy-based angle of arrival estimation via switch antenna array for indoor localization," in *Proc. IEEE 23rd Int. Conf. Inf. Fusion (FUSION)*, Jul. 2020, pp. 1–6.

[42] Z. Hajiakhondi-Meybodi, M. Salimibeni, A. Mohammadi, and K. N. Plataniotis, "Bluetooth low energy-based angle of arrival estimation in presence of Rayleigh fading," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2020, pp. 3395–33400.

[43] M. Salimibeni, Z. Hajiakhondi-Meybodi, P. Malekzadeh, M. Atashi, K. N. Plataniotis, and A. Mohammadi, "IoT-TD: IoT dataset for multiple model BLE-based indoor localization/tracking," in *Proc. 28th Eur. Signal Process. Conf. (EUSIPCO)*, Jan. 2021, pp. 1697–1701.

[44] M. Atashi, P. Malekzadeh, M. Salimibeni, Z. Hajiakhondi-Meybodi, K. N. Plataniotis, and A. Mohammadi, "Orientation-matched multiple modeling for RSSI-based indoor localization via BLE sensors," in *Proc. 28th Eur. Signal Process. Conf. (EUSIPCO)*, Jan. 2021, pp. 1702–1706.

[45] A. M. Al-Samman, T. Abd. Rahman, T. Al-Hadhrami, A. Daho, M. N. Hindia, M. H. Azmi, K. Dimyati, and M. Alazab, "Comparative study of indoor propagation model below and above 6 GHz for 5G wireless networks," *Electronics*, vol. 8, no. 1, p. 44, Jan. 2019.

[46] B. Ravi Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," 2020, *arXiv:2002.00444*. [Online]. Available: http://arxiv.org/abs/2002.00444

[47] K. Li, T. Zhang, and R. Wang, "Deep reinforcement learning for multiobjective optimization," *IEEE Trans. Cybern.*, early access, Mar. 18, 2020, doi: 10.1109/TCYB.2020.2977661.

[48] K. Van Moffaert, M. M. Drugan, and A. Nowe, "Scalarized multi-objective reinforcement learning: Novel design techniques," in *Proc. IEEE Symp. Adapt. Dyn. Program. Reinforcement Learn. (ADPRL)*, Singapore, Apr. 2013, pp. 191–199.

[49] R. S. Burachik, C. Y. Kaya, and M. M. Rizvi, "A new scalarization technique to approximate Pareto fronts of problems with disconnected feasible sets," *J. Optim. Theory Appl.*, vol. 162, no. 2, pp. 428–446, Aug. 2014.

[50] X. Lu, L. Xiao, C. Dai, and H. Dai, "UAV-aided cellular communications with deep reinforcement learning against jamming," *IEEE Wireless Commun.*, vol. 27, no. 4, pp. 48–53, Aug. 2020.

[51] M. A. Aref and S. K. Jayaweera, "Robust deep reinforcement learning for interference avoidance in wideband spectrum," in *Proc. IEEE Cognit. Commun. for Aerosp. Appl. Workshop (CCAAW)*, Cleveland, OH, USA, Jun. 2019, pp. 1–5.

[52] C. Wu, S. Shi, S. Gu, L. Zhang, and X. Gu, "Deep reinforcement learning-based content placement and trajectory design in urban cache-enabled UAV networks," *Wireless Commun. Mobile Comput.*, vol. 2020, pp. 1–11, Aug. 2020.

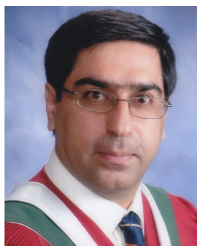
[53] T. Zhang, Y. Wang, Y. Liu, W. Xu, and A. Nallanathan, "Cache-enabling UAV communications: Network deployment and resource allocation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7470–7483, Nov. 2020.

[54] R. Ahmad, E. A. Sundararajan, and A. Khalifeh, "A survey on femtocell handover management in dense heterogeneous 5G networks," *Telecommun. Syst.*, vol. 75, no. 4, pp. 481–507, Dec. 2020.

[55] Y. Wang, H. He, and X. Tan, "Robust reinforcement learning in POMDPs with incomplete and noisy observations," 2019, *arXiv:1902.05795*. [Online]. Available: http://arxiv.org/abs/1902.05795

ZOHREH HAJIAKHONDI-MEYBODI (Graduate Student Member, IEEE) received the B.Sc. degree in communication engineering and the M.Sc. degree (Hons.) in communication systems engineering from Yazd University, Yazd, Iran, in 2013 and 2017, respectively. She is currently pursuing the Ph.D. degree in electrical and computer engineering (ECE) with Concordia University, Montreal, QC, Canada. Since 2019, she has been an Active Member of the I-SIP Laboratory, Concordia University. Her research interests include general areas of wireless communication networks with a particular emphasis on Femtocaching, the Internet of Things (IoT), indoor localization, optimization algorithms, and multimedia wireless sensor networks (WMSN).

ARASH MOHAMMADI (Senior Member, IEEE) is currently an Associate Professor with the Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada. Prior to joining Concordia University and for two years, he was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada. He is also a registered Professional Engineer in Ontario. He is also the Director-Membership Developments of the IEEE Signal Processing Society (SPS), the General Co-Chair of 2021 IEEE International Conference on Autonomous Systems (ICAS), and the Guest Editor of *IEEE Signal Processing Magazine (SPM)* Special Issue on Signal Processing for Neurorehabilitation and Assistive Technologies. He is also serving as an Associate Editor on the Editorial Board for IEEE Signal Processing Letters. He was the Co-Chair of Symposium on Advanced Bio-Signal Processing and Machine Learning for Assistive and Neuro-Rehabilitation Systems as part of 2019 IEEE GlobalSIP, and Symposium on Advanced Bio-Signal Processing and Machine Learning for Medical Cyber-Physical Systems, as a part of IEEE GlobalSIP'18, The Organizing Chair of 2018 IEEE Signal Processing Society Video and Image Processing (VIP) Cup, and the Lead Guest Editor of IEEE Transactions on Signal and Information Processing Over Networks Special Issue on Distributed Signal Processing for Security and Privacy in Networked Cyber-Physical Systems. He was a recipient of several distinguishing awards, including the Eshrat Arjomandi Award for outstanding Ph.D. dissertation from the Electrical Engineering and Computer Science Department, York University, in 2013, the Concordia President's Excellence in Teaching Award in 2018, and the 2019 Gina Cody School of Engineering and Computer Science's Research and Teaching Awards in the new scholar category.



JAMSHID ABOUEI (Senior Member, IEEE) received the B.Sc. degree in electronics engineering and the M.Sc. degree in communication systems engineering (Hons.) from the Isfahan University of Technology (IUT), Iran, in 1993 and 1996, respectively, and the Ph.D. degree in electrical engineering from the University of Waterloo, Waterloo, ON, Canada, in 2009. He joined with the Department of Electrical Engineering, Yazd University, Iran, in 1996 (as a Lecturer) and was promoted to an Assistant Professor in 2010, and an Associate Professor in 2015. From 1998 to 2004, he served as a Technical Advisor and a Design Engineer with the Research and Development Center and the Cable Design Department, SGCC, Iran. From 2009 to 2010, he was a Postdoctoral Fellow with the Multimedia Laboratory, Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada, and worked as a Research Fellow with the Self-Powered Sensor Networks (ORF-SPSN) consortium. During his sabbatical, he was an Associate Researcher with the Department of Electrical, Computer, and Biomedical Engineering, Ryerson University, Toronto. He was the International Relations Chair of 27th ICEE2019 Conference, Iran, in 2019. He currently directs the Research Group with the Wireless Networking Laboratory (WINEL), Yazd University. His research interests include the next-generation of wireless networks (5G) and wireless sensor networks (WSNs), with a particular emphasis on PHY/MAC layer designs, including the energy efficiency and optimal resource allocation in cognitive cell-free massive MIMO networks, multi-user information theory, mobile edge computing, and femtocaching. He is a member of the IEEE Information Theory. He has received several awards and scholarships, including FOE and IGSA awards for excellence in research in the University of Waterloo, Waterloo, MSRT Ph.D. Scholarship from the Ministry of Science, Research, and Technology, Iran, in 2004, the Distinguished Researcher Award in province of Yazd, Iran, in 2011, and THE Distinguished Researcher Award in Electrical Engineering Department, Yazd University, in 2013. He was a recipient of the Best Paper Award for the IEEE Iranian Conference on Electrical Engineering (ICEE 2018).