Received 20 June 2023; accepted 9 July 2023. Date of publication 18 July 2023; date of current version 16 August 2023.

Digital Object Identifier 10.1109/OJCOMS.2023.3296378

HAPS-UAV-Enabled Heterogeneous Networks: A Deep Reinforcement Learning Approach

ATEFEH HAJIJAMALI ARANI[1], PENG HU[2,3] (Senior Member, IEEE), AND YEYING ZHU[1]

[1]Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada

[2]Digital Technologies Research Center, National Research Council of Canada, Waterloo, ON N2L 3G1, Canada

[3]Cheriton School of Computer Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada

CORRESPONDING AUTHOR: P. HU (e-mail: Peng.Hu@nrc-cnrc.gc.ca)

This work was supported in part by the High-Throughput and Secure Networks Challenge Program of National Research Council Canada under Grant CH-HTSN-418, and in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant RGPIN-2022-03364.

ABSTRACT The integrated use of non-terrestrial network (NTN) entities such as the high-altitude platform station (HAPS) and low-altitude platform station (LAPS) has become essential elements in the space-air-ground integrated networks (SAGINs). However, the complexity, mobility, and heterogeneity of NTN entities and resources present various challenges from system design to deployment. This paper proposes a novel approach to designing a heterogeneous network consisting of HAPSs and unmanned aerial vehicles (UAVs) being LAPS entities. Our approach involves jointly optimizing the three-dimensional trajectory and channel allocation for aerial base stations, with a focus on ensuring fairness and the provision of quality of service (QoS) to ground users. Furthermore, we consider the load on base stations and incorporate this information into the optimization problem. The proposed approach utilizes a combination of deep reinforcement learning and fixed-point iteration techniques to determine the UAV locations and channel allocation strategies. Simulation results reveal that our proposed deep learning-based approach significantly outperforms learning-based and conventional benchmark models.

INDEX TERMS Deep reinforcement learning, high-altitude platform station, resource allocation, fairness, unmanned aerial vehicles, non-terrestrial networks.

I. INTRODUCTION

RECENTLY, the integrated use of non-terrestrial network (NTN) entities such as high-altitude platform stations (HAPSs) and low-altitude platform stations (LAPSs) has become essential elements in the space-air-ground integrated networks (SAGINs). These entities can complement the space segments to provide high-quality network access to global users. For example, the use of HAPSs together with unmanned aerial vehicles (UAVs) being LAPSs can be well suited for meeting capacity and coverage demands, such as temporary events-driven coverage, greenfield coverage, terrestrial backhaul, and white spot reduction [1], with the capability of keeping the round-trip-time latency down to within 10 ms. However, the great promises of such HAPSs and LAPSs in a SAGIN come with challenges. One challenge in an important and generic scenario is fairness assurance in the overall quality of experience (QoE) for ground users. The heterogeneous NTN entities, resources, and dynamics of UAV trajectories add much complexity to the system modeling and solutions.

Most recent works have proposed to address UAVs and HAPS separately. For UAVs, the recent results focus on trajectory and resource management within a UAV network where multiple UAVs are employed. The optimization of UAV trajectory for quality of service (QoS) performance and coverage is discussed in [2], [3]. Deep Q-network (DQN) [4], [5] and reinforcement learning methods [6] have been applied to UAV trajectory optimization, while deep learning methods [7], [8], [9] for optimal resource management have been proposed. HAPS has been studied as a standalone system providing uplink and downlink to the ground users [10] and as part of a SAGIN system [11], [12].

In order to address the dynamic nature of an integrated system consisting of HAPS and LAPS entities, Q-learning is considered an effective technique for solving an optimal solution in system modeling. However, it poses limitations

when the mobility of ground users and UAVs is considered. Furthermore, it must deal with the exponential growth of states and actions when exploring an optimal solution in a high-dimensional space. A new approach is needed to solve the theoretical limits while meeting generic QoE or QoS requirements in an integrated system setting. In the context of a HAPS-UAV-enabled heterogeneous network, such an approach needs to be applied to the fundamental challenge in resource allocation and UAV trajectory planning, considering practical deployment configurations. This challenge has hardly been well addressed in the current works.

This work proposes a deep reinforcement learning-based algorithm for aerial base stations (ABSs) to provide network services in a highly dynamic environment where the mobility of ground users and UAVs presents a challenge for conventional reinforcement learning techniques such as Q-learning. This is due to the potential for failure caused by the curse of dimensionality. The proposed algorithm uses neural networks to approximate Q-value functions to address this issue, allowing the UAVs to operate autonomously and intelligently adapting to rapidly changing conditions. In particular, we make the following contributions:

- We construct a high dynamic scenario of an aerial heterogeneous network in a diverse environment, considering both HAPSs and UAVs while considering user mobility.
- We employ the deep reinforcement learning algorithm to intelligently optimize the trajectory and transmit channel of UAVs.
- Our proposed solution considers loads of ABSs and incorporates them into the trajectory design and resource allocation process, which determines the average resource utilization at ABSs and the system's ability to provide sufficient QoS to users. Furthermore, we optimize fairness among users in the system.
- The proposed approach is the joint utilization of deep reinforcement learning and fixed-point iteration techniques for addressing the complex and dynamic nature of our problem.
- We compare the performance in terms of fairness, rate, and outage between the proposed and reinforcement learning-based benchmarks.

The remainder of the paper is structured as follows. Section II overviews the related work. Section III presents the system model and problem statement. Section VI discusses the Q-learning and our proposed DQN-based scheme for a joint resource management and trajectory design. Section V evaluates the proposed scheme in comparison with the typical algorithms and variations.

II. RELATED WORK

In recent years, the integration of HAPSs and UAVs into communication networks has gained significant attention as a promising solution for extending wireless coverage and providing access to remote areas. HAPS provides a high-altitude persistent coverage that can reduce the number of cell towers required, resulting in lower capital and operational costs. Furthermore, the mobility of UAVs allows for dynamic deployment in areas with high user density, thereby improving the overall network capacity. The use of multiple HAPSs and UAVs in a network can also provide improved reliability. Most of the works focus on optimizing the trajectory of UAVs to enhance network performance and coverage. These studies have proposed various trajectory design algorithms based on non-learning and learning algorithms, with the aim of maximizing the network's coverage area and enhancing user throughput. In [2], a trajectory design algorithm based on deep reinforcement learning for a single UAV is proposed. The solution aims at maximizing the uplink sum rate of users. To maximize the spectral efficiency of a network composed of a ground base station (BS) and UAVs, a deep reinforcement algorithm to optimize the locations of UAVs is developed in [3]. Moreover, it is assumed that users have different QoS. In [4], a UAV is employed for emergency communication support for users. The objective is to maximize the number of served users and uplink data rate by optimizing the UAV trajectory and transmission power of users. A DQN-based algorithm is proposed to solve the UAV trajectory problem. Additionally, a successive convex approximation-based algorithm is proposed for power control at the level of users, based on the optimized UAV trajectory. To optimize the trajectory of a single UAV for mobile edge computing, a double deep Q-network algorithm is proposed in [5]. The authors in [7] utilize a deep learning algorithm for dynamically allocating radio resources for uplink and downlink. In [8], the authors propose a reinforcement learning approach to address the challenge of traffic offloading in an aerial network. The proposed solution employs a double Q-learning algorithm with an improved delay-sensitive replay memory mechanism to train the nodes to make intelligent offloading decisions based on both local and neighboring historical information. Additionally, they utilize a joint information collection technique and an offline training mechanism to further enhance the efficiency of the algorithm. In [6], the authors propose an energy-efficient UAV path planning based on reinforcement learning and satisfaction algorithms. To maximize the throughput of an aerial network, learning-based mechanisms are implemented in [9], [13]. In [14], the learning algorithms are surveyed in UAV-assisted SAGINs. However, the existing studies are restricted to only UAV networks and do not consider HAPS.

On the other hand, the integration of HAPSs in UAV networks can enhance the capabilities of aerial networks, providing a cost-effective solution to meet the increasing demands for high-speed and reliable communication. In [15], the authors propose a transmission scheme that combines the HAPS and the ground-to-space transmission to improve terrestrial communication and reduce transmission power. They develop a transmission control strategy, where ground users can switch between the two transmission schemes with a probability, which is determined to maximize overall

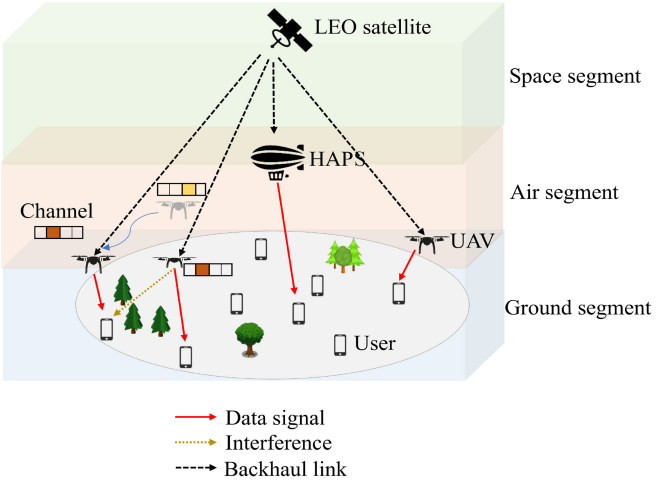

FIGURE 1. An illustration of the system model.

throughput. In [16], a solution is developed to improve the reliability of uplink communications by fusing free-space optics (FSO) and radio frequency (RF) technologies. The proposed solution utilizes a HAPS, as a relay station. Furthermore, two system models, single-hop and SAGIN-based dual-hop are investigated for uplink communication with hybrid FSO/RF links. The impact of HAPS deployments on terrestrial networks is investigated in [10]. It analyzes both co-channel and adjacent channel deployment scenarios with both unsynchronized and synchronized time-division duplexing (TDD). The results indicate that the synchronized TDD scenario requires a smaller inter-system distance than the unsynchronized case for the co-channel case. In the adjacent channel case, the interference-to-noise ratio is always below a certain threshold for both unsynchronized and synchronized scenarios. Furthermore, results for full buffer and bursty traffic models under different traffic loads are considered. In [11], low Earth orbit (LEO) satellites and HAPSs are used to provide access and data backhaul to remote area users which aims at maximizing the revenue of LEO satellites. The problem is formulated as a mixed integer nonlinear programming. To solve the problem, matching algorithms are proposed. In [12], the authors consider a system composed of a HAPS and a set of UAVs, in which the locations of all the ABSs are fixed. To solve the problem of power and sub-carrier allocation, a heuristic greedy algorithm is used. However, the aforementioned work focused on HAPSs does not take into account the fairness issue in the system and most studies consider static scenarios for users in the system.

III. SYSTEM MODEL AND PROBLEM STATEMENT

A. SYSTEM MODEL

In this section, we present the system formulation and the problem statement. As depicted in Fig. 1, the considered HAPS-UAV-enabled heterogeneous network is heterogeneously constructed with HAPSs and low altitude platforms (LAPs) or UAVs. We consider the downlink transmission of the system composed of a set of UAVs $\mathcal{U}$ and a set of HAPSs $\mathcal{M}$ as ABSs. Let $\mathcal{B} = \mathcal{U} \cup \mathcal{M}$ denote the set of total ABSs in the system. Furthermore, we assume that LEO satellites provide backhaul connectivity for the ABSs. The set of total users and the set of users associated with ABS $b \in \mathcal{B}$ at time instant t are represented by $\mathcal{K}$ and $\mathcal{K}_b(t) \in \mathcal{K}$, respectively. The three-dimensional (3D) location of ABS b is denoted by $z_b^{\text{ABS}}(t) = (x_b(t), y_b(t), h_b(t))$, where $(x_b(t), y_b(t))$ and $h_b(t)$ are the horizontal coordinate and the altitude of ABS b at time instant t, respectively. Generally, discrete-time sampling is adopted to update the system configuration. We consider a discrete-time setting $\mathcal{N} = \{0, 1, 2, \ldots, N\}$. We assume that the HAPSs are fixed and the UAVs fly at a fixed speed v_{U}. Therefore, the location of UAV $u \in \mathcal{U}$ is updated as follows:

$$z_u^{\text{ABS}}(t+1) = z_u^{\text{ABS}}(t) + v_{\text{U}}(t) T_{\text{s}}, \tag{1}$$

where T_{s} and $z_u^{\text{ABS}}(t)$ are the duration of each time slot and the location of UAV u at time instant t, respectively.

B. USER MOBILITY MODEL

We assume that the users move according to a random walk mobility model [17]. Let $z_k^{\text{UE}}(t) = (x_k(t), y_k(t), h_k)$ denote the coordinate of user $k \in \mathcal{K}$ at time instant $t \in \mathcal{N}$, where $(x_k(t), y_k(t))$ and h_k are the horizontal coordinate and the height of user k at time instant t, respectively. Obviously, the heights of the users are fixed. In this model, the users change their speeds and movement directions with zero pause time at each time slot. At each time, the speed of user k, $v_k(t)$, is randomly determined from the predefined ranges $[v_{\text{UE}}^{\min}, v_{\text{UE}}^{\max}]$ following a uniform distribution, where $v_{\text{UE}}^{\min}$ and $v_{\text{UE}}^{\max}$ denote the minimum and maximum speed of the users, respectively. Furthermore, the movement direction for user k, $\phi_k(t)$, is randomly chosen from the ranges $[0, 2\pi]$ according to a uniform distribution. Therefore, for each user $k \in \mathcal{K}$, the velocity vector is $[v_k(t)\cos\phi_k(t), v_k(t)\sin\phi_k(t)]$.

C. RADIO PROPAGATION AND SIGNAL QUALITY

We assume that at each time slot, the network topology is quasi-static, and the channel state information is constant. We adopt the International Telecommunications Union (ITU) path loss model between the users and the ABSs. The path loss model between HAPS $m \in \mathcal{M}$ and user $k \in \mathcal{K}$ includes the free space path loss (FSPL) model which can be expressed as [18]

$$L_{m,k}(t) = 32.44 + 20\log_{10} f_{\text{HAPS}} + 20\log_{10} d_{m,k}(t) \quad [\text{dB}], \tag{2}$$

where f_{HAPS} and $d_{m,k}(t)$ are the HAPS' operating frequency in Mega Hertz (MHz) and the distance in kilometers between user k and HAPS m at time t, respectively.

To model a channel between user k and UAV u, we consider the model described in (3) which includes line-of-sight (LoS) and non-LoS components. The probability of having a LoS link between user k and UAV u depends on the

environmental characteristics and it can be written as [6]

$$\mathrm{pr}_{u,k}^{\mathrm{LoS}}(t) = \prod_{n=0}^{J}\left[1 - \exp\left(-\frac{\left[h_u(t) - \frac{\left(n+\frac{1}{2}\right)(h_u(t)-h_k)}{J+1}\right]^2}{2\xi^2}\right)\right], \tag{3}$$

where $J = \lfloor \frac{r_{u,k}(t)\sqrt{\alpha\beta}}{1000} - 1 \rfloor$, and α, β and ξ represent statistical environment-dependent parameters [19, Table 1]. Here, parameter α represents the ratio of land area covered by buildings to total land area, β denotes the mean number of buildings per unit area, and ξ is the distribution of building height. This blockage model can be used for air-to-ground transmissions with any transmitter/receiver heights and for a broad spectrum range [20]. Here, $r_{u,k}(t) = \sqrt{(x_u(t) - x_k(t))^2 + (y_u(t) - y_k(t))^2}$ denotes the horizontal distance between UAV $u \in \mathcal{U}$ and user $k \in \mathcal{K}$ at time t. Therefore, the probability of having a non-LoS link at time t can be determined as $\mathrm{pr}_{u,k}^{\mathrm{NLoS}}(t) = 1 - \mathrm{pr}_{u,k}^{\mathrm{LoS}}(\mathrm{t})$.

Let $d_{u,k}(t) = \sqrt{r_{u,k}^2(t) + (h_u(t) - h_k)^2}$ be the 3D distance between UAV u and user k at time t. The channel gain between UAV u and user k can be written as [21]

$$L_{u,k}^{z}(t) = \delta_u^z + \eta_u^z \log_{10} d_{u,k}(t) + \chi_u^z \quad [\mathrm{dB}], \tag{4}$$

where superscript $z \in \{\mathrm{LoS}, \mathrm{NLoS}\}$ denotes a LoS or non-LoS component. Parameters δ_u^z and η_u^z represent the reference path loss and the path loss exponent, respectively. Here, χ_b^z denotes a zero-mean Gaussian random variable with a standard deviation $\sigma_{b,\mathrm{SF}}^z$ in dB.

We assume that the HAPSs transmit over the orthogonal channels and also there is no interference between the HAPSs and the UAVs (spectrum overlay access). However, multiple UAVs can transmit over the same channel and cause co-channel interference. Let ω_{H} and ω_{U} denote the total bandwidth for the HAPSs and the UAVs, respectively. The total bandwidth ω_{U} (or ω_{M}) is divided into $|\mathcal{Q}_{\mathrm{U}}|$ (or $|\mathcal{Q}_{\mathrm{M}}|$) orthogonal channels with bandwidth $\omega_{\mathrm{U}}/|\mathcal{Q}_{\mathrm{U}}|$ (or $\omega_{\mathrm{M}}/|\mathcal{Q}_{\mathrm{M}}|$), where $\mathcal{Q}_{\mathrm{U}}$ (or $\mathcal{Q}_{\mathrm{M}}$) is the set of available channels for the UAVs (or the HAPSs). Let p_u and $g_{u,k}(t)$ denote the transmit power of UAV u and the channel gain between UAV u and user k at time instant t, respectively. Therefore, the maximum achievable data rate to user k provided by UAV u can be expressed as

$$C_{u,k}(t) = \frac{\omega_{\mathrm{U}}}{|\mathcal{Q}_{\mathrm{U}}|}\log_2\left(1 + \gamma_{u,k}(t)\right) \quad [\mathrm{bps}], \tag{5}$$

where $\gamma_{u,k}(t)$ denotes the signal to interference plus noise ratio (SINR) at the receiver of user k associated to UAV u, which can be written as

$$\gamma_{u,k}(t) = \frac{I_{u,k}(t)p_u g_{u,k}(t)}{\sum_{u' \in \mathcal{U}\setminus u} p_{u'} g_{u',k}(t)\rho_{u'}(t)\mathbb{1}_{(q_u(t)=q_{u'}(t))} + \sigma_0^2}, \tag{6}$$

where $q_u(t)$ and σ_0^2 is the transmit channel of UAV u at time t and the noise power, respectively. Here, $\rho_u(t)$ represents the load of UAV u at time t. This model takes into account the load-coupling effect in Long-Term Evolution (LTE) networks, which is represented by a system of non-linear equations based on the joint stationary distribution of active flows in all cells [22]. The proposed technique requires solving a system of linear equations whose dimension increases exponentially with the number of cells. The load coefficient is added to the denominator of the SINR to capture the impact of cell loads on interference and accurately assess the performance of the network. It acknowledges the interdependence between cells and their load factors, which reflect the utilization of available resources within each cell. A low load factor implies sufficient network capacity to meet the demand, while a high load factor indicates congestion and an increased service outage.

The binary element $I_{u,k}(t) \in \{0, 1\}$ indicates the association between UAV u and user k at time t which can be defined as follows:

$$I_{u,k}(t) = \begin{cases} 1, & \text{if user } k \text{ is associated to UAV } u \text{ at time } t, \\ 0, & \text{o.w.} \end{cases} \tag{7}$$

The achievable rate for user k associated to HAPS m is given by

$$C_{m,k}(t) = \frac{\omega_{\mathrm{M}}}{|\mathcal{Q}_{\mathrm{M}}|}\log_2\left(1 + \gamma_{m,k}(t)\right) \quad [\mathrm{bps}], \tag{8}$$

where $\gamma_{m,k}(t)$ denotes the SINR at the receiver of user k associated to HAPS m, which can be defined as

$$\gamma_{m,k}(t) = \frac{I_{m,k}(t)p_m g_{m,k}(t)}{\sigma_0^2}, \tag{9}$$

where p_m and $g_{m,k}(t)$ denote the transmit power of HAPS m and the channel gain between HAPS m and user k, respectively. $I_{m,k}(t) \in \{0, 1\}$ represents the association between HAPS m and user k at time t which can be defined as follows:

$$I_{m,k}(t) = \begin{cases} 1, & \text{if user } k \text{ is associated to HAPS } m \text{ at time } t, \\ 0, & \text{o.w.} \end{cases} \tag{10}$$

Let $\mathcal{K}_m(t)$ and $\mathcal{K}_u(t)$ denote the set of associated users to HAPS $m \in \mathcal{M}$ and UAV $u \in \mathcal{U}$, respectively. According to $I_{u,k}(t)$ and $I_{m,k}(t)$ defined in (7) and (10), we can define $\mathcal{K}_u(t)$ and $\mathcal{K}_m(t)$ as follows:

$$\mathcal{K}_u(t) = \left\{k | k \in \mathcal{K}, I_{u,k}(t) = 1\right\}, \tag{11}$$

and

$$\mathcal{K}_m(t) = \left\{k | k \in \mathcal{K}, I_{m,k}(t) = 1\right\}. \tag{12}$$

D. LOAD AND USER-ABS ASSOCIATION POLICY

Now, we define the load of ABS $b \in \mathcal{B}$ at time instant t as follows [23]:

$$\rho_b(t) = \sum_{k \in \mathcal{K}_b(t)} \frac{\vartheta_k}{\zeta_k} \frac{1}{C_{b,k}(t)} \triangleq f_b(\boldsymbol{\rho}(t)), \tag{13}$$

where ϑ_k and $1/\zeta_k$ are the packet arrival rate and the mean packet size of user k, respectively. Here, ϑ_k/ζ_k represents the user rate requirement. Under this definition, we can consider heterogeneous users, which have different user rate requirements. Vector $\boldsymbol{\rho}(t) = (\rho_1(t), \ldots, \rho_{|\mathcal{B}|}(t))$ denotes the load

vector which comprises the load of all the ABSs in the system. Let $\boldsymbol{f}(\boldsymbol{\rho}(t)) = (f_1(\boldsymbol{\rho}(t)), \ldots, f_{|\mathcal{B}|}\boldsymbol{\rho}(t))^T$. Thus, we can express (13) in the form of a vector as follows [24]:

$$\boldsymbol{\rho}(t) = \boldsymbol{f}(\boldsymbol{\rho}(t)). \tag{14}$$

Due to the inter-cell interference, load conditions at the ABSs are dependent. Indeed, the accurate measurement of load can be a complex task, especially when the network is subject to rapid changes and varying traffic patterns. In our proposed approach, we address this challenge by employing an approximation method to calculate the load on ABSs. However, obtaining real-time and precise load measurements in dynamic network conditions can be challenging. Therefore, we adopt a practical approach to estimate the load on ABSs using a fixed-point algorithm. This algorithm computes approximate ABSs' loads by leveraging the concept of average interference [22].

It is worth noting that $f(\boldsymbol{\rho}(t))$ is a standard interference function. Therefore, the non-linear load coupling equation (14) can be solved by the fixed point iteration algorithm starting from an arbitrary initial ABS load vector $\boldsymbol{\rho}^0 > 0$ as follows [9]:

$$\boldsymbol{\rho}^{\iota} = \min\Big(\boldsymbol{f}\Big(\boldsymbol{\rho}^{\iota-1}\Big), 1\Big), \tag{15}$$

where $\boldsymbol{\rho}^{\iota}$ denotes the load vector at iteration $\iota \in \{1, \ldots, N_{\text{FP}}\}$, and N_{FP} is the total number of fixed point iterations. To ensure the system is stable, we need to guarantee loads of the ABSs not exceed the value one. However, in the case that a load of an ABS b exceeds the threshold one, it would drop some of its associated users to achieve $\rho_b \leq 1$ [25].

Definition 1: A function $f(\boldsymbol{n})$ is called a standard interference function if for all $n \geq 0$, the following properties are satisfied [26]:

1) *Positivity:* $f(\boldsymbol{n}) > 0$,
2) *Monotonicity:* $\boldsymbol{n} \geq \boldsymbol{n}' \Rightarrow f(\boldsymbol{n}) \geq f(\boldsymbol{n}')$,
3) *Scalability:* $\alpha f(\boldsymbol{n}) > f(\alpha \boldsymbol{n})$ *for* $\alpha > 1$.

Lemma 1 indicates that the BS load vector $\boldsymbol{\rho}^{N_{\text{FP}}}$ converges to the fixed point solution of (14).

Lemma 1: If the fixed point of (14) exists, then it is unique, and can be iteratively obtained by (15) as N_{FP} goes to infinity.

Proof: In the Appendix, it is proved that $f_b(\boldsymbol{\rho}(t))$ is a standard interference function [22]. Furthermore, [26, Th. 7] prove that $\min(f_b(\boldsymbol{\rho}), 1)$ is a standard interference function. Then, by using [26, Th. 2], the convergence is proved. ■

The assignment of the users to the ABSs needs to be addressed. Due to the mobility of the users in the system, they are expected to periodically assess their performance and make necessary adjustments. If a user is not satisfied with its current ABS association, it may change its serving ABS and establish a new association. Therefore, new users and users that are currently experiencing an outage require to initiate new association procedures in order to be associated with new ABSs. Given the fixed locations and the transmit channels of the ABSs, each user is associated with an ABS based on the following user association policy:

$$b_k^*(t) = \arg\max_{b \in \mathcal{B}} \{p_b g_{b,k}(t)\}. \tag{16}$$

E. PROBLEM FORMULATION

Given the described system, the objective is to maximize fairness among the users while minimizing the load of the ABSs under the constraint of load. The optimized parameters are the UAVs' trajectories and the transmission channels. By considering the load of UAVs, we can optimize resource utilization while also ensuring that UAVs are not overloaded, which can lead to decreased QoS and network performance. For a dynamic system captured by a flow-level queuing model M/M/1, the average number of flows at UAV u is given by $\frac{\rho_u}{1-\rho_u}$. From Little's formula, minimizing the average number of flows is equivalent to minimizing the average delay experienced by a typical flow. Moreover, we aim to strike a balance between fairness and resource utilization in aerial heterogeneous networks, which we believe is critical for ensuring a robust and reliable network infrastructure that can support a wide range of applications and user groups. Here, we introduce the fairness factor to the objective function named Jain's fairness index, the most widely-used fairness metric in wireless networks' applications. The Jain's fairness index at time t can be defined as follow [27], [28], [29]:

$$\mathcal{F}(t) = \frac{\big(\sum_{k \in \mathcal{K}} \bar{C}_k(t)\big)^2}{|\mathcal{K}|\big(\sum_{k \in \mathcal{K}} \bar{C}_k(t)^2\big)}. \tag{17}$$

where $\bar{C}_k(t)$ is the total data rate for user k until time instant t expressed as follows:

$$\bar{C}_k(t) = \sum_{\tau \leq t} \sum_{b \in \mathcal{B}} C_{b,k}(\tau). \tag{18}$$

The definition in (17) reveals that the fairness index $\mathcal{F}(t)$ is continuous so that a change in a user rate results in a change in the fairness index. Furthermore, it is applicable to any size of users' sets in the system. Besides, it is bounded between $\frac{1}{|\mathcal{K}|}$ and 1, in which a totally fair system has a Jain index of 1 while $\frac{1}{|\mathcal{K}|}$ corresponds to the least fair system. Therefore, the higher value of the fairness index is the result of the smaller differences among the total data rates of the users $\{\bar{C}_k(t)\}_{k \in \mathcal{K}}$. Note that Jain's fairness index takes into consideration all the users in the system, not only the users with poor performance [30]. In addition, it is mostly used for assessing long-term fairness performance. Furthermore, fairness and loads of the ABSs are unitless metrics, and they are the functions of the locations of the ABSs and the resource allocation procedure. Thus, we can combine them to define a reward function. Furthermore, the configuration of the system can be determined by the transmit channels of the ABSs $\boldsymbol{q}(t) = (q_1(t), \ldots, q_{|\mathcal{B}|}(t))$, the locations of the ABSs $\boldsymbol{Z}^{\text{ABS}}(t) = (\boldsymbol{z}_1^{\text{ABS}}(t), \ldots, \boldsymbol{a}_{|\mathcal{B}|}^{\text{ABS}}(t))$, and the association indicators $\boldsymbol{I}(t) = \{I_{u,b}\}_{b \in \mathcal{B}, k \in \mathcal{K}}$.

Our goal is to maximize an objective function which captures both fairness and load of the ABSs. In this regard, the optimization problem can be expressed as follows:

$$\max_{\boldsymbol{q}(t),\boldsymbol{Z}^{\mathrm{ABS}}(t)} \quad \sum_{t\in\mathcal{N}}\sum_{b\in\mathcal{B}}\sum_{k\in\mathcal{K}_b(t)}\Big(\phi_b\mathcal{F}(t)+\psi_b(1-\rho_b(t))\Big) \tag{19a}$$

$$\text{s.t.} \quad x_u(t)\in[x_{\min},x_{\max}], \quad \forall u\in\mathcal{U}, \tag{19b}$$

$$y_u(t)\in\big[y_{\min},y_{\max}\big], \quad \forall u\in\mathcal{U}, \tag{19c}$$

$$h_u(t)\in[h_{\min},h_{\max}], \quad \forall u\in\mathcal{U}, \tag{19d}$$

$$q_u(t)\in\mathcal{Q}_{\mathrm{U}}, \ \forall u\in\mathcal{U}, \tag{19e}$$

$$\rho_b(t)=f_b(\boldsymbol{\rho}), \ \forall b\in\mathcal{B}, \tag{19f}$$

$$0\le\rho_b(t)\le 1, \quad \forall b\in\mathcal{B}, \tag{19g}$$

$$I_{b,k}(t)\in\{0,1\}, \ \forall b\in\mathcal{B},\forall k\in\mathcal{K}, \tag{19h}$$

$$\sum_{b\in\mathcal{B}}I_{b,k}(t)\le 1, \ \forall k\in\mathcal{K}, \tag{19i}$$

where ϕ_b and ψ_b indicate the weight parameters for the fairness index and the load of ABS b on the objective function, respectively. $x_{\min}$ and $x_{\max}$ are the minimum and the maximum point of horizontal ordinate in a Cartesian coordinates of the system, respectively. $y_{\min}$ and $y_{\max}$ denote the minimum and the maximum point of vertical ordinate in a Cartesian coordinates of the system, respectively. $h_{\min}$ and $h_{\max}$ indicate the minimum and the maximum altitude of the UAVs, respectively. The constraints in (19b)–(19d) determine the feasible area in the 3D space for the locations of the UAVs at each time instant t in the system. The constraint in (19e) represents the constraint on the set of available channels for the UAVs. The constraints in (19f)–(19g) guarantee the limitation on the load of the ABSs. The constraints in (19h)–(19i) ensure each user k is associated with at most one ABS at each time instant t.

The following remarks characterize the difficulties in solving the problem formulated in (19). First, due to the presence of binary association indicators $\boldsymbol{I}(t)=\{I_{u,b}\}_{b\in\mathcal{B},k\in\mathcal{K}}$ and non-convex optimization problem, the problem in (19) is NP-hard. Moreover, due to the mobility of the users and the inherent highly dynamic nature of the system, the problem is very difficult to solve and it is intractable to find a globally optimal solution. Given the non-convexity and high complexity of the problem in (19), our pragmatic target is to find a high-performance solution in a reasonable amount of time.

Due to the inherent hyper-heterogeneity characteristics of SAGINs, we can use a hybrid method combination of a centralized and distributed approach. In this regard, we take advantage of both approaches. The advanced hardware processing units with fast computation speed and compatibility with various algorithms make to utilize the DQN algorithm in a distributed manner at the levels of the UAVs. The benefits of distributed approaches in wireless networks, such as reducing the signaling overhead and robustness to failures and attacks, have been widely recognized in the literature [8], [31]. For the centralized part, we assume that there is a cloud radio access network (C-RAN) for sharing information regarding the data rates of the users to calculate fairness [32]. In this regard, at the beginning of each time slot, the C-RAN broadcasts the calculated Jain's fairness index to the UAVs. Then, the UAVs are allowed to employ the broadcasted data and process their own information. Thus, Once a new UAV is launched into the system, it will first listen to the beacons, and then will start the action selection process. At the end of the time slot, each ABS calculates the data rates of its associated users and send these values to the C-RAN. This procedure results in a more adaptive and flexible system and can reap the benefit of both centralized and distributed approaches.

IV. DEEP REINFORCEMENT LEARNING-BASED LINK OPTIMIZATION

In this section, we first present an overview of Q-learning and DQN. Then, a DQN-based scheme for resource management and trajectory design is proposed. It utilizes both load and fairness using a replay memory method to achieve the formulated objective function which is described in (19). Since the load balancing and fairness optimization problem is a high dimensional and high state/action problem, we must employ novel and state-of-the-art methods such as DQN algorithm. The proposed algorithm enables UAVs to learn the entire network environment to adjust their positions jointly while determining their transmit channel. Finally, we present a detailed state, action, and reward function design.

A. LEARNING MODEL

The use of learning methods in wireless networks has received unprecedented attention, in which they show significant improvements over traditional mechanisms. Among them, reinforcement learning (RL), e.g., Q-learning, has achieved remarkable success for different problems in complex and highly dynamic systems. In RL, agents interact with the environment and take action. Then, they observe the consequences of their actions which can lead to learning their optimal policies. This success is due to the procedure of effectively finding the optimal policy for a finite Markov decision process (MDP). The MDP can be expressed as a four-tuple $<\mathcal{S},\mathcal{A},\mathcal{R},P>$, where $\mathcal{S}$ implies the observable environment states, $\mathcal{A}$ is the set of alternative actions. $\mathcal{R}$ indicates the reward function for taking action $a\in\mathcal{A}$ in state $s\in\mathcal{S}$ [33], [34], [35]. $P:\mathcal{S}\times\mathcal{A}\times R\to[0,1]$ is the state transition probability distribution function. The actions of an agent are selected based on a policy $\pi:\mathcal{S}\to\mathcal{A}$, which is a mapping from the state space to the action space. RL algorithms aim at learning an optimal policy $a=\pi(s)\in\mathcal{A}$. The agent will adjust its policy π in order to maximize its long-term expected return $E[G_n]$, which is given by [36]:

$$G(n)\triangleq\sum_{k=0}^{\infty}\gamma^k R(n+k), \tag{20}$$

where $G(n)$ is the accumulated discounted reward, and $0\le\gamma\le 1$ is the discount factor of future reward, which makes

trade-off immediate rewards with the rewards generated in future time instants. Let $Q_\pi(s,a)$ represent the action-value function of executing action a under state s following policy π as the average cumulative discount reward. The action-value function can be defined as follows:

$$Q_\pi(s,a) = E_\pi[G(n)|S(n)=s, A(n)=a]. \tag{21}$$

In Q-learning, an agent in a state takes an action and observes a reward. To select an action, it has two options: choosing an action with the highest Q-value or selecting a random action. Then, it updates the Q-table based on the observed reward. Q-learning is an off-policy reinforcement learning algorithm, in which it gradually improves its strategies with its accumulation of experience and strives to find the best action at any state. To evaluate the quality of an action-state pair, the algorithm updates the Q-value function using the Bellman equation according to the weighted average of the current Q-value function and the reward as follows [37]:

$$Q\big(s(t),a(t)\big) \leftarrow Q\big(s(t),a(t)\big) + \alpha\Big(r(t) + \gamma \max_a Q\big(s(t+1),a\big) - Q\big(s(t),a(t)\big)\Big) \tag{22}$$

where $Q(s(t),a(t))$ is the Q-value function for state $s(t)$ and action $a(t)$ at time t, and α is the learning rate. Note that this table-based reinforcement learning method is suitable for problems with limited action-state space. Despite the great empirical success of Q-learning, it is less applicable to real-world problems. This is due to the fact that most real-world problems are complex and have large or continuous action-state spaces so that they remain unaddressed hindering the deployment of Q-learning-based solutions. Therefore, using the table-based Q-learning algorithms to solve those problems is challenging and it is not feasible to apply them directly to complex and highly dynamic environments. To practically use RL algorithms for problems with large or continuous action-state space, the function approximation method can be employed. DQN is an extension of the Q-learning algorithm that combines deep neural networks with a reinforcement learning framework. In the DQN algorithm, a deep neural network is employed to approximate Q-values instead of using a Q-table to represent $Q(s(t),a(t))$ which can allow us to deal with large action-state spaces. However, the Q-Network will take the state as an input and return the expected Q-values for every action. Thus, $Q(s(t),a(t);\theta)$ is the estimated Q-value function during the iterative process, which is approximated by the neural network with the weights of θ. In the training process, $Q(s(t),a(t);\theta)$ is updated by adjusting weights θ. In our system, we choose to represent the state as a multi-dimensional array that contains the information of the 3D location and transmit channel of a UAV, in which the location is normalized and a one-hot decoder is used for the channel. The action space includes the movement direction and the transmit channel of UAVs. We will discuss further the elements of our proposed model in Section IV-B.

Given the environment, each UAV learns to take the best action depending on the current state during the training phase. In the Q-learning method, it updates its Q-table according to the returned reward value which shows how good it is to take a given action in a given state. On the other hand, in a DQN method, the model is not represented using a table, but it is represented by a set of weights and biases in a neural network referred to as a Q-network compared to the Q-table. The DQN is composed of two neural networks including the policy and the target network. To train the models, the weights and biases of policy and target networks are initialized randomly. To optimize the learning process, a replay memory, shown in Fig. 2, is incorporated for updating the Q-network [38]. Replay memory is an efficient technique to reuse previous experiences, and it allows the agent to learn from earlier memories. In this regard, experiences are stored in a memory buffer with a fixed size. When the replay memory is full, the oldest memories are erased [39]. Furthermore, to update the agent's parameters, a random batch of experiences is sampled from the replay memory. Using replay memory can address the issues relevant to the temporal correlations and enhances data usage and computation efficiency. It stores the agent's instances which include the past state, selected action, reward, and the next state given the selected action. Let $< s(t), a(t), r(t), s(t+1) >$ represent a sample from the replay memory. Then, the agent randomly samples a batch from the replay memory. To take an action, an ϵ-greedy model is used, which allows the agent to explore its action space, and it can be defined as follows:

$$a(t) = \begin{cases} \text{a random action}, & \text{with probability } \epsilon \\ \arg\max_{a\in\mathcal{A}} Q(s(t+1),a;\theta), & \text{with probability } 1-\epsilon, \end{cases} \tag{23}$$

where $\epsilon > 0$ is an exploring ratio which is adaptively updated according to the following expression:

$$\epsilon \leftarrow \epsilon_{\text{end}} + (\epsilon_{\text{start}} - \epsilon_{\text{end}})\exp\big(-\tau/\epsilon_{\text{decay}}\big), \tag{24}$$

where ϵ_{start} and ϵ_{end} denote the start value and the end value for the ϵ-greedy threshold, respectively. ϵ_{decay} is the threshold decay, and τ indicates as many as steps done for selecting an action based on the $Q(s(t),a;\theta)$. The output of the policy network $Q(s(t),a;\theta)$ is used as the decision of the agent, whereas the output of the target network is used to update the networks through computing a loss function which compares the outputs of the policy and target networks. To choose action $a(t)$, at time instant t, state $s(t)$ is fed into the neural network with weights θ, and $a(t)$ is obtained as $a(t) = \arg\max_a Q(s(t),a;\theta)$ or through a random selection according to (23), where $Q(s(t),a;\theta)$ denotes the outputs of the neural network corresponding to all possible actions a. After taking action $a(t)$, the agent received reward $r(t)$ and moves to the next state $s(t+1)$. Then, the DQN is trained by minimizing the prediction error of $Q(s(t),a(t);\theta)$ using the loss function $L_\delta(y,\hat{y})$. We use the Huber loss to minimize the loss so that when the loss is small, it acts as the mean

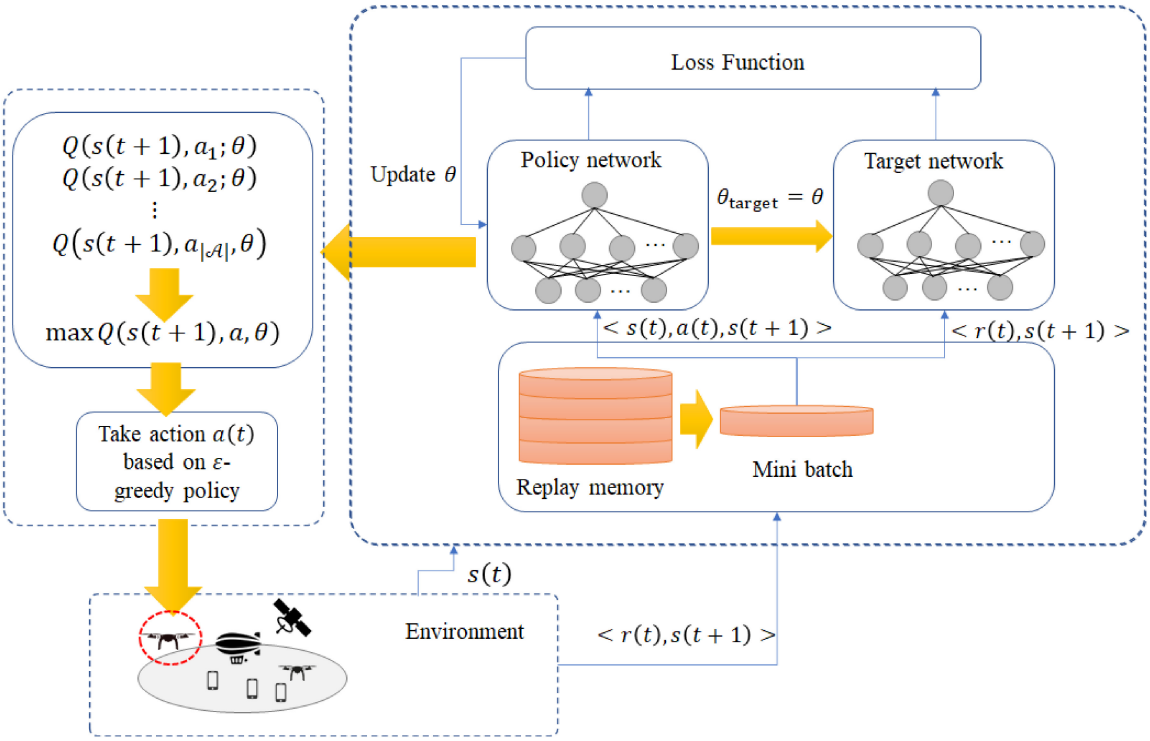

FIGURE 2. Overview of the DQN structure.

squared error (i.e., L2 loss), whereas the loss is large, it acts as the mean absolute error (i.e., L1 loss) which makes it more robust to outliers for the noisy estimations of the neural networks [40]. The loss is calculated over a batch of transitions sampled from the replay memory as follows [8]:

$$L_\delta(y, \hat{y}) = \begin{cases} \frac{1}{2}(y - \hat{y})^2, & \text{if} |y - \hat{y}| \le \delta \\ \delta |y - \hat{y}| - \frac{1}{2}\delta^2, & \text{otherwise,} \end{cases} \tag{25}$$

where y and $\hat{y}$ denote the output of the learning system, i.e., $Q(s(t), a(t); \theta)$, and the target value, respectively. The target value $\hat{y}$ can be estimated as

$$\hat{y} = r(t) + \gamma \max_a Q\big(s(t+1), a(t); \theta_{\text{target}}\big). \tag{26}$$

Parameter $\delta > 0$ specifies the threshold at which to change between delta-scaled L1 and L2 loss. Here, the target value $\hat{y}$ is computed based on the obtained reward and predicted discounted reward $\gamma \max_a Q(s(t+1), a(t); \theta_{\text{target}})$ given by the target network, where θ_{target} denotes the weights of the target network. Unlike the policy network, which continuously updates its weights based on the observed rewards and actions, the weights of the target network are not updated iteratively. Instead, they are periodically updated by copying the weights of the policy network after a specified time interval [41]. Note that to calculate the loss function, the agent picks a random batch from the replay memory rather than using a single sample which leads to improving the learning stability. After calculating the loss, it is fed into an optimizer to update the weights and biases of the neural networks. In our model, we use RMSprop optimizer which is an adaptive algorithm to evaluate gradient updates [42]. The update rules in RMSprop are as follows:

$$E\big[g^2\big]_t = \eta E\big[g^2\big]_{t-1} + (1 - \eta) g_t^2, \tag{27}$$

$$\theta(t+1) = \theta(t) - \frac{\alpha}{\sqrt{e\big[g^2\big]_t + \epsilon}} g_t, \tag{28}$$

where g_t is the gradient at time t. Parameters η and α denote the constant forgetting factor and the initial learning rate, respectively. θ is the weights and biases in the neural networks, respectively. Then, these updates are applied to the model. Fig. 2 illustrates a structure of a DQN approach for a UAV in an aerial network.

B. DQN-ASSISTED UAV OPERATION ALGORITHM

In the proposed approach, UAVs are seen as agents which interact with the system environment in a sequence of discrete time instances. At each time t, each UAV u observes the state $s_u(t)$, takes action $a_u(t)$ and receives the reward $r_u(t)$. Then, it moves to the new state $s_u(t+1)$ at time $t+1$. Furthermore, each UAV utilizes a replay memory $\mathcal{D}_u$ with a certain capacity to store the transition sample $< s_u(t), a_u(t), r_u(t), s_u(t+1) >$. In the context of the described problem, we define the state $s_u(t)$, action $a_u(t)$, and reward $r_u(t)$ for UAV u at time instant t as follows:

- State representation $\boldsymbol{s}_u(t)$: each UAV $u \in \mathcal{U}$ determines state $\boldsymbol{s}_u(t)$ from its location, i.e., $z_u^{\text{ABS}}(t) = (x_u(t), y_u(t), h_u(t))$, and transmit channel $q_u(t)$. Here, we introduce an encoder to encode the UAV's transmitted

channel information into a unique vector using a one-hot code. In one-hot encoding, a variable is represented by a one-hot vector, e.g., $1 \rightarrow [0, 1, 0, 0]$, $4 \rightarrow [0, 0, 0, 0, 1]$. More precisely, one-hot encoding is a process, which is used to convert categorical variables into a suitable form feeding to Q-networks [43]. Thus, the UAV translates each state into a $0-1$ string, and then it sends the state vector into the Q-network. The UAV's location is normalized by the minimum and the maximum values of the UAV's altitude, point of horizontal, and vertical ordinate. In this regard, the state of UAV u can be expressed as $\boldsymbol{s}_u(t) = \{\bar{\boldsymbol{x}}_u(t), \bar{q}_u(t)\}$. Here, $\bar{\boldsymbol{x}}_u(t) = (\frac{x_u(t)}{x_{\max}-x_{\min}}, \frac{y_u(t)}{y_{\max}-y_{\min}}), \frac{h_u(t)}{h_{\max}-h_{\min}})$ denotes the normalized value of the UAV's location $z_u^{\text{ABS}}(t)$, and $\bar{q}_u(t)$ is the one-hot encoded of the transmit channel $q_u(t) \in \mathcal{Q}$.

- Action: For each UAV $u \in \mathcal{U}$, action $\boldsymbol{a}_u(t) = \{z_u(t), q_u(t)\}$, where $z_u(t) \in \mathcal{Z}$ and $q_u(t) \in \mathcal{Q}$ denote the movement direction and transmit channel of UAV u at time t, respectively. The set of movement directions is defined as [44], [45], [46], [47], [48], [49], [50]

$$\mathcal{Z} = \{\text{up, down, left, right, forward, backward, fixed}\}. \tag{29}$$

Although UAVs can fly in arbitrary directions, modeling all possible movements can be computationally expensive and complex. By assuming a constant velocity and coordinated turns, the movement model can be simplified to a smaller number of directions [48]. Although this simplification may not capture all possible movement directions, it allows us to balance between accuracy and computational complexity which provides a more tractable and computationally efficient model. Therefore, the action space for each UAV $u \in \mathcal{U}$ can be described as

$$\mathcal{A} = \{\boldsymbol{a}_u(t) | \boldsymbol{a}_u(t) = \{z_u(t), q_u(t)\}, z_u(t) \in \mathcal{Z}, q_u(t) \in \mathcal{Q}\}. \tag{30}$$

- Reward: the reward is the objective of the dynamic resource management and trajectory design problem. This function is consistent with the mathematical formulation of our optimization problem. Thus, for each UAV $u \in \mathcal{U}$, the reward function is related to load and fairness, and according to (19), it can be defined as follows:

$$r_u(t) = \phi_u \mathcal{F}(t) + \psi_u(1 - \rho_u(t)). \tag{31}$$

After taking action $a_u(t)$ by UAV u, it receives the reward $r_u(t)$ and moves to the new state $s_u(t+1)$.

To approximate the Q-function values, each UAV utilizes two deep networks for policy and target networks with the same four fully connected layers while they have different weights and biases. From Fig. 3, we can see that the neural network is composed of three parts, including the input layer, hidden layers, and output layer. In our model, we

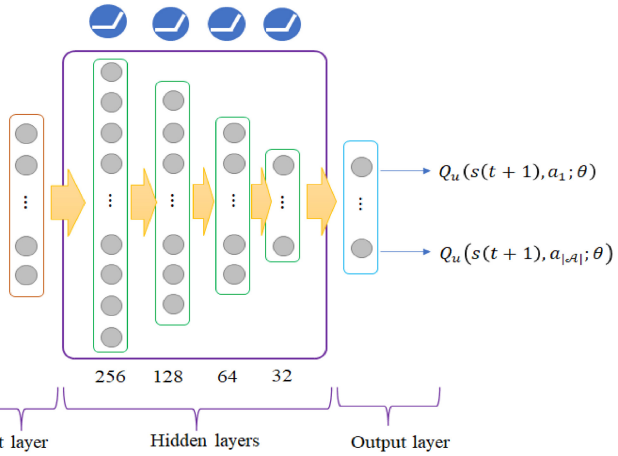

FIGURE 3. Structure of the policy network.

employ 4 hidden layers with 256, 128, 64, and 32 nodes. For the activation function, ReLU is selected. Furthermore, after each layer except the output layer, we apply layer normalization and drop out with probability 0.2. The layer normalization technique can enhance the performance and stability of neural networks [51]. It normalizes the inputs to a layer, thereby enabling the utilization of higher learning rates and faster convergence. The dropout technique operates by randomly disconnecting the connections between neurons in connected layers based on a certain dropout rate to reduce the dependency between neurons [52]. The input of the neural network corresponds to the state of the UAV, and the output corresponds to action-value approximations. For the policy network, the input is the current state-action pair $(s_u(t), a_u(t))$ and the output is the predicted value $Q(s_u(t), a_u(t); \theta)$. For the target network, the input is the next state $s_u(t+1)$ and the output is the maximum Q-value of the next state-action pair so that the target value of $(\boldsymbol{s}_u(t), \boldsymbol{a}_u(t))$ for UAV u can be calculated as follows:

$$\hat{y}_u(\boldsymbol{s}_u(t), \boldsymbol{a}_u(t)) = r_u(t) + \gamma \max_a Q_u\big(\boldsymbol{s}_u(t+1), \boldsymbol{a}_u(t); \theta_{\text{target}}\big). \tag{32}$$

To address the challenges of slow learning and increased sample complexity in our learning method, we employ several strategies. These strategies include the use of replay memory to break temporal correlations, the introduction of a target network to stabilize learning, reward, state and reward normalization techniques, and a balanced exploration-exploitation trade-off using an exploring ratio. By carefully designing our algorithm and tuning parameters, we aim to optimize learning efficiency, achieve satisfactory performance, and strike a balance between exploration and exploitation. These collective strategies can help to overcome the challenges and improve the effectiveness of our learning approach.

It is important to note that each UAV has its own DQN network, with its own unique set of neural network weights, distinct from the other UAVs. Algorithm 1 presents the

Algorithm 1 DQN-Based Algorithm for 3D Trajectories and Resource Management in Aerial HetNets

1: **Input**: a differentiable Q-value function parameterization $Q_u(s,a;\theta)$ and $\theta_{\text{target}}=\theta$, $\forall u\in\mathcal{U}$
2: **Initialization**: a replay memory $\mathcal{D}_u, \forall u\in\mathcal{U}$
Initialization of the UAVs' locations:
3: $\boldsymbol{f}^{\text{ABS}}(0)\Leftarrow\boldsymbol{f}_{\text{H}}$, $\mathcal{B}^*\Leftarrow\mathcal{H}$, $u=0$, $\boldsymbol{f}_u^{\text{UAV}}(0)=\{\}, \forall u\in\{1,\ldots,|\mathcal{U}|\}$
4: **while** $u<|\mathcal{U}|$ **do**
5: **for** $\forall l\in\mathcal{L}$ **do**
6: **for** $\forall b\in\mathcal{B}^*$ **do**
7: $r_{l,b}=||\boldsymbol{f}_l-\boldsymbol{f}_b^{ABS}||$
8: **end for**
9: $r_l^{\min}=\min_{b\in\mathcal{B}^*} r_{l,b}$
10: **end for**
11: $l^*=\arg\max_{l\in\mathcal{L}} r_l^{\min}$
12: $\mathcal{L}\Leftarrow\mathcal{L}\backslash\{l^*\}$, $\boldsymbol{f}^{\text{ABS}}(0)\Leftarrow\boldsymbol{f}^{\text{ABS}}(0)\cup\{\boldsymbol{f}_{l^*}\}$, $\mathcal{B}^*\Leftarrow\mathcal{B}^*\cup\{u\}$, $\boldsymbol{f}_u^{\text{UAV}}(0)=\boldsymbol{f}_{l^*}$
13: $u\Leftarrow u+1$
14: **end while**
Learning procedure:
15: **for** episode: $=1, N_{\text{episode}}$ **do**
16: **while** $t<N$ **do**
17: $t\leftarrow t+1$
18: **for each** $k\in\mathcal{K}$ **do**
19: Update $z_k^{\text{UE}}(t)$ based on the random walk mobility model described in Section III-B
20: Associate user k to an ABS according to (16)
21: Update the user association indicators according to (7) and (10)
22: **end for**
23: **for each** $u\in\mathcal{U}$ **do**
24: Select an action according to (23)
25: Update the location $z_u^{\text{ABS}}(t)$ based on $a_u(t)$ and (2)
26: Calculate reward $r_u(t)$ according to (31) and move to the next state $s_u(t+1)$
27: Store the transition sample $<s_u(t),a_u(t),r_u(t),s_u(t+1)>$ into $\mathcal{D}_u$
28: Sample a stochastic minibatch of samples from $\mathcal{D}_u$
29: Compute target value according to (32)
30: Update weights θ by minimizing the loss (25)
31: Update the target network parameters θ_{target} every N_{T} steps as $\theta_{\text{target}}=\theta$
32: **end for**
33: **end while**
34: **end for**

pseudocode for our proposed approach. It is noteworthy that we apply the heuristic-based initialization for our proposed DQN-based approach. In this algorithm, the horizontal location of a new UAV is determined based on the furthest distances from the other BSs in the system [53] (lines 3-14). We define the set of all predefined locations for the UAVs as $\mathcal{L}$ and a single location in this set as l. The two-dimensional (2D) coordinate of a location l is represented by $\boldsymbol{f}_l$ while the vector composed of the locations of the ABSs in the system is represented by $\boldsymbol{f}^{\text{ABS}}(0)$. The initial ABS locations, $\boldsymbol{f}^{\text{ABS}}(0)$, are determined by the HAPSs, as $\boldsymbol{f}^{\text{ABS}}(0)\Leftarrow\boldsymbol{f}_{\text{H}}$, where $\boldsymbol{f}_{\text{H}}=(\boldsymbol{f}_1^{\text{ABS}},\ldots,\boldsymbol{f}_{|\mathcal{H}|}^{\text{ABS}})$ represents the 2D locations of all HAPSs. $f_u^{\text{UAV}}(0)$ denotes the selected location for UAV u. The set of current ABSs in the system, $\mathcal{B}^*$, is initialized with the set of HAPSs. At each iteration, the algorithm determines the initial location of a new UAV. The 2D distance, $r_{l,b}$, between each ABS b in $\mathcal{B}^*$ and location l is calculated (lines 6-8). Then, the distance between location l and the nearest ABS in set $\mathcal{B}^*$, which is denoted by $r_l^{\min}$ is calculated (line 9). Finally, the location l^* with the farthest distance from ABSs in $\mathcal{B}^*$ is selected as the UAV location, denoted as l^* (line 11). Then, the location l^* is removed from $\mathcal{L}$ and its coordinate $\boldsymbol{f}_{l^*}$ is added to the ABS locations in $\boldsymbol{f}^{\text{ABS}}(0)$ (line 12). To initialize the transmit channels of the UAVs, we adopt a random selection, in which the UAVs choose their channels from a uniform distribution, i.e., $\pi_{u,q}=\frac{1}{|\mathcal{Q}|}$ for $\forall u\in\mathcal{U}$ and $\forall q\in\mathcal{Q}$, where $\pi_{u,q}$ is the probability assigned channel $q\in\mathcal{Q}$ for UAV $u\in\mathcal{U}$.

Based on Alg. 1, we can derive the time complexity of the proposed DQN scheme. If we consider the steps 5-13 takes t_0, steps 19-21 takes t_1, steps 24-31 takes t_2, then the total time taken can be expressed as $t_0\cdot|\mathcal{U}|+N_{\text{episode}}\cdot(N\cdot(|\mathcal{K}|\cdot t_1+|\mathcal{U}|\cdot t_2))$. The main term affecting the execution time in this expression is $N\cdot N_{\text{episode}}$. Therefore, the time complexity can be derived as $O(N\cdot N_{\text{episode}})$. For the Q-learning algorithm, the worst-case complexity for action executions has a bound of $O(n_s^3)$ [54], where n_s is the size of the state space. Therefore, the Q-learning in our paper has the time complexity as $O(N\cdot n_s^3)$. In addition, because the use of HAPSs mainly affects the initialization phase in Alg. 1, i.e., ABS locations, the complexity upper bounds will be kept the same for the DQN and Q-learning schemes without using HAPSs.

V. SIMULATION RESULTS

In the simulation scenario, a 1000×1000 m^2 area is considered, and a set of users are uniformly distributed throughout the area. Furthermore, a HAPS is located at the center of the area at a height of 20 km from the ground [55]. Table 1 summarizes the system parameters employed in the simulations. The simulation results are obtained by averaging over numerous independent runs with variations using practical configurations. Furthermore, the performance of our proposed DQN scheme is evaluated through comparison with several benchmark algorithms as follows:

- *DQN-No HAPS:* To demonstrate the advantages of incorporating HAPS, the DQN-No HAPS scheme is implemented. In this approach, only UAVs are employed

TABLE 1. System-level simulation parameters.

System Parameters	
Parameter	**Value**
Height of the HAPS	20 km [55]
$h_{\min}, h_{\max}$	22.5 m, 150 m
Height of users	1.5 m
Carrier frequency UAV and HAPS	28, 2.11 GHz
$\lvert\mathcal{Q}_{\mathrm{U}}\rvert, \lvert\mathcal{Q}_{\mathrm{M}}\rvert$	4, 1
$\omega_{\mathrm{U}}, \omega_{\mathrm{M}}$	56, 14 MHz
Noise power spectral density	-174 dBm/Hz
Number of HAPSs	1
Height of HAPSs	20 km
Total number of iterations (N)	5740
T_s	1 sec
Fixed point iterations (N_{FP})	500
ρ_b^0	0.5
α, β, ξ	0.1, 750, 8
$v_{\mathrm{UE}}^{\min}, v_{\mathrm{UE}}^{\max}$	0, 1.3 m/sec
v_{U}	10 m/sec
ϑ_k/ζ_k	1.8 Mbps
ϕ_b, ψ_b	0.5, 0.5
Transmit power of the HAPS and UAVs	43, 24 dBm
Reference path loss	61.4 [20]
Path loss exponent LoS/NLoS	2, 3
Shadowing standard deviation LoS/NLoS	5.8, 8.7
γ	0.999
Batch size	128
Repaly memory size	5000
Minimum repaly memory size	264
Target network frequency update (N_{T})	10
$\epsilon_{\mathrm{start}}, \epsilon_{\mathrm{end}}, \epsilon_{\mathrm{decay}}$	0.9, 0.5, 200

for data transmission to the users, without the presence of HAPS. The UAVs optimize their trajectories and transmit channels using the proposed DQN algorithms.

- *Q-learning:* In the Q-learning approach, both UAVs and HAPSs are deployed to provide service to the users. The 2D positions and the transmit channels of the UAVs are optimized using a Q-learning technique. The altitude of the UAVs is set at $h_{\max}$.
- *Q-learning-No HAPS:* In this benchmark algorithm, no HAPSs are employed, and only UAVs provide service for users. The UAVs optimize their trajectories and transmit channels using a Q-learning technique, flying at the fixed altitude of $h_{\max}$.

Fig. 4 presents the impact of the number of UAVs on Jain's fairness index defined in (17), which is used to quantify the distribution of resources among the users in the system. It shows that as the number of UAVs increases, Jain's fairness index improves. The main reason is that the additional UAVs provide more resources and coverage to the network which leads to a more fair distribution of resources among users and ensures an enhanced user experience and improved network performance. Furthermore, the DQN approach significantly outperforms the benchmark algorithms. This improved performance is due to the ability of the DQN approach to learn from experience and adapt to changing conditions in the system. However, for a system with a single UAV, the Q-learning (or Q-learning-No HAPS) approach slightly performs better than DQN (or

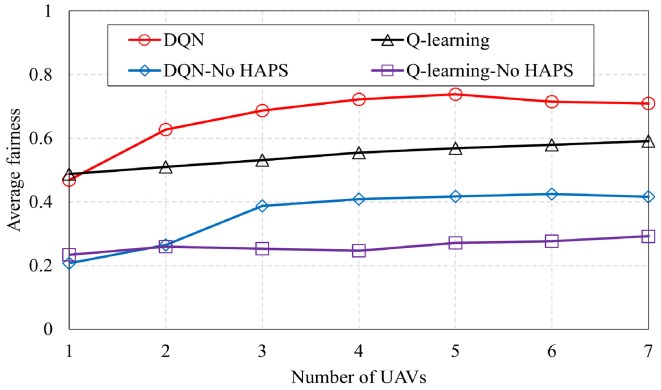

FIGURE 4. Average fairness versus the number of UAVs for a system with 200 users.

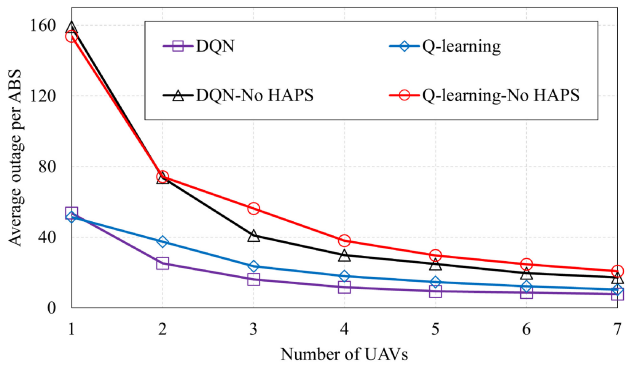

FIGURE 5. Average outage per ABS versus the number of UAVs for a system with 200 users.

DQN-No HAP) mechanism. This is due to the fact that in the Q-learning algorithm, the altitude of the UAV is set at the maximum altitude. Thus, it can cover more area and support more users due to providing a high probability of LoS links.

As shown in Fig. 5, the performance of the proposed DQN approach is compared with the benchmark algorithms in terms of outage users. The figure illustrates the relationship between the number of UAVs and the average number of outage users and the scalability of our proposed DQN approach. Outage users refer to users which experience disconnection or a drop in the received data rate. Therefore, it is imperative for network operators and service providers to effectively monitor and manage the number of outage users to ensure the sustainability and reliability of the network. In addition, the number of outage users is a critical performance metric and can be used to assess the efficacy of network optimization strategies and resource allocation algorithms. From Fig. 5, it can be observed that, as the number of UAVs increases, the average number of outage users per ABS decreases for all methods. However, the proposed DQN approach outperforms the benchmark algorithms, demonstrating its effectiveness in reducing the number of outage users and improving service coverage. This result highlights the effectiveness of the proposed DQN approach in improving the resource allocation and 3D trajectory design

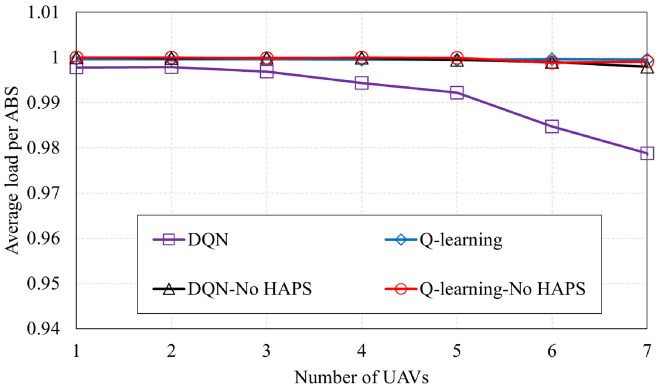

FIGURE 6. Average load per ABS versus the number of UAVs for a system with 200 users.

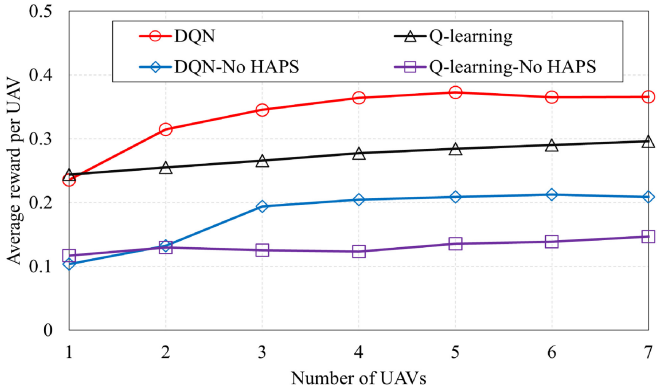

FIGURE 7. Average reward per UAV versus the number of UAVs for a system with 200 users.

for UAV-based communication systems. The reason for the decrease in the average number of outage users as the number of UAVs increases is due to the improved resource allocation and more efficient utilization of the available UAVs. With a larger number of UAVs, the loads are balanced over the ABSs, and thus more users can be served which reduces the number of users without service. Additionally, having more UAVs with effective interference management methods enables a more flexible design and better serves the users. Note that increasing the number of ABSs in the system may cause more interference if the resource is not allocated properly. The benchmark algorithms without HAPSs which only employ UAVs, degrade the performance in terms of outage users. This is due to the limited coverage area of UAVs which leads to inadequate service quality for some users, especially in dense areas. In contrast, the proposed DQN approach which leverages both UAVs and HAPSs, provides a larger coverage area and improved service quality, thus it reduces the number of outage users.

Fig. 6 shows the average load per ABS as the number of UAVs increases. The results indicate that in the proposed approach, as the number of UAVs increases, the average load per ABS decreases. This behavior helps to alleviate the overloading of the ABSs and ensures efficient and stable service provided to the users. Furthermore, the results indicate that the benchmark methods are not capable of effectively balancing the load in the system, in which with the increasing number of UAVs, there is a limited decrease in average load. Specifically, for the dense deployment of UAVs, the proposed DQN approach shows an improvement in terms of load balancing compared to the benchmark algorithms. The gap between the proposed approach and the benchmark algorithms becomes larger as the number of UAVs increases which demonstrates the effectiveness of the DQN approach in ensuring a balanced distribution of load among the ABSs in densely deployed UAVs scenarios by managing the resource and optimizing the 3D locations of the UAVs.

The average reward, defined in (31), per UAV as a function of the number of UAVs is depicted in Fig. 7. The average reward can be considered as a suitable performance metric to assess the success of the methods in optimizing the system's objective. A higher reward value indicates that the algorithm is successful in satisfying the objective function, while a lower reward value indicates that the algorithm is encountering difficulties to achieve desired outcomes. As illustrated in Fig. 7, the DQN approach outperforms the benchmark algorithms in terms of the average reward achieved by the UAVs. The improvement in reward achieved by the proposed DQN approach is a result of the decreased load of the ABSs and improved fairness among users. By optimizing the UAV's trajectories and transmission channels, the DQN approach ensures an equitable distribution of resources, thereby improving both load balancing and fairness. Since the reward function captures both load and fairness, thus improving both parameters results in a higher overall reward compared to the benchmark algorithms. However, it should be noted that for scenarios involving a single UAV, the Q-learning approach performs slightly better than the DQN approach. This is due to the fact that in the Q-learning method, the altitude of the UAV is set at the maximum altitude, enabling it to cover a larger area with a high probability of LoS. Additionally, in the scenario with a single UAV, due to the lack of interference, setting the altitude of the UAV at the maximum altitude results in improving the performance of the Q-learning method. However, employing more UAVs may increase interference in the system, which requires critical factors such as load balancing and fairness provisioning to be optimized dynamically and intelligently. Additionally, the scenarios without the utilization of the HAPS result in a decreased reward compared to the scenarios that employ the HAPS. The main reason is that the HAPS provides an additional layer of support for service coverage, which leads to improved fairness.

Fig. 8 illustrates the average rate per user versus the number of UAVs deployed in the system. This figure shows a comparison of the performance of the proposed DQN approach with the benchmark algorithms and provides insights into the impact of the number of UAVs on the system performance in terms of users' rates. It can be observed that with increasing the number of UAVs, the average rate per user tends to improve. This is due to providing more resource for the users, and thus they have more opportunities

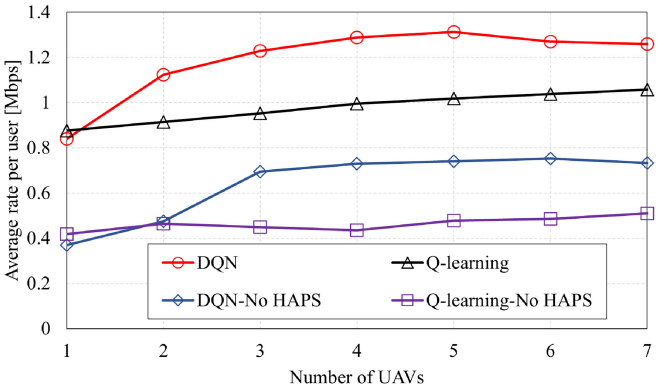

FIGURE 8. Average rate per user versus the number of UAVs for a system with 200 users.

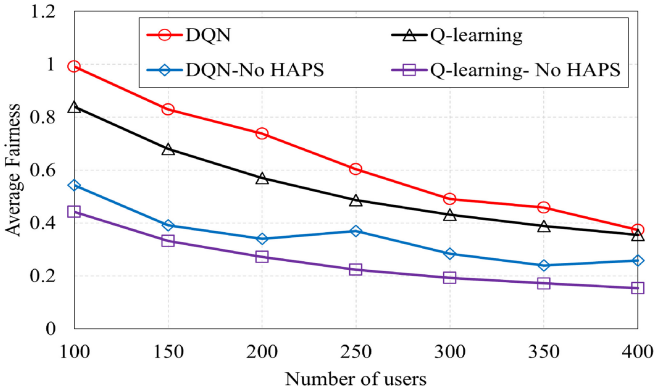

FIGURE 9. Average fairness versus the number of users for a system with 5 UAVs.

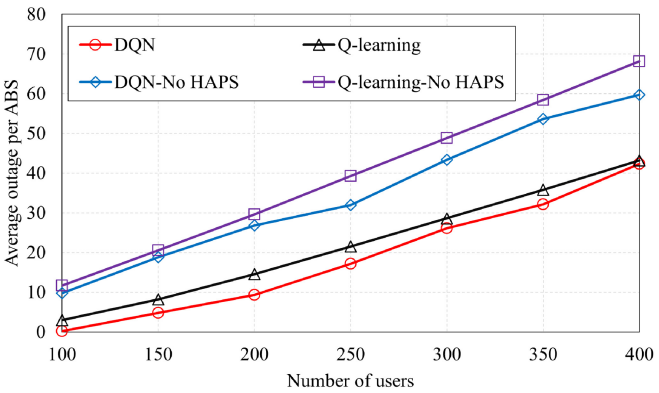

FIGURE 10. Average outage per ABS versus the number of users for a system with 5 UAVs.

to select their serving ABSs which lead to offloading outage users from highly loaded ABSs to lightly loaded ABSs. In addition, the DQN approach manages interference efficiently in the system and this can lead to an increase in the number of users served by the ABSs which results in higher user rates. Similarly, by optimizing the location of UAVs and resource allocation, it is possible to reduce interference and improve user rates. Note that increasing the number of UAVs may cause increasing interference and the overloading of ABSs. However, the joint channel allocation and trajectory optimization techniques allow us to effectively manage interference in the UAV-assisted networks. Thus, by strategically deploying a moderate number of UAVs, we can effectively enhance the overall network capacity and improve user experience. In addition, for the initialization of UAVs' locations, we use an algorithm based on the furthest distances from the other ABSs in the system to reduce the interference among the ABSs in the system. Furthermore, this figure shows the integration of the HAPS and UAVs can improve the user rate significantly compared to conventional aerial communication systems. For instance, the proposed DQN approach enhances the user rate up to about 77% compared to the DQN-No HAPS method for 2 UAVs. It is important to note that the improvement in user rate depends on the deployment scenario, resource allocation, and the number of ABSs used.

Fig. 9 shows the performance of the DQN approach and the benchmark algorithms in terms of Jain's fairness index versus different numbers of users. This figure can measure the distribution of resource among the users in the system. We can observe that the DQN scheme achieves improved performance in terms of fairness compared to the benchmark algorithms. This can provide valuable insight into the scalability, flexibility, and ability of the DQN method to allocate resources fairly for a varying number of users based on the states of the system. Furthermore, the performances of all methods decrease as the number of users in the system increases. This is due to the fact that as the number of users increases, the availability of resources in the system becomes limited. Thus, it shows the importance of effectively and fairly allocating resources among the users. Furthermore, in the absence of the HAPS, only the UAVs provide service for the users which can lead to a decrease in Jain's fairness index as resources are not distributed fairly among users.

Fig. 10, illustrates the average number of outage users for the DQN method and the benchmark algorithms. It can be seen that the DQN approach yields better performance compared to the benchmark algorithms. Obviously, for a fixed number of ABSs, as the number of users in the system increases, the demand for resources also increases, potentially leading to a higher number of outage users. Moreover, Fig. 10 demonstrates the contribution of the HAPS to the reduction of outage users. The deployment of HAPSs has the potential to significantly decrease the number of outage users and can help to alleviate resource scarcity in the dense system. In the absence of HAPSs, the system relies solely on UAVs, which can lead to limited network coverage, resulting in a higher number of outage users.

In Fig. 11, the average reward is plotted versus the number of users in the system to evaluate the performance of all the methods under different load conditions. As the number of users in the system increases, the UAVs are faced with a greater challenge in balancing the load and distributing resources fairly and efficiently. We can observe that the DQN algorithm demonstrates higher rewards compared to the benchmark algorithms due to its improved performance in load balancing and fairness. Compared to the benchmark

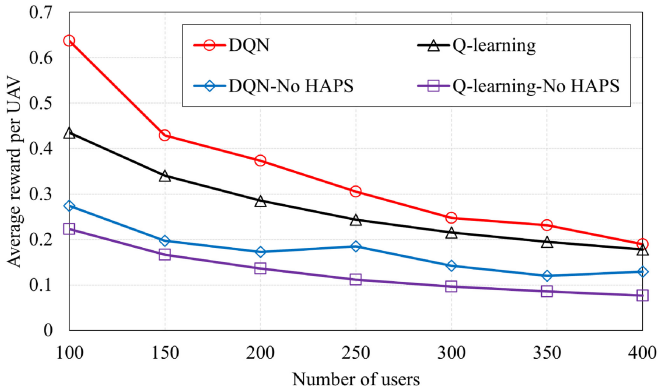

FIGURE 11. Average reward per UAV versus the number of users for a system with 5 UAVs.

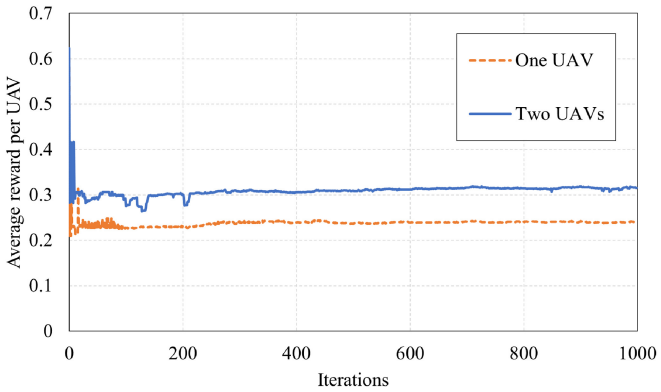

FIGURE 12. The convergence behavior of the DQN approach for the scenarios with one and two UAVs and 200 users.

algorithms, the DQN algorithm achieves better load balancing by dynamically adjusting resource allocation through channel allocation and 3D trajectory design based on the current state of the system.

Fig. 12 shows the convergence behavior of our proposed approach for the systems composed of one UAV and two UAVs with 200 users. We can observe that the DQN approach converges within reasonable numbers of iterations for these scenarios. For the case of a single UAV, the DQN algorithm converges after approximately 100 iterations. However, for the scenario with two UAVs, it converges at around 220 iterations. Note that the convergence behavior is influenced by the number of UAVs in the system, in which increasing the number of UAVs can lead to slower convergence. This is due to the fact that with increasing the number of UAVs, interactions between multiple UAVs (i.e., agents) increase which results in growing the complexity of the problem. It is important to note that the convergence characteristics can be impacted by the problem's complexity, network conditions, and parameter settings.

VI. CONCLUSION

In this paper, we have addressed an important problem of joint trajectory and resource management design in HAPS-UAV-enabled heterogeneous networks composed of HAPSs and UAVs as ABSs. To solve the problem, we have employed a DQN algorithm which is able to handle the complexity of the problem. Moreover, we have utilized a fixed-pint iteration method to find the load of ABSs. Simulation results have shown that the integration of HAPSs and UAVs can significantly improve the performance of the network compared to conventional communication systems and a Q-learning-based mechanism in terms of fairness, user rate, and outage.

APPENDIX
PROOF OF SIF FOR LOAD

To demonstrate that ρ_b in our paper satisfies the three conditions of a standard interference function (SIF), we refer to [25, Proposition 1], which states that concave functions are SIFs. Our goal is to establish that ρ_b is indeed a concave function. To begin the proof, let to write the load function as the composition $\rho_b = \sum_{k\in\mathcal{K}_b}(F \circ G_b)(., k)$ with

$$F(z) = \frac{1}{\log_2\left(1 + z^{-1}\right)} \tag{33}$$

and

$$G_b(k, \boldsymbol{\rho}) = \frac{1}{I_{b,k} p_b g_{b,k}}\Big(\sum_{b'\in\mathcal{B}\setminus b} p_{b'} g_{b',k} \rho_{b'} \mathbb{1}_{(q_b(t)=q_{b'}(t))} + \sigma_0^2\Big). \tag{34}$$

Since summation over the users is a linear function and function $G_b(k, \boldsymbol{\rho})$ is an affine function. Thus, it is needed to show that function $F(z)$ is concave and the second derivative of function $F(z)$ for $z > 0$ is negative. The second derivative of function $F(z)$ is as follows:

$$F''(z) = -\frac{\ln(2)\left((2z+1)\ln\left(1+\frac{1}{z}\right) - 2\right)}{z^2(z+1)^2\ln^3\left(1+\frac{1}{z}\right)}. \tag{35}$$

The condition for f(z) to be concave (i.e., $F''(z) < 0$) is satisfied by fulfilling the following inequality:

$$\ln\left(1 + \frac{1}{z}\right) > \frac{2}{2z+1}, \quad z > 0. \tag{36}$$

Now, we need to show that (36) is fulfilled. In this regard, we define the following functions, which are presented in (36), as follows:

$$F_1(z) = \ln\left(1 + \frac{1}{z}\right) \tag{37}$$

and

$$F_2(z) = \frac{2}{2z+1}. \tag{38}$$

Fig. 13 shows functions $F_1(z)$ and $F_2(z)$. In the limit, when $z \to 0$, we have

$$\lim_{z\to 0} F_1(z) = +\infty, \qquad \lim_{z\to 0} F_2(z) = 2. \tag{39}$$

Thus, in the case that $z \to 0$, $\lim_{z\to 0} F_1(z) > \lim_{z\to 0} F_2(z)$. Furthermore, the limits for $z \to \infty$ are

$$\lim_{z\to\infty} F_1(z) = 0, \qquad \lim_{z\to\infty} F_2(z) = 0. \tag{40}$$

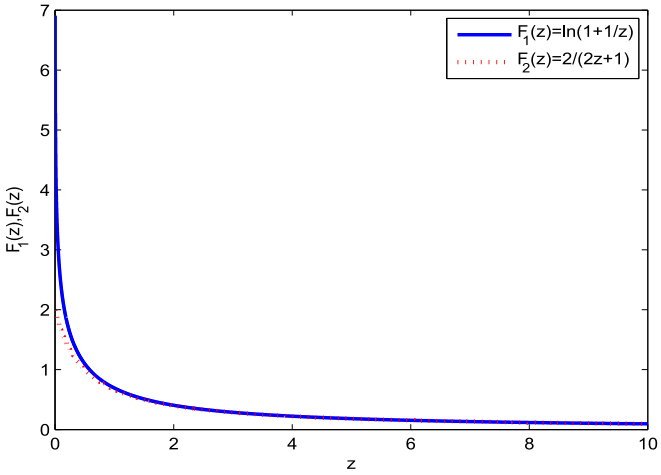

FIGURE 13. $F_1(z)$ and $F_2(z)$.

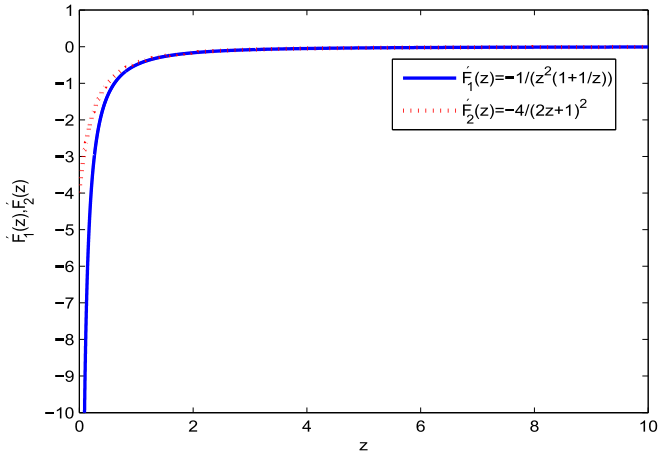

FIGURE 14. $F_1'(z)$ and $F_2'(z)$.

Therefore, in the case that $z \to \infty$, the limits of both functions, $F_1(z)$ and $F_2(z)$, approach to zero. To show the validity of $F_1(z) > F_2(z)$ for $z > 0$, we obtain the derivative of both functions as follows:

$$F_1'(z) = -\frac{1}{z^2\left(1+\frac{1}{z}\right)} < 0, \qquad F_2'(z) = -\frac{4}{(2z+1)^2} < 0 \tag{41}$$

Fig. 14 shows functions $F_1'(z)$ and $F_2'(z)$. Now, using the recursive method, we investigate the validity of $F_1'(z) < F_2'(z)$. Thus, we assume that this is valid for $z \neq \infty$. Thus, we have

$$-\frac{1}{z^2\left(1+\frac{1}{z}\right)} < -\frac{4}{(2z+1)^2} \Rightarrow 4z^2+4z < 4z^2+4z+1 \Rightarrow 1 > 0 \tag{42}$$

We can see that (42) is trivial, indicating that all the equations are recursive and the proposition $F_1'(z) < F_2'(z)$ holds for $z \neq \infty$ values.

REFERENCES

[1] "High altitude platform systems-towers in the skies," GSMA, U.K., Rep. Feb. 2022. [Online]. Available: https://www.gsma.com/futurenetworks/wp-content/uploads/2021/06/GSMA-HAPS-Towers-in-the-skies-Whitepaper-2021.pdf

[2] G. Hao, W. Ni, H. Tian, and L. Cao, "Mobility-aware trajectory design for aerial base station using deep reinforcement learning," in *Proc. Int. Conf. Wireless Commun. Signal Process. (WCSP)*, 2020, pp. 1131–1136.

[3] J. Guo et al., "3D aerial vehicle base station (UAV-BS) position planning based on deep Q-learning for capacity enhancement of users with different QoS requirements," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, 2019, pp. 1508–1512.

[4] L. Wang, K. Wang, C. Pan, X. Chen, and N. Aslam, "Deep Q-network based dynamic trajectory design for UAV-aided emergency communications," *J. Commun. Inf. Netw.*, vol. 5, no. 4, pp. 393–402, Dec. 2020.

[5] Q. Liu, L. Shi, L. Sun, J. Li, M. Ding, and F. Shu, "Path planning for UAV-mounted mobile edge computing with deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5723–5728, May 2020.

[6] A. H. Arani, M. M. Azari, P. Hu, Y. Zhu, H. Yanikomeroglu, and S. Safavi-Naeini, "Reinforcement learning for energy-efficient trajectory design of UAVs," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 9060–9070, Jun. 2022.

[7] F. Tang, Y. Zhou, and N. Kato, "Deep reinforcement learning for dynamic uplink/Downlink resource allocation in high mobility 5G HetNet," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 12, pp. 2773–2782, Dec. 2020.

[8] F. Tang, H. Hofner, N. Kato, K. Kaneko, Y. Yamashita, and M. Hangai, "A deep reinforcement learning-based dynamic traffic offloading in space-air-ground integrated networks (SAGIN)," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 276–289, Jan. 2022.

[9] A. H. Arani, P. Hu, and Y. Zhu, "Re-envisioning space-air-ground integrated networks: Reinforcement learning for link optimization," in *Proc. IEEE Int. Conf. Commun.*, 2021, pp. 1–7.

[10] S. Yuan, F. Hsieh, S. Rasool, E. Visotsky, M. Cudak, and A. Ghosh, "Interference analysis of HAPS coexistence on terrestrial mobile networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2022, pp. 2494–2499.

[11] Z. Jia, M. Sheng, J. Li, D. Zhou, and Z. Han, "Joint HAP access and LEO satellite backhaul in 6G: Matching game-based approaches," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 4, pp. 1147–1159, Apr. 2021.

[12] H. Ahmadinejad and A. Falahati, "Forming a two-tier heterogeneous air-network via combination of high and low altitude platforms," *IEEE Trans. Veh. Technol.*, vol. 71, no. 2, pp. 1989–2001, Feb. 2022.

[13] A. H. Arani, P. Hu, and Y. Zhu, "Fairness-aware link optimization for space-terrestrial integrated networks: A reinforcement learning framework," *IEEE Access*, vol. 9, pp. 77624–77636, 2021.

[14] A. H. Arani, P. Hu, and Y. Zhu, "UAV-assisted space-air-ground integrated networks: A technical review of recent learning algorithms," 2022, *arXiv:2211.14931*.

[15] X. Cao, B. Yang, C. Yuen, and Z. Han, "HAP-reserved communications in space-air-ground integrated networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 8, pp. 8286–8291, Aug. 2021.

[16] S. R. Shubha Sharma, N. Vishwakarma, and A. S. Madhukumar, "HAPS-based relaying for integrated space-air-ground networks with hybrid FSO/RF communication: A performance analysis," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 3, pp. 1581–1599, Jun. 2021.

[17] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Commun. Mobile Comput.*, vol. 2, no. 5, pp. 483–502, 2002.

[18] Y. Shibata, W. Takabatake, K. Hoshino, A. Nagate, and T. Ohtsuki, "Two-step dynamic cell optimization algorithm for HAPS mobile communications," *IEEE Access*, vol. 10, pp. 68085–68098, 2022.

[19] J. Holis and P. Pechac, "Elevation dependent shadowing model for mobile communications via high altitude platforms in built-up areas," *IEEE Trans. Antennas Propag.*, vol. 56, no. 4, pp. 1078–1084, Apr. 2008.

[20] G. Fontanesi, A. Zhu, and H. Ahmadi, "Outage analysis for millimeter-wave Fronthaul link of UAV-aided wireless networks," *IEEE Access*, vol. 8, pp. 111693–111706, 2020.

[21] M. M. Azari, G. Geraci, A. Garcia-Rodriguez, and S. Pollin, "UAV-to-UAV communications in cellular networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 9, pp. 6130–6144, Sep. 2020.

[22] A. J. Fehske and G. P. Fettweis, "Aggregation of variables in load models for interference-coupled cellular data networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2012, pp. 5102–5107.

[23] S. Samarakoon, M. Bennis, W. Saad, and M. Latva-aho, "Dynamic clustering and on/off strategies for wireless small cell networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 3, pp. 2164–2178, Mar. 2016.

[24] A. Hajijamali Arani, M. J. Omidi, A. Mehbodniya, and F. Adachi, "Minimizing base stations' ON/OFF switchings in self-Organizing heterogeneous networks: A distributed satisfactory framework," *IEEE Access*, vol. 5, pp. 26267–26278, 2017.

[25] R. L. G. Cavalcante, S. Stańczak, J. Zhang, and H. Zhuang, "Low complexity iterative algorithms for power estimation in ultra-dense load coupled networks," *IEEE Trans. Signal Process.*, vol. 64, no. 22, pp. 6058–6070, Nov. 2016.

[26] R. D. Yates, "A framework for uplink power control in cellular radio systems," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 7, pp. 1341–1347, Sep. 1995.

[27] R. Jain, A. Durresi, and G. Babic, "Throughput fairness index: An explanation," Dept. CIS Columbus, Ohio State Univ., Columbus, OH, USA, document ATM_Forum/99-0045 1999.

[28] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe, "A quantitative measure of fairness and discrimination," Eastern Res. Lab., Digit. Equip. Corp., Hudson, MA, USA, 1984.

[29] H. Shi, R. V. Prasad, E. Onur, and I. G. M. M. Niemegeers, "Fairness in wireless networks: Issues, measures and challenges," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 5–24, 1st Quart., 2014.

[30] A. B. Sediq, R. H. Gohary, R. Schoenen, and H. Yanikomeroglu, "Optimal tradeoff between sum-rate efficiency and jain's fairness index in resource allocation," *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 3496–3509, Jul. 2013.

[31] A. H. Arani, A. Mehbodniya, M. J. Omidi, F. Adachi, W. Saad, and I. Güvenç, "Distributed learning for energy-efficient resource management in self-organizing heterogeneous networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 10, pp. 9287–9303, Oct. 2017.

[32] L. Tsipi, M. Karavolos, and D. Vouyioukas, "An unsupervised machine learning approach for UAV-aided offloading of 5G cellular networks," *Telecom*, vol. 3, no. 1, pp. 86–102, 2022.

[33] S. Fu et al., "Energy-efficient UAV-enabled data collection via wireless charging: A reinforcement learning approach," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 10209–10219, Jun. 2021.

[34] H. Huang, Y. Yang, H. Wang, Z. Ding, H. Sari, and F. Adachi, "Deep reinforcement learning for UAV navigation through massive MIMO technique," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 1117–1121, Jan. 2020.

[35] D. Deng, C. Wang, and W. Wang, "Joint air-to-ground scheduling in UAV-aided vehicular communication: A DRL approach with partial observations," *IEEE Commun. Lett.*, vol. 26, no. 7, pp. 1628–1632, Jul. 2022.

[36] V. Saxena, J. Jalden, and H. Klessig, "Optimal UAV base station trajectories using flow-level models for reinforcement learning," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 4, pp. 1101–1112, Dec. 2019.

[37] S. A. Al-Ahmed, M. Z. Shakir, and S. A. R. Zaidi, "Optimal 3D UAV base station placement by considering autonomous coverage hole detection, wireless backhaul and user demand," *J. Commun. Netw.*, vol. 22, no. 6, pp. 467–475, Dec. 2020.

[38] Y. Liu, J. Yan, and X. Zhao, "Deep-reinforcement-learning-based optimal transmission policies for opportunistic UAV-aided wireless sensor network," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 13823–13836, Aug. 2022.

[39] R. Liu and J. Zou, "The effects of memory replay in reinforcement learning," in *Proc. 56th Annu. Allerton Conf. Commun. Control Comput. (Allerton)*, 2018, pp. 478–485.

[40] Q. Sun, W.-X. Zhou, and J. Fan, "Adaptive Huber regression," *J. Amer. Statist. Assoc.*, vol. 115, no. 529, pp. 254–265, 2020.

[41] T. Zhang, J. Lei, Y. Liu, C. Feng, and A. Nallanathan, "Trajectory optimization for UAV emergency communication with limited user equipment energy: A safe-DQN approach," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 3, pp. 1236–1247, Sep. 2021.

[42] A. Graves, "Generating sequences with recurrent neural networks," 2013, *arXiv:1308.0850*.

[43] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[44] I. A. Nemer, T. R. Sheltami, S. Belhaiza, and A. S. Mahmoud, "Energy-efficient UAV movement control for fair communication coverage: A deep reinforcement learning approach," *Sensors*, vol. 22, no. 5, p. 1919, 2022.

[45] R. Polvara et al., "Toward end-to-end control for UAV autonomous landing via deep reinforcement learning," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, 2018, pp. 115–123.

[46] C. Han, A. Liu, X. Liang, L. Ruan, and K. Cheng, "UAV trajectory control against hostile jamming in satellite-UAV coordination networks," in *Proc. IEEE 6th Int. Conf. Comput. Commun. (ICCC)*, 2020, pp. 701–705.

[47] A. Khalili, E. M. Monfared, S. Zargari, M. R. Javan, N. M. Yamchi, and E. A. Jorswieck, "Resource management for transmit power minimization in UAV-assisted RIS HetNets supported by dual connectivity," *IEEE Trans. Wireless Commun.*, vol. 21, no. 3, pp. 1806–1822, Mar. 2022.

[48] X. Liu, Y. Liu, Y. Chen, and L. Hanzo, "Trajectory design and power control for multi-UAV assisted wireless networks: A machine learning approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7957–7969, Aug. 2019.

[49] X. Liu, Y. Liu, and Y. Chen, "Reinforcement learning in multiple-UAV networks: Deployment and movement design," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8036–8049, Aug. 2019.

[50] X. Liu, Y. Liu, and Y. Chen, "Deployment and movement for multiple aerial base stations by reinforcement learning," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2018, pp. 1–6.

[51] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[52] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[53] F. Lagum, I. Bor-Yaliniz, and H. Yanikomeroglu, "Strategic Densification with UAV-BSs in cellular networks," *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 384–387, Jun. 2018.

[54] S. Koenig and R. Simmons, "Complexity analysis of real-time reinforcement learning," in *Proc. 11th Nat. Conf. Artif. Intell. (AAAI)*, Jul. 1993, pp. 99–105.

[55] "RESOLUTION 247 (WRC-19): Facilitating mobile connectivity in certain frequency bands below 2.7 GHz using high-altitude platform stations as international mobile telecommunications base stations," Int. Telecommun. Union, Geneva, Switzerland, 2019.

ATEFEH HAJIJAMALI ARANI received the Ph.D. degree in electrical engineering communication systems from the Isfahan University of Technology, Iran, in 2018. She is currently a Postdoctoral Fellow with the University of Waterloo, Canada. Her research interests include machine learning, resource management, heterogeneous, and aerial networks.

PENG HU (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from Queen's University, Canada. He is a Research Officer with National Research Council Canada and an Adjunct Professor with the Cheriton School of Computer Science, University of Waterloo. His current research interests include satellite-terrestrial integrated networks, non-terrestrial networks, autonomous networking, and the industrial Internet of Things systems. He currently serves as an Associate Editor of the IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY and the Co-Chair of the Technology Working Group of the IEEE LEO Satellites and Systems Project. He has served as an Associate Editor of the *Canadian Journal of Electrical and Computer Engineering*, a member of the IEEE Sensors Standards Committee, and on the Organizing and Technical Program Committees of Industry Consortia and International Conferences/Workshops at IEEE ICC'23, IEEE GLOBECOM'23, IEEE PIMRC'17, and IEEE AINA'15.

YEYING ZHU received the Ph.D. degree in statistics from Pennsylvania State University, USA. She is currently an Associate Professor with the Department of Statistics and Actuarial Science with the University of Waterloo. Her research interest lies in the interface between causal inference and machine learning methods.