CHALMERS
UNIVERSITY OF TECHNOLOGY

5G Network on Wings: A Deep Reinforcement Learning Approach to the UAV-Based Integrated Access and Backhaul

IEEE ComSoc | IEEE COMPUTER SOCIETY | IEEE Signal Processing Society | VTS | IEEE Transactions on Machine Learning in Communications and Networking

Received 26 May 2023; revised 26 December 2023 and 11 June 2024; accepted 8 August 2024.
Date of publication 13 August 2024; date of current version 20 August 2024.

The associate editor coordinating the review of this article and approving it for publication was J. J. P. C. Rodrigues.

Digital Object Identifier 10.1109/TMLCN.2024.3442771

5G Network on Wings: A Deep Reinforcement Learning Approach to the UAV-Based Integrated Access and Backhaul

HONGYI ZHANG[1], ZHIQIANG QI[2], JINGYA LI[2], ANDERS ARONSSON[2], JAN BOSCH[1], AND HELENA HOLMSTRÖM OLSSON[3]

[1]Department of Computer Science and Engineering, Chalmers University of Technology, 412 96 Gothenburg, Sweden
[2]Ericsson Research, Ericsson, 164 80 Stockholm, Sweden
[3]Department of Computer Science and Media Technology, Malmö University, 205 06 Malmö, Sweden

CORRESPONDING AUTHOR: H. ZHANG (hongyiz@chalmers.se)

This work was supported by the Chalmers AI Research Center and Software Center.

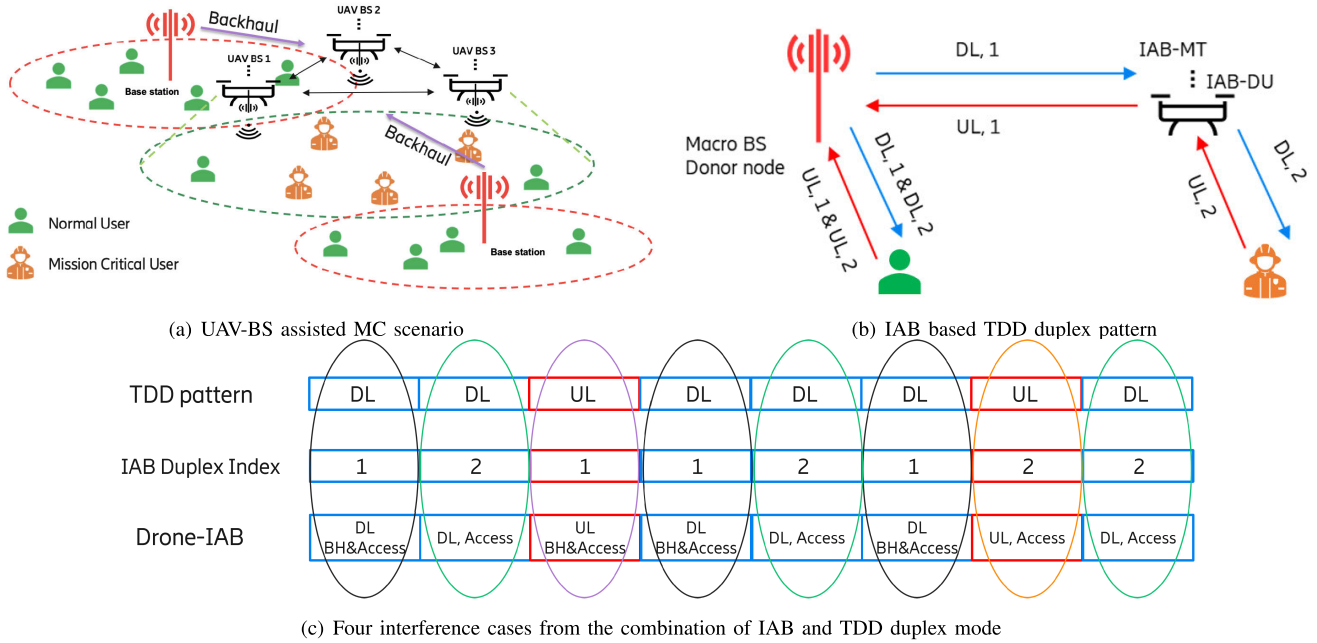
ABSTRACT Fast and reliable wireless communication has become a critical demand in human life. In the case of mission-critical (MC) scenarios, for instance, when natural disasters strike, providing ubiquitous connectivity becomes challenging by using traditional wireless networks. In this context, unmanned aerial vehicle (UAV) based aerial networks offer a promising alternative for fast, flexible, and reliable wireless communications. Due to unique characteristics such as mobility, flexible deployment, and rapid reconfiguration, drones can readily change location dynamically to provide on-demand communications to users on the ground in emergency scenarios. As a result, the usage of UAV base stations (UAV-BSs) has been considered an appropriate approach for providing rapid connection in MC scenarios. In this paper, we study how to control multiple UAV-BSs in both static and dynamic environments. We use a system-level simulator to model an MC scenario in which a macro-BS of a cellular network is out of service and multiple UAV-BSs are deployed using integrated access and backhaul (IAB) technology to provide coverage for users in the disaster area. With the data collected from the system-level simulation, a deep reinforcement learning algorithm is developed to jointly optimize the three-dimensional placement of these multiple UAV-BSs, which adapt their 3-D locations to the on-ground user movement. The evaluation results show that the proposed algorithm can support the autonomous navigation of the UAV-BSs to meet the MC service requirements in terms of user throughput and drop rate.

INDEX TERMS Reinforcement learning, multi-agent, integrated access and backhaul (IAB), 5G NR, wireless backhaul, UAV-BS.

I. INTRODUCTION

TRADITIONAL cellular infrastructure provides fast and reliable connectivity in most use cases. However, when a natural disaster happens, such traditional wireless base stations (BSs) can be damaged and therefore they cannot provide mission-critical (MC) services to the users in the disaster area. In this context, further enhancements of the cellular networks are needed to enable temporary connectivity and on-demand coverage for MC users in various challenging scenarios.

Vehicular networking can be enabled by various vehicle types including not only cars but also buses, trucks and UAVs. By equipping with a cellular tower and transceiver on a truck or trailer, cell-on-wheels have fewer cruising duration constraints and can transmit with a higher power to provide a relatively large coverage area [1]. However, cell-on-wheel placement may be less flexible for MC operations in rural areas with complex environments, such as forest firefighting, mountain search and rescue. UAV-BS (cell-on-wings) on the other hand, can be deployed in a more flexible and mobile manner. Specifically, UAVs can be used to carry deployable BSs to provide additional or on-demand coverage to users, thanks to their good mobility and higher chances of light-of-sight (LOS) propagation. However, there are a number of challenges when implementing UAV-BS assisted wireless communication networks in practice [2], [3]. The system performance and user experience are significantly



(a) UAV-BS assisted MC scenario

(b) IAB based TDD duplex pattern

(c) Four interference cases from the combination of IAB and TDD duplex mode

FIGURE 1. A UAV-BS assisted wireless network design enabled by a half-duplex IAB operation.

impacted by the deployment and configuration of UAV-BSs, including the UAV's 3-D position, operation time, antenna capabilities, transmit power, etc [4]. Using wireless backhaul, UAV-BSs can connect to the on-ground BSs (e.g., cell-on-wheels or macro BSs) and be integrated into the cellular system. Hence, it is necessary to jointly optimize the configuration parameters for the access links (between UAV-BS and on-ground users) and the backhaul links (between UAV-BSs and on-ground BSs), when optimizing UAV-BS based wireless communication systems. The optimization problem becomes even more complicated when considering different system loads and user movement on the ground. In some cases where multiple UAV-BSs are needed to cover a wide area, the complexity of providing reliable and scalable backhaul links will further increase.

Despite the fact that there are numerous applications for UAV-based reinforcement learning algorithms, the fundamental drawback of classic RL is its low performance in a changing environment. If the environment changes (the environmental values observed by the agent change), the agent usually has to retrain the entire algorithm to keep up with the environmental changes [5], [6]. In our case, user mobility would have a major impact on the system performance in terms of MC user throughput and drop rate. As a result, to ensure good service quality, a triggering mechanism needs to be implemented for algorithm adaptation and analysis. The dynamic environment, in this case, indicates that the states (user throughout and drop rate) that the agent observed will vary substantially due to wireless communication environment changes and user movement.

A. RELATED WORK

In recent years, UAV-BS assisted wireless communication networks have attracted significant attention from both industry and academia [7], [8], [9], [10], [11]. To guarantee a robust wireless connection between the UAV-BSs and the core network, more and more research work has started working on improving the wireless backhaul link [12], [13], [14], [15], [16]. Authors in [12] assume that all the UAV-BSs are flying at a fixed height, and a robust backbone network among UAV-BSs is guaranteed by ensuring that there is always at least one path between any UAV-BS and a BS on the ground. Then they investigate the rapid UAV deployment problem by minimizing the number of UAVs to provide on-demand coverage for as many users as possible. In [13], optimal 3-D deployment of a UAV-BS is investigated to maximize the number of connected users with different service requirements by considering the limitation of wireless backhaul links. In [14], the limitation of backhaul and access capacities is also considered, and a heuristic algorithm is proposed to optimize the UAV navigation and bandwidth allocation. Similar to [13], the authors in [16] also investigate a coverage improvement problem enabled by UAV-BS with backhaul limitation but with a machine learning (ML) based solution.

Enabled by 5G new radio (NR), the integrated access and backhaul (IAB) feature can be applied to wirelessly integrate multiple UAV-BSs to an existing cellular network seamlessly [17]. Figure 1 shows an example of UAV-BS assisted network deployment using IAB technology. The macro-BSs who have connections with the core network are serving the normal users, and some of them can also be acting as

donor-BSs, who can provide wireless backhaul connections to the flying UAV-BS. Based on the wireless backhaul link, the UAV-BS is acting as an IAB node, which can be deployed at different locations to provide on-demand services to MC users and/or normal users who are out of the coverage of the existing mobile network. To evaluate the performance of the UAV-assisted wireless system enabled by IAB, authors in [15] propose a dedicated dynamic algorithm based on the particle swarm optimization (PSO) method to optimize the throughput and user fairness. By intertwining different spatial configurations of the UAVs with the spatial distribution of ground users, [18] proposes an interference management algorithm to jointly optimize the access and backhaul transmissions. Their results prove that both coverage and capacity can be improved.

Due to the characteristics of revealing implicit features in large amounts of data, the ML methodology draws growing attention and has been extensively applied in various fields. As a sub-field of ML, agent-based reinforcement learning (RL) features in interacting with the external environment and providing an optimized action strategy. Hence, it has been used to solve complicated optimization problems that are difficult to be addressed by traditional methods. As two of the promising technologies for the next-generation wireless communication networks, it is natural to combine ML with deployable UAV-BS to solve high complexity optimization problems [19], [20].

Specifically, ML is frequently used to solve problems on deployment [16], [21], [22], scheduling [23], [24], [25], [26], trajectory [27], [28], [29], [30], [31], [32], [33], [34], [35] and navigation [36], [37], [38], [39] in UAV assisted network. In [21], a deep RL-based method is proposed for UAV control to improve coverage, fairness, and energy efficiency in a multi-UAV scenario. To solve the scheduling problem in a high mobility environment, the authors in [23] develop a dynamic time-division duplex (TDD) configuration method to perform intelligent scheduling. Based on the experience replay mechanism of deep Q-learning, the proposed algorithm can adaptively adjust the TDD configuration and improve the throughput and packet loss rate. From the perspective of distributed learning, [24] proposes a framework based on asynchronous federated learning in a multi-UAV network, which enables local training without transmitting a significant amount of data to a central server. In this framework, an asynchronous algorithm is introduced to jointly optimize UAV deployment and scheduling with enhanced learning efficiency.

For ML-based trajectory and navigation, the authors in [27] investigate a trajectory strategy for a UAV-BS by formulating the uplink rate optimization problem as a Markov decision process without user-side information. The authors in [31] introduces a UAV-based downlink communication model that addresses UAVs' limited energy resources by using simultaneous wireless information and power transfer technology. By optimizing UAV trajectory, power splitting ratio, and communication scheduling through a deep reinforcement learning framework, the approach significantly enhances energy efficiency and communication quality, outperforming conventional methods. Paper [32] proposes a novel adaptable integrated sensing and communication mechanism in UAV-enabled systems, optimizing communication and sensing beamforming along with UAV trajectory to maximize system throughput while ensuring quality-of-service. Authors in [33] proposes a UAV trajectory optimization scheme based on reinforcement learning to maximize energy efficiency and network resource utilization through load balancing. Based on deep reinforcement learning, authors in [34] and [35] both propose solutions to jointly optimize the UAV trajectory and resource scheduling in UAV-assisted network. To enable UAV autonomous navigation in large-scale complex environments, an online deep RL-based method is proposed in [36] by mapping UAV's measurement into control signals. Furthermore, to guarantee that the UAV always navigates towards the optimal direction, authors in [37] enhance the deep RL algorithm by introducing a sparse reward scheme and the proposed method outperforms some existing algorithms.

Additionally, the limited battery life of a UAV restricts its flying time, which in turn affects the service availability that can be provided by the UAV. Therefore, many works have been focusing on designing energy-efficient UAV deployment or configuration schemes either with non-ML [7], [40], [41] or ML methodologies [30], [42], [43].

B. CONTRIBUTIONS

In this paper, we consider a scenario with multiple macro-BSs covering a large area, but due to disaster, one of the macro-BSs is damaged, which creates a coverage hole where the first responders execute their MC operations. The deployable UAV-BSs are set up to fill the coverage hole and provide temporary connectivity for these MC users. Compared with the related works and our previous paper [44] navigating only one single UAV-BS, we propose in this paper a novel RL algorithm combined with adaptive exploration and value-based action selection algorithms to autonomously and efficiently deploy multiple UAV-BSs based on the requirements. Furthermore, to extend the algorithm in a scalable manner, a decentralized architecture is proposed for the collaboration of multiple UAV-BSs. More specifically, the contributions of this paper include the following aspects:

1) We propose the framework to support applying RL algorithm for the considered use case in an IAB network architecture.
2) We applied two strategies, i.e., adaptive exploration control and value-based action selection for the RL algorithm so that the algorithm itself can adapt to a dynamic environment (e.g., MC user movement and change of channel characteristics) in a fast and efficient way.
3) We demonstrated deployment in a decentralized method for supporting multiple UAV-BSs deployment to respond to varied industrial scenarios.
4) We validate the proposed RL algorithm in a continuously changing environment with consecutive MC user

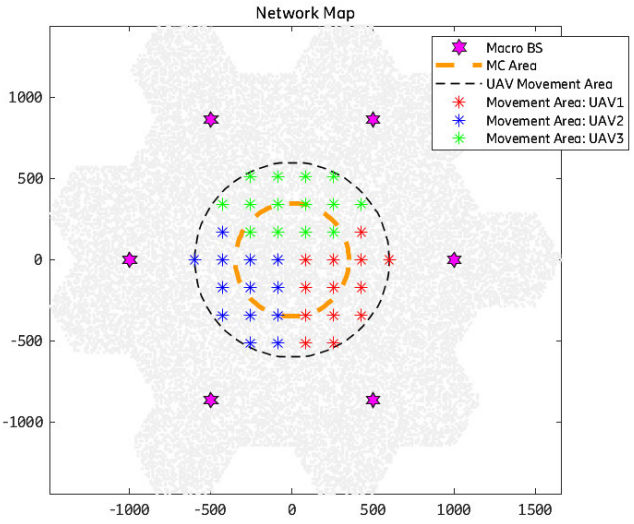


FIGURE 2. System model: UAV-BSs assisted network deployment.

movement phases. Our results show that the proposed algorithm can create a generalized model and assist in updating the decision-making on UAV-BSs and navigation in a dynamic environment.

The remainder of this paper is structured as follows. Section II introduces the system model considered in this paper. In section III, we propose a framework to enable ML in an IAB network architecture. Section IV discusses our proposed ML algorithm. Section V presents the system-level simulation results and evaluates the proposed RL algorithm. In Section VI, we summarize our findings and discuss future works.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. SYSTEM DESCRIPTION

For the system model, we consider a multi-cell mobile cellular network, consisting of a public network and a deployable network, as shown in Figure 2. Initially, seven macro-BSs are serving users uniformly distributed in the whole area. However, one of the macro-BSs in the center of the scenario is damaged due to, e.g., a natural disaster that creates a coverage hole. For users in the central emergency area with a predefined radius (marked as an orange dashed circle), they might have very limited or no connectivity with the public network. Hence, multiple UAV-BSs, which are integrated into the public network using IAB technology, can be set up to provide temporary or additional coverage to the users in this emergency area, which is also the research target of this paper. In this paper, the UAV-BSs are limited only to stay at the discrete points indicated by the colored stars in Figure 2. The total number of discrete points is selected based on the criteria that the simulation data is large enough to train the proposed model but not too much to spend an excessive amount of simulation time. Hence, the navigation in the scope of this paper refers to trajectory optimization among these discrete location points. To avoid the UAV-BSs staying too close to cause strong interference to each other, we split the whole UAV-moving area (inside the black dashed circle) into three non-overlapping areas denoted by colored discrete points. For example, UAV-BS 1 is only allowed to move between location points marked as red. In the considered scenario, there are two types of users: The users located in the MC area are marked as MC users, while the others are normal users. User equipment (UE), either an MC user or a normal user, can select either a macro-BS or a UAV-BS as its serving-BS, based on the wireless link qualities between the UE and these BSs.

For the traffic pattern design, we apply a FTP-based dynamic traffic model, which is commonly used in the Third Generation Partnership Project (3GPP) [45]. All the users are randomly dropped in the scenario. For each time slot, the users are activated with a predefined arrival rate. Only these activated users can be scheduled and initiate fixed-size data transmission based on the link quality (both access and backhaul links) and system load for downlink and uplink, respectively. When the data transmission is completed, the user will leave the system and wait to be activated again. Then the user throughput can be calculated with actually served traffic and consumed time to deliver the traffic.

As mentioned before, the UAV-BSs work as the IAB nodes in the current scenario. They will measure the wireless link to all macro-BSs and select one with the best link quality as their donor-BSs. Once the wireless backhaul link between the UAV-BSs and their donor-BSs is established, the three sectors of the UAV-BSs will share this wireless backhaul link and provide access service to both normal users and MC users. For the users served by the UAV-BSs, the corresponding throughput depends not only on the access link but also on the wireless backhaul link. While selecting the access links, the users with too bad link quality, for instance, below a certain threshold, will be dropped. To reduce the complexity and the load-bearing of the UAV-BSs, it is assumed that the same antenna configuration is applied for both access and backhaul antennas of the UAV-BSs. The reason to make such an assumption also includes that the positions of UAV-BSs have more impact on the key performance metrics (e.g., user throughput and drop rate) than varying the tilt of access and backhaul antennas. The proposed model is also applicable in the real world. The number of discrete points for UAV-BSs can be selected according to the computing capability of the target system in certain scenarios. If the UAV-BS supports two separate antenna panels for access and backhaul links, the antenna configuration (e.g., antenna tilt) for access and backhaul links can be adjusted respectively to further improve the performance when the same configuration is applied for both access and backhaul antennas. The proposed model can handle such cases by adding antenna tilt as a new input feature.

The system operates under a TDD model, and the time slot pattern consists of downlink (DL), DL, uplink (UL),

and DL, which is repeated with a periodicity of 2 ms [46]. The system bandwidth is 100 MHz, and it is shared between backhaul and access links. The time slots assigned for UL/DL Access/Backhaul links are shown in Figure 1(b) and (c). Two full TDD periods are required to cover all eight UL/DL Access/Backhaul combinations, which lead to four interference cases, denoted as: DL1, DL2, UL1, and UL2. As shown in Figure 1(b), for the UAV-IAB node, DL1 and UL1 are reserved for backhaul link transmission, while DL2 and UL2 are reserved for providing access services for users. For the donor-BS and all other macro-BSs, all the time slots can be used for access link transmission. In each interference case, the interfering nodes for users are different from those in other interference cases. For example, in the DL1 case where the time slots are used for both backhaul and access links, the UAVs are acting as users and the interfering nodes in this case only include marco-BSs. While in the DL2 case where the time slots are only used for access links, the UAVs are acting as BSs to serve users. In this case, the interfering nodes in DL include both macro-BSs and UAV-BSs. These features are all captured in the proposed ML model by importing the performance metrics(e.g., user throughput and drop rate) in the reward function, which will be introduced in detail in Section III.

To validate the performance of the proposed algorithm in adapting to a dynamic environment, we established five distinct phases in the time domain. Data traces required for training were captured at the beginning of each phase while the user movement (i.e. changes in user locations) was considered between phases. Upon entering a new phase, each user moves a random distance along both the horizontal and vertical directions in two-dimensional space, with the moving distances uniformly selected from 0 to 10 meters. In addition to the changes in user locations, the configuration related to channel conditions including fading and multi-path components are updated when switching phases. Consequently, the state of each phase maps to a different set of feature values across these five phases. Due to these environmental changes, the optimal state may vary in each phase, requiring the UAV-BS to adjust its decision-making model to adapt to the dynamic environments.

B. TRANSMISSION MODEL

For the public network in this paper, we use an urban-macro propagation model [47], while a refined aerial model from 3GPP standardization is used for UAV-BS [45]. It is assumed that the network consists of D macro-BS, M UAV-BSs and N users, denoted by $\mathcal{D} = \{1, 2, \ldots, D\}$, $\mathcal{M} = \{1, 2, \ldots, M\}$ and $\mathcal{N} = \{1, 2, \ldots, N\}$. The whole available bandwidth W is divided into K sub-channels and each one has a bandwidth denoted by $B_k = \frac{W}{K}$. Assuming that the three dimensional coordinates of the m^{th} UAV-BS and the n^{th} user are (x_m, y_m, h_m) and (x_n, y_n, h_n), respectively. Based on 3GPP channel model [47], the following formula is applied to represent the probability of LOS propagation between UAV-BS m and user n:

$$Pr_{LOS}^{(m,n)} = \begin{cases} 1, & d_{2D}^{(m,n)} \le d_{2D}^{Th} \\ [\frac{18}{d_{2D}^{(m,n)}} + \exp(\frac{-d_{2D}^{(m,n)}}{63}) \\ \quad \times (1 - \frac{18}{d_{2D}^{(m,n)}})] \\ \quad \times (1 + \frac{5}{4} e^{-6} \times C'(h_n) d_{2D}^{(m,n)^3} \\ \quad \exp \frac{-d_{2D}^{(m,n)}}{150}) & d_{2D}^{(m,n)} > d_{2D}^{Th} \end{cases} \tag{1}$$

where,

$$C'(h_n) = \begin{cases} 0, & h_n \le 13 \\ \left(\frac{h_n - 13}{10}\right)^{1.5}, & 13 < h_n \le 23 \end{cases} \tag{2}$$

$h_n \in \left[h_n^{min}, h_n^{max}\right]$ denotes the height of user n with meter as a unit, while h_n^{min} and h_n^{max} denote the minimal and maximal height of a user, respectively. $d_{2D}^{(m,n)} = \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2}$ is the horizontal distance between UAV-BS m and user n. d_{2D}^{Th} is a 2D distance threshold and its value is 18 meters. The path loss between UAV-BS m and user n in the case of LOS propagation and NLOS propagation can also be derived based on [47]:

$$PL_{LOS}^{(m,n)} = 28 + 22 log_{10}\left(d_{3D}^{(m,n)}\right) + 20 log_{10}(f_c) \tag{3}$$

$$PL_{NLOS}^{(m,n)} = 13.54 + 39.08 log_{10}\left(d_{3D}^{(m,n)}\right) + 20 log_{10}(f_c) - 0.6\,(h_n - 1.5) \tag{4}$$

where $d_{3D}^{(m,n)} = \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2 + (h_m - h_n)^2}$ denotes the distance between the antennas of UAV-BS m and user n, while f_c is the carrier frequency. Hence the average path loss between UAV-BS m and user n can be denoted as:

$$PL_{MN}^{(m,n)} = Pr_{LOS}^{(m,n)} \times PL_{LOS}^{(m,n)} + \left(1 - Pr_{LOS}^{(m,n)}\right) \times PL_{NLOS}^{(m,n)} \tag{5}$$

Similarly, $PL_{DN}^{(d,n)}$ denotes the average path loss between macro-BS d and user n, while $PL_{DM}^{(d,m)}$ denotes the average path loss between macro-BS d and UAV-BS m. To indicate whether a sub-channel is occupied by a UAV-BS/macro-BS to serve the users, an occupy indicator is defined, and setting c_i^k as 1 implies that the sub-channel k is occupied by UAV-BS/macro-BS i. Meanwhile, another indicator is defined where $\alpha_m^n = 1$ indicating that user n is served by UAV-BS m. Hence, the SINR between UAV-BS m and user n on sub-channel k can be denoted as:

$$\Upsilon_{(n,m)}^k = \frac{c_m^k \alpha_m^n \left(P_m - PL_{MN}^{(m,n)}\right)}{N_0 B_k + \sum_{i \ne m} c_i^k \left(P_i - PL_I^{(i,n)}\right)} \tag{6}$$

where P_m and P_i represent the transmit power of UAV-BS m and interfering node i, respectively. N_0 is the power spectral

density of the additive Gaussian noise. When the UAV-BSs are serving users, the interference may not only come from the other UAV-BSs serving users but also from the macro-BSs serving other UAV-BSs/users. Therefore, $PL_I^{(i,j)}$ generally denotes the path loss between user/UAV-BS j and interfering node (UAV-BS/macro-BS) i. Based on the above-mentioned expressions, the achieved throughput for MC user can be obtained by:

$$C_n = \sum_{k}^{K} \lambda_n B_k log_2 \left(1 + \Upsilon_{(n,m)}^{k}\right) \tag{7}$$

where λ_n is the user drop indicator. The user n with an SINR lower than Υ_{min} will be dropped and its corresponding user drop indicator λ_n equals zero. For the drop rate of MC users which will be used in the following sections, it is defined as:

$$\beta_{MC} = \frac{\sum_{n}^{N_{MC}} \lambda_n}{N_{MC}}, \lambda_n \tag{8}$$

where N_{MC} is the number of MC users.

The intention of this paper is to generate a generic model, which can be used to navigate the UAV-BSs to serve MC users in typical MC scenarios. That is why the statistic model is applied in the transmission model mentioned above to calculate the user SINR. In comparison, the real map scenario models the physical objects in a specific environment, which can capture the blockage effect in the network [48]. If there is a need to apply this generic model in a specific scenario, the generic model can be further refined to accommodate such scenario, which is also the next step in our future research.

C. PROBLEM FORMULATION

In a multi-network scenario consisting of both existing BSs on the ground and temporarily deployed UAV-BS, the deployment of the UAV-BS play a critical role in guaranteeing the performance of the target users/services (e.g., MC users/services). It can also impact the overall system performance. As the UAV-BS is connected to the core network using wireless backhaul, it is important to ensure the good quality of both the backhaul and access links when performing this system optimization. Furthermore, the optimal solution depends on many factors like network traffic load distribution, quality of service (QoS) requirements and user movements on the ground. Therefore, jointly optimizing these parameters of UAV-BS is a complex system-level optimization problem that needs to be solved in a dynamic changing environment.

In order to best serve target users while also maintaining a good backhaul link quality between UAV-BSs and their donor-BSs, we aim to solve the following research problems: 1) Design an RL algorithm to jointly optimize the 3-D locations of the UAV-BSs. 2) Find the movement strategy of a UAV-BS to accommodate the dynamically changing user distribution.

Based on the system model introduced in the previous subsection, the target problem we intend to solve is optimizing the 3-D locations of the UAV-BSs to maximize a weighted sum of the following system key performance metrics for the MC users:

- Backhaul link rate for UAV-BS: On one hand, the backhaul link rate reflects the link quality when the UAV-BS is served as a user via its donor-BS. On the other hand, it also affects the end-to-end throughput performance of its associated users since the throughput of UAV-served users is calculated by considering the quality of both the access link and backhaul link.
- The 5-percentile and 50-percentile of the cumulative distribution function (CDF) of MC user throughput: The 5-percentile MC user throughput represents the performance of the cell-edge MC users, i.e., the MC users with the "worst" throughput performance, while the 50% throughput indicates the average MC user performance in the simulation area.
- Drop rate for MC users: The ratio of MC users that cannot be served with the required services. This is an important performance metric for MC scenarios, since for MC users, keeping reliable connectivity broadly is more important than guaranteeing high-demand services for specific users in most MC cases.

Although the performance metrics of normal users is also critical to evaluate the overall performance of the network even in an MC scenario, in this paper, we only focus on improving the performance of MC users because the performance of normal users is nearly not impacted by broken macro-BS and deployed UAV-BSs in the system model.

III. ML-BASED SOLUTION

In this section, we describe how we transform and model the considered use case in an ML environment. Three important components, including the state space, action space, and reward function, are constructed in order to design an RL algorithm to jointly optimize the 3-D position of multiple UAV-BSs in an IAB network.

A. MODELING OF ML ENVIRONMENT

1) STATE SPACE

In our case, a UAV-BS state at a given time instance t has three dimensions, namely a UAV-BS's 3-D position.

We use $\mathcal{P}_t = \{x_t, y_t, z_t\}$ to denote the 3-D position of a UAV-BS at time t. Then, a UAV-BS's state at a given time instance t is denoted as $s_t = \{x_t, y_t, z_t\}$. Table 1 shows the candidate values for each UAV-BS:

It should be noted in Table 1 that the available height range for all UAV-BSs is limited between 10 m and 20 m, rather than deploying the UAV-BSs into a higher altitude. The reason is that, in the scenario considered in this paper, the UAV-BSs tend to stay at a lower height to maintain good backhaul links to on-ground donor-BS and also provide better access links to serve on-ground MC users, which makes the current height range selection reasonable.

The candidate values of 2-D space location x and y axis cover the disaster area shown in Figure 2. The location

TABLE 1. Candidate values for each UAV-BS in the simulation environment.

3-D position $\mathcal{P}$ Space	UAV1 Candidate Values
x	$[85, 257, 428, 600]$ meters
y	$[-514, -342, -171, 0]$ meters
z	$[10, 20]$ meters
3-D position $\mathcal{P}$ Space	**UAV2 Candidate Values**
x	$[-600, -428, -257, -85]$ meters
y	$[-514, -342, -171, 0]$ meters
z	$[10, 20]$ meters
3-D position $\mathcal{P}$ Space	**UAV3 Candidate Values**
x	$[-428, -257, -85, 85, 257, 428]$ meters
y	$[171, 342, 514]$ meters
z	$[10, 20]$ meters

options are selected by three deployed UAV-BSs. The 2-D MC area has been divided into 3 parts, with each UAV-BS covering one part of the area. For UAV1, x and y axis options are $[85, 257, 428, 600]$ and $[-514, -342, -171, 0]$ meters. For UAV2, x and y axis options are $[-600, -428, -257, -85]$ and $[-514, -342, -171, 0]$ meters. For UAV3, x and y axis options are $[-428, -257, -85, 85, 257, 428]$ and $[171, 342, 514]$ meters. And the height options for all three UAV-BSs in the z axis are $[10, 20]$ meters. As a result, the total number of state combinations in this environment is 18928. The computation complexity will be linearly increased $O(n)$ based on the total number of input states combination.

2) ACTION SPACE

In order to enable a UAV-BS to control its state, for each state dimension, we defined three potential action options and the UAV-BSs chose an action from three candidate options. These three alternative action options are denoted by the three digits: $-1, 0, 1$, where "-1" indicates that a UAV-BS decreases the status value at this state dimension by one step from its current value; "0" indicates that a UAV-BS does not need to take any action at this state dimension and keeps its current value; "1" indicates that a UAV-BS increases the status value at this state dimension by one step from its current value.

For example, if the x-axis value of the UAV1 (i.e. the value of the x_t dimension) equals 257 meters, an action coded by "-1" for this dimension means that the UAV-BS will select an action to reduce the position value to 85 meters, an action coded by "0" implies that the UAV-BS will hold the current position (257 meters), and an action coded by "1" implies that the UAV-BS will increase the position value to 428 meters. The same policy is applied to all dimensions of the state space.

Since there are three action alternatives for each space state, the action pool for 3-D position space, the pool has 27 action candidates that may be programmed to an action list $\mathscr{P} = [(-1, -1, -1), (-1, -1, 0), (-1, -1, 1), (-1, 0, -1) \ldots, (1, 1, 1)]$. As a result, if we combine the action of

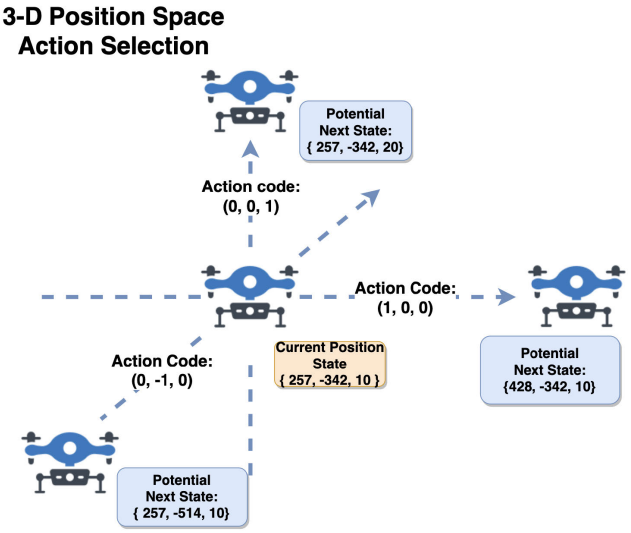


FIGURE 3. Example of the UAV1's state transition from current state {257, −342, 10} meters.

the 3-D position space, at any given moment t, a UAV-BS can thus choose an action a_t from these 27 alternatives. Figure 3 depicts a state transition from the specified state $s_t = \{257, -342, 10\}$ meters.

3) REWARD FUNCTION DESIGN

It is more critical to serve as many MC users as possible with appropriate service quality than to maximize the peak rate of a subset of MC users. In the MC context, ensuring a seamless and reliable communication service for all users is paramount. The user experience is intricately tied to two key performance metrics: drop rate and throughput, each addressing distinct aspects of communication quality.

1) Drop Rate (β): The drop rate metric is a critical indicator of connection reliability. It reflects the percentage of users who remain connected without disruptions in both uplink (UL) and downlink (DL) communication. In MC scenarios, where ubiquitous connectivity is essential, minimizing the drop rate is synonymous with ensuring that every user remains connected to the communication network. A low drop rate implies a higher level of reliability and availability of the communication service. This is particularly crucial in MC scenarios where universal access takes precedence over-optimizing the communication quality for a specific subset of users. By minimizing the drop rate, the model prioritizes the requirement of connecting every user in the disaster area.
2) Throughput (α): Throughput metrics, on the other hand, provide insights into the quality of the communication service. The 50th percentile and 5th percentile throughput values represent the average and "worst" performance of MC users, respectively, in both UL and DL. These metrics delve into the actual service quality experienced by users. In a mission-critical context,

optimizing communication quality is crucial to meet the diverse needs of users. Throughput metrics ensure that not only are users connected, but the quality of their communication experience is also considered. This is particularly relevant when users in the disaster area may have continuous communication demands, and the network must adapt to dynamically changing conditions.

Together, drop rate and throughput metrics provide a holistic view of the user experience in mission-critical scenarios. A low drop rate ensures universal connectivity, meeting the fundamental requirement of MC scenarios, while throughput metrics delve into the aspects of service quality. Balancing both aspects is essential for delivering a comprehensive and reliable user experience that aligns with the needs of MC users in disaster areas. It's worth noting that there may be trade-offs between connection reliability and service quality. Striking the right balance ensures that the communication network not only connects all users but also provides satisfactory service quality, acknowledging the dynamic nature of MC scenarios. Therefore, a reward function is constructed for measuring the overall user experience of current service settings.

As the reward function reflects the overall user experience of the MC users, the aggregated reward metrics are produced for the reward function design of the reinforcement learning algorithm to take into account both the impact of other drones' actions as well as the quality of services at the local drone. The reward is calculated using the average of the performance indicators of local and neighbouring agents. We have selected six key performance metrics for each local agent to highlight the local quality of service for MC users, including:

- The drop rates of MC users for UL and DL (β_{ul}, β_{dl}), which reflect the percentage of unserved MC users.
- The 50% throughput values of MC users for both UL and DL ($\alpha_{ul-50\%}$, $\alpha_{dl-50\%}$), which represent the average performance of the MC users, and
- The 5% throughput values of MC users for both UL and DL ($\alpha_{ul-5\%}$, $\alpha_{dl-5\%}$), which represent the ''worst'' performance of the MC users.

The choice of performance metrics in our study is linked to the unique challenges and priorities inherent in MC scenarios, where the primary objective is to establish and maintain reliable communication services for all users within the disaster area. Unlike conventional scenarios that may prioritize maximizing the peak rate for specific users, our focus is on universal service delivery and quality. This distinctive prioritization is a direct response to the critical nature of MC scenarios, where seamless communication can be a matter of life and death. The selected metrics serve as crucial indicators to address the specific needs of emergency situations and ensure the effective deployment of UAV-BSs.

The reward function is built as a weighted sum of these six feature values to balance these critical performance indicators, as shown below. The reason why the backhaul link rate is not considered here is that the values of the six features all rely on the quality of the backhaul link between the UAV-BS and its donor-BS. Before the model, all characteristics are normalized using min-max normalization, thus the values are constrained within the range [0, 1].

$$R_s = \omega_1 \times \frac{(1-\beta_{dl})+(1-\beta_{ul})}{2} + \omega_2 \times \frac{(\alpha_{ul-5\%} + \alpha_{dl-5\%})}{2} + \omega_3 \times \frac{(\alpha_{ul-50\%} + \alpha_{dl-50\%})}{2} \tag{9}$$

Furthermore, we set weighting coefficients $\omega_1 + \omega_2 + \omega_3 = 1$ to normalize the reward value such that R_s is between [0, 1]. To emphasize the significance of supporting all MC users, we assign higher weights to user drop rates and 5% MC-user throughput metrics. This is because, in the MC use cases, we must first prioritize that all users have access to the communication service rather than focusing on optimizing the communication quality of a small subset. In this paper, our method uses the weight values $\omega_1 = 0.5$, $\omega_2 = 0.3$ and $\omega_3 = 0.2$. The weighting of these metrics in the reward function emphasizes our commitment to supporting all MC users, as opposed to optimizing the communication quality for a select group. This intentional emphasis aligns with the core principle that in MC scenarios, every user's access to communication services is of paramount importance. This normalization ensures that the reward values are representative and comparable across diverse scenarios. The reward function's formulation involves a balance of weights assigned to key performance metrics to optimize the algorithm for mission-critical scenarios. Drop rate (ω_1) bears the highest weight, underscoring its critical role in minimizing service interruptions and prioritizing universal access. The weight for 5th percentile throughput (ω_2) is selected to address the trade-off between reducing drop rates and ensuring quality for the worst-performing users. Similarly, the weight for 50th percentile throughput (ω_3) is determined to strike a balance between providing quality for the majority and not compromising the performance of the most disadvantaged users. The optimal combination of these weights is identified through grid search, ensuring the reward function aligns with the unique priorities of mission-critical scenarios and achieves peak system performance. In order to know the influence of each UAV-BS, the reward function will also aggregate the reward values of the neighbour agents. Hence, the following is the reward function applied in the algorithm:

$$R_s = \frac{\sum_{c=1}^{C} \mathcal{M}^c + \mathcal{M}}{len(C)+1} \tag{10}$$

Assuming that C is the set of register neighbours, $\mathcal{M}$ represents the current agent's local system performance, and the $\mathcal{M}^c$ indicates the local system performance of its neighbour ID c.

B. RL ALGORITHM DESIGN

In this section, we design an RL algorithm to solve the optimization problem of the considered use case. RL is distinct

from supervised and unsupervised learning in the field of ML in that supervised learning is performed from a training set with annotations provided by an external supervisor (task-driven), whereas unsupervised learning is typically a process of discovering the implicit structure in unannotated data (data-driven). RL is suitable for this case since the method provides a unique feature: the trade-off between exploration and exploitation, in which an intelligence agent must benefit from prior experience while still subjecting itself to trial and error, allowing for a larger action selection space in the future (i.e., learning from mistakes).

In order to achieve better self-control decisions for our scenario, we applied deep Q-network (DQN) as our base RL algorithm. The algorithm was first proposed by Mnih et al. [49], [50] by combining convolutional neural networks with Q-learning algorithms [51] in traditional RL. The approach has been frequently used in gaming and static environments. However, the original approach is incapable of adapting to our MC situation due to environmental changes. To address these issues, we have proposed two significant schemes in our autonomous UAV-BSs control algorithm (Algorithm 1): adaptive exploration control and value-based action selection.

Algorithm 1 Deep Reinforcement Learning in Each UAV-BS With Adaptive Exploration and Value-Based Action Selection

Initialize the agent's replay memory Buffer $\mathcal{D}$ to capacity M
Initialize action-value function Q with two random sets of weights θ, θ'
Initialize exploration probability ε to 1
Set previous reward value r_p to 0
for *Iteration* $= 1, N$ **do**
  for $t = 1, T$ **do**
    $\mathscr{T}_t, \mathscr{P}_t \leftarrow$ Action_Selection(r_p, r_t, ε)
    $a_t = \{\mathscr{T}_t, \mathscr{P}_t\}$
    Set $r_p = r_t$
    Decode a_t to action options in four state dimensions and execute the actions
    Collect reward r_t and observe the agent's next state $\mathcal{P}_{t+1} \leftarrow \{x_{t+1}, y_{t+1}, z_{t+1}\}$
    Set $s_{t+1} = \{\mathcal{T}_{t+1}, \mathcal{P}_{t+1}\}$
    Store the state transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}$
    $\mathcal{A}_s, \mathcal{A}_o \leftarrow$ Action_Grouping(a_t)
    Sample mini-batch of transitions (s_j, a_j, r_j, s_{j+1}) from buffer $\mathcal{D}$
    if s_{j+1} is terminal **then**
      Set $y_j = r_j$
    else
      Set $y_j = r_j + \gamma \max_{a'} Q(s_{j+1}, a'; \theta')$
    end if
    Perform a gradient descent step using targets y_j with respect to the online parameters θ
    Set $\theta' \leftarrow \theta$
    $\varepsilon \leftarrow$ Adaptive_Exploration(r_p, r_t, ε)
  end for
end for

1) ADAPTIVE EXPLORATION (AE)

Because of the environmental changes, the original DQN model needs to be updated to accommodate feature value changes. As a result, we create a dynamic exploration probability triggered by a substantial decline in reward value. Following the completion of each learning iteration, the final reward value is checked and compared to the pre-defined reward drop and upper reward thresholds. Based on the outcome, the exploration probability ϵ will be adjusted.

Each UAV-BS initially explores the state space and then performs Q-value iterations at each training episode. When deciding whether to take an action that gives the maximum reward value or randomly explore a new state, a ϵ-greedy exploration is used. The parameter ϵ determines the likelihood of exploration. Each training step's data is saved in a replay batch $\mathcal{D}$. Each row of $\mathcal{D}$ holds the tuple (s_t, a_t, r_t, s_{t+1}), which represents the current state, action, reward, and next state for a training step. Samples will be chosen at random and used to update the Q value model.

The most recent reward value is reviewed and compared to a pre-defined reward-drop threshold and an upper reward threshold. Then, the exploration probability is updated by checking the following three conditions: (Algorithm 2):

- If the most recent reward value is less than the prior reward, and the difference is greater than the reward drop threshold, the exploration probability is increased to 0.1.
- If the most recent reward value exceeds the higher reward threshold, we can conclude that the algorithm has already located the optimal zone capable of delivering a reliable connection to MC users. The likelihood of exploration will be matched to the probability of completion.
- Otherwise, the exploration probability will multiply by an exploration decay and fall linearly after each learning cycle.

2) VALUE-BASED ACTION SELECTION (VAS)

Although the ϵ-greedy algorithm can strike a reasonable balance between exploration and exploitation, in some cases the approach utilized for exploration is redundant and time-consuming. The algorithm will choose actions at random throughout the searching stage, which may lengthen the search time. However, when dealing with a large action and state space, random action selection is clearly not an effective strategy and may cause decision-making to be delayed, which is unacceptable in most time-critical businesses. Therefore, we propose a novel value-based action selection strategy (Algorithm 3) which can lead to fast decision-making for a UAV-BS when determining its 3-D space location.

As described in the previous section, an agent's 3-D position state at a given time instance t is denoted as $\mathcal{P}_t = \{x_t, y_t, z_t\}$. Since each position state has three

Algorithm 2 Adaptive Exploration Algorithm (AE)

Set restarting exploration probability $\varepsilon_{Restart}$ to 0.1
Set ending exploration probability ε_{End} to 0.0001
Set exploration decay ϱ to 0.995
Function `Adaptive_Exploration`(r_p, r_t, ε)**:**
  if r_p - r_t > Drop threshold **then**
    Set $\varepsilon = \varepsilon_{Restart}$
  else if r_t > Upper reward threshold **then**
    Set $\varepsilon = \varepsilon_{End}$
  else
    Set $\varepsilon = \varepsilon \times \varrho$
  end if
  return ε

Algorithm 3 Value-Based Action Selection (VAS)

Set grouping threshold β to 0
Function `Action_Grouping`(a_t)**:**
  $\mathscr{P}_t \leftarrow a_t = \mathscr{T}_t, \mathscr{P}_t$
  for all potential next action $\mathscr{P}_{t+1}$ **do**
    if $\vec{\mathscr{P}}_t \cdot \vec{\mathscr{P}}_{t+1} > \beta$ **then**
      Append $\vec{\mathscr{P}}_{t+1}$ to $\mathcal{A}_s$
    else
      Append $\vec{\mathscr{P}}_{t+1}$ to $\mathcal{A}_o$
    end if
  end for
  return $\mathcal{A}_s, \mathcal{A}_o$

Function `Action_Selection`(r_p, r_t, ε)**:**
  if $r_t \geq r_p$ **then**
    Select a random action $\mathscr{P}_t$ with probability ε from the same consequence 3-D position action pool $\mathcal{A}_s$
  else
    Select a random action $\mathscr{P}_t$ with probability ε from the opposite consequence 3-D position action pool $\mathcal{A}_o$
  end if
  Otherwise, select $a_t = \arg\max_a Q(s_t, a; \theta)$
  return $\mathscr{T}_t, \mathscr{P}_t$

dimensions and each state dimension has three action options, the action pool contains in total 27 action candidates that can be programmed to a list of action space $[(-1,-1,-1),(-1,-1,0),(-1,-1,1),\ldots(1,1,1)]$. Each element in this list can then be regarded as an action vector.

Figure 4 depicts a probable set of next actions with the same or opposite consequence. The consequence is defined as the reward value (or monitored performance metrics) change after an action has been executed. The algorithm will analyze the outcome of past actions. If the prior action decision has a positive outcome (the reward value increases or monitored performance metrics become better) as defined above,

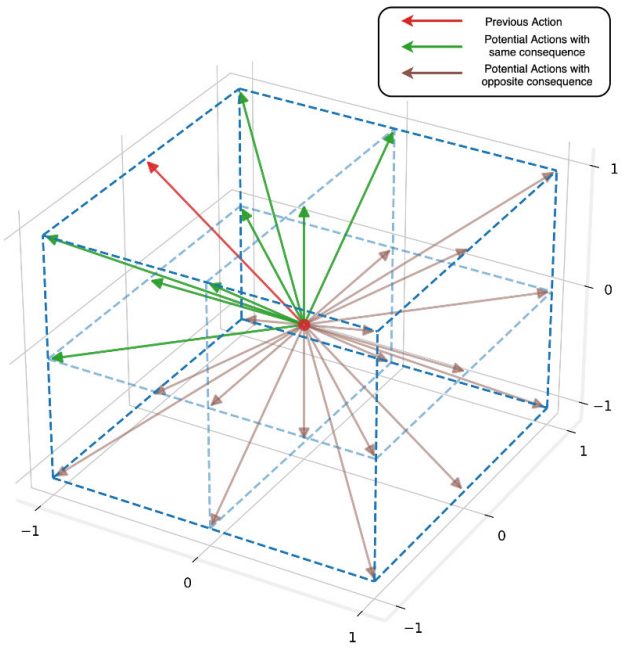


FIGURE 4. Diagram of a potential set of next actions with same or opposite consequence.

the algorithm will choose actions from a pool of following actions with the same consequence. The dot product between two action vectors determines the result. If the dot product is larger than 0, this action vector can be assumed to have the same outcome as the prior action option.

If the previous action decision results in a negative consequence (the reward value decreases or monitored performance metrics become worse), the algorithm will select actions from the pool consisting of potential next actions with the opposite consequence. The opposite consequence is determined by the dot product of two action vectors that is smaller than or equal to 0. The actions in this pool will result in an opposite consequence compared with the previous action decision. Assume that the previous action vector is $\vec{\mathscr{P}}_t$ while the next potential action vector is $\vec{\mathscr{P}}_{t+1}$:

$$\begin{cases} \vec{\mathscr{P}}_t \cdot \vec{\mathscr{P}}_{t+1} > 0 & \text{Same consequence as previous} \\ \vec{\mathscr{P}}_t \cdot \vec{\mathscr{P}}_{t+1} \leq 0 & \text{Opposite consequence as previous} \end{cases} \tag{11}$$

In Figure 4, the red vector represents the previous action decision. The angle between the previous action vector (red vector) and the green vectors is less than $\frac{\pi}{2}$, which can be represented by a dot product greater than zero. As a result, the green vectors represent actions that may result in the same consequence as the red vector. Similarly, the angle between the previous action vector (red vector) and the brown vectors is greater than or equal to $\frac{\pi}{2}$, which is represented by a dot product value less than 0. As a result, the brown vectors may have the opposite consequence.

During the UAV-BSs deployment, the algorithm monitors a set of critical system performance values (the reward value).

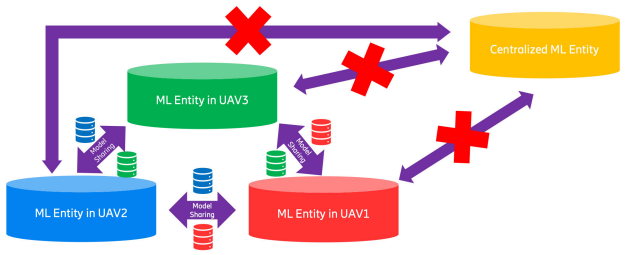


FIGURE 5. Decentralized architecture for multi-UAV coordination.

Based on the current and a set of previous performance values, the algorithm will evaluate the consequences caused by the previous action. The algorithm will thus select the action set which will potentially result in positive consequences.

C. DECENTRALIZED REINFORCEMENT LEARNING

In some circumstances, a single UAV is not capable of being extended to cover a larger area. As shown in Figure 1(a), multiple UAV-BSs are deployed to work together to service the MC users. An extensible decentralized method for deploying numerous UAV-BSs is therefore designed. The concept is illustrated in Figure 5 where we relocate the central server operation function from the central entity and attach it to the edge entity on UAV, as opposed to the typical single-agent reinforcement learning algorithms, to achieve decentralized characteristics.

Algorithm 4 Transmission Functions of the Decentralized Reinforcement Learning Algorithm (DecRL)

for *Iteration* $= 1, N$ **do**
  for $t = 1, T$ **do**
    After action selections:
    for each client $c \in C$ **in parallel do**
      send $\{s_{t+1}, \mathcal{M}\}$;
      receive $\{s^c_{t+1}, \mathcal{M}^c\}$
    end for$s_{t+1} = \{(x_{t+1}, y_{t+1})^c \; for \; c \; in \; C\}$
    Store the state transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}$
    Learning in each UAV-BS: $f_{AE\&VAS}(s_t, a_t, r_t, s_{t+1})$
  end for
  if $t \bmod f == 0$ **then**
    for each client $c \in C$ **in parallel do**
      send θ'_{t+1};
      receive ${\theta'_{t+1}}^c$;
    end for
    $\theta'_{t+1} \longleftarrow \sum_{c=1}^{C} \frac{1}{len(C)} {\theta'_{t+1}}^c$;
  end if
end for
End Function

The system has two different kinds of data for exchanging information, namely the system-related data (including location information and system KPIs) and the model data. The location will communicate with nearby drones regarding the connection performance and UAV-BS system-related data. These kinds of data can assist each drone in understanding how their movements affect the others and in being aware of one another's surroundings. Following each UAV-BS decision, the information will be continuously exchanged and used as a guide for the subsequent choice. The local model of each UAV-BS will be shared with its neighbours via the model data channel. Each UAV-BS has a separate procedure to train, communicate, and receive model weights and service metrics during the learning process. Each UAV-BS will share its learning experiences as a result, and the others can gain information from the experiences of the others. Information is exchanged asynchronously through active listening to neighboring UAV-BSs for receiving models and service metrics, rather than requesting them. This push-based communication mechanism helps avoid interruptions from malfunctioning or slow neighbors. After multiple training epochs, the UAV-BSs can swap their model with their neighbours under the control of a frequency parameter. The process is described in Algorithm 1. The procedures can be summarized as follows:

Step 1: Each training episode will begin with each UAV-BS exploring and locating its neighbours before moving on to exploring the environment and doing Q-value iterations. When deciding whether to choose the best action or to randomly explore the new state, a ϵ-greedy exploration is used. The parameter ϵ specifies the likelihood of exploration.

Step 2: After making a choice, each UAV-BS will notify its neighbours of the state and local performance indicators. The agent will simultaneously listen to the other neighbours, and get ready to receive their states s^c_{t+1} and local performance metrics $\mathcal{M}^c$. A global system metric value that can direct each UAV-BS to take future actions will be formed when all metrics have been received and the reward has been calculated based on the reward function R_s.

Step 3: A replay batch $\mathcal{D}$ contains the data for each training stage. The tuple (s_t, a_t, r_t, s_{t+1}), or the current state, action, reward, and next state for a training step, is contained in each row of $\mathcal{D}$. For the purpose of updating the Q value model, samples will be chosen at random. The current state reward pairs will also be distributed to the other agents after each decision round.

Step 4: A UAV-BS will send the updated model results, θ', to its registered neighbours for model aggregation after it has reached the predetermined exchanging iteration. Each UAV-BS will simultaneously listen to its neighbours in order to receive models and service metrics instead of requesting models from the others, which results in a push-based communication mechanism to avoid interruptions from malfunctioning neighbours.

Step 5: Each node executes aggregation by averaging all updated models depending on the aggregation function, $\theta'_{t+1} \longleftarrow \sum_{c=1}^{C} \frac{1}{len(C)} {\theta'_{t+1}}^c$;, after receiving all the models from the registered neighbours.

Step 6: The updated model is used by the edge device to replace the outdated one and to carry out additional local training. We'll repeat the steps from above.

D. COMPLEXITY AND ROBUSTNESS ANALYSIS

The Adaptive Exploration algorithm takes three parameters: r_p (prior reward), r_t (current reward), and ε (exploration probability). The function contains three conditional branches based on reward comparisons. Each branch contains constant time operations: setting ε to a constant value or updating it using multiplication. The time complexity of this function is constant, i.e., O(1).

The Action Selection algorithm takes three parameters: r_p (prior reward), r_t (current reward), and ε (exploration probability). It performs conditional branching based on reward comparisons and selects actions accordingly. The time complexity is O(1) since the operations inside each branch are constant.

Both AE and VAS algorithms have constant time complexities for their core functions. The overall complexity is dominated by the number of iterations in the main training loop, which is specified as N. Therefore, the overall time complexity of the algorithms is O(1) for each iteration, and O(N) for the entire training process.

The overall time complexity of the DecRL algorithm is influenced by the number of iterations N, the number of clients C, and the frequency parameter f. The transmission functions contribute a significant portion of the complexity, especially when considering parallel communication with each client. The model update and aggregation steps also have a complexity that depends on the number of clients and the frequency of model updates. The algorithm's complexity is not fixed and can vary based on the specific values chosen for parameters.

Furthermore, the Decentralized Reinforcement Learning (DecRL) algorithm's robustness against communication failures and potential UAV-BS malfunctions is underpinned by the push-based communication approach. This approach ensures dynamic adaptation as UAV-BSs actively listen to updates from the network, avoiding the situation that one needs to wait for the response from a malfunctioning edge, which can avoid accidental disconnections and maintain system engagement. In the face of communication disruptions, the push-based strategy facilitates adaptive reconfiguration, allowing UAV-BSs to adjust positions and communication parameters for continuous connectivity. Additionally, the monitoring function can be enabled by the push-based model aids in detecting UAV-BS malfunctions, prompting dynamic decision-making to redistribute tasks or optimize resource deployment.

IV. SIMULATION RESULTS AND ANALYSIS

In this section, the simulation configuration and scenario deployment are introduced firstly. We then investigate the impact of the 3-D location of multiple UAV-BSs on the performance of MC users in terms of backhaul link rate, throughput, and drop rate based on system-level simulations. Finally, we present the results of proposed RL algorithms for autonomous UAV-BS navigation.

A. SIMULATION CONFIGURATION

To evaluate the performance of the proposed RL algorithm in solving the formulated problem, we build a multi-cell scenario by considering the predefined system model, and a simulation is executed with a system-level simulator. With the output of the simulation, the proposed RL algorithm can be applied for UAV-BS to build a well-trained model, based on which optimal UAV-BS position and antenna configuration can be found rapidly.

In the simulation, we drop 500 users in the area as shown in Figure 2. The circle area with a 350m radius around the UAV-BS is defined as the MC area. The users located in the MC area are marked as MC users, while the others are normal users. All users follow an arrival model and only arrived users can be considered as activated.

To investigate how a well-trained RL model performs in a dynamic environment, we design a set of different user distributions to simulate the case of slow-moving users.

The detailed simulation parameters are shown in Table 2. Specifically, the typical inter-site-distance (ISD) is 500 meters for the mid-band urban-macro scenario in NR. To create a coverage hole and clearly show the impact of introducing UAV-BSs to cover bad-quality MC users, the ISD is set to 1000 meters and a macro-BS is removed from the center of the map due to malfunction. For the radius of the MC area, 350 meters is selected to create an area that is large enough for three UAV-BSs to jointly serve but not too large to introduce excessive candidate positions which significantly impacts the simulation efficiency. The user arriving rate per simulation area is used to control the system load in the network, and its value is selected by keeping the drop rate of the users in a reasonable range between 0% and 10%. The simulation time denotes the time duration between the network beginning and stopping to serve users with one set of configuration parameters, which comprises the 3D positions of three UAV-BSs. Hence, in this paper, one epoch means running a 2-second simulation with one specific combination of positions of three UAV-BSs. During this time, the users are activated based on a predefined arrival rate, as defined in the system model. The value of other parameters is selected based on the typical setting used in a mid-band urban-macro scenario.

B. SYSTEM-LEVEL PERFORMANCE EVALUATION

To evaluate the impact of UAV-BSs' positions on the user performance, Figure 6 shows the range of achieved average backhaul link rates when the three UAV-BSs are deployed at all possible combinations of different candidate positions. Around the MC area denoted by orange circle, 40 candidate 2D-positions of UAV-BS (colored stars shown in Figure 2) are selected and mapped to the centers of the colored circles in Figure 6. Each colored UAV-BS can only be deployed at the position marked with the same color. The radius of

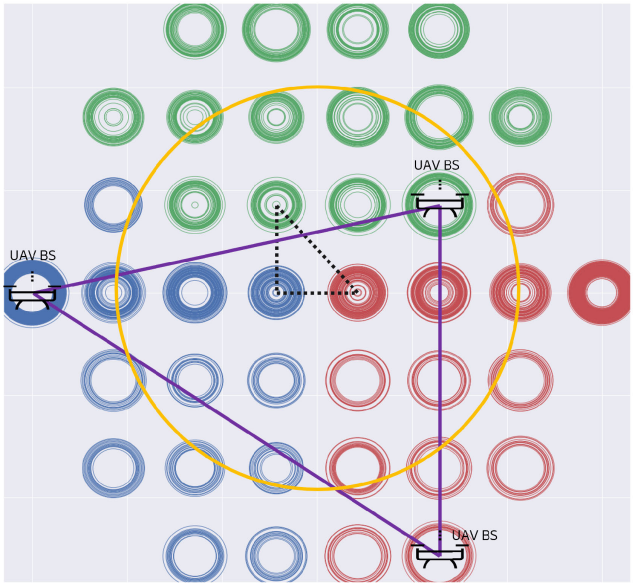


FIGURE 6. **Impact of UAV-BSs' positions on average backhaul link rate.**

TABLE 2. **Simulation parameters.**

Parameter	Value
Carrier Frequency	3.5 GHz
Bandwidth	100 MHz
Duplex Mode	TDD
TDD DL/UL Configuration	DDUD
Inter-Site-Distance (ISD)	1000 m
Radius of MC Area	350 m
Number of BSs	Macro-BS: 6; UAV-BS: 3
BS Transmit Power	Macro-BS: 46 dBm; UAV-BS: 40 dBm
Noise Figure	7 dB
BS Height	Macro-BS: 32 m; UAV-BS: 10-300 m
Number of Sectors per Site	3
Number of MC&Normal Users	500
User Arriving Rate per Simulation Area	270 users/s
User Speed	3 km/s
Minimum Distance between BS and Users	30 m
Simulation Time	2 s

each circle denotes the normalized average backhaul link rate of three UAV-BSs who are located at current positions. For one specific candidate position in blue where one UAV-BS is deployed, the other two UAV-BSs can be placed at all possible combinations of green and red candidate positions. This explains why there are multiple circles at each candidate position. Hence the smallest and largest circle radiuses at one candidate position denote the lower and higher bounds of the backhaul link rate when the UAV-BS is placed at the current location. If connecting the centers of three circles with the largest radiuses which represents the highest backhaul link rate, one triangle marked with a solid purple line can be observed. As shown in Figure 6, if three UAV-BSs are deployed at the vertexes of the purple triangle, the average backhaul link rate in the system is the highest. Based on the distribution pattern of three UAV-BSs, it seems that the optimal UAV-BSs' positions to maximize the backhaul link rate tend to be near the edge of the MC area. This is because the UAV-BSs can keep good backhaul link quality when located near the donor-BSs. Similarly, if three UAV-BSs are deployed at the vertexes of the black triangle, the lowest average backhaul link rate is reached.

Since the backhaul link rate decides the level of user throughput, the users can not be served well if the backhaul link quality is poor. Thus the UAV-BS can not only pursue a high backhaul link rate while ignoring the user throughput, and vice versa. By considering the weight for each metric in the reward function, the proposed RL algorithm can provide optimal locations of UAV-BSs to achieve the highest reward value. For example, if the weight of 5th percentile throughput or drop rate of MC users is high, the UAV-BSs navigated by the proposed RL algorithm tend to move to locations where the performance of low-quality users can be improved as much as possible.

Based on the above observation, the optimal UAV-BSs' positions are different if different performance metrics are considered with different weights in the reward function. Hence, optimizing the performance metrics in the reward function to achieve global optimization by adjusting UAV-BSs' positions is a complicated problem, for which ML-based solutions can be applied to find the implicit structure from the collected data.

C. MACHINE LEARNING PERFORMANCE EVALUATION

In this section, we present the experimental results of Decentralized Reinforcement Learning with Adaptive Exploration and Value-based Action Selection (DecRL-AE&VAS) for autonomously controlling multiple UAV-BSs. For the UAV-BSs deployment, different from the single UAV case introduced in the previous section, more candidate positions are allocated around the MC area for the multi-UAV case. As shown in Figure 2, the available positions for each UAV-BS don't overlap with others, which means each UAV-BS covers a certain geographical area with a total of 18928 combinations. The result is evaluated using two criteria: (1) six features (specified in Section IV - Modeling of ML Environment) that demonstrate link service quality, and (2) model learning quality in each phase as demonstrated by system reward value. The results are compared to several baseline models. As we described before, the experiment contains five validation phases incorporating MC user mobility. When entering a new phase, the algorithms will use the data collected during the new phase to learn and update themselves. During the simulation, the DecRL-AE&VAS method trains the model from scratch in the training phase and then

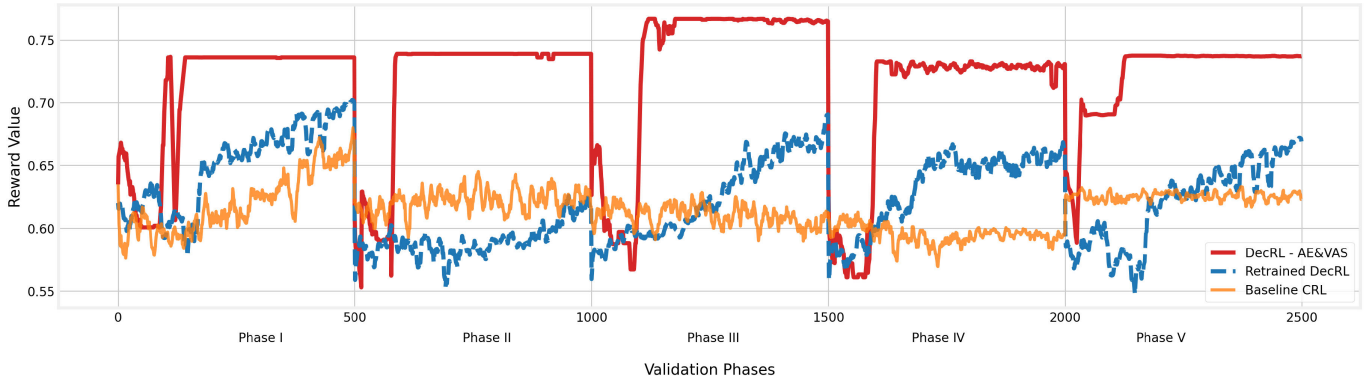


FIGURE 7. **Reward value with the number of learning iterations in five consecutive validation phases.**

continuously improves itself in the succeeding validation phases. The previously learned experience will not be removed from the subsequent sessions. During the model learning, for each agent, the test bed provides an 8-core Intel Xeon Processor with 8 GB memory. The average CPU consumption is lower than $\sim 1\%$ while the memory usage is about 300 MB. The communication overhead of one agent within one epoch is $\sim$ 410 KB while the learning time of each epoch takes $\sim$ 0.3 second, which is suitable for quick response to real-time user movement and can be easily implemented on resource-constraint devices.

There are two baseline methods used to compare the performance of DecRL-AE&VAS and demonstrate the effectiveness of the decentralized architecture and the convergent efficiency in the training phase, namely, baseline CRL and baseline IRL method. For the baseline IRL algorithm, these baseline models are explicitly trained using each edge UAV-BS. There will not be any model or information exchange between the edge and central nodes during training, in contrast to decentralized reinforcement learning. The service quality performance can be compared to the decentralized reinforcement Learning model to demonstrate how it can outperform those locally trained individual models. For the baseline CRL, this baseline model is trained using the centralized learning strategy. Prior to model training, all data from the edge are collected into a single server, and learn the action strategy step by step.

In the validation phase, in order to prove the performance in a dynamic environment, two baselines are utilized, namely, the retrained DecRL and baseline CRL. The retrained DecRL algorithm removes the past information and randomizes the ML model parameters but with the same decentralized setup. When entering a new phase, the algorithm will retrain the model from scratch. The difference between retrained DecRL and DecRL-AE&VAS is the utilization of AE and VAS strategies. These strategies have been proven to be useful when deploying UAV-BSs into dynamic environments. Last but not least, the baseline CRL algorithm in the validation phases will constantly learn and employ the new data when entering the new phase but with a centralized algorithm that controls all deployed UAV-BSs but without improved strategies.

For the learning hyper-parameters of our method, we explored various sets of combinations using random search in order to achieve the best results. Random search efficiently explores a wide range of hyper-parameter combinations, making it more likely to discover effective configurations compared to a more constrained grid search. During the training, the network architecture for a DQN is set to $[4 - 16 - 16 - 81]$. The architecture choice balances the complexity of the model with the capacity to represent the relationship between state and action. The exploration probability decay is set to 0.995. The exploration probability decay determines how fast the exploration probability ϵ decreases over time. A high decay rate is selected in our case to allow the agent to explore more in the early stages of training and gradually shift towards exploitation as training progresses. The learning rate is set to 5×10^{-5}. The learning rate determines the step size during the weight updates in the training process. The selected learning rate strikes a balance between convergence speed and stability. Lastly, the number of learning iterations at the training phase and validation phases equals 1500 and 500, respectively. The number of iterations determines how much exposure the algorithm has to the training data. The values are selected to balance the learning time and the model quality.

We first analyze the algorithm's convergence performance during the model training phase. Figure 8 depicts the reward value as a function of the number of learning iterations. Compared to the baseline CRL algorithm, we can see that the VAS method can help the UAV-BS quickly discover the near-optimal position and converge at a high-quality level.

Training the algorithm in a decentralized way can also be converged and reach a high reward value before 400 training rounds. The convergent reward value of the DecRL-AE&VAS is approximately 5% higher than that of the centralized training method in the initial learning iterations which results in higher training efficiency. When compared to the independent learning method, the algorithm does not converge after

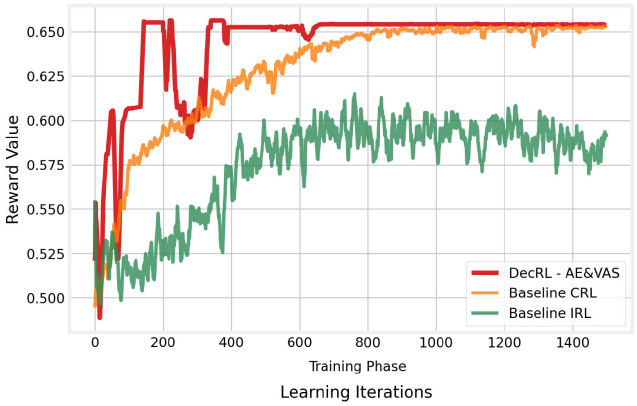


FIGURE 8. **Reward value with the number of learning iterations in the training phase.**

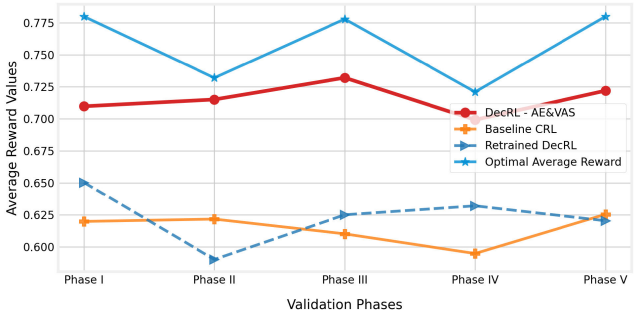


FIGURE 9. **Average reward value in each validation phase.**

1500 learning iterations. It is difficult for agents to share knowledge and collaborate since the algorithm stops them from exchanging information. Throughout the training procedure, the Baseline IRL algorithm performs poorly. Because the reward value represents the overall system performance of the three UAV-BSs, we can conclude that the DecRL-AE&VAS algorithm enables the UAV-BSs to provide the best wireless connection service when compared to the other two frequently utilized baseline approaches in the training phase.

When entering the validation phases (Figure 7 Phase I - V), the proposed algorithm can learn from the past and finally reach the optimal state that provides the highest reward value in the validation scenarios using the AE method. Even if the environment has changed and the quality has dropped dramatically, the algorithm can assist the UAV-BS in quickly adjusting and returning to ideal performance. When we look at the baseline CRL approach, the method failed to respond to environmental changes in a short period due to the slow-paced state exploration. When we compared the baseline results to the retrained DecRL and DecRL-AE&VAS, the results showed that our suggested VAS-AE approach can enable UAV-BS to quickly converge in most of the validation phases. However, with the retrained DecRL, a significant impact may occur on the algorithm and service stability if not using previous existing knowledge. When we integrated RL with adaptive exploration and the value-based action selection strategy, the algorithm showed the best performance in terms of convergence speed, the ability to adapt to environmental changes, and stable service quality of our suggested DecRL-AE&VAS method.

The average reward value changes during the validation phases are depicted in Figure 9. Because the reward value encompasses all of the essential criteria that must be evaluated during the deployment of the UAV-BS, the value clearly demonstrates the model's quality at each stage. As shown in the figure, our proposed DecRL-AE&VAS method can assist the UAV-BS in maintaining the ideal service quality, but the baseline model failed to discover a state that can give reliable and satisfactory service to MC users. The suggested DecRL-AE&VAS algorithm gives a reward value (a weighted sum of the six assessed performance measures) of only about 5% to 6% less than the global best solution in each validation phase.

V. DISCUSSION

In summary, in this paper, we show that when compared to the baseline RL approaches, the DecRL-AE&VAS model can efficiently assist the UAV-BS in finding the near-optimal location and converging at a high-quality level. Utilizing the model-sharing and information-exchange technique, the proposed algorithm can learn from the past and eventually arrive at the optimal state that provides the best reward value, the proposed DecRL-AE&VAS method can achieve the same or even higher levels of system quality than the Baseline CRL approach in both training and validation phases. Because the reward value shows the system's performance, we may conclude that the DecRL-AE&VAS can help UAV-BS provide better service than the other baseline models. Our findings show that the algorithm has the capacity to link MC users swiftly and adequately, allowing drone systems to self-learn without centralized interaction. In Figure 10, we also demonstrate the effectiveness after deploying multiple UAV-BSs in the MC use case. The "7 Macro-BSs" denotes the scenario where 7 marco-BSs are serving the whole scenario, while the "6 Macro-BSs without UAV-BS" denotes the scenario where one macro-BS is broken and no UAV-BS is deployed. The case "6 Macro-BSs with 3 UAV-BSs RL" describes the situation which three UAV-BSs are deployed to fill the coverage hole created in case "6 Macro-BSs without UAV-BS". The results in case "6 Macro-BSs with 3 UAV-BSs Opt" is derived based on grid search to be compared with the RL case. It can be observed that when one macro-BS breaks down, the MC users experience severe performance degradation. Compared with the "6 Macro-BSs without UAV-BS" case, deploying 3 UAV-BSs as in the case "6 Macro-BSs with 3 UAV-BSs RL" can improve about 80% system performance (including 5%, 50% throughput and drop rate) for MC users in both DL and UL. Furthermore, the suggested DecRL-AE&VAS algorithm gives a throughput of 10 Mbps and a drop rate of only about 2% to 3% less than the global best solution indicated in the "6 Macro-BSs with 3 UAV-BSs Opt" case. In conclusion,

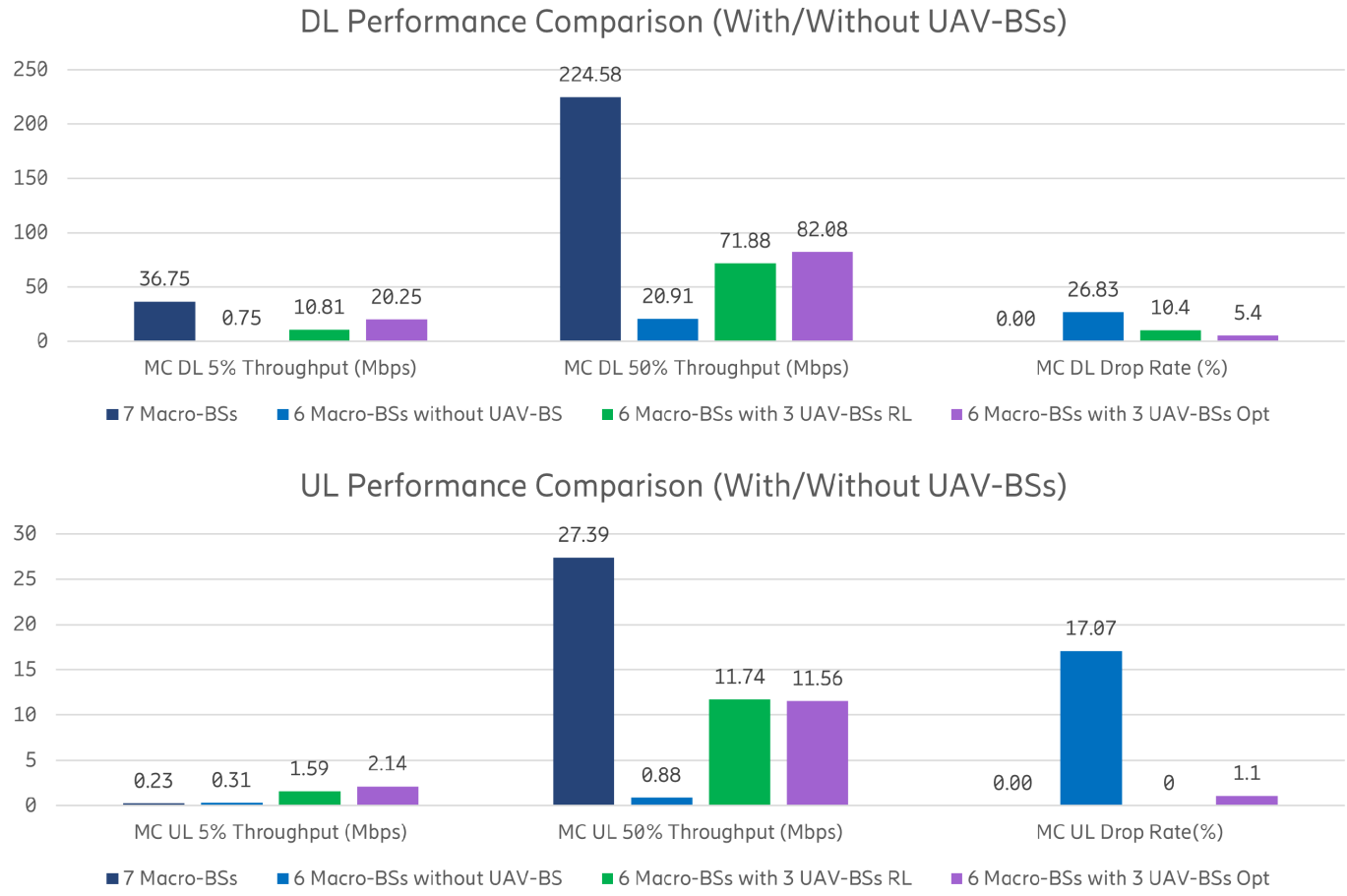


FIGURE 10. **DL & UL performance comparison with/without UAV-BSs in the mission-critical use case.**

deploying UAV-BS can greatly improve the performance of MC users who are experiencing coverage loss. However, due to the BS capability difference, deploying 3 UAV-BS, in this case, can not guarantee that the MC users have similar performance in the case when no emergency happens. The only exception is that the UL 5% throughput performance of MC users served by 3 UAV-BSs is better than that served by one macro-BS before the disaster happens. This is because three UAV-BSs can be deployed in dispersed locations which increases the probability for an MC user to be close to its serving BS and the UL 5% throughput performance of MC users is improved accordingly. The 5th percentile, 50th percentile, and drop rate of normal users are also investigated, but they are nearly not impacted by broken macro-BS and newly deployed UAV-BSs. That is because most normal users are not served by the broken macro-BS before the malfunction. In some cases, deploying UAV-BSs even leads to lower 5th or 50th percentile throughput of normal users compared to the case where there are no UAV-BSs, due to the introduced interference from UAV-BSs.

The proposed model, initially designed for UAV-based communication in mission-critical scenarios, holds promise for adaptation and scalability across diverse disaster types and alternative use cases. Examining these dimensions is pivotal for the model's real-world applicability beyond the specific disaster context considered in this study. When considering different disaster types, such as natural disasters, man-made incidents, and public health emergencies, it is imperative to assess the adaptability of the model. Varied communication needs and environmental challenges associated with each type of disaster may require nuanced adjustments to the model's parameters and strategies. By understanding how the model can flexibly adapt to a range of disaster scenarios, its practical utility can be enhanced.

VI. LIMITATIONS

While our proposed deep reinforcement learning algorithm for the three-dimensional placement of UAV-BSs in MC scenarios demonstrates promising results, it is crucial to acknowledge several limitations and challenges associated with the decentralized nature of the approach. The utilization of decentralized architecture and model delivery mechanism introduces unique considerations that require further research.

Communication Overhead: The independence of edge entities in completing tasks introduces communication overhead among the UAV-BSs. As each UAV-BS makes decisions based on local observations and interacts with others, the need for frequent information exchange arises. This can lead to increased communication latency and potential congestion in scenarios with a large number of UAV-BSs. The need for frequent information exchange arises and follows equation $n^2 + n$ where n equals the number of participated edge entities. This can lead to increased communication latency and potential congestion in scenarios with a large number of UAV-BSs. In our paper, we have parameters to control the number of UAV-BSs communicated in each learning iteration and the frequency of model exchange. However, the trade-off between the control parameter and the quality of the model requires further research.

Coordination Complexity: The decentralized approach requires efficient coordination mechanisms among UAV-BSs to avoid conflicts and optimize their collective behavior, especially when the number of participated UAV-BSs is largely scaled up. Ensuring seamless interaction without a central orchestrator poses challenges, particularly in dynamic environments where the on-ground user movement and network conditions may change rapidly.

In conclusion, while our approach showcases the benefits of decentralized control in UAV-BS placement, the outlined limitations underscore the need for further research and development. Addressing these challenges will be crucial for the practical implementation of our proposed solution in real-world scenarios, particularly in dynamic environments that require a large number of UAV-BSs deployed.

VII. CONCLUSION AND FUTURE WORK

In this paper, we presented a data collection system and machine learning applications for an MC use case, a novel RL algorithm, as well as a decentralized architecture to autonomously pilot multiple UAV-BS in order to offer users temporary wireless access. Two novel strategies, i.e., adaptive exploration and value-based action selection, are developed to help the proposed RL algorithms work efficiently in a dynamic real-world context, incorporating MC user movements and a decentralized architecture to support multi UAV-BSs deployment. Note that the number of participated UAV-BSs can be further extended based on the industrial requirements due to the characteristics of the decentralized architecture. We show that the proposed RL algorithm can monitor the MC service performance and quickly respond to environmental changes via self-adapting exploration probability. In addition, it requires far fewer model training iterations by reusing previous experiences and the value-based action selection strategy. Therefore, the proposed method can well serve the MC users by autonomously navigating multiple UAV-BSs despite environmental changes.

In the future, we will consider separating the configuration for access and backhaul antennas of the UAV-BS, as well as modelling drone rotation in the horizontal domain as an additional parameter for the UAV-BS configuration. We also intend to examine other hyper-parameters and reward function combinations based on different service requirements and further refine the model to accommodate specific real-map scenarios. Last but not least, we will also investigate the energy efficiency problem for deploying UAVs with machine learning components in the real-world context.

REFERENCES

[1] H. Shakhatreh, K. Hayajneh, K. Bani-Hani, A. Sawalmeh, and M. Anan, "Cell on wheels-unmanned aerial vehicle system for providing wireless coverage in emergency situations," *Complexity*, vol. 2021, no. 1, Jan. 2021, Art. no. 8669824.

[2] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36–42, May 2016.

[3] J. Li et al., "Towards providing connectivity when and where it counts: An overview of deployable 5G networks," 2021, *arXiv:2110.05360*.

[4] Z. Qi, A. Lahuerta-Lavieja, J. Li, and K. K. Nagalapur, "Deployable networks for public safety in 5G and beyond: A coverage and interference study," in *Proc. IEEE 4th 5G World Forum (5GWF)*, Oct. 2021, pp. 346–351.

[5] G. Dulac-Arnold, D. Mankowitz, and T. Hester, "Challenges of real-world reinforcement learning," 2019, *arXiv:1904.12901*.

[6] Z. Ding and H. Dong, "Challenges of reinforcement learning," in *Deep Reinforcement Learning: Fundamentals, Research and Applications*. Springer, 2020, pp. 249–272.

[7] J. Li et al., "5G new radio for public safety mission critical communications," *IEEE Commun. Standards Mag.*, vol. 6, no. 4, pp. 48–55, Dec. 2022.

[8] S. A. R. Naqvi, S. A. Hassan, H. Pervaiz, and Q. Ni, "Drone-aided communication as a key enabler for 5G and resilient public safety networks," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 36–42, Jan. 2018.

[9] K. P. Morison and J. Calahorrano. (2020). *FirstNet Case Study: How FirstNet Deployables Are Supporting Public Safety*. [Online]. Available: https://www.policeforum.org/assets/FirstNetDeployables.pdf

[10] A. Merwaday, A. Tuncer, A. Kumbhar, and I. Guvenc, "Improved throughput coverage in natural disasters: Unmanned aerial base stations for public-safety communications," *IEEE Veh. Technol. Mag.*, vol. 11, no. 4, pp. 53–60, Dec. 2016.

[11] L. Ferranti, L. Bonati, S. D'Oro, and T. Melodia, "SkyCell: A prototyping platform for 5G aerial base stations," in *Proc. IEEE 21st Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Aug. 2020, pp. 329–334.

[12] H. Wang, H. Zhao, W. Wu, J. Xiong, D. Ma, and J. Wei, "Deployment algorithms of flying base stations: 5G and beyond with UAVs," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10009–10027, Dec. 2019.

[13] E. Kalantari, M. Z. Shakir, H. Yanikomeroglu, and A. Yongacoglu, "Backhaul-aware robust 3D drone placement in 5G+ wireless networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2017, pp. 109–114.

[14] C. T. Cicek, H. Gultekin, B. Tavli, and H. Yanikomeroglu, "Backhaul-aware optimization of UAV base station location and bandwidth allocation for profit maximization," *IEEE Access*, vol. 8, pp. 154573–154588, 2020.

[15] N. Tafintsev et al., "Aerial access and backhaul in mmWave B5G systems: Performance dynamics and optimization," *IEEE Commun. Mag.*, vol. 58, no. 2, pp. 93–99, Feb. 2020.

[16] 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.

[17] C. Madapatha et al., "On integrated access and backhaul networks: Current status and potentials," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 1374–1389, 2020.

[18] A. Fouda, A. S. Ibrahim, İ. Güvenç, and M. Ghosh, "Interference management in UAV-assisted integrated access and backhaul cellular networks," *IEEE Access*, vol. 7, pp. 104553–104566, 2019.

[19] M.-A. Lahmeri, M. A. Kishk, and M.-S. Alouini, "Artificial intelligence for UAV-enabled wireless networks: A survey," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 1015–1040, 2021.

[20] A. Ly and Y.-D. Yao, "A review of deep learning in 5G research: Channel coding, massive MIMO, multiple access, resource allocation, and network security," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 396–408, 2021.

[21] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, "Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2059–2070, Sep. 2018.

[22] C. Madapatha, B. Makki, A. Muhammad, E. Dahlman, M.-S. Alouini, and T. Svensson, "On topology optimization and routing in integrated access and backhaul networks: A genetic algorithm-based approach," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 2273–2291, 2021.

[23] 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.

[24] H. Yang, J. Zhao, Z. Xiong, K.-Y. Lam, S. Sun, and L. Xiao, "Privacy-preserving federated learning for UAV-enabled networks: Learning-based joint scheduling and resource management," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 10, pp. 3144–3159, Oct. 2021.

[25] C. Zhou et al., "Deep reinforcement learning for delay-oriented IoT task scheduling in SAGIN," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 911–925, Feb. 2021.

[26] S. Tan, C. Dun, F. Jin, and K. Xu, "UAV control in smart city based on space-air-ground integrated network," in *Proc. Int. Conf. Internet, Educ. Inf. Technol. (IEIT)*, Apr. 2021, pp. 324–328.

[27] L. Zhang, A. Celik, S. Dang, and B. Shihada, "Energy-efficient trajectory optimization for UAV-assisted IoT networks," *IEEE Trans. Mobile Comput.*, vol. 21, no. 12, pp. 4323–4337, Dec. 2022.

[28] S. Yin, S. Zhao, Y. Zhao, and F. R. Yu, "Intelligent trajectory design in UAV-aided communications with reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8227–8231, Aug. 2019.

[29] S. Yin and F. R. Yu, "Resource allocation and trajectory design in UAV-aided cellular networks based on multiagent reinforcement learning," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2933–2943, Feb. 2022.

[30] H. Wu, F. Lyu, C. Zhou, J. Chen, L. Wang, and X. Shen, "Optimal UAV caching and trajectory in aerial-assisted vehicular networks: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 12, pp. 2783–2797, Dec. 2020.

[31] Y. Yang and X. Liu, "Deep reinforcement learning based trajectory optimization for UAV-enabled IoT with SWIPT," *Ad Hoc Netw.*, vol. 159, Jun. 2024, Art. no. 103488.

[32] C. Deng, X. Fang, and X. Wang, "Beamforming design and trajectory optimization for UAV-empowered adaptable integrated sensing and communication," *IEEE Trans. Wireless Commun.*, vol. 22, no. 11, pp. 8512–8526, Nov. 2023.

[33] S. M. M. Abohashish, R. Y. Rizk, and E. I. Elsedimy, "Trajectory optimization for UAV-assisted relay over 5G networks based on reinforcement learning framework," *EURASIP J. Wireless Commun. Netw.*, vol. 2023, no. 1, p. 55, Jul. 2023.

[34] W. Liu, D. Li, T. Liang, T. Zhang, Z. Lin, and N. Al-Dhahir, "Joint trajectory and scheduling optimization for age of synchronization minimization in UAV-assisted networks with random updates," *IEEE Trans. Commun.*, vol. 71, no. 11, pp. 6633–6646, Nov. 2023.

[35] C. Zhang, Z. Li, C. He, K. Wang, and C. Pan, "Deep reinforcement learning based trajectory design and resource allocation for UAV-assisted communications," *IEEE Commun. Lett.*, vol. 27, no. 9, pp. 2398–2402, Sep. 2023.

[36] C. Wang, J. Wang, Y. Shen, and X. Zhang, "Autonomous navigation of UAVs in large-scale complex environments: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2124–2136, Mar. 2019.

[37] C. Wang, J. Wang, J. Wang, and X. Zhang, "Deep-reinforcement-learning-based autonomous UAV navigation with sparse rewards," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6180–6190, Jul. 2020.

[38] 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.

[39] Y. Zeng, X. Xu, S. Jin, and R. Zhang, "Simultaneous navigation and radio mapping for cellular-connected UAV with deep reinforcement learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 7, pp. 4205–4220, Jul. 2021.

[40] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3747–3760, Jun. 2017.

[41] S. Ahmed, M. Z. Chowdhury, and Y. M. Jang, "Energy-efficient UAV relaying communications to serve ground nodes," *IEEE Commun. Lett.*, vol. 24, no. 4, pp. 849–852, Apr. 2020.

[42] C. Zhao, J. Liu, M. Sheng, W. Teng, Y. Zheng, and J. Li, "Multi-UAV trajectory planning for energy-efficient content coverage: A decentralized learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 10, pp. 3193–3207, Oct. 2021.

[43] B. Zhu, E. Bedeer, H. H. Nguyen, R. Barton, and J. Henry, "UAV trajectory planning in wireless sensor networks for energy consumption minimization by deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9540–9554, Sep. 2021.

[44] H. Zhang et al., "Autonomous navigation and configuration of integrated access backhauling for UAV base station using reinforcement learning," 2021, *arXiv:2112.07313*.

[45] *Study on Enhanced LTE Support for Aerial Vehicles*, document TR 36.777, Version 15.0.0, 3rd Generation Partnership Project (3GPP), Dec. 2017. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/36_series/36.777/36777-f00.zip

[46] *Radio Resource Control (RRC) Protocol Specification*, document TR 38.331, Version 16.8.0, 3rd Generation Partnership Project (3GPP), Mar. 2022. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/38_series/38.331/38331-g80.zip

[47] *Study on Channel Model for Frequencies From 0.5 to 100 GHz*, document 38.901, Version 16.1.0, 3rd Generation Partnership Project (3GPP), 2020. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/38_series/38.901/38901-g10.zip

[48] B. Galkin, E. Fonseca, R. Amer, L. A. DaSilva, and I. Dusparic, "REQIBA: Regression and deep Q-learning for intelligent UAV cellular user to base station association," *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 5–20, Jan. 2022.

[49] V. Mnih et al., "Playing Atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.

[50] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[51] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. thesis, King's College, Cambridge, U.K., 1989.