Multi-UAV Aided Energy-Aware Transmissions in mmWave Communication Network : Action-Branching QMIX Network

Quang Tuan Do[a], Thien Duc Hua[b], Anh-Tien Tran[a], Dongwook Won[a], Geeranuch Woraphonbenjakul[a], Wonjong Noh[c,*], Sungrae Cho[a,*]

[a]*Department of Computer Science and Engineering, Chung-Ang University, Seoul, 06974, South Korea*
[b]*Centre for Wireless Innovation (CWI), Queen's University Belfast, United Kingdom, United Kingdom*
[c]*School of Software, Hallym University, Chuncheon, 24252, South Korea*

Abstract

Advancements in drone technology and high-frequency millimeter-wave communications are transforming unmanned-aerial-vehicles(UAV)-aided networks, expanding their potential across diverse applications. Despite the advantages of broad frequency bandwidth and enhanced line of sight connectivity in the UAV-aided millimeter-wave networks, it is challenging to provide high network performance because of the inherent limitations of limited UAV energy and millimeter-wave's large path loss. This challenge becomes more important in dynamically changing multi-UAV environments. To address this challenge in multi-UAV networks, we propose a novel approach based on multi-agent deep reinforcement learning called action-branching QMIX. Our method determines nearly optimal codebook-based discrete beamforming vectors and UAV trajectories while maintaining a balance between communication efficiency and energy consumption. The proposed approach employs a new Long Short-Term Memory module to control long sequences effectively and enables it to adapt to changing environmental variables in real-time. We thoroughly evaluate the proposed control with a real-world measurement-based channel model. The evaluation confirms that the proposed control converges stably and consistently, and provides enhanced performance in terms of downlink data rate, success rate of reaching the destination, and service duration when compared to traditional benchmark multi-agent reinforcement learning schemes. These results emphasize the enhanced energy sustainability, robustness, and stability of the proposed approach in dynamically changing multi-UAV environments when compared to the existing benchmark algorithms.

Keywords: Energy management, millimeter-wave communication, multiagent reinforcement learning (MARL), uncrewed aerial vehicle (UAV), UAV-based communication systems

1. Introduction

The emergence of unmanned aerial vehicles (UAVs), paired with advances in millimeter-wave (mmWave) communications and the widespread availability of wireless technology, has accelerated the development of UAV-aided mmWave communication systems. These systems are becoming increasingly recognized as promising alternatives for providing high-speed connectivity in rural or disaster-prone places where conventional communication infrastructures fail. The wide range of mmWave bands enables large data rates and low latency, which are critical for effective UAV communication. Furthermore, the elevation at which UAVs operate usually improves line-of-sight connectivity to ground units, which is essential for mmWave transmissions.

Despite their benefits, UAV-based mmWave communication networks encounter significant challenges that hinder their effectiveness. The first challenge is the limited onboard energy resources of UAVs, which requires the development of efficient energy management strategies to extend their operational lifespan and maintain uninterrupted service [1]. Additionally, mmWave communications are highly prone to interference from other aerial or terrestrial signals and have inherent limitations from the free-space path loss. Advanced and sophisticated beamforming (BF) techniques are therefore essential. However, integrating multiple antennas, necessary for effective beamforming and spatial multiplexing, is restricted by the UAVs' limited size and weight capacity. Furthermore, managing a network of multiple UAVs to efficiently fulfill on-demand user requests presents logistical challenges, especially in deploying multiple UAVs over a target area cost-effectively compared to a single high-power UAV system.

Prior research has explored various strategies to overcome these challenges, utilizing approaches such as genetic algorithms, game theory, and deep reinforcement learning (DRL) [2–6]. However, existing research often overlooks the complexities of enhancing both energy and throughput efficiency within multi-UAV networks, and the majority of studies concentrate primarily on scenarios involving only a single UAV. This work introduces a novel control mechanism designed to enhance both the energy and throughput performance of mmWave commu-

*Corresponding author
email: dqtuan@uclab.re.kr (Quang Tuan Do), dhua01@qub.ac.uk (Thien Duc Hua), attran@uclab.re.kr (Anh-Tien Tran), dwwon@uclab.re.kr (Dongwook Won), geeranuch@uclab.re.kr (Geeranuch Woraphonbenjakul), wonjong.noh@hallym.ac.kr (Wonjong Noh), srcho@cau.ac.kr (Sungrae Cho)

nication networks supported by multiple UAVs. By accounting for critical factors such as UAV mobility, channel conditions, energy status, and communication demands, this study ensures adaptability to dynamically changing environments. To achieve this, we have developed a multi-agent reinforcement learning (MARL) framework that synergistically integrates reinforcement learning (RL) with multi-UAV systems, proposing solutions that are both innovative and applicable in real-world scenarios.

1.1. Backgrounds

We provide background information essential for comprehending this research. Placing this study within the larger settings of technology and theory highlights the importance of establishing advanced control mechanisms, such as the proposed MARL framework. This research advances UAV-assisted mmWave communication systems by addressing both their promising potential and substantial challenges.

Unmanned Aerial Vehicles (UAVs). UAVs, commonly known as drones, are increasingly being integrated into communication networks to provide critical services in areas where traditional infrastructure is limited or non-existent. Coupled with millimeter-wave (mmWave) technology, which operates at frequencies typically within the 30 GHz to 300 GHz range, UAVs can offer high-speed, high-capacity communications [7]. This combination is particularly promising for the deployment of fifth-generation (5G) and future telecommunications networks [8].

Millimeter-Wave Communications: Advantages and Challenges. Millimeter-wave (mmWave) communications are characterized by their ability to transmit large amounts of data quickly due to their high frequency. However, these waves have shorter wavelengths, which can lead to higher losses through absorption and scattering in the atmosphere [9]. This makes line-of-sight communication essential for maintaining strong connections. The integration of mmWave technology with UAVs helps overcome some of these challenges by providing elevated platforms that enhance line-of-sight probability [10].

Relevance of UAVs in Communication Networks. Because of their mobility and versatility, UAVs are ideal for rapidly extending network coverage to remote areas. They can be deployed swiftly after natural disasters to restore communications and facilitate emergency responses [11]. Despite their advantages, UAVs face operational challenges, including limited battery life and payload capacity, which restrict the equipment they can carry [12].

The Role of Beamforming. Beamforming is a signal processing technique used in mmWave communications to direct the transmission and reception of radio signals to specific devices, thus improving the signal's strength and reducing interference [13]. Advanced beamforming techniques are crucial in UAV networks to ensure efficient energy use and robust communication links.

Multi-Agent Reinforcement Learning (MARL) in UAV Networks. The dynamic nature of UAV communication systems, coupled with the challenges posed by mmWave technology, requires adaptive and intelligent control mechanisms. Multi-agent reinforcement learning (MARL) offers a framework through which multiple UAVs can learn to coordinate their movements and communication strategies effectively, adapting to the environment's complexities [14]. This approach allows UAVs to operate autonomously with minimal human intervention, optimizing network performance in real-time. Section 3.1 will further elaborate on this.

QMIX: Optimizing Multi-Agent Cooperation. QMIX is a significant advancement in the field of multi-agent reinforcement learning, designed to optimize agent cooperation in complex and dynamic environments [15]. This algorithm improves the ability of individual agents, such as UAVs in communication networks, to make decisions that benefit the overall system performance. QMIX accomplishes this by allowing each agent to act based on local observations while ensuring that their actions are coordinated toward a common goal, making it ideal for scenarios in which agents must adapt to rapidly changing conditions. However, a notable drawback of QMIX is its reliance on a centralized training process, which may be impractical in scenarios requiring full decentralization or when communication limitations exist. Furthermore, QMIX assumes a fully cooperative environment, which can be restrictive in scenarios involving competitive elements or mixed objectives. Despite these limitations, QMIX's approach to integrating individual learning processes into a cohesive strategy significantly improves both individual agent autonomy and overall network operating efficiency, particularly in UAV communication systems. Section 3.1 will go into further detail on QMIX and its use in UAV networks, emphasizing its role in enhancing communication reliability and operating efficiency.

1.2. Related Works

Recently, many researchers [16–19] have investigated energy problems in the UAV networks. Zhang et al. [16] emphasized energy efficiency in UAV-assisted emergency communication networks by optimizing the UAV trajectory and power allocation to extend the battery life of user devices. They addressed the challenge of providing reliable communication services in post-disaster scenarios where traditional networks may be compromised and where energy resources are scarce. Using convex optimization and a DRL method called the soft actor-critic algorithm, they enhanced the energy sustainability of user devices, ensuring more extended operational periods and improving reliability in critical communication tasks.

In addition, Ding et al. [17] investigated energy efficiency and user fairness by optimizing three-dimensional (3D) trajectories and frequency band allocation of quad-rotor UAVs using a deep deterministic policy gradient-based DRL algorithm. Further, Mohamed et al. [18] and Zhou et al. [19] explored energy-aware UAV trajectory optimizations. They developed an algorithm focusing on maximizing data rates while minimizing energy usage. The algorithm demonstrated superiority

in balancing energy efficiency and communication quality and targeting urban hotspots in mmWave communications. Moreover, Zhou et al. [19] introduced a constrained soft actor-critic algorithm for minimizing mission time while adhering to energy limits. The algorithm is noted for its adaptability to generate optimal trajectories in dynamic environments and improve energy efficiency in data collection.

Moreover, recently, some literature [20–24] has investigated the UAV BF design problem. For example, Susarla et al. [20] examined the role of UAVs in sixth-generation (6G) millimeter-wave wireless networks, emphasizing the importance of fast beam alignment for efficient mmWave communications with base stations (BSs). They leveraged the hierarchical deep Q-network (H-DQN) for UAV-BS beam alignment in uplink communications. The framework, operating in fifth-generation (5G) new radio (NR) BS coverage with 3D beams, uses UAV data and reduces the beam search complexity via location information and a fixed antenna spatial arrangement. Liu et al. [22] proposed a deep learning-based location-aware predictive BF scheme, employing a long short-term memory (LSTM)-based recurrent neural network (RNN) that addresses challenges in accurate beam alignment with ground BSs. The scheme predicts the UAV location, enabling effective and rapid beam alignment in the next time slot for reliable UAV-to-BS communication. Vaezy et al. [23] explored the use of UAVs as wireless service providers, using a uniform linear array to improve the quality of service for downlink users. The research introduces a BF design method for mmWave communication, aiming to maximize user coverage and consider human body blockage. The optimal beam direction problem is modeled as a multi-armed bandit and can identify the optimal beam angle within 10 iterations. Paper [24] introduced a DQN-based RL framework for beam alignment in UAVs employing mmWave uplink communications. The framework leverages a specialized BF codebook optimized by UAV location data to streamline the beam selection process. This method achieves quicker alignment and reduces search complexity, demonstrating enhanced performance and adaptability over traditional techniques, such as multiarmed bandit and exhaustive search methods, particularly in dynamic UAV environments.

Although prior studies predominantly addressed UAV BF challenges in cellular-connected communication, recent research [25–28] has increasingly recognized the need to integrate trajectory optimization into the framework. The joint optimization of the UAV trajectory and BF presents a refined approach, acknowledging the interplay between UAV movement and beam alignment to enhance the overall system performance. Abdalla et al. [25] proposed using UAVs as mobile aerial relays to counter passive eavesdropping in wireless communications. The method clusters users, optimizes multiuser BF, and employs the DQN to optimize the 3D position, BF, and transmission power of the UAV. Muy et al. [26] explored a wireless power transfer network with UAVs that charge ground devices. They introduced a DRL framework to optimize the UAV trajectory, BF, and power transmission, demonstrating better performance than hover-and-fly algorithms. Paper [27] developed a path-planning approach for UAVs in 5G NR BSs using mmWave technology. They employed DRL and the DQN for path planning and beam tracking, offering improved connectivity and beam-tracking efficiency. Dong et al. [28] studied UAV-enabled mmWave communications for physical layer security. They jointly optimized BF, UAV trajectory, and user scheduling using a DRL method to balance secure transmissions and energy efficiency.

In contrast, in UAV-enabled communication systems, the significance of a multi-UAV system model surpasses that of a single-UAV system model. Although single-UAV models provide valuable insight, the complex dynamics and challenges inherent in modern wireless networks necessitate a more comprehensive understanding afforded by multiple UAVs. As such, recent studies and advancements [29–32] in the field have shifted to embracing the potential of multi-UAV system models to unlock the full capabilities of UAV-enabled communication networks. For example, Khalili et al. [29] presented a method to enhance network performance using multiple UAVs with RISs, optimizing the transmission power, UAV trajectory, and beamformer. They employed dueling DQN and successive convex approximation methods, achieving a 6 d Bm reduction in transmission power while maintaining the quality of service. Chiang et al. [30] proposed a machine learning solution for mmWave hybrid BF in multi-UAV networks. They used Q-learning for analog beam tracking and digital BF to maximize the signal-to-interference-plus-noise ratio (SINR), enhancing data transmission and beam efficiency in dynamic environments. Chiang et al. [31] studied hybrid analog-digital BF for UAVs in 3D space, focusing on low-latency and directional mmWave communications. They used Q-learning for beam prediction and digital weight optimization to maximize the SINR. Zhang et al. [32] explored a cooperative jamming approach using UAVs to protect against eavesdroppers, using multiagent DRL (MADRL), specifically multiagent deep deterministic policy gradient (MADDPG), to optimize the UAV trajectory and power in order to reduce jamming. They introduced continuous action-attention MADDPG for enhanced learning and convergence, which performed better than the standard MADDPG. In addition, Cui et al. [33] offered a MARL strategy for multi-UAV networks, allowing UAVs to learn coordination tactics to increase the overall system performance. Park et al. [34] introduced an energy-efficient framework for multi-UAV networks using a modified version of communication neural network (CommNet) for collaborative optimization of flight paths and power usage, reducing energy consumption and ensuring robust network coverage. This approach enables autonomous UAV coordination, adapting to environmental changes and user needs and highlighting the role of the multiagent system in enhancing UAV network efficiency.

1.3. Motivations, Contributions, and Organizations

Previous studies have contributed to energy-aware UAV communication; however, scarce literature has studied the resource and behavior control of mmWave-based multi-UAV networks in a distributed environment. The main contributions of this work are summarized as follows, and Table 1 lists the main differences and features from the representative references.

- First, we formulate a mixed-integer optimization problem that maximizes the total system rate with stringent energy consumption constraints in a multi-UAV-aided mmWave communication network. We leverage the capabilities of multiple UAVs equipped with advanced multi-antenna systems to implement directional BF. Under UAV energy limitations, this method significantly boosts the network capacity by directing transmitted energy toward specific ground users (GUs), improving signal quality while minimizing interference—a critical challenge in mmWave communication. The solution strategically and jointly optimizes UAV positioning and BF, ensuring optimal resource usage and enhanced network performance in a distributed manner.

- Second, we propose a novel MARL framework as a model-free approach to solve the non-convex sequential optimization problem. The proposed distributed MARL model, referred to as action-branching QMIX (AB-QMIX), determines nearly optimal codebook-based discrete BF vectors and UAV trajectories while maintaining a balance between communication efficiency and energy consumption. The proposed AB-QMIX employs a new LSTM module to control long sequences effectively. Using the proposed framework, the UAVs cooperatively move, hover, and provide downlink communication for all GUs in a distributed environment.

- Extensive simulations employ a real-world measurement-based channel model to validate the effectiveness of the proposed AB-QMIX framework in terms of network capacity and underscore its efficiency in energy utilization. The framework demonstrates stable convergence and performance enhancement over conventional MARL approaches, achieving over 90% of the theoretical upper limit in terms of energy efficiency, specifically the maximum possible service duration of UAVs.

The rest of this paper is structured as follows. Section 2 presents a system model comprising the architecture and components of the proposed multi-UAV energy-aware system and the problem formulation. Then, Section 3 describes the proposed algorithm and training technique. Next, Section 4 provides the performance evaluation. Finally, Section 5 concludes the study by reviewing the findings and prospective future research topics.

2. System Model

This section presents the models for the network scenario, antenna, UAV mobility, UAV-user association, energy consumption, and channel of the proposed system.

2.1. Network Model

We considered a downlink mmWave wireless communication network, where M UAVs are deployed as aerial BSs to serve a set of K single antenna GUs, as illustrated in Fig. 1. Specifically, the sets of UAVs and GUs are denoted by $\mathcal{M} = \{1, 2, \ldots, M\}$ and $\mathcal{K} = \{1, 2, \ldots, K\}$, respectively. The UAVs were deployed from an initial position, where they cooperatively move, hover, and provide downlink communication for all GUs. The position of the m-th UAV at time step t is represented by $\mathbf{b}_m(t) = [X_m(t), Y_m(t), H]$, where $X_m(t)$ and $Y_m(t)$ denote the horizontal coordinates of the UAV, indicating its x- and y-axis positions, respectively, whereas H represents the constant altitude of the UAV, maintaining a fixed height above ground to ensure stable service coverage. Conversely, the 3D location of the k-th GU at each time step is given by $\mathbf{u}_k(t) = [x_k(t), y_k(t), 0]$, with $x_k(t)$ and $y_k(t)$ marking the GU position and the zero value indicating the GU altitude at ground level. Each GU is equipped with a global positioning system (GPS) and periodically broadcasts its position to all the UAVs, facilitating accurate and efficient UAV positioning and communication link optimization.

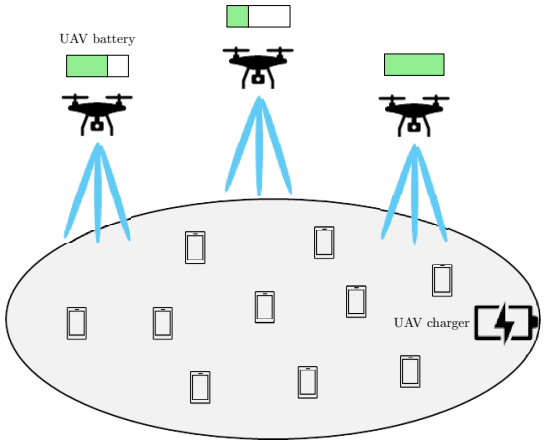

Figure 1: Downlink millimeter-wave communication system aided by multiple uncrewed aerial vehicles.

2.2. Antenna Model

Each UAV is equipped with N antennas in a uniform linear array structure, which helps UAVs construct BF designs to improve the downlink transfer signal for GUs. Figure 2 illustrates the detailed BF architecture of each UAV, adopting an analog BF structure. The structure consists of one radio frequency chain that is fully connected with N antennas to provide BF gain. In analog BF systems, where the signal amplitude adjustment is constrained, the constant-modulus constraint is essential for ensuring uniform signal transmission power across all antennas, as in the following equation:

$$|[\mathbf{W}_m](n)| = \frac{1}{\sqrt{N}}, \ \forall m \in \mathcal{M}, \ \forall n \in \mathcal{N}, \tag{1}$$

where $|[\mathbf{W}_m](n)|$ specifies the magnitude of the BF weight for the n-th antenna element on the m-th UAV, which is set to $\frac{1}{\sqrt{N}}$ to ensure each antenna transmits with equal power. This constraint simplifies the analog BF design, focusing on phase adjustments for directional control and maximizing system efficiency by equally distributing power across the UAV antennas.

Table 1: Comparison of the proposed and previous approaches.

Reference	Beamforming	Trajectory Design	Multi-UAV	Energy Problem	Method Used
[16]		○		○	CO + SAC
[17]		○		○	DDPG
[18]	○	○		○	MAB
[19]		○		○	C-SAC
[20]	○				H-DQN
[21]	○				DQN-PER
[22]	○				LSTM Net
[23]	○				MAB
[24]	○				DQN
[25]	○	○			DQN
[26]	○	○			DQN
[27]	○	○			DQN
[28]	○	○			PPO
[29]	○	○	○		D-DQN
[30]	○	○	○		DQN
[31]	○	○	○		DQN
[32]	○	○	○		A-MADDPG
[33]		○	○	○	DQN
[34]			○	○	Comm-Net
Proposed	○	○	○	○	AB-QMIX

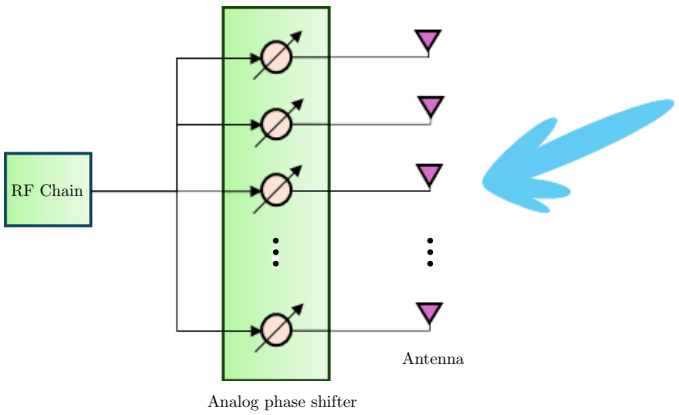

Figure 2: Uncrewed aerial vehicle beamforming architecture.

2.3. UAV Mobility Model

The mobility model of each UAV m is crucial for optimizing network performance and ensuring efficient energy usage within the UAV-assisted mmWave communication system. In each time step, the movement of UAV m is determined by its flight speed $V_m(t)$ within the range $[0, V_{\max}]$, and its azimuth angle $\varphi_m(t)$ within $(0, 2\pi]$. These parameters dictate the UAV's trajectory and are fundamental to managing both communication coverage and energy consumption dynamically. The following kinematic equations govern the position updates:

$$X_m(t + \Delta t) = X_m(t) + V_m(t)\cos(\varphi_m(t))\Delta t, \tag{2}$$

$$Y_m(t + \Delta t) = Y_m(t) + V_m(t)\sin(\varphi_m(t))\Delta t. \tag{3}$$

Collision avoidance is critical, particularly in dense UAV deployments. It is enforced through the following constraint, ensuring a minimum separation distance of $2r$ between any two UAVs:

$$\|\mathbf{b}_m(t) - \mathbf{b}_{m'}(t)\|_2 > 2r, \ \forall m \neq m', \forall t, \tag{4}$$

In the above model, r represents the safety radius around each UAV. This radius is a critical parameter in the collision avoidance mechanism, ensuring that all UAVs maintain a safe distance from each other to prevent accidents. The value of r is assumed to be homogeneous across all UAVs, reflecting standardization in UAV design and operational protocols. This uniform approach helps simplify the computational requirements of the collision avoidance algorithm and ensures consistent safety margins across the entire fleet. Additionally, operational boundaries are set for the airspace in which the UAVs can operate, defined as follows:

$$X_m(t) \in [0, X_{\max}], \ \forall m, t, \tag{5}$$

$$Y_m(t) \in [0, Y_{\max}], \ \forall m, t. \tag{6}$$

To ensure sustainable operation, UAVs are equipped with a finite energy reserve, primarily depleted by propulsion and operational demands. This reserve mandates strategic energy management to allow safe return to charging stations upon nearing depletion. This aspect is critical as it dictates the UAVs' operational endurance and their ability to maintain continuous service. The return to the charging station at the end of a mission is mandated as follows:

$$\mathbf{b}_m(T) = \mathbf{b}_d, \ \forall m, \tag{7}$$

where $\mathbf{b}_m(T)$ denotes the UAV's position at the mission's conclusion, aligning with the designated charging station coordinates, $\mathbf{b}_d$.

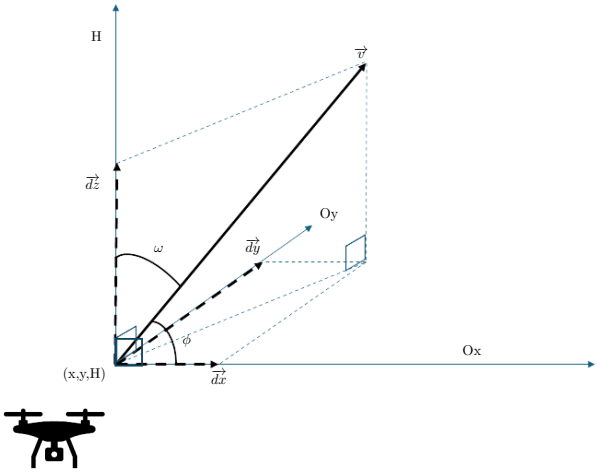

Figure 3: Uncrewed aerial vehicle mobility model.

2.4. UAV-User Association Model

In the realm of UAV operations, especially where multiple UAVs are deployed within the same airspace, establishing an efficient and reliable communication system between the UAVs and their users is paramount. Time-division multiple access (TDMA) offers a structured method to facilitate this communication by dividing the communication channel into distinct time slots. In TDMA, each user is assigned a UAV in a specific time slot, within which the user can receive data from the UAV, ensuring that a limited number of UAVs can serve every user. This model can be more precisely articulated through binary variables to indicate the allocation of UAVs to users during specific time steps. A binary variable that takes the value of 1 if UAV m is serving user k at time step t is denoted by $\alpha_{m,k}(t)$ and is 0 otherwise:

$$\alpha_{m,k}(t) = \{0, 1\}, \quad \forall t, \forall m, \forall k. \tag{8}$$

The objective is to ensure that precisely one UAV serves each user at any given time step and that each UAV can only serve one user at a time, adhering to the constraints of TDMA to avoid interference and optimize service delivery. Thus, we have the following two constraints:

$$\sum_{m=1}^{M} \alpha_{m,k}(t) \leq 1, \quad \forall t, \forall m, \forall k \tag{9}$$

$$\sum_{k=1}^{K} \alpha_{m,k}(t) \leq 1, \quad \forall t, \forall m, \forall k. \tag{10}$$

This paper simplifies the association decision by adopting a proximity-based assignment strategy, wherein each UAV is designated to serve the nearest GU. The overall association matrix can be easily solved using a linear assignment solver based on the 3D distance between all UAV-user pairs to minimize the total distance between paired UAVs and users. Thus, the possibility of interference due to multiple UAVs serving the same GU is reduced, and the quality and reliability of the communication links are significantly enhanced.

2.5. Energy Consumption Model

In this system, we assumed that the UAV has a limited battery pool with a maximum capacity of E_{max}. The battery level of the m-th UAV at each time step t is denoted as $E_m(t)$. All UAVs start their flights with a full battery level and end their service once their battery level reaches a pre-defined critical level $E_{critical}$. Thus, we have the following energy constraints:

$$E_m(1) = E_{max}, \ \forall m \tag{11}$$

$$E_m(T) \leq E_{critical}, \ \forall m. \tag{12}$$

The consumed energy of each UAV is primarily from the processes of propulsion and transmission. In particular, the propulsion process is the action of flying and hovering, whereas the transmission process is the action of providing mmWave communication to the GU. We disregard the energy consumption of the transmission process because, in practice, it is insignificant compared to the propulsion energy cost, as demonstrated in previous work [35]. We can compute the propulsion energy cost using the following formula [36]:

$$P_m(V) = P_i \sqrt{\sqrt{1 + \frac{V^4}{4v_0^2}} - \frac{V^2}{2v_0^2}} + P_b\left(1 + \frac{3V^2}{U_{tip}^2}\right) + \frac{1}{2} d_0 \rho s A V^3. \tag{13}$$

The energy required for propulsion is determined by three main factors of power use: induced drag, represented as $P_i \sqrt{\sqrt{1 + \frac{V_m^4}{4v_0^2}} - \frac{V^2}{2v_0^2}}$, blade profile drag, denoted by $P_b\left(1 + \frac{3V^2}{U_{tip}^2}\right)$, and parasite drag. Here, P_i and P_b are constants that stand for the power consumed by induced drag and blade profile drag, respectively, when the UAV is hovering. When the UAV ceases motion and remains stationary in the air (i.e., $V_m = 0$), the propulsion power simplifies to $P_m(0) = P_i + P_b$. Additionally, v_0 is the average rotor-induced velocity when hovering, and U_{tip} is the speed at the tip of the UAV's blades, measured in meters per second. The term d_0 refers to the fuselage drag ratio, while ρ, s, and A represent the air density, the rotor's radius, and the rotor disk area, respectively. Table 2 summarizes the parameters related to energy consumption modeling. Thus, the m-th UAV energy consumption at each time step is $P_m(V)(t)\Delta t$. As each UAV m has a limited battery pool, we updated its current battery pool at each time step t, denoted by $E_m(t)$, using the following formula:

$$E_m(t + \Delta t) = E_m(t) - P_m(V)(t)\Delta t. \tag{14}$$

2.6. Channel Model

Akdeniz et al. [37] developed a statistical model based on real-world measurements in New York City. This model derives a meticulous spatial statistical assessment of mmWave wireless communication networks in a densely populated urban environment for important channel metrics. These metrics include path loss, spatial cluster count, angular dispersion, and outage probability. Given the study's vast and practical insight,

Table 2: Energy consumption modeling parameters.

Notation	Meaning
W	UAV weight
ρ	Air density
R	Rotor radius
A	Rotor disk area
Ω	Angular velocity of UAV blades
U_{tip}	Tip speed of UAV blades
s	Rotor solidity
d_0	Fuselage drag ratio
k	Correction factor for induced power
v_0	Mean rotor-induced velocity during hover
δ	Profile drag coefficient
P_i	Induced drag power consumption during hovering status
P_b	Profile blade drag power consumption during hovering status

including these known models in this research enables a more accurate representation of channel features in similar urban environments.

Remark 1. *Combining UAV platforms and mmWave cellular networks provides significant benefits. First, the broad bandwidth inherent in the mmWave frequency spectrum enables managing ultra-high data traffic and supports a broad range of UAV applications. Another advantage is the short wavelength of mmWave, allowing the integration of more antennas in a smaller space. As the number of antennas increases, so does the potential for increased BF gain, improving the channel quality in the system. Furthermore, an advantage of mmWave communication is its spatial sparsity, facilitated by pencil-like beams that allow directional communication, improving the efficient reuse of spectrum resources in the spatial domain. Finally, the combination of mmWave communication and a multi-antenna array allows for flexible BF. The flexibility of mmWave BF offers the degrees of freedom to address interference effectively in the spatial domain.*

2.6.1. LoS, NLoS, and outage probabilities

The evaluation methods currently endorsed by the Third-Generation Partnership Project (3GPP), such as those documented in [38], typically employ a statistical approach where each communication link is categorized as either LoS or non-LoS, with the likelihood of each state being dependent on the distance. The characteristics of path loss and other aspects of the link are then determined based on whether the condition is LoS or NLoS.

However, for millimeter-wave (mmWave) communications, it's necessary to introduce an additional state to account for situations where a link might not just be LoS or NLoS, but completely non-existent due to outage. In such an outage state, the link is considered to have no connectivity between the transmitter and receiver, resulting in an infinite path loss. By incorporating this third state, which signifies a total signal loss with a certain random probability, the model more accurately represents the potential for outages that are a distinct characteristic of mmWave communication systems. This statistical model thus adjusts to include probabilities for these three distinct states: LoS, NLoS, and outage.

$$p_{out}(d) = max(0, 1 - e^{-a_{out}d+b_{out}}), \tag{15}$$

$$p_{LoS} = (1 - p_{out}(d))e^{-a_{los}d}, \tag{16}$$

$$p_{NLoS} = 1 - p_{out}(d) - p_{LoS}(d), \tag{17}$$

In this context, the constants a_{los}, a_{out}, and b_{out} are specific parameters that have been determined through analysis of gathered data and vary according to the particular environment being studied. The model for predicting outage probability, as referenced in 15, bears a resemblance to the model used by the 3GPP for non-line-of-sight (NLoS) conditions between suburban relay stations and user equipment, as detailed in [38]. The methodology for calculating the probability of line-of-sight (LoS) conditions, as mentioned in 16, is developed from the principles of random shape theory, a concept explored in depth in [39].

2.6.2. Distance-based path loss

We investigated the air-to-ground channel model for the communication link between UAVs and GUs, including the conditional probability of the LoS and NLoS components [40]. Because the transmission range is one of the bottlenecks that degrade the mmWave communication link, the path loss comprises distance-based path loss plus excessive path loss due to shadowing:

$$PL = \beta_1 + 10\beta_2 \log_{10}(d) + \xi\ [dB], \tag{18}$$

where d denotes the 3D Euclidean distance between the UAV and GU. Under the assumption that each GU knows its geometrical location information with the help of global positioning systems (GPS) and broadcasts its position information periodically, m-th UAV can calculate the distance between itself and the k-th GU by using the following formula:

$$d_{m,k} = \sqrt{(X_m - x_k)^2 + (Y_m - y_k)^2 + H^2} \tag{19}$$

Meanwhile, β_1 and β_2 represent parameters related to the transmission environment, differing between LoS and NLoS links. The symbol ξ is the excessive path loss resulting from lognormal shadowing, with the value depending only on whether the current transmission link is LoS or NLoS: $\xi_{LoS} \sim \mathcal{N}(0, \sigma^2_{LoS})$ and $\xi_{NLoS} \sim \mathcal{N}(0, \sigma^2_{NLoS})$. We define the path loss PL based on the probabilistic state of the communication link as follows:

$$PL = \begin{cases} +\infty & \text{with probability } p_{\text{outage}}, \\ p_{LoS}\, PL_{LoS} & \text{with probability } p_{LoS}, \\ p_{NLoS}\, PL_{NLoS} & \text{with probability } p_{NLoS}. \end{cases} \tag{20}$$

In mapping out the spatial distribution of the antenna's pattern, we utilized a conventional approach aligned with the

3GPP/International Telecommunication Union specifications for multiple input, multiple output technologies, as outlined in [38] and [41]. This approach posits that the channel consists of a random assortment C of "path clusters," each signifying a significant scattering path at a macro level. The description of each path cluster encompasses its proportion of the total power, the primary angles of departure, the spread of beams around these central angles, and the delay in cluster propagation time, all of which are considered hidden variables in this analysis.

The statistical characterization of the channel is based on attributes of the clusters, specifically their quantity and the distribution of power among them. Initially, the quantity of clusters is determined to adhere to a Poisson distribution:

$$C \sim max(Poisson(\lambda), 1), \tag{21}$$

with λ being the average number of clusters observed. Next, the distribution of power for each cluster c, within the range of $[1, 2, \ldots, C]$, relative to the total power of the signal, is formulated as:

$$\Gamma_c = U_c^{r_\tau - 1} 10^{-0.1 Z_c}, U_c \sim U[0, 1], Z_c \sim \mathcal{N}(0, \zeta^2), \tag{22}$$

where r_τ and ζ^2 serve as the parameters for this model.

2.6.3. Small-Scale Fading Model

In the small-scale fading model, each path cluster, denoted by Z, can be broken down into L subpaths. Within each cluster, indexed by c ranging from 1 to C, and each subpath, indexed by l from 1 to L, there are specific angles of departure. These angles are derived from a Gaussian distribution centered around the cluster's core angles, with the standard deviation defined by the cluster's angular spreads (denoted as *rms*). Consequently, the narrowband channel gain for the connection between the m-th UAV and the k-th user can be depicted as follows (refer to [42] for an in-depth explanation):

$$H_{m,k}(t) = \frac{1}{\sqrt{L}} \sum_{c=1}^{C} \sum_{l=1}^{L} \frac{g_{c,l}(t) \mathbf{u}(\theta_{c,l})}{\sqrt{PL}}, \tag{23}$$

where $g_{c,l}(t)$ and $\theta_{c,l}$ are the complex small-scale fading gain and angle of departure of the l-th subpath of the c-th cluster, respectively. Additionally, $\mathbf{u}(.)$ denotes the steering vector of the transmission antenna arrays given the angles of departure. The small-scale coefficients are given by

$$g_{c,l}(t) = \bar{g}_{c,l} e^{2\pi j \Delta t}, \bar{g}_{c,l} \sim \mathcal{CN}(0, \Gamma_c 10^{-0.1 PL}). \tag{24}$$

2.7. Problem Formulation

This study aims to optimize the downlink sum rate of all K GUs. We jointly optimized the analog transmit BF $\mathbf{W}_m$ and the UAV 3D trajectory design $\mathbf{b}_m$. The problem is formulated as follows:

$$\begin{aligned} (\mathbf{P1}): \quad & \max_{\mathbf{W}_m, \mathbf{b}_m} \quad \sum_{t=0}^{T} \sum_{m=1}^{M} \sum_{k=1}^{K} R_{m,k}(t) \\ & \text{s.t.} \quad (1), \\ & \quad (4), (5), (6), (7), \\ & \quad (8), (9), (10), \\ & \quad (11), (12). \end{aligned} \tag{25}$$

In (25), the achievable data rate is calculated as follows:

$$R_{m,k}(t) = \log_2(1 + \mathrm{SINR}_{m,k}(t)), \tag{26}$$

where the SINR for each GUs k served by UAV m is calculated as follows:

$$\mathrm{SINR}_{m,k}(t) = \alpha_{m,k}(t) \frac{H_{m,k}(t) \mathbf{w}_m(t)}{\sum_{m' \neq m} H_{m',k}(t) \mathbf{w}_{m'}(t) + \sigma^2}, \tag{27}$$

where $\mathbf{w}_m(t)$ is the BF vector of the m-th UAV at time step t. The term $\sum_{m' \neq m} H_{m',k}(t) \mathbf{w}_{m'}(t)$ is the total interference acting on user k from all other UAVs, and σ^2 is the additive white Gaussian noise.

Additionally, the constraint in (1) is the constant-modulus constraint of the analog BF matrix for each UAV. The constraint in (4) represents the collision constraint between all UAVs in the system. Moreover, (5) and (6) are the boundary constraints of the system model, where the UAVs can only operate inside this target area. The constraint in (7) indicates that all UAVs must reach the predefined final destination point at the end of the service duration. In addition, (8) to (10) are UAV-user association constraints. Each GU can only be served by a single UAV at a time, and each UAV can only serve, at most, one GU at a time. Additionally, (11) and (12) are energy-related constraints. The constraint in (11) indicates the initial energy level of all UAVs at the beginning of the service duration, whereas (12) indicates that all UAVs are only considered to have finished the service duration when the battery level falls below a threshold $E_{critical}$.

In practice, the problem (P1) is mathematically challenging due to its non-convexity. In addition, it must be jointly optimized in multiple UAV scenarios, making the problem even more difficult to solve using conventional mathematic tools. Determining the optimal solution is infeasible using an exhaustive search, which must be performed over all 3D coordinates of all UAVs for every possible UAV-user association. When the number of UAVs increases, the computational cost significantly increases.

Nonetheless, MADRL-based approaches can address large-scale optimization problems because they can approximate sufficient solutions by exploring the environment and can quickly execute online once they finish the offline training session. This research investigates a QMIX-based multi-agent machine learning approach to address this optimization problem. In the following section, we present the proposed controls to solve the multi-UAV problem.

3. Proposed Joint Control of Three-Dimensional Trajectory and Beamforming

3.1. Preliminaries for Multiagent Deep Reinforcement Learning

As a sequential optimization problem, the proposed problem can be solved using the Markov decision process (MDP) formulation. Moreover, each UAV is treated as a unique agent that cannot directly interact with others throughout the service and can only obtain its state via local observations. Each agent monitors its state in the environment and acts according to its learned optimal policy. Therefore, this problem can be modeled as a decentralized, partially observable MDP (Dec-POMDP). The goal of this Dec-POMDP problem is to maximize the expected cumulative reward over a finite (or infinite) number of steps. However, due to the limited information on the state transition probability and reward, instead of solving this Dec-POMDP directly, we suggest a MADRL solution as a model-free control. Before presenting the proposed control, we introduce preliminary information on MADRL.

3.1.1. Overview of multi-agent deep reinforcement learning

One of the most common machine learning methods is RL, which assists the agent in determining an optimal policy by learning from its interactions with the environment. Furthermore, deep learning is a powerful machine learning technique in which numerous processing unit layers extract increasingly complex properties from input data. In addition, DRL combines RL and deep learning, which can make judgments based on unstructured input data without manually building the state space. This method is also preferred due to its ability to process large volumes of data and select optimal actions to achieve a goal. Particularly, MADRL is a sub-field of DRL that deals with the behavior of many learning agents interacting in the same environment. To build a MADRL model, we considered the following elements:

- Agents: Each agent acts as an independent entity that can perceive the information from its environment and take appropriate actions to achieve a goal.
- Environment: The medium contains agents that interact with the environment to gain information.
- State: At each time step t, the environment has an overall state denoted by $s(t)$ consisting of all the information related to itself. The set of all possible states that can occur in a configured environment is regarded as the state space and is denoted by $\mathcal{S}$.
- Observation: At each time step t, the observation $o_m(t)$ represents the information that can be observed from the current state $s(t)$ of the environment by the m-th agent. The agent can use that information to reconstruct its view of the surrounding world.
- Action: At each time step t, an action $a_m(t)$ derived from a policy $\pi(.)$ is performed by the m-th agent after it observes the environmental information. After executing that action, the agent makes changes to the environment. Thus, it receives the following observation $s_m(t+1)$ of the next step. Moreover, $\mathcal{A}$ is the joint action space from all agents, which is the set of all possible action combinations in a multi-agent system.
- Reward: A reward function $R(s, a)$ assigns numerical values to state-action pairs based on their outcomes, guiding the agent toward goals by distinguishing beneficial actions. This feedback mechanism enables the agent to learn optimal behaviors through trial and error, maximizing the cumulative rewards.
- Policy: A policy $\pi_m(.)$ is a function that maps information from the observation space to the action space. The policy acts as a guideline for the agent to take action after each observation.
- State-action-value function (also called the Q-function): This function maps a state-action pair (s, a) to a scalar value representing the expected reward the agent receives after it executes the specific action following the policy $\pi(.)$ after that specific state.
- Finally, $\gamma \in [0, 1)$ is the discount factor used to maintain a finite sum in the infinite-horizon case.

At every discrete time step t, each agent m executes an action a_i chosen from a set of possible actions $\mathcal{A}_i$. The system's state then changes according to the transition function $P(s'|s, a)$, which depends on the present state and the combined actions of all agents. Subsequently, each agent receives an observation determined by the observation function $O(o|s', a)$, which utilizes the subsequent state and collective action. Following this, a collective reward is allocated to the entire group based on the reward function $R(s, a)$.

3.1.2. Multi-agent deep reinforcement learning with the QMIX network

In multi-agent cooperative systems, all agents pursue a common goal instead of seeking only to maximize their gains; however, this system encounters challenges. The first problem is the curse of dimensionality, where the joint action space grows exponentially by the number of agents, preventing the system from being scalable. Another problem is that, in most real-world scenarios, each agent cannot access the full state information of the environment and can only rely on partial observation with communication constraints. Fixing these problems requires decentralized policy algorithms, and the most common method is centralized training and decentralized execution.

In the QMIX architecture, concerning fully cooperative behavior with centralized training and decentralized execution, the joint action-value function Q_{tot} can be decomposed into M different Q-functions for M agents, where each Q-function Q_m measures how good each action is, given a state, for the agents following a policy [15]. The fundamental concept of this method is that training consistency can be achieved if a

global argmax performed on Q_{tot} yields the same result as a set of individual argmax operations performed on each Q_m:

$$\underset{\mathbf{a}}{\text{argmax}}\, Q_{tot}(\boldsymbol{\tau}, \mathbf{a}) = \begin{pmatrix} \text{argmax}_{a^1}\, Q_1(\tau^1, a^1) \\ \vdots \\ \text{argmax}_{a^n}\, Q_m(\tau^m, a^m). \end{pmatrix} \tag{28}$$

This method allows each agent m to participate in a decentralized execution solely by choosing greedy actions with respect to Q_m. To satisfy (28), we can enforce monotonicity through a constraint on the relationship between Q_{tot} and each Q_m:

$$\frac{\partial Q_{tot}}{\partial Q_m} \geq 0, \forall m \in \mathcal{M}. \tag{29}$$

The QMIX architecture employs a structure to compute the total action-value function Q_{tot} that includes an agent network, a mixing network, and hypernetworks:

- Agent Network: Each agent m operates a distinct Q-network that calculates its action-value function based on the current observation and the action taken in the preceding time step, yielding a Q-value Q_m.
- Mixing Network: This network is a feedforward type that aggregates the Q_m values from all agents into the comprehensive action-value function Q_{tot}. It is designed to ensure monotonicity through the application of nonnegative weight constraints.
- Hypernetwork: Hypernetworks generate weights from the global state s_t for use in the mixing network. Each consists of a linear layer followed by a ReLU activation function, which guarantees the nonnegativity of the weights for the mixing network.

3.2. Proposed Control: Action-Branching QMIX Network

This section presents the proposed AB-QMIX method for joint trajectory and BF design to maximize the system sum rate.

3.2.1. Definitions for multi-agent deep reinforcement learning

State. For each agent m, its state consists of a variable indicating its identification ID to differentiate between itself and other UAVs, the 3D position of all UAVs $\boldsymbol{V}(t)$, the horizontal distance between it and the charging station $d_{0,m}(t)$, the current battery level $E_m(t)$, and the 2D locations of all GUs $\boldsymbol{U}(t)$. Thus, the state information for each UAV m in each time step t is formulated as follows:

$$s_i(t) = \{\text{ID}, \boldsymbol{V}(t), d_{0,m}(t), E_m(t), \boldsymbol{U}(t)\}. \tag{30}$$

Action. We considered the joint trajectory design and BF design; thus, the action output from each agent m in each time step t consists of the next position and the BF design vector. However, because the proposed algorithm is based on the DQN method where the action space must be discretized, we discretized the target region into a grid system, where each grid is equally sized, denoted by ς. We considered the trajectory design for the agent to choose one of the following actions $a_m^{trajectory} \in$ (`hover, up, down, left, right, up_left, up_right, down_left, and down_right`). To calculate the UAV movement corresponding to each of these control actions, we converted the discrete action into the corresponding V and φ and applied them to eqs. (2) and (3):

1. Hover: $V = 0$ and $\varphi = 0$
2. Up: $V = \varsigma\Delta t$ and $\varphi = \pi/2$
3. Down: $V = \varsigma\Delta t$ and $\varphi = -\pi/2$
4. Left: $V = \varsigma\Delta t$ and $\varphi = \pi$
5. Right: $V = \varsigma\Delta t$ and $\varphi = 0$
6. Up left: $V = \sqrt{2}\varsigma\Delta t$ and $\varphi = 3\pi/4$
7. Down left: $V = \sqrt{2}\varsigma\Delta t$ and $\varphi = -3\pi/4$
8. Up right: $V = \sqrt{2}\varsigma\Delta t$ and $\varphi = \pi/4$
9. Down right: $V = \sqrt{2}\varsigma\Delta t$ and $\varphi = -\pi/4$

For the BF design, we considered that the agent m selects its BF vector a_m^{bf} from a beam-steering-based BF codebook $\mathcal{F}$, with the nth element in this codebook defined as follows:

$$\boldsymbol{f}_n(\phi) = \frac{1}{\sqrt{N}}\left[1, e^{j\frac{2\pi}{\lambda}dcos(\phi_n)}, \ldots, e^{j\frac{2\pi}{\lambda}d(N-1)cos(\phi_n)}\right]^T, \tag{31}$$

where ϕ_n is the controlled angle of departure, and we select it from a pool consisting of multiple predefined angles with cardinality $|\mathcal{A}_{bf}|$. This paper considers the angle pool to be equally quantized from the range $(0, 2\pi)$, making the BF selection action the following:

$$a_m^{bf} \in \left\{0, \frac{2\pi}{|\mathcal{A}_{bf}|}, 2\frac{2\pi}{|\mathcal{A}_{bf}|}, 3\frac{2\pi}{|\mathcal{A}_{bf}|}, ..., 2\pi\right\}. \tag{32}$$

Therefore, we designed the action space for each agent m at each time step t to be

$$a_m(t) = \left\{a_m^{trajectory}(t), a_m^{bf}(t)\right\}. \tag{33}$$

Reward. In each time slot, after executing the action according to the policy, each system agent receives the corresponding individual reward from the environment. We considered using the sum rate to be the main reward for the agents:

$$r_{1,m} = \sum_{m=1}^{M} R_m. \tag{34}$$

Also, we considered a step penalty function that scales with the consumed energy to encourage the agent to be efficient with its energy consumption:

$$r_{2,m} = P_m(V)\Delta t. \tag{35}$$

When the terminal state occurs, if the UAV fails to reach the destination, it is punished by a constant penalty. Instead, if the UAV successfully reaches the destination, it is rewarded with a large reward that scales with the service duration:

$$r_{3,m} = \begin{cases} -w_3^{fail}, & \text{agent } m \text{ does not reach the destination} \\ w_3^{success} \times T, & \text{agent } m \text{ reaches the destination.} \end{cases} \tag{36}$$

Afterward, the total reward for each agent is summed from these three rewards:

$$r_m = w_1 r_{1,m} - w_2 r_{2,m} + r_{3,m}, \tag{37}$$

where w_1, w_2, w_3^{fail}, and w_3^{win} are hyperparameters controlling the trade-off weights between the three reward parts.

3.2.2. Proposed architecture and algorithm

The proposed learning model consists of the agent network, mixing network, and hypernetwork. Each agent m has an agent network to represent its action-value function. We propose using the action-branching dueling DQN (D-DQN) to prevent exponential action space growth due to the high dimensions of the action space. Instead of outputting the action-value function Q_m for all possible action combinations, we separated them into J independent action branches, where each branch is responsible for controlling one subaction [43]. Then, we output one Q action-value function for each action branch $Q_{m,1}, Q_{m,2}, \ldots, Q_{m,J}$ and take the sum of all branches to obtain the final Q-value, $Q_m = \sum_{j=1}^{J} Q_{m,j}$.

The mixing network requires input from an RNN; therefore, we further modified the action-branching D-DQN by adding an LSTM module to the architecture. For a specific agent m at the current step t, with the current local observation o_m^t and previous subaction a_m^{t-1}, we can calculate the Q-value for the optimal subaction $a_{m,d}^t$ at the current step by combining the value from the common state estimator $V_m(\tau_m^t)$ and the subaction advantage estimator $A_m^j(\tau_m^t, a_{m,j}^t)$ as follows:

$$\begin{aligned} &Q_{m,j}(\tau_m^t, a_{m,j}^t) \\ =&V_m(\tau_m^t) + \left(A_{m,j}\left(\tau_m^t, a_{m,j}^t\right) - \frac{1}{J} \sum_{a'_{m,j} \in \mathcal{A}_{m,j}} A_{m,j}\left(\tau_m^t, a'_{m,j}\right) \right), \end{aligned} \tag{38}$$

where $\tau_m^t = (o_m^t, a_m^{t-1})$. Afterward, the mixing network takes all outputs from the agent network of all agents as input to calculate the total Q-value for the whole system. Inside the mixing network, hypernetworks also work to transform global state information s^t for the current time step t into the weights of the mixing network. Each hypernetwork consists of a single linear layer, followed by a rectified linear unit activation function to ensure that the mixing network weights are non-negative, helping to maintain the monotonicity of the mixing structure. The mixing network can be trained by calculating the following loss function:

$$L(\theta) = \sum_{i=1}^{b} \left[\left(y_i^{tot} - Q_{tot}\left(\tau, a, s; \theta_M\right) \right)^2 \right], \tag{39}$$

where b is the total number of sampled transitions from the replay buffer memory, θ denotes the main mixing network parameters, and $y_{tot} = r + \gamma max_{a'} Q_{tot}(\tau', a', s'; \theta_M^-)$, where θ^- represents the parameters of the target mixing network.

The primary mixing network has a corresponding target network, which has static parameters and only updates parameters by copying them directly from the primary network after a fixed number of training steps. The reason is that, for DQN-based methods, many state-action values are updated per time step, affecting the action values for the very next state of the agents instead of guaranteeing their stability. The algorithm stability can be improved using a fixed target network that is only updated occasionally because the target networks change at a much slower speed than the primary network. Figure 4 presents the overall architecture of the proposed algorithm.

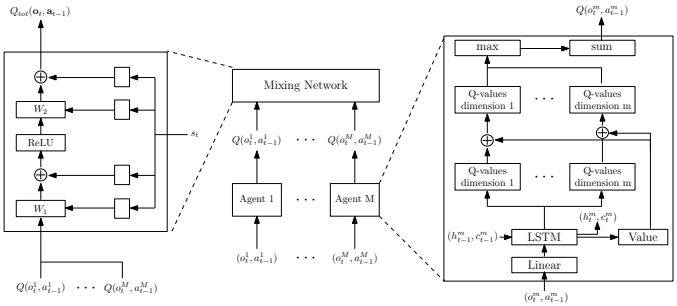

Figure 4: Action-branching QMIX architecture.

The proposed framework uses an action filter to filter out all unavailable actions to ensure that all selected actions by the agents are not illegal (out-of-bounds or colliding with other UAVs). For the implementation, at every time step, the proposed framework checks for all the unavailable actions and filters out their corresponding Q-value so that the agent only selects the optimal action from the remaining pool.

For policy selection, we chose the ϵ-greedy action selection policy, with these two modes:

1. Exploration: the action is randomized to help agents explore the environment randomly and have a better chance of discovering effective action.
2. Exploitation: the agents choose the action that maximizes the state-action-value function based on previous experience.

In the ϵ-greedy policy, the agent performs exploration with a probability of ϵ and performs exploitation with a probability of $1-\epsilon$, where $\epsilon \in [0, 1]$ is a hyperparameter that adjusts the trade-off between exploration and exploitation. Usually, for best practice, $\epsilon = 1$ at the start of each training epoch, and the value gradually reduces as the training epoch continues to shift the agent behavior from exploring the environment to exploiting the trained optimal policy from the networks. The advantage of using the ϵ-greedy policy is enhancing the trade-off between exploration and exploitation.

Upon successfully selecting the next action according to the ϵ-greedy policy, the state transition information tuple $(s_t, \mathbf{o}_t, \mathbf{a}_t, r_t, s_{t+1}, \mathbf{o}_{t+1})$ is stored into a replay buffer memory $\mathcal{D}$. Once the buffer $\mathcal{D}$ reaches a certain size C_{min}, the proposed framework starts performing the training process by randomly sampling a batch of data from B episodes in the buffer. Each sampled episode from this buffer has a sequence length s_l, used as the data to update the parameters for the proposed networks. The buffer memory has a fixed upper limit of C and emits the oldest data whenever the maximum size is reached. The overall

training process for the proposed algorithm is demonstrated in Fig. 5, with further step-by-step details summarized in Algorithm 1.

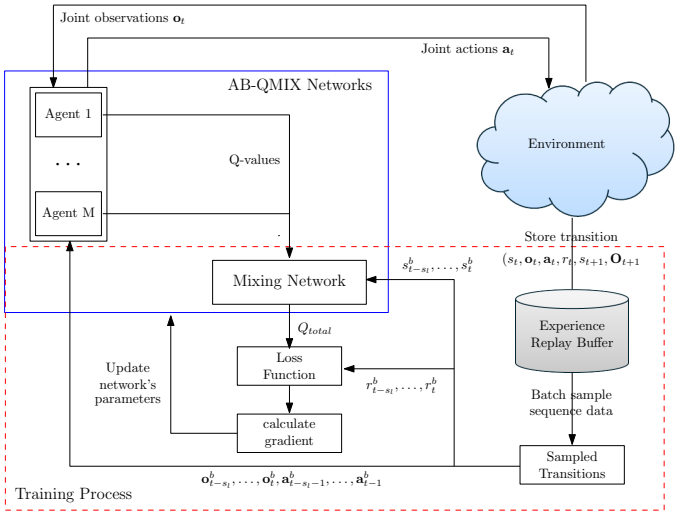

Figure 5: Overall training process of the action-branching QMIX algorithm.

Algorithm 1 Action-Branching QMIX Training Procedure

Initialize the mixing network, target mixing network, agent networks, target agent networks, and hypernetwork
Set the total training episodes E, and the maximum steps per ep $step_{max}$
Set the batch size B, sequence length s_l, ϵ, ϵ decay $\Delta\epsilon$, total number of agents M, discount factor γ, target network copy frequency L
Initialize a buffer memory $\mathcal{D}$ with a maximum capacity of C, and minimum buffer size to start sampling C_{Min}
for each episode **do**
  Get the initial state s_0, reset ϵ to the initial value
  while not last step **do**
    for each agent m **do**
      $\epsilon = max(0, \epsilon - \Delta\epsilon)$
      if probability ϵ **then**
        u^t_m = random action
      else if probability $1 - \epsilon$ **then**
        $u^t_m = argmax_{u^t_m} Q_m(\tau^t_m, a^t_m)$
      end if
    end for
    Get reward r^t, next state s^{t+1}, and next joint observation $\mathbf{o}^{t+1}$
    Store the tuple $(s_t, \mathbf{o}_t, \mathbf{a}_t, r_t, s_{t+1}, \mathbf{o}_{t+1})$ in the buffer $\mathcal{D}$
    if current step $t > C_{Min}$ **then**
      Randomly sample a minibatch of transitions from $\mathcal{D}$
      $Q_{tot} = Mixingnetwork(Q_1, ..., Q_M; Hypernetwork(s_t))$
      $Q'_{tot} = Mixingnetwork'(Q'_1, ..., Q'_M; Hypernetwork'(s_t))$
      $y_{tot} = r^b + \gamma Q'_{tot}$
      loss = MSE(y_{tot}, Q_{tot})
      Update network parameters with the gradient descent
    end if
    if the target network update interval passed **then**
      Copy main network parameters to target network parameters
    end if
  end while
end for

4. Simulation Results

In this simulation, we implemented and evaluated the proposed model using the PyTorch library with Python 3.

4.1. Simulation Setup

We considered a target area of 300×300 m divided into 30×30 equal-sized grids, where $M = 3$ UAVs, each equipped with $N = 48$ antennas, are deployed to serve $K = 20$ GUs, uniformly distributed in the target area. The UAV location is randomly generated close to the origin, and the charging station is located at the opposite corner of the target area. The flying speed of the UAV is assumed to be at a constant speed of one grid at a time, and only the flying direction is decided by the RL agent for each time step, with each time step having a length of $\Delta t = 0.5$ s.

The agent network consists of three hidden layers, with the first hidden layer having 256 neurons. The second layer is an LSTM layer, and the third hidden layer has 128 hidden neurons. For the reward function, we considered $w_1 = 1.2, w_2 = 1.0, w_3^{fail} = 2,500$, and $w_3^{success} = 1.0$. The discount factor for calculating the reward function is set to $\gamma = 0.99$. We trained the model with 1,000 episodes, where each episode lasts until all the UAVs reach their terminal state, either successfully reaching the destination when the battery level is lower than the critical level or failing to reach the destination and running out of energy. For the other training parameters, we set the learning rate to $lr = 0.0001$ and the buffer memory capacity to $|C| = 500,000$. To balance the exploration and exploitation trade-off, we used the ϵ-greedy policy, with ϵ starting at 1.0 and gradually decreasing by $\Delta\epsilon = 5 \times 10^{-6}$ until it reaches the minimum $\epsilon_{min} = 0.1$. At each time slot, the agents either perform a random action with the probability of ϵ or perform the best possible action, outputting from the agent network with the probability of $1 - \epsilon$. Afterward, they store the state transition information in the buffer memory. Once the current memory size reaches a minimum threshold of $C_{min} = 10,000$, the agent network parameters are updated through backpropagation by randomly sampling a batch of data from $B = 4$ episodes in the memory, where each sampled episode has a sequence length of $s_l = 500$. Furthermore, to ensure a stable training session for each of the primary agent networks, we also have a copy version called the target network, where the parameters are fixed, and we only update them once every fixed interval of 10,000 main network parameter updates.

Regarding propulsion energy consumption, each UAV has a fixed energy pool starting at $E_{max} = 200,000$, which is depleted to maintain its flying status. The propulsion energy cost is a function of its current flying velocity. Table 3 provides the physical parameters related to the propulsion energy consumption for the UAV flights. With the provided physical variables, Fig. 6 illustrates the propulsion energy cost for various flying speeds.

The simulations were conducted using a custom-built simulation environment in Python, utilizing the PyTorch framework for implementing the MARL algorithms. The simulations were executed on a machine with the following specifications to ensure high computational performance and accuracy in the multi-agent reinforcement learning simulations: (1) Operating System: Windows 10 Education 64-bit, (2) Processor: 12th Gen Intel(R) Core(TM) i7-12700 CPU @ 2.1GHz (20 CPUs), (3) Memory: 32 GB RAM, and (4) Graphics Card:

NVIDIA GeForce RTX 3070, with a total of 24 GB of memory (8 GB GDDR6 VRAM directly available to GPU). This hardware setup provided the computational power necessary to handle the complex real-time data processing and learning algorithms involved in the simulations, ensuring that the simulation environment was robust and reflective of real-world operational conditions.

Table 3: Simulation parameters related to energy consumption.

Notation	Meaning	Value
W	UAV weight	100 N
ρ	Air density	1.225 kg/m^3
R	Rotor radius	0.5 m
A	Rotor disk area	0.79 m^2
Ω	Angular velocity of UAV blades	400 rad/s
U_{tip}	Tip speed of UAV blades	200 m/s
s	Rotor solidity	0.05
d_0	Fuselage drag ratio	0.3
k	Correction factor for induced power	0.1
v_0	Mean rotor-induced velocity during hover	7.2
δ	Profile drag coefficient	0.012
P_i	Induced drag power consumption during hovering status	88.6279
P_b	Profile blade drag power consumption during hovering status	79.8563

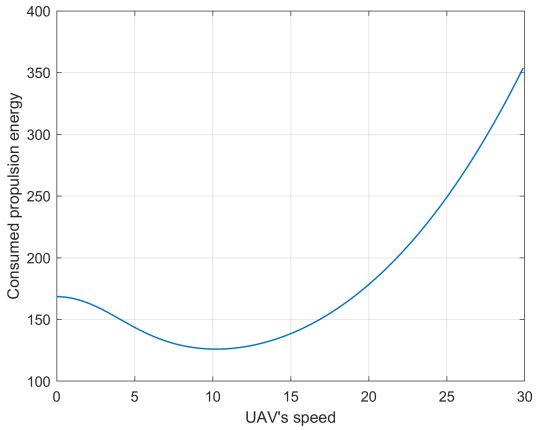

Figure 6: Propulsion energy consumption at various velocities.

This work uses the following standard benchmark algorithms to illustrate the effectiveness of the proposed algorithm:

- Standard DQN (S-DQN) [44]: The S-DQN is a groundbreaking RL algorithm that combines Q-learning with deep neural networks. The critical innovation of DQN is using a deep convolutional neural network to approximate the Q-value function, which assesses the quality of actions in given states within an environment. This approach enables addressing high-dimensional sensory inputs, making it suitable for tasks with complex input spaces, such as video games or robotic control systems. The original DQN algorithm introduced such techniques as experience replay and target networks to stabilize the learning process and improve convergence.

- Dueling DQN [45]: The D-DQN enhances the architecture of the standard DQN by separately estimating two components of the Q-value function: the value function $V(s)$ that estimates the expected reward of being in a particular state s and the advantage function $A(s,a)$ that reflects the additional benefit of taking a specific action a in state s over others. The D-DQN combines these two streams at the final layer to produce the Q-value. This architectural tweak allows the network to learn which states are (or are not) valuable without having to learn the effect of each action for each state, providing a more robust estimate of state values, often leading to better policy evaluation, especially where the action choice does not significantly affect the outcome.

- Hierarchical DQN (H-DQN) [46]: The H-DQN introduces a two-tiered system of agents to manage tasks with hierarchical structures. The higher-level agent sets abstract goals, which are subtasks that guide the lower-level agent's actions. By dividing the problem into a hierarchy of decision-making processes, H-DQN can manage complex tasks that require long-term planning and decision-making and learn policies involving sequences of actions and subgoals. This hierarchical approach is beneficial when the environment has a natural hierarchical structure or when tasks can be decomposed into meaningful subtasks, leading to more efficient learning and better performance on complex problems.

4.2. Numerical Analysis

4.2.1. Convergence analysis

Figure 7 provides insight into the convergence rate through the reward optimization aspect of algorithms with over 1,000 training episodes. The graph displays a general trend of increasing rewards, indicative of the improving performance of the algorithms as they learn from the environment. The proposed algorithm displays a higher and more stable reward-increase trajectory, signifying that it is consistently making better decisions over time. Among the benchmark algorithms, S-DQN and H-DQN also slowly converge but at a slower rate than the proposed method. In addition, D-DQN also experienced a sudden reward spike around episodes 600 to 800. However, it dropped back to its previous state afterward, indicating a sudden change in its optimal policy, leading to the agent being stuck in a new local optimum. This outcome underscores the enhanced stability of the proposed approach, illustrating its superior robustness in comparison to the existing benchmark algorithms.

The same pattern as above is also depicted in Fig. 8 because the reward is calculated using the overall system sum rate. This figure indicates that the proposed algorithm also exhibits an upward trend, indicative of its ability to optimize network

throughput over time. Initially, all algorithms emphasize exploring the state space. This exploration phase results in seemingly random movements within the target area, experiencing a decrease in the overall system throughput. Furthermore, during this early period, the agent struggles to optimize its energy consumption, resulting in a lower service duration and leading to a lower overall total throughput in each episode. As the training episode progresses, the ϵ value slowly decreases, and the agents slowly switch from exploration to exploitation mode. They start using their learned strategy to control their flight path and BF to slowly optimize the total throughput of the system while ensuring the possibility of reaching the destination before running out of energy. As the ϵ value continues decreasing, the proposed method quickly recovers and continues to improve, demonstrating resilience and an effective learning strategy.

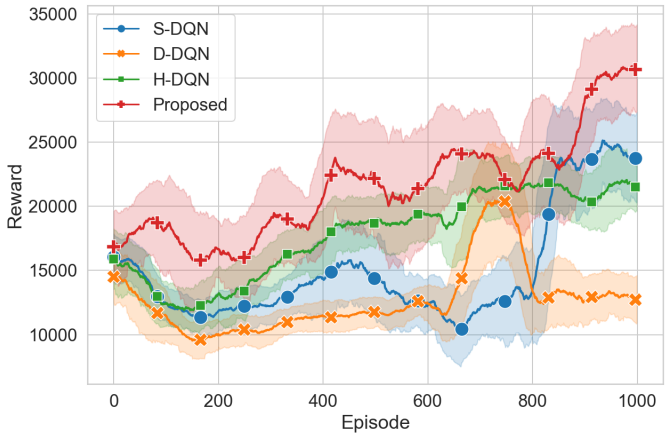

Figure 7: Convergence of rewards per training episode.

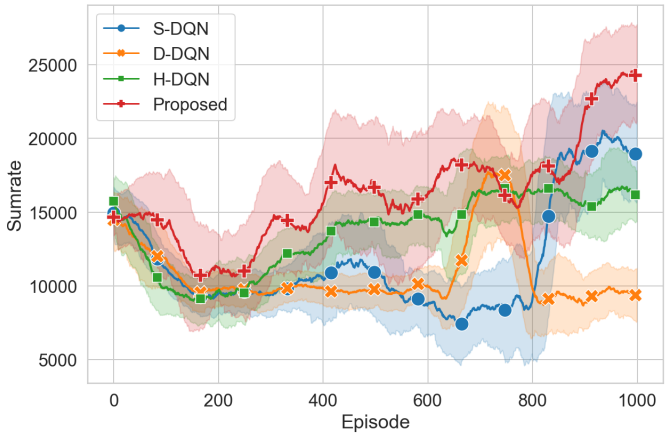

Figure 8: Convergence of sum rates per training episode.

4.2.2. *Probability of reaching the destination and service duration*

Figure 9 illustrates the average success rate over the training episodes. Initially, a fluctuation is observed across all methods, which is related to the early exploration phase when all algorithms are learning how to explore the environment. However, once the agent slowly moves away from the random exploration mode, the proposed algorithm rapidly ascends to a high success rate, asymptotically approaching 1.0, indicating swift convergence and an effective learning strategy. The proposed method outperforms the benchmarks by reaching a higher success rate more rapidly. This enhanced performance is owing to the use of an LSTM structure in the proposed algorithm. The LSTM is recognized for its memory capabilities and proficiency in learning long sequences of data. This attribute helps the proposed agent learn the necessary sequence of optimal control that leads it to the destination. Moreover, benchmark algorithms have lower success rates, with S-DQN and D-DQN depicting intermediate performance, suggesting a potential inefficiency in dealing with complex or high-dimensional action spaces. In contrast, H-DQN outperforms both S-DQN and D-DQN and comes close to the proposed algorithm, underscoring the advantage of hierarchical approaches in complex decision-making scenarios.

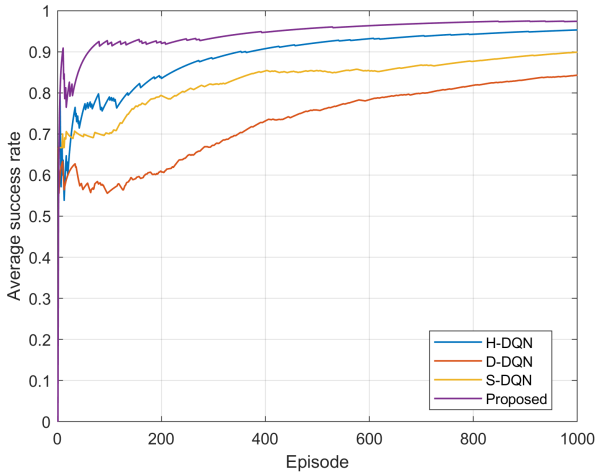

Figure 9: Average probability of reaching the destination per training episode.

In Fig. 10, the average service duration is provided alongside the upper bound of the possible service duration calculated from simulation variables. Initially, all agents randomly explore following the random exploration policy; thus, they perform in the same way. However, as more episodes pass and the agent shifts to using its learned optimal policy, the proposed algorithm starts rapidly increasing its service duration, especially during the first 200 episodes. The proposed algorithm experienced a drop in service duration around episode 200 $\sim$ 400 but gradually rebounded throughout the remaining episodes. Before this episode window, the proposed algorithm focuses too much on obtaining a higher session terminal reward because the reward at the terminal state is set equal to the service duration. However, afterward, the proposed algorithm shifts its optimal policy to optimize the sum-rate reward in each time slot, exemplifying the superiority of the LSTM memory mechanism at memorizing and learning from long data sequences. For the remaining episodes, the proposed method tries to op-

timize the sum-rate-related reward and destination-reaching-related reward simultaneously, resulting in a much higher overall reward. The benchmark algorithms also converge to the upper bound service duration but at a slower speed.

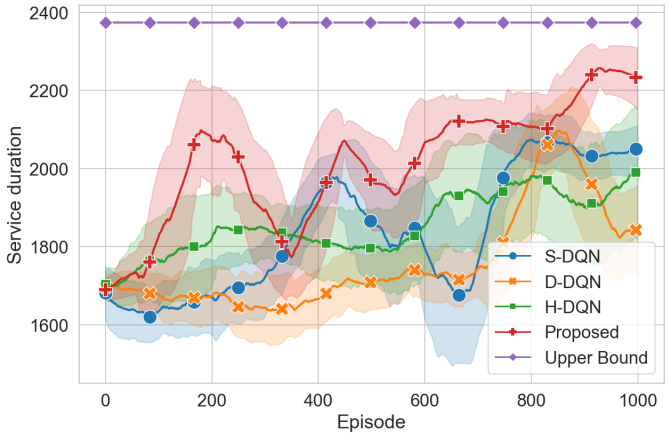

Figure 10: Average service duration per training episode.

4.2.3. Hyperparameter comparison

Figure 11 compares the average data rate against various UAV altitudes, specifically for $H \in \{5, 8, 11, 14, 17, 20\}$. The figure reveals that the proposed method maintains a higher data rate across various altitudes, suggesting robustness to altitude changes, a crucial attribute for UAV networks. Additionally, the downward trend across all algorithms implies that altitude negatively affects the data rate, consistent with the attenuation effects in mmWave communications. Among the three benchmark algorithms, S-DQN and D-DQN perform similarly, with S-DQN having a slight edge at lower altitudes, whereas H-DQN performs the worst among all three benchmarks.

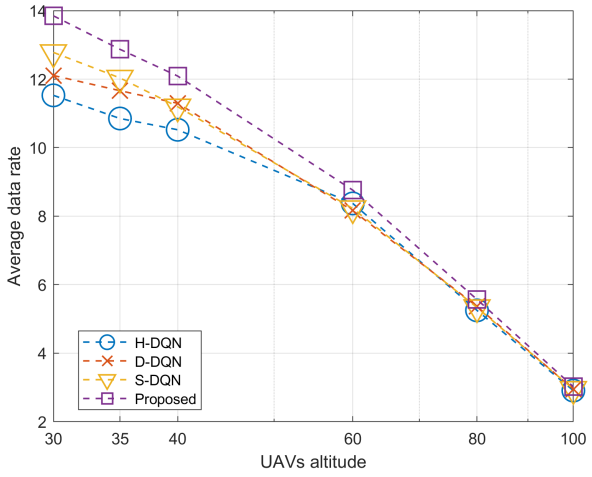

Figure 11: Average data rate for various uncrewed aerial vehicle altitudes when $K = 20$ and $N = 48$.

Figure 12 presents the average data rate according to the number of served GUs. The graph indicates a general upward trend in the average data rate as the number of users increases until reaching a certain threshold. The reason for the data rate drop for the lower number of users is that users are more sparsely distributed in the target area. Hence, the overall communication distance between the UAV and the target user becomes longer, further degrading the mmWave signal. However, as the number of users reaches the threshold $K = 14$, the performance for all the algorithms starts converging. Generally, the proposed algorithm still outperforms benchmark algorithms for sufficiently high K settings. Among the benchmark algorithms, D-DQN performs well when K is not high, but once the number of users is sufficiently high, S-DQN starts performing slightly better than D-DQN. In addition, H-DQN demonstrates a weaker data-rate performance, as the H-DQN agent is more focused on flying quickly toward the destination while focusing less attention on optimizing the transfer data rate.

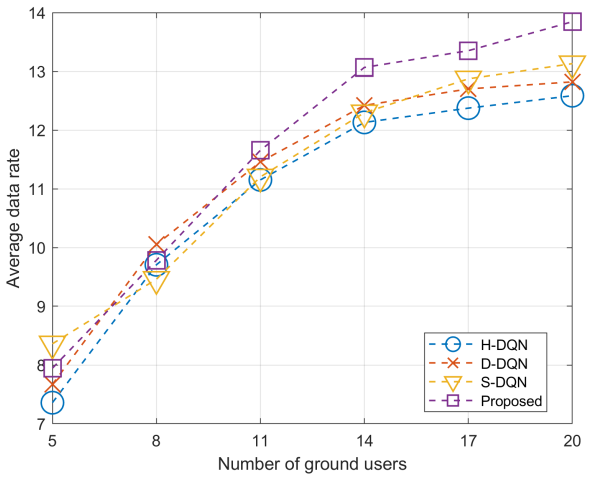

Figure 12: Average data rate for different number of ground users when $H = 30$ and $N = 48$

Lastly, we observe the average data rate according to the number of UAV antennas in Figure 13. The proposed method demonstrates a consistent increase in the data rate with more antennas, highlighting its ability to exploit additional hardware capabilities effectively. The other algorithms also benefit from increased antennas, but the improvement rate is less pronounced than that of the proposed method. This outcome suggests that the proposed method may have a more sophisticated strategy for leveraging increased antenna counts, which could be critical for enhancing communication network capacities.

4.2.4. Execution time analysis

Figure 14 demonstrates the execution time per step among the proposed algorithm and three benchmark algorithms, with conditions set at $K = 20$ and $N = 48$. The y-axis quantifies the execution time per step in milliseconds, providing a measure of each algorithm's computational efficiency. Notably, the proposed algorithm exhibits a slightly higher median execution time relative to the benchmarks, suggesting a more computationally intensive process. Nevertheless, the tighter distribution of the proposed algorithm's execution times implies enhanced

stability in its processing. This slight increase in execution time is justifiable given the substantial improvements in communication quality and system performance delivered by the proposed algorithm. The proposed algorithm employs the LSTM module, which efficiently maintains long sequences and dynamically adapts to environmental changes, ensuring robust and stable communication capabilities. This sophistication contributes to slightly longer execution times but results in significant improvements in network capacity and energy efficiency. These advantages validate the trade-off between execution speed and extensive functionality, underlying the algorithm's capability to manage complicated, dynamic communication environments.

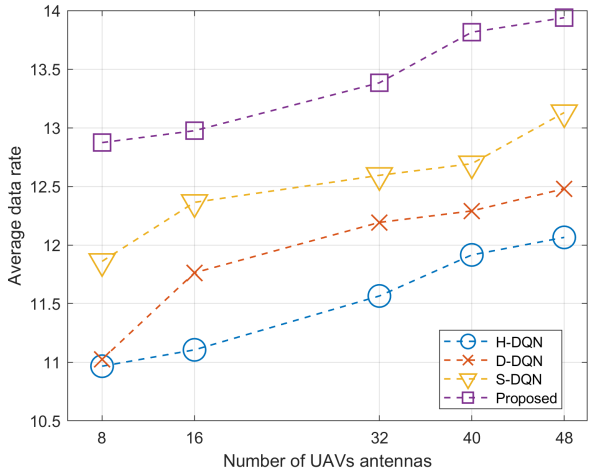

Figure 13: Average data rate for various numbers of UAV antennas when $K = 20$ and $H = 30$.

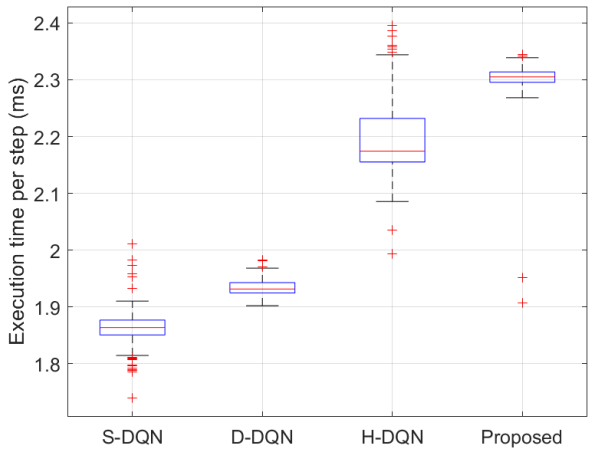

Figure 14: The box plot distribution of execution time per step (in milliseconds) when $K = 20$ and $N = 48$.

4.3. Simulation Summary and Discussion

The extensive simulations demonstrate the effectiveness of the proposed multi-agent reinforcement learning (MARL) framework, Action-Branching QMIX (AB-QMIX), in optimizing the performance of UAV-assisted mmWave communication networks. The results underscore several key contributions:

- Enhanced Network Performance: The implementation of AB-QMIX led to significant improvements in network throughput and efficiency. By optimizing UAV trajectories and beamforming strategies in real-time, the system achieved nearly 90% of the theoretical upper bound for energy efficiency. This is a substantial improvement over benchmark algorithms, highlighting the adaptability and robustness of the proposed approach.

- Energy Efficiency: The proposed framework effectively manages the limited energy resources of UAVs, extending their operational life and maintaining consistent communication capabilities even in challenging environments. This is crucial for ensuring reliable service in remote or disaster-stricken areas without access to traditional communication infrastructure.

- Adaptability to Dynamic Environments: The use of the MARL-based proposed approach allows each UAV in the network to adapt independently to changes in the environment, including variable user demands and physical obstacles. This adaptability is critical for maintaining high levels of service quality across diverse deployment scenarios.

- Practical Implications: The real-world measurement-based channel model used in the simulations provides a robust basis for validating the proposed approach, suggesting its applicability and effectiveness in actual deployment scenarios. This enhances the potential for practical implementation and sets the stage for future on-field deployments.

5. Conclusion

This work proposes a novel MARL framework, called the AB-QMIX network, that maximizes the total system rates under energy and mobility constraints in a UAV-aided mmWave communication network. The proposed AB-QMIX network employs a new LSTM module to control long sequences effectively and adeptly manage the intricacies of the UAV trajectory optimization and analog BF design within the constraints of limited energy resources. Using the proposed framework, the UAVs cooperatively move, hover, and provide downlink communication for all GUs in a distributed environment. The extensive simulations, employing a real-world measurement-based channel model to validate its effectiveness, confirmed that the proposed control provides significant network throughput and energy performance enhancement, in particular, around 90% of the upper bound performance in terms of energy sustainability of UAVs, compared to legacy benchmark MARL schemes. Future research directions include extending from analog BF to hybrid digital-analog BF optimization and assessing the fairness between users based on data demand.

CRediT Authorship Contribution Statement

Quang Tuan Do: Conceptualization, investigation, methodology, software, formal analysis, validation, writing the original draft, writing the review, and editing. **Thien Duc Hua**: Conceptualization, methodology, and editing. **Anh-Tien Tran**: Methodology and software. **Dongwook Won**: Conceptualization and software. **Geeranuch Woraphonbenjakul**: software. **Wonjong Noh**: Project administration, supervision, reviewing, and validation. **Sungrae Cho**: Project administration, supervision, reviewing, and validation. All authors have read and approved the final manuscript.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was supported in part by the MSIT (Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2024-RS-2022-00156353) supervised by the IITP(Institute for Information Communications Technology Planning Evaluation) and in part by the National Research Foundation of Korea (NRF) grants funded by the Korea government (MSIT) (RS-2023-00209125).

References

[1] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, M. Debbah, A tutorial on uavs for wireless networks: Applications, challenges, and open problems, IEEE communications surveys & tutorials 21 (3) (2019) 2334–2360.

[2] E. Eldeeb, J. M. de Souza Sant'Ana, D. E. Pérez, M. Shehab, N. H. Mahmood, H. Alves, Multi-uav path learning for age and power optimization in iot with uav battery recharge, IEEE Transactions on Vehicular Technology (2022).

[3] C. Zhan, Y. Zeng, Energy minimization for cellular-connected uav: From optimization to deep reinforcement learning, IEEE Transactions on Wireless Communications 21 (7) (2022) 5541–5555.

[4] W. Huang, Z. Yang, C. Pan, L. Pei, M. Chen, M. Shikh-Bahaei, M. Elkashlan, A. Nallanathan, Joint power, altitude, location and bandwidth optimization for uav with underlaid d2d communications, IEEE Wireless Communications Letters 8 (2) (2018) 524–527.

[5] T. P. Truong, N.-N. Dao, S. Cho, et al., Flyreflect: Joint flying irs trajectory and phase shift design using deep reinforcement learning, IEEE Internet of Things Journal (2022).

[6] W. Noh, S. Cho, et al., Sparse cnn and deep reinforcement learning-based d2d scheduling in uav-assisted industrial iot networks, IEEE Transactions on Industrial Informatics (2023).

[7] B. C. Nguyen, N. T. Xuan, H. T. Manh, H. L. T. Thanh, P. T. Hiep, Performance analysis for multi-ris uav noma mmwave communication systems, Wireless Networks 29 (2) (2023) 761–773.

[8] Y. Liu, H.-N. Dai, Q. Wang, M. K. Shukla, M. Imran, Unmanned aerial vehicle for internet of everything: Opportunities and challenges, Computer communications 155 (2020) 66–83.

[9] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, F. Gutierrez, Millimeter wave mobile communications for 5g cellular: It will work!, IEEE access 1 (2013) 335–349.

[10] M. Gapeyenko, D. Moltchanov, S. Andreev, R. W. Heath, Line-of-sight probability for mmwave-based uav communications in 3d urban grid deployments, IEEE Transactions on Wireless Communications 20 (10) (2021) 6566–6579.

[11] D.-H. Tran, V.-D. Nguyen, S. Chatzinotas, T. X. Vu, B. Ottersten, Uav relay-assisted emergency communications in iot networks: Resource allocation and trajectory optimization, IEEE Transactions on Wireless Communications 21 (3) (2021) 1621–1637.

[12] C. H. Liu, Z. Chen, J. Tang, J. Xu, C. Piao, Energy-efficient uav control for effective and fair communication coverage: A deep reinforcement learning approach, IEEE Journal on Selected Areas in Communications 36 (9) (2018) 2059–2070.

[13] S. Kutty, D. Sen, Beamforming for millimeter wave communications: An inclusive survey, IEEE communications surveys & tutorials 18 (2) (2015) 949–973.

[14] K. Zhang, Z. Yang, T. Başar, Multi-agent reinforcement learning: A selective overview of theories and algorithms, Handbook of reinforcement learning and control (2021) 321–384.

[15] T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, S. Whiteson, Monotonic value function factorisation for deep multi-agent reinforcement learning, Journal of Machine Learning Research 21 (178) (2020) 1–51.

[16] C. Zhang, X. Li, C. He, X. Li, D. Lin, Trajectory optimization for uav-enabled relaying with reinforcement learning, Digital Communications and Networks (2023).

[17] R. Ding, F. Gao, X. S. Shen, 3d uav trajectory design and frequency band allocation for energy-efficient and fair communication: A deep reinforcement learning approach, IEEE Transactions on Wireless Communications 19 (12) (2020) 7796–7809.

[18] E. M. Mohamed, S. Hashima, K. Hatano, Energy aware multiarmed bandit for millimeter wave-based uav mounted ris networks, IEEE Wireless Communications Letters 11 (6) (2022) 1293–1297.

[19] X. Zhou, X. Zhang, H. Zhao, J. Xiong, J. Wei, Constrained soft actor-critic for energy-aware trajectory design in uav-aided iot networks, IEEE Wireless Communications Letters 11 (7) (2022) 1414–1418.

[20] P. Susarla, Y. Deng, M. Juntti, O. Sílven, Hierarchial-dqn position-aided beamforming for uplink mmwave cellular-connected uavs, in: GLOBECOM 2022-2022 IEEE Global Communications Conference, IEEE, 2022, pp. 1308–1313.

[21] I. Ahmad, R. Narmeen, Z. Becvar, I. Guvenc, Machine learning-based beamforming for unmanned aerial vehicles equipped with reconfigurable intelligent surfaces, IEEE Wireless Communications 29 (4) (2022) 32–38.

[22] C. Liu, W. Yuan, Z. Wei, X. Liu, D. W. K. Ng, Location-aware predictive beamforming for uav communications: A deep learning approach, IEEE Wireless Communications Letters 10 (3) (2020) 668–672.

[23] H. Vaezy, M. S. H. Abad, O. Ercetin, H. Yanikomeroglu, M. J. Omidi, M. M. Naghsh, Beamforming for maximal coverage in mmwave drones: A reinforcement learning approach, IEEE Communications Letters 24 (5) (2020) 1033–1037.

[24] P. Susarla, B. Gouda, Y. Deng, M. Juntti, O. Silvén, A. Tölli, Learning-based beam alignment for uplink mmwave uavs, IEEE Transactions on Wireless Communications 22 (3) (2022) 1779–1793.

[25] A. S. Abdalla, A. Behfarnia, V. Marojevic, Uav trajectory and multi-user beamforming optimization for clustered users against passive eavesdropping attacks with unknown csi, IEEE Transactions on Vehicular Technology (2023).

[26] S. Muy, J.-R. Lee, Joint optimization of trajectory, beamforming, and power allocation in uav-enabled wpt networks using drl combined with water-filling algorithm, Vehicular Communications 43 (2023) 100632.

[27] P. Susarla, Y. Deng, G. Destino, J. Saloranta, T. Mahmoodi, M. Juntti, O. Sílven, Learning-based trajectory optimization for 5g mmwave uplink uavs, in: 2020 IEEE International Conference on Communications Workshops (ICC Workshops), IEEE, 2020, pp. 1–7.

[28] R. Dong, B. Wang, J. Tian, T. Cheng, D. Diao, Deep reinforcement learning based uav for securing mmwave communications, IEEE Transactions on Vehicular Technology 72 (4) (2022) 5429–5434.

[29] A. Khalili, E. M. Monfared, S. Zargari, M. R. Javan, N. M. Yamchi, E. A. Jorswieck, Resource management for transmit power minimization in uav-assisted ris hetnets supported by dual connectivity, IEEE Transactions on Wireless Communications 21 (3) (2021) 1806–1822.

[30] H.-L. Chiang, K.-C. Chen, W. Rave, M. K. Marandi, G. Fettweis,

Machine-learning beam tracking and weight optimization for mmwave multi-uav links, IEEE Transactions on Wireless Communications 20 (8) (2021) 5481–5494.
[31] H.-L. Chiang, K.-C. Chen, W. Rave, M. K. Marandi, G. Fettweis, Multi-uav mmwave beam tracking using q-learning and interference mitigation, in: 2020 IEEE International Conference on Communications Workshops (ICC Workshops), IEEE, 2020, pp. 1–7.
[32] Y. Zhang, Z. Mou, F. Gao, J. Jiang, R. Ding, Z. Han, Uav-enabled secure communications by multi-agent deep reinforcement learning, IEEE Transactions on Vehicular Technology 69 (10) (2020) 11599–11611.
[33] J. Cui, Y. Liu, A. Nallanathan, Multi-agent reinforcement learning-based resource allocation for uav networks, IEEE Transactions on Wireless Communications 19 (2) (2019) 729–743.
[34] C. Park, H. Lee, W. J. Yun, S. Jung, C. Cordeiro, J. Kim, Cooperative multi-agent deep reinforcement learning for reliable and energy-efficient mobile access via multi-uav control, arXiv preprint arXiv:2210.00945 (2022).
[35] Y. Zeng, R. Zhang, Energy-efficient uav communication with trajectory optimization, IEEE Transactions on Wireless Communications 16 (6) (2017) 3747–3760.
[36] Y. Zeng, J. Xu, R. Zhang, Energy minimization for wireless communication with rotary-wing uav, IEEE Transactions on Wireless Communications 18 (4) (2019) 2329–2345.
[37] M. R. Akdeniz, Y. Liu, M. K. Samimi, S. Sun, S. Rangan, T. S. Rappaport, E. Erkip, Millimeter wave channel modeling and cellular capacity evaluation, IEEE journal on selected areas in communications 32 (6) (2014) 1164–1179.
[38] 3GPP, Evolved Universal Terrestrial Radio Access (E-UTRA); Further advancements for E-UTRA physical layer aspects, Technical Report (TR) 36.814, 3rd Generation Partnership Project (3GPP), release 9 (2010).
[39] T. Bai, R. W. Heath, Coverage and rate analysis for millimeter-wave cellular networks, IEEE Transactions on Wireless Communications 14 (2) (2014) 1100–1114.
[40] C. Yan, L. Fu, J. Zhang, J. Wang, A comprehensive survey on uav communication channel modeling, IEEE Access 7 (2019) 107769–107792.
[41] ITU-R, Requirements related to technical performance for IMT-Advanced radio interface(s), Technical Report (TR) M.2134, International Telecommunication Union (ITU) (2008).
[42] D. Tse, P. Viswanath, Fundamentals of Wireless Communication, Cambridge University Press, Cambridge, 2005.
[43] A. Tavakoli, F. Pardo, P. Kormushev, Action branching architectures for deep reinforcement learning, in: Proceedings of the aaai conference on artificial intelligence, Vol. 32, 2018.
[44] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing atari with deep reinforcement learning, arXiv preprint arXiv:1312.5602 (2013).
[45] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, N. Freitas, Dueling network architectures for deep reinforcement learning, in: International conference on machine learning, PMLR, 2016, pp. 1995–2003.
[46] T. D. Kulkarni, K. Narasimhan, A. Saeedi, J. Tenenbaum, Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation, Advances in neural information processing systems 29 (2016).