sensors

MDPI

Article

Deep Reinforcement Learning for Joint Trajectory Planning, Transmission Scheduling, and Access Control in UAV-Assisted Wireless Sensor Networks

Xiaoling Luo [1,2], Che Chen [3,4], Chunnian Zeng [1], Chengtao Li [2], Jing Xu [5,*] and Shimin Gong [4]

[1] School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China
[2] China Three Gorges Corporation, Wuhan 430010, China
[3] School of Computer Sciences, Minnan Normal University, Zhangzhou 363000, China
[4] School of Intelligent Systems Engineering, Sun Yat-sen University, Shenzhen 518107, China
[5] School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China
* Correspondence: xujing@hust.edu.cn

Abstract: Unmanned aerial vehicles (UAVs) can be used to relay sensing information and computational workloads from ground users (GUs) to a remote base station (RBS) for further processing. In this paper, we employ multiple UAVs to assist with the collection of sensing information in a terrestrial wireless sensor network. All of the information collected by the UAVs can be forwarded to the RBS. We aim to improve the energy efficiency for sensing-data collection and transmission by optimizing UAV trajectory, scheduling, and access-control strategies. Considering a time-slotted frame structure, UAV flight, sensing, and information-forwarding sub-slots are confined to each time slot. This motivates the trade-off study between UAV access-control and trajectory planning. More sensing data in one time slot will take up more UAV buffer space and require a longer transmission time for information forwarding. We solve this problem by a multi-agent deep reinforcement learning approach that takes into consideration a dynamic network environment with uncertain information about the GU spatial distribution and traffic demands. We further devise a hierarchical learning framework with reduced action and state spaces to improve the learning efficiency by exploiting the distributed structure of the UAV-assisted wireless sensor network. Simulation results show that UAV trajectory planning with access control can significantly improve UAV energy efficiency. The hierarchical learning method is more stable in learning and can also achieve higher sensing performance.

Keywords: UAV; multi-agent deep reinforcement learning; trajectory planning; access control

Citation: Luo, X.; Chen, C.; Zeng, C.; Li, C.; Xu, J.; Gong, S. Deep Reinforcement Learning for Joint Trajectory Planning, Transmission Scheduling, and Access Control in UAV-Assisted Wireless Sensor Networks. *Sensors* **2023**, *23*, 4691. https://doi.org/10.3390/s23104691

Academic Editor: Sergio Toral Marín

Received: 3 April 2023
Revised: 6 May 2023
Accepted: 8 May 2023
Published: 12 May 2023

1. Introduction

Nowadays, with the development of unmanned aerial vehicles (UAVs) and the increasing traffic demand on future wireless networks, UAVs can be integrated into wireless networks and used to build an air–ground integrated wireless sensing network for the Internet of Things (IoT), e.g., [1–3]. Traditionally, direct links between ground users (GUs) and a remote base state (RBS) can be unreliable due to channel blockage, GU mobility, and limited energy supply. Thanks to enhanced air-to-ground direct channel conditions and UAVs' fast mobility, UAVs can play an important role assisting GU data sensing and information forwarding to the RBS. UAVs can be used as aerial access points to enhance service provisioning to the GUs or as relay nodes to assist data transmissions beyond the RBS's service coverage area [4,5]. For example, by leveraging their flexibility in fast deployment, UAVs can serve as mobile access points for emergency rescue [6,7].

Currently, there are still some limitations to joint control of UAV trajectory and transmission-control strategies due to the complexity of high-dimensional optimization,

the lack of centralized coordination, and unknown dynamics of network environments, e.g., [8–10]. To exploit the performance gain of UAV-assisted wireless networks, UAV trajectory planning is one of the most beneficial design problems to make use of UAV mobility and reshape the network structure dynamically in favor of data transmission, e.g., [11–22]. There are many existing works that focus on the trajectory planning problem in UAV-assisted wireless networks. The GUs' uplink-data transmission strategy is also a critical design aspect for efficient data collection and transmission in UAV-assisted sensing networks. Due to variations in UAV coverage at different locations, GUs have to be smartly divided among different UAVs as a trade-off between interference and network coverage [23,24]. When some UAVs have a low altitude and are closer to the GUs, the UAV may have a restricted coverage area and only serve a limited number of GUs; however, it will have better channel conditions for the GUs under its coverage. In other cases, when more GUs are covered by the same UAV, the sensing information can be of a large amount and thus take up more of the UAV's buffer space. This implies more sensing time and higher transmission power for the UAV to forward all information to the RBS. Such a performance trade-off motivates us to optimize the UAVs' access-control strategy jointly with UAV trajectory planning. It is clear that UAV access control depends on UAV trajectories in each time slot and on the time-varying network environment, including the GUs' spatial distribution, channel conditions, traffic demands, and energy supply. Most of the existing works in the literature focus on energy and spectrum efficiency in UAV-assisted sensing networks by designing UAV trajectories and effective scheduling strategies [25–29].

In this paper, we focus on the joint optimization of UAV trajectory, transmission-scheduling, and access-control strategies in a wireless powered sensor network. The GUs are low-power sensor devices with limited energy supply, but they can harvest and convert RF signals into energy supply. As the UAVs fly over their trajectories, they not only collect the sensing data from the GUs but also adapt their access-control strategies to balance GU energy harvesting and consumption. This can help sustain the GUs' sensing activities and prolong the lifetime of the sensor network. In particular, we consider a time-slotted frame structure for the UAVs to sense and report GU sensing information. In each time slot, the UAVs decide the optimal hovering locations and the transmission-scheduling strategy for information forwarding. Given the UAVs' locations, each GU can upload its sensing data via either low-power backscatter communications or conventional RF communications with a higher transmission rate. The GUs' mode selection between backscatter and RF communications can be optimized to balance the GUs' energy consumption and traffic demands. The GUs' access-control strategy can be further optimized at each UAV to balance the sensing and transmission overhead. Considering the non-convexity and complexity in such a high-dimensional control problem, we first propose the multi-agent DRL approach to jointly adapt UAV trajectory and transmission-scheduling and GU mode-selection and access-control strategies via continuous interactions with the network environment. To improve the multi-agent learning efficiency, we further propose a hierarchical learning framework to decompose the control variables into two parts. Based on the UAVs' local observations, the UAV trajectory and scheduling strategy is firstly updated by the upper-layer MADDPG algorithm. Then, given the fixed sensing locations, the GU mode-selection and access-control strategies can be further adapted by the lower-layer DQN method. Our simulation results demonstrate that the hierarchical learning framework has more preferable convergence performance and achieves a significantly higher reward than the conventional MADDPG algorithm.

2. Related Works

2.1. Multi-UAV-Assisted Wireless Networks

Many traditional optimization methods are applied to solve the problems of trajectories, resource allocation, and scheduling in UAV-assisted wireless sensor networks. To jointly optimize UAV trajectory, resource allocation, and power-allocation strategies, a non-convexity and combinatorial problem was formulated in [11], wherein the authors derived

an approximate and iterative algorithm to solve it. The authors in [12] aimed to maximize the minimum average data collection rate of all sensing nodes (SNs). However, the problem lacks a closed-form solution for effective power control. Instead, a data regression method was employed to approximate the optimal solution by the block coordinate descent (BCD) method. To maximize the GUs' sum rate, the author in [13] proposed using an intelligent reflecting surface (IRS) to improve channel conditions. Similarly, the BCD method was used to optimize resource allocation, IRS phase shift, UAV trajectory planning, and transmission power in an iterative manner. Optimization methods typically require complete network information to adapt UAV trajectory-planning and resource-allocation strategies. This becomes inflexible in a dynamic wireless network as the UAVs frequently change their sensing locations. The overhead for information exchange can be extremely high. Additionally, trajectory optimization in the spatial–temporal domain essentially relies on dynamic programming, which is computational demanding in a large-scale UAV-assisted network.

2.2. Multi-Agent DRL for UAV-Assisted Wireless Networks

Compared to traditional optimization methods, the recent application of DRL can make the UAVs more adaptive to a dynamic network environment with incomplete information, e.g., imperfect channel conditions and unknown traffic demands. The authors in [14] studied the joint IoTD association, partial offloading, and communication-resource-allocation problem. A multi-agent DDPG algorithm was proposed to maximize the service satisfaction of the IoTD while minimizing its total energy consumption. The authors in [15] proposed an air computing system to provide computing services for ground equipment. Multi-agent proximal policy optimization (MAPPO) was employed to maximize the number of computing tasks within the heterogeneous QoS requirements by jointly optimizing the UAV resource-allocation and task-offloading strategies. The authors in [16] leveraged the twin-delayed deep deterministic policy gradient (TD3) algorithm to plan UAV trajectories and achieve the goal of minimizing task completion delay. The authors in [17] considered complicated spatial- and temporal-coupling in UAV trajectory planning and network formation. A heuristic algorithm was proposed to update the UAVs' network formation while optimizing UAV trajectories by using the multi-agent deep deterministic policy gradient (MADDPG) algorithm. In particular, each UAV can collect and cache GU sensing data first and then forward the cached data to the next UAV when they meet each other on their trajectories. The authors in [18] proposed a federated multi-agent deep deterministic policy gradient (F-MADDPG) algorithm for UAV trajectory planning to maximize the average spectral efficiency. Federated averaging (FA) is used to eliminate the isolation of data and thus accelerate the convergence of learning. The distributed F-MADDPG (DF-MADDPG) method is further designed to reduce the communication overhead in the distributed architecture. The design idea of a layered learning algorithm appears in many publications. For example, the authors in [19] aimed to minimize the UAV's total energy consumption. A two-layer hybrid learning algorithm was designed to adapt the UAV's trajectory by the DRL method in the top layer and then optimize the underlying resource allocation by using a model-based optimization method. The authors in [20] adopted the hierarchical multi-agent DRL (H-MADRL) framework to improve overall energy efficiency in a mobile edge-computing system by jointly optimizing a high-level access point's beamforming strategy and the low-level users' offloading decisions. The authors in [21] proposed a hierarchical DRL framework to minimize the age of information in two steps. The first step is to determine the users' transmission-scheduling strategy through the outer-loop DRL method, and the second step aims to adapt the uplink and downlink transmission strategies of all nodes through an inner-loop optimization method. Different from the above hierarchical learning frameworks, our method in this paper includes two DRL learning layers instead of a hybrid learning and optimization framework. The upper-layer MADDPG is used to solve the UAV trajectory-planning problem, while the lower-layer DQN is used to solve the GU access-control strategy. The authors in [22] studied UAV network formation and trajectory optimization by a hierarchical learning

approach. The network formation aims to adapt the UAV-to-UAV links to improve the UAVs' transmission capabilities. In the outer-loop, a heuristic algorithm is used to adapt the UAVs' network formation. Given the fixed network formation strategy, UAV trajectory planning is adapted by the multi-agent DDPG algorithm, which is further enhanced by the Bayesian optimization method. Different from [22], our work in this paper assumes that all UAVs are required to report information directly to the base station, and we focused on UAV trajectory planning, GU transmission scheduling, and access control, which were not considered in [22]. Additionally, we design a two-layer learning algorithm in this paper instead of the outer-loop heuristic algorithm, which can be inflexible for a large-scale UAV-assisted wireless network.

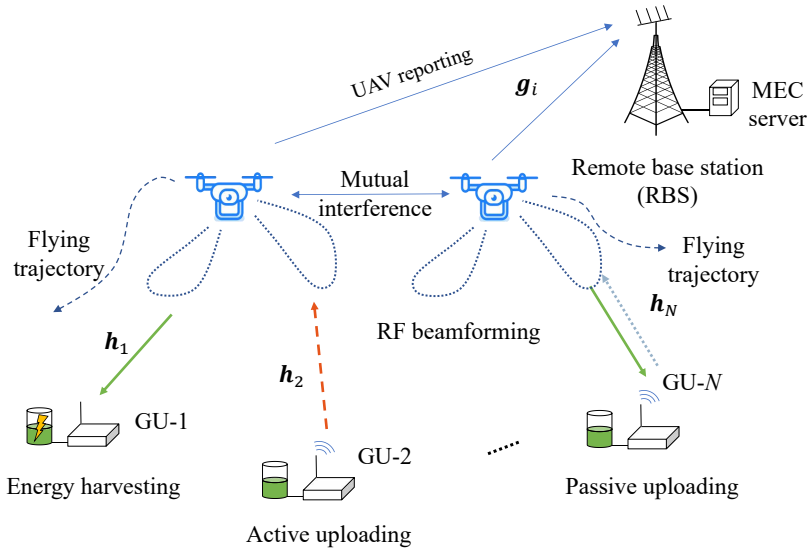
2.3. UAV-Assisted Sensing Scheduling and Access Control

Given UAVs' high mobility, it becomes an important task to adaptively update the the GUs' sensing-scheduling and access-control policies according to the time-varying network environment. The authors in [29] employed UAVs to assist with downlink transmissions in cellular networks. To maximize the users' sum achievable rate subject to limited fronthaul capacity, mixed-integer nonlinear programming (MINLP) was proposed to jointly design the UAVs' positions, transmission beamforming, and the UAV-UE association strategies. The authors in [26] proposed a framework for charging scheduling and energy management for UAVs. To maximize charging efficiency, UAVs have to be properly scheduled to fly back to the charging tower. A multi-agent DRL method was developed to achieve collaborative energy sharing between the UAVs and the charging tower. The authors in [27] aimed to collect the latest information from the GUs by minimizing the averaged age-of-information (AoI). A UAV was employed as a relay node to assist with information transmission to the receiver. The authors in [28] used a UAV as an edge cloud that provides data processing services for IoT devices. The goal was to minimize the UAV's energy consumption while meeting quality-of-service (QoS) requirements. The authors in [25] aimed to navigate a swarm of UAVs to provide optimal communication coverage for mobile users under partial observation. They proposed a stochastic DRL strategy, namely the soft deep recurrent graph network (SDRGN) approach, to reduce the training cost through distributed online learning. Considering the non-convexity and the unavailability of channel state information due to the UAVs' movement, the deep Q-learning algorithm was used to update the UAVs' locations, while the difference of convexity algorithm is used to iteratively update the UAVs' transmission beamforming and UAV-UE association. The authors in [30] considered rate-splitting multiple access (RSMA) to serve multiple GUs simultaneously in a UAV-assisted wireless network, with the goal of maximizing the overall capacity. The authors in [31] also considered RSMA for a multi-UAV-assisted downlink wireless network to maximize the multi-user ergodic sum rate. The authors in [32] considered using a UAV as a flying base station to serve multiple GUs. The GUs' uplink information transmissions to the UAV followed the non-orthogonal multiple access (NOMA) strategy to improve spectrum efficiency. GUs' NOMA transmissions were also studied in [33], which considered a multi-UAV-assisted vehicular communication network.

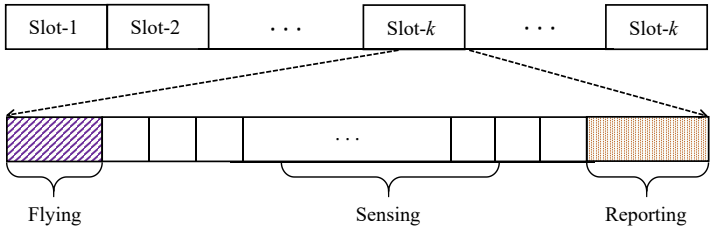
3. System Model

We consider a UAV-assisted wireless network with one RBS and multiple UAVs to serve multiple GUs, as shown in Figure 1a. The set of UAVs is denoted as $\mathcal{N} = \{1, 2, \ldots, N\}$, and the set of GUs is denoted as $\mathcal{M} = \{1, 2, \ldots, M\}$. Due to blockage or large distances between GUs and the RBS, direct links between GUs and the RBS are unavailable. The UAVs can fly over the GUs, collect the GUs' sensing data, and then carry the information to the RBS. Each GU can harvest energy from the UAVs' RF beamforming signals to charge its battery and sustain its operations. Each UAV has F antennas, while the GU has a single antenna. Via beamforming optimization, the UAV can control its energy transfer to different GUs and also adapt the uplink transmission rates. Each GU's sensing data can be uploaded to the UAV by either active RF communications or passive backscatter communications [34],

depending on its energy status, the channel condition, and traffic demand. After collecting the GUs' sensing information, the UAV then forwards the information to the RBS.



(**a**) UAV-assisted sensing network



(**b**) Time allocation for flying, sensing, and reporting phases

Figure 1. UAV-assisted downlink wireless power transfer and uplink information transmission.

3.1. UAV Trajectory Planning

UAV trajectory planning is realized in a time-slotted frame structure, as shown in Figure 1b. Each time slot has a fixed length τ, which is further divided into three sub-slots for flying, sensing, and reporting phases. The UAV can fly to a preferable location in the first sub-slot τ_f, can collect the GUs' information during sensing sub-slot τ_s, and can then report the information to the RBS in sub-slot τ_d. In sensing sub-slot τ_s, the UAVs adopt a time-division protocol to collect GU information. In particular, each GU under the UAV's coverage is allocated a mini-slot τ_z. All GUs can upload their information to the UAV one-by-one via active or passive communications. Additionally, each GU can harvest RF energy when the other GUs are actively transmitting. The third sub-slot τ_d is used for the UAV to report its information to the RBS. We assume that the UAV-GU and the UAV-RBS channel conditions are constant in each time slot and may change over different time slots as the UAVs fly their trajectories.

Similar to [17], each UAV-i's trajectory can be defined as a set of locations over different time slots, i.e., $\boldsymbol{\mathcal{L}}_i = [\boldsymbol{\ell}_i(t)]_{t\in\mathcal{T}}$. Each location is specified by a 3-dimensional (3D) coordinate, i.e, $\boldsymbol{\ell}_i(t) = (x_i(t), y_i(t), z_i(t))$. Let $\boldsymbol{\ell}_0(t)$ denote the RBS's location and $d_{i,0}(t)$ denote the distance between UAV-i and the RBS in slot-t. Given that UAV-i moves in direction $\boldsymbol{d}_i(t)$ with limited speed $v_i(t) \leq v_{\max}$, UAV-i's location in the next time slot can be

updated as $\boldsymbol{\ell}_i(t+1) = \boldsymbol{\ell}_i(t) + v_i(t)\tau_f \boldsymbol{d}_i(t)$. We have the following inequalities to regulate UAV mobility:

$$d_{i,j}(t) \triangleq ||\boldsymbol{\ell}_i(t) - \boldsymbol{\ell}_j(t)|| \geq d_{\min}, \text{ and } ||\boldsymbol{\ell}_i(t+1) - \boldsymbol{\ell}_i(t)|| \leq v_{\max}\tau_f, \tag{1}$$

where $d_{\min}$ denotes the minimum allowable distance between two UAVs to ensure safety.

Given the location $\boldsymbol{\ell}_m^u$ of GU-m on the ground, its distance to UAV-i is given by $d_{m,i}(t) = ||\boldsymbol{\ell}_i(t) - \boldsymbol{\ell}_m^u||$. We consider a realistic channel model consisting of both line-of-sight (LOS) and non-line-of-sight (NLOS) components. Let $\mathbf{h}_{m,i}(t) \in \mathbb{C}^{F\times 1}$ denote the channel vector between UAV-i and GU-m at the t-th slot, which can be modeled as $\mathbf{h}_{m,i}(t) = \sqrt{\psi_{m,i}(t)}\tilde{\mathbf{h}}_{m,i}(t)$, where $\psi_{m,i}(t) = \omega_0(d_{m,i}(t))^{-\alpha}$ denotes large-scale fading, while small-scale fading is characterized as follows:

$$\tilde{\mathbf{h}}_{m,i}(t) = \sqrt{\frac{K}{1+K}}\bar{\mathbf{h}}_{m,i}(t) + \sqrt{\frac{1}{1+K}}\hat{\mathbf{h}}_{m,i}(t).$$

The first term $\bar{\mathbf{h}}_{m,i}(t)$ accounts for the LOS component, and the second term $\hat{\mathbf{h}}_{m,i}(t)$ denotes the NLOS component. The Rician factor K sets different weights for the LOS and NLOS components. Similarly, we can define $\mathbf{g}_i(t)$ as the channel vector from UAV-i to the RBS.

3.2. GU Access Control and Mode Selection

Given the UAVs' hovering locations in sensing sub-slot τ_s, there may be multiple GUs under the same UAV's coverage. Some GUs may have worse channel conditions, and thus the data rate for information uploading can be low. This implies that the UAV has to design an access-control strategy to improve the energy efficiency for uplink information transmission. Let $\mathcal{M}_i(t) \subset \mathcal{M}$ denote the set of all GUs under UAV-i's coverage. Let $\mathcal{M}_i^a(t) \subset \mathcal{M}_i(t)$ denote the set of GUs that are allowed to upload sensing information to UAV-i. The other part of the GUs in set $\mathcal{M}_i(t) \setminus \mathcal{M}_i^a(t)$ suspend their data transmission in the current time slot due to insufficient energy or undesirable channel conditions. They can resume data transmission when their channel conditions improve. Let $x_{m,i}(t) \in \{0,1\}$ denote the access-control strategy of GU-m to UAV-i in the t-th time slot, i.e., $\mathcal{M}_i^a(t) = \{m \in \mathcal{M}_i(t) : x_{m,i}(t) = 1\}$. We require $\sum_{i=1}^{N} x_{m,i}(t) \leq 1$ to ensure that GU-m only accesses to one UAV in each time slot.

For all GU-m in the set $\mathcal{M}_i^a(t)$, we consider a time division protocol to upload their sensing data. In particular, sensing sub-slot τ_s can be further divided into $|\mathcal{M}_i^a(t)|$ mini-slots with equal length τ_z. Each mini-slot is assigned to one GU in $\mathcal{M}_i^a(t)$ and can be used for RF active transmission or low-power backscatter communications. For active RF transmission, the received signal at UAV-i can be denoted as $\boldsymbol{\varrho}_{m,i}^a(t) = \sqrt{p_m}\mathbf{h}_{m,i}(t)\varpi_m(t) + \mathbf{v}_0$, where p_m is GU-m's transmission power, $\varpi_m(t)$ is the information symbol with unit power, and $\mathbf{v}_0$ denotes the noise signal. Then, the data rate in RF communications is given by

$$r_{m,i}^a = \tau_z \log_2\left(1 + p_m|\mathbf{h}_{m,i}|^2\right), \tag{2}$$

where we assume a normalized noise power. In passive data uploading, GU-m relies on UAV-i's beamforming signals to backscatter its own information symbols [34]. Let $\mathbf{u}_{m,i}(t) = \sqrt{p_i^A}\mathbf{w}_{m,i}s$ denote UAV-i's beamforming signals in the t-th mini-slot, where $\mathbf{w}_{m,i}$ denotes the normalized beamforming vector for GU-m, p_i^A denotes the fixed transmission power of UAV-i, and s is a random symbol with unit power. After GU-m's backscattering, the data rate in passive transmission can be approximated as follows:

$$r_{m,i}^b = \tau_z \log_2\left(1 + p_i^A|\Gamma_o|^2||\mathbf{h}_{m,i}||^2|\mathbf{h}_{m,i}^H\mathbf{w}_{m,i}|^2\right),$$

where Γ_o is an antenna-specific constant coefficient [35]. For simplicity, we assume that UAV-i uses the maximum ratio combining (MRC) scheme when detecting GU-m's information. Hence, we have $\mathbf{w}_{m,i} = \mathbf{h}_{m,i}/||\mathbf{h}_{m,i}||$, and then we can simplify $r^b_{m,i}$ as follows:

$$r^b_{m,i} = \tau_z \log_2\Big(1 + p^A_i |\Gamma_o|^2 ||\mathbf{h}_{m,i}||^4\Big). \tag{3}$$

Similar to [35], we allow each GU to optimally select its transmission mode based on the energy status and channel conditions. Let $z_m(t) \in \{0, 1\}$ denote GU-m's transmission mode selection in the t-th time slot, i.e., GU-m chooses backscatter communication when $z_m(t) = 0$ and switch to RF active communication when $z_m(t) = 1$. Let $s_{m,i}(t)$ denote the size of sensing data uploaded from GU-m to UAV-i in mini-slot τ_z, which can be evaluated as follows:

$$s_{m,i}(t) = z_m(t) r^a_{m,i}(t) + \Big(1 - z_m(t)\Big) r^b_{m,i}(t).$$

3.3. UAV Transmission Scheduling and Buffer Dynamics

In the reporting phase, we use $y_i(t) \in \{0, 1\}$ to indicate whether UAV-i is scheduled to forward its data to the RBS. To avoid interference among UAVs, we require $\sum_{i=1}^{N} y_i(t) \leq 1$ to ensure that only one UAV can be scheduled to transmit its data in each time slot. Hence, we expect a dynamic update of each UAV's data buffer over different time slots. Let $A_m(t)$ denote the size of sensing data arriving at GU-m at the beginning of the t-th time slot. For each GU-m, we assume that $A_m(t) \in [A_{m,\min}, A_{m,\max}]$ is independent and identically distributed (i.i.d) with mean value λ_m. Let $(\zeta_m(t), Q_i(t))$ denote the sizes of remaining data in GU-m's and UAV-i's buffers, respectively, which can be updated as follows:

$$\zeta_m(t+1) = \left[\zeta_m(t) - \sum_{i \in \mathcal{N}} x_{m,i}(t) s_{m,i}(t) + A_m(t)\right]^+, \tag{4}$$

$$Q_i(t+1) = \left[Q_i(t) + \sum_{m \in \mathcal{M}^a_i(t)} s_{m,i}(t) - y_i(t) O_i(t)\right]^+, \tag{5}$$

where $[X]^+ \triangleq \max\{0, X\}$, and $O_i(t)$ denotes the size of sensing data forwarded to the RBS when UAV-i is allowed to transmit in the t-th time slot, i.e., $y_i(t) = 1$.

$$O_i(t) = \tau_d \log\Big(1 + p_{i,r}(t) ||\mathbf{g}_i||^2\Big), \tag{6}$$

where $p_{i,r}(t)$ is UAV-i's transmission power in the t-th time slot. It is clear that $O_i(t)$ depends on the distance $d_{i,0}(t)$ and the channel condition $\mathbf{g}_i$ between UAV-i and the RBS.

4. Learning for Energy-Efficiency Maximization

We aim to maximize the energy efficiency of the UAV-assisted sensing network by jointly optimizing UAV trajectory, access-control, and transmission-scheduling strategies, as well as GU mode-selection strategies. The overall energy consumption in each time slot includes UAV operation energy consumption during flying and hovering and UAV RF energy consumption during sensing and reporting. For simplicity, we assume that UAV operation energy consumption $e_{i,o}$ is a constant that depends on the overall length of time of flying and hovering. UAV RF energy consumption in sensing $e_{i,s}(t)$ depends on UAV signal beamforming in different mini-slots. Given a fixed beamforming power p^A_i, UAV RF energy consumption can be evaluated as follows: $e_{i,s}(t) = \sum_{m \in \mathcal{M}^a_i(t)} p^A_i \tau_z (1 - z_m(t))$, where τ_z is the fixed length of each mini-slot. Note that $e_{i,s}(t)$ relates to the GUs' access-control strategy $\mathcal{M}^a_i(t)$ and model selection $\{z_m\}_{m \in \mathcal{M}^a_i(t)}$ in each time slot. For example, when the GUs have insufficient energy supply and rely on backscatter communications more often, the UAVs have to consume more energy for signal beamforming. When a larger set of GUs are

allowed to upload their sensing data, this depletes the GUs' energy faster, especially for those GUs with the worst channel conditions. UAV RF energy consumption in reporting $e_{i,r}(t) = y_i(t)p_{i,r}\tau_d$ can be simply modeled as a linear function of the data transmission time τ_d and UAV transmission power $p_{i,r}(t)$ when $y_i(t) = 1$.

When GU-m is associated with UAV-i, its active RF communication relies on the energy harvested from UAV-i. Let $E_m^h(t)$ denote the energy harvested by GU-m in the t-th time slot. Considering a linear energy harvesting model, the harvested energy $E_m^h(t)$ can be estimated as follows:

$$E_m^h(t) = \sum_{n \in \mathcal{M}_i^a(t), n \neq m} \mu p_i^A \tau_z \Big(1 - z_n(t)\Big) \mathbb{E}\Big[|\mathbf{h}_{m,i}^H \mathbf{w}_{n,i}(t)|^2\Big], \tag{7}$$

where μ is the energy conversion efficiency. Note that the energy harvesting model in (7) can be easily extended to a more practical nonlinear model. When some other GU-n is backscattering its information to UAV-i, i.e., $z_n(t) = 0$, GU-m can harvest RF power from UAV-i's beamforming signal $\mathbf{u}_{n,i}(t) = \sqrt{p_i^A}\mathbf{w}_{n,i}(t)s$. Therefore, we have the following energy budget constraint:

$$z_m(t)p_m\tau_z \leq \min\Big\{E_m(t) + E_m^h(t), E_m^{\max}\Big\}, \tag{8}$$

where $E_m(t)$ denotes the energy status at the beginning of the t-th time slot and $E_m^{\max}$ is the maximum battery capacity.

Up to this point, we have defined the energy efficiency Ξ as the time-averaged ratio between the overall throughput received by the RBS and the UAVs' energy consumption:

$$\Xi \triangleq \lim_{|\mathcal{T}|\to\infty} \frac{1}{|\mathcal{T}|} \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{N}} \frac{y_i(t)O_i(t)}{e_{i,o} + e_{i,s}(t) + e_{i,r}(t)}, \tag{9}$$

which depends on the GUs' access and transmission-control strategies as well as the UAV trajectory-planning and scheduling strategies. Let $\mathbf{z} = \{z_m(t)\}_{m\in\mathcal{M},t\in\mathcal{T}}$ denote the GUs' transmission mode-selection strategy. Let $\mathbf{X} = \{x_{m,i}(t)\}_{m\in\mathcal{M},i\in\mathcal{N},t\in\mathcal{T}}$ denote the GUs' access-control strategy. Let $\boldsymbol{\mathcal{L}} = \{\boldsymbol{\mathcal{L}}_i\}_{i\in\mathcal{N}}$ and $\mathbf{y} = \{y_i(t)\}_{i\in\mathcal{N},t\in\mathcal{T}}$ denote UAV trajectory-planning and transmission-scheduling strategies, respectively. Therefore, we can formulate the energy efficiency maximization problem as follows:

$$\max_{\mathbf{z},\mathbf{X},\mathbf{y},\boldsymbol{\mathcal{L}}} \Xi(\mathbf{z},\mathbf{X},\mathbf{y},\boldsymbol{\mathcal{L}}) \quad s.t. \quad (1)\text{–}(8). \tag{10}$$

For simplicity, we consider a fixed beamforming strategy in Problem (10). Thus, the uplink transmission rate and downlink energy transfer to each GU only depend on the channel conditions. The inequalities in (1) define the UAVs' feasible trajectory-planning strategies. The equalities in (2) and (3) denote the uplink data rates in different transmission modes. The constraints in (4)–(6) describe the buffer dynamics of both UAVs and GUs. The constraints in (7) and (8) ensure sustainable operation of the sensing network. Practically, UAV operation energy consumption $e_{i,o}$ is much larger than the sensing power $e_{i,s}(t)$ and the reporting power $e_{i,r}(t)$, which can be ignored in objective (9).

Problem (10) is a combinatorial optimization problem and is difficult to solve optimally. To simplify this problem, we reformulate it into a Markov decision process (MDP), which can adapt the GU access-control and mode-selection strategies as well as the UAV trajectory and scheduling strategies based on continuous interaction with the network environment. Considering that each UAV needs to make decisions independently, we regard each UAV as a decision-making agent and leverage the multi-agent DRL (MADRL) algorithm to solve it. MADRL can effectively coordinate the interactions among multiple agents with large state and action spaces by using a centralized training and decentralized execution scheme [36]. It is built on multiple pairs of actor and critic networks designed for different agents, i.e.,

the UAVs in this paper, as shown in Figure 2. During the training phase, each critic network needs not only the local observation and action but also the actions of all other agents. This requires information exchange among all UAVs. In online learning, each UAV's actor network generates its own actions based on local observations, which enables decentralized implementation.

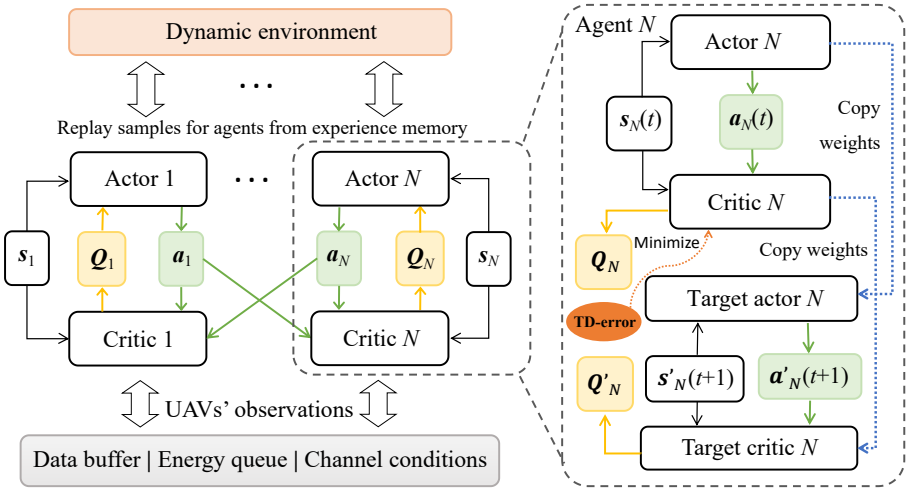


Figure 2. MADDPG framework for UAV centralized training and decentralized execution.

We denote the UAVs' state in each time slot as $\mathbf{s}_t = (\mathbf{s}_1(t), \mathbf{s}_2(t), \ldots, \mathbf{s}_N(t))$, which includes all UAVs' energy storage and data buffers and the channel conditions in the network. Let $\boldsymbol{\chi}_i = (\mathbf{E}_i, \boldsymbol{\zeta}_i, Q_i)$ denote UAV-i's local state. The vector $\mathbf{E}_i$ represents the energy states of UAV-i and all GUs in the set $\mathcal{M}_i^a$ under its coverage. The vector $\boldsymbol{\zeta}_i$ denotes the GU buffer states for all GUs in the set $\mathcal{M}_i^a$, and Q_i represents UAV-i's buffer state, as shown in (4) and (5), respectively. UAV-i's channel conditions are denoted as $\boldsymbol{\psi}_i = (\mathbf{h}_i, \mathbf{g}_i)$, where $\mathbf{h}_i$ denotes the GU-UAV channels for uplink information transmission and $\mathbf{g}_i$ is the channel vector from UAV-i to the RBS. Hence, for each UAV-i, its state can be denoted as $\mathbf{s}_i(t) \triangleq (\boldsymbol{\chi}_i, \boldsymbol{\psi}_i)$. We assume that all the states can be measured at the beginning of each sensing slot. The UAVs' actions in each time slot are defined as the vector $\mathbf{a}_t = (\mathbf{a}_1(t), \mathbf{a}_2(t), \ldots, \mathbf{a}_N(t))$. Motivated by the optimization problem in (10), each UAV-i's action will include the GUs' access control $\mathbf{X}_i = [x_{m,i}(t)]_{m \in \mathcal{M}_i^a}$ and the mode selection $\mathbf{z}_i = [z_m(t)]_{m \in \mathcal{M}_i^a}$, as well as UAV-$i$'s schedule $y_i(t)$ and trajectory $\boldsymbol{\ell}_i(t)$. For simplicity, we denote $\mathbf{a}_i(t) \triangleq (\mathbf{z}_i, \mathbf{X}_i, y_i, \boldsymbol{\ell}_i)$ as the action vector for each UAV.

We denote UAV-i's long-term reward as $R_i = \sum_{t=0}^{T} \varsigma^t r_i(t)$, where $\varsigma \in (0,1)$ is a discounting factor and $r_i(t)$ denotes the instant reward in each time slot. In our problem, we aim to collect all GU sensing data as much as possible and forward them to the RBS with minimum delay. Hence, the throughput includes two parts, i.e., one part denotes the size of uplink data transmission and the second part denotes the size of data forwarded to the RBS. Additionally, a penalty term $r_p(t)$ can be added to the reward to avoid interference and collision between UAVs. As such, we define each UAV-i's reward as follows:

$$r_i(t) = \frac{\sum_{m \in \mathcal{M}_i^a} x_{m,i} s_{m,i} + \gamma y_i O_i}{e_{i,o} + e_{i,s} + e_{i,r}} - \eta r_p(t). \tag{11}$$

We omit the time index in (11) for notational convenience. The constant weight parameter γ puts different priority on the throughput in two parts. The penalty term is defined as $r_p(t) = \sum_{j \in \mathcal{N}, j \neq i} \mathbf{I}(d_{i,j}(t) < d_{\min})$, where $\mathbf{I}(\cdot)$ is an indicator function and η can be a large positive value to avoid collision.

After defining the state, action, and reward for each DRL agent, we can proceed to train the actor and critic networks in the MADDPG framework [36], which implements

the DDPG algorithm for each agent. Let $\boldsymbol{\mu}_i(\mathbf{s}_i, \mathbf{a}_i|\boldsymbol{\theta}_i)$ denote UAV-i's policy, parameterized by the deep neural network (DNN) (i.e., the actor network) with weight parameter $\boldsymbol{\theta}_i$. A delayed copy of the actor network with weight parameter $\boldsymbol{\theta}_i'$ is also maintained to ensure smooth learning. Similarly, there are also two sets of critic networks with parameters $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_i'$ to estimate the Q-values for each state–action pair. Both $\boldsymbol{\theta}_i'$ and $\boldsymbol{\omega}_i'$ are delayed copies of $\boldsymbol{\theta}_i$ and $\boldsymbol{\omega}_i$, respectively. We denote $\boldsymbol{\theta}_t = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_N)$ and $\boldsymbol{\omega}_t = (\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \ldots, \boldsymbol{\omega}_N)$ as the sets of DNN parameters for all UAV actor and critic networks, respectively. It is clear that each agent-i's expected reward $J_i(\boldsymbol{\mu}_i) \triangleq \mathbb{E}_{\boldsymbol{\mu}_i}[R_i]$ becomes a function of the weight parameters $(\boldsymbol{\theta}_i, \boldsymbol{\omega}_i)$. By the policy gradient theorem [37], the maximum reward can be evaluated based on the gradient in terms of $\boldsymbol{\theta}_i$:

$$\nabla_{\boldsymbol{\theta}_i} J(\boldsymbol{\mu}_i) = \nabla_{\boldsymbol{\theta}_i} \boldsymbol{\mu}_i(\mathbf{s}_i, \mathbf{a}_i|\boldsymbol{\theta}_i) \nabla_{\mathbf{a}_i} Q_i^{\boldsymbol{\mu}_i}(\mathbf{s}_t, \mathbf{a}_t|\boldsymbol{\omega}_i). \tag{12}$$

Note that agent-i's policy $\boldsymbol{\mu}_i(\mathbf{s}_i, \mathbf{a}_i|\boldsymbol{\theta}_i)$ depends on the local state $\mathbf{s}_i$, while its Q-value estimation $Q_i^{\boldsymbol{\mu}_i}(\mathbf{s}_t, \mathbf{a}_t|\boldsymbol{\omega}_i)$ following the current policy $\boldsymbol{\mu}_i$ relates to all UAV actions and states $(\mathbf{s}_t, \mathbf{a}_t)$. The update to the critic network's DNN parameter $\boldsymbol{\omega}_i$ also follows a gradient descent approach to minimize the squared error between the Q-value estimation and the one-step look-ahead target Q-value:

$$\min \mathbb{E}\left[|Q_i^{\boldsymbol{\mu}_i}(\mathbf{s}_t, \mathbf{a}_t|\boldsymbol{\omega}_i) - v_i|^2 \right] \tag{13}$$

where the target Q-value is given by $v_i = r_i(t) + \gamma Q_i^{\boldsymbol{\mu}_i'}(\mathbf{s}_t', \mathbf{a}_t'|\boldsymbol{\omega}_i')$ and $\boldsymbol{\mu}_i'$ denotes the target actor network. The complete solution procedure is shown in Algorithm 1. After centralized training, each UAV-i can follow its actor network to generate its action $\mathbf{a}_i = \boldsymbol{\mu}_i(\mathbf{s}_i|\boldsymbol{\theta}_i) + \mathbf{n}_o$, where $\mathbf{n}_o$ denotes random noise to trade-off between exploitation and exploration. The action includes all UAVs' trajectory and scheduling decisions $(\boldsymbol{\ell}_i, y_i)$ as well as the access-control and mode-selection decisions $(\mathbf{X}_i, \mathbf{z}_i)$ for all GUs under its coverage. Then, all GUs follow UAV-i's decisions $(\mathbf{X}_i, \mathbf{z}_i)$ to upload their sensing data, as shown in lines 12–16 of Algorithm 1. After sensing, UAV-i either forwards its data to the RBS or holds on until the next time slot, depending on scheduling decision y_i, as shown in lines 17–21 of Algorithm 1. When all UAVs and GUs have updated their actions, the overall reward is evaluated and used to drive the update of actor networks.

The above MADDPG algorithm provides a general solution framework for high-dimensional optimization problems, as shown in (10). However, it is still challenging to deploy in practice due to the requirement for information exchange and large-scale training. The MADDPG algorithm relies on a centralized training and decentralized execution scheme that requires each UAV to report its local system observation to the RBS, including the channel conditions, energy status, and the offloading decisions of the GUs under its coverage. With a large number of UAVs and GUs, the state and action spaces in the MADDPG algorithm will increase drastically. The speed of convergence will slow due to the high-dimensional state and action spaces in a multi-UAV-assisted sensing network. The cost of the information exchange also becomes significant as the number of UAVs increases. Information collection from a large set of UAVs inevitably suffers from excessive delays and slows the training process.

Algorithm 1 MADDPG for multi-UAV trajectory planning, transmission scheduling, access control, and mode selection

Initialize all data buffer and energy storage queues
Initialize channel conditions and UAVs' observations
%**UAVs' trajectory planning**
for each UAV $i \in \mathcal{N}$ **do**
    Collect state information $\mathbf{s}_i$
    Generate the action $\mathbf{a}_i = \boldsymbol{\mu}_i(\mathbf{s}_i|\boldsymbol{\theta}_i) + \mathbf{n}_o$
    Update the UAV's trajectory point $\boldsymbol{\ell}_i$
    Update the UAV's scheduling decision y_i
    Update the GUs' access-control decision $\mathbf{X}_i$
    Update the GUs' mode-selection decision $\mathbf{z}_i$
    Distribute $\mathbf{X}_i$ and $\mathbf{z}_i$ to GUs in the set $\mathcal{M}_i^a$
    %**UAV access control**
    for each GU $m \in \mathcal{M}_i^a$ **do**
        Transmit in RF mode if $z_{m,i} = 1$, otherwise transmit in backscatter mode
        Update GU-m's data queue and energy states
    end for
    %**UAVs' reporting schedule**
    if UAV-i is scheduled with $y_i = 1$ **do**
        Forward buffered data with size O_i to RBS
        Update the UAV's data queue
    end for
    Evaluate the UAV's reward function $r_i(t)$
end for
Evaluate all UAVs' rewards
Update the target actor and critic networks
Loop back to step (3)

5. A Hierarchical Learning Approach

In this part, we intend to improve the learning efficiency and performance of the conventional MADDPG algorithm by designing a hierarchical framework to reduce the state and action spaces and to avoid frequent information exchange among GUs, UAVs, and RBS. Note that a multi-UAV-assisted wireless network naturally has a hierarchical structure. The RBS is the information receiver and the coordinator of all UAVs. For more efficient sensing, the RBS can deploy and dispatch different UAVs to collaboratively accomplish sensing over a large geographical area. Due to channel fading, the UAVs may have little interference with each other when they are separated in different service regions and aiming to collect sensing information from different sets of GUs. As such, each UAV only cares about its own GUs under its coverage. This implies that the UAV's local decisions, e.g., the beamforming and scheduling strategy, only affect the local GUs' access-control and mode-selection strategies. When the UAVs are far apart, they can be viewed as independent devices making their own decisions based on local observations.

5.1. Hierarchical Multi-Agent Learning Framework

The above observations motivate us to design a hierarchical learning framework to decompose Problem (10) into two sub-problems that can be solved individually and iteratively. The overall learning algorithm includes the upper-layer learning loop for UAV trajectory planning and the lower-layer learning loop for GU access control. Each layer only focuses on a part of the control variables with reduced dimensionality. In particular, we employ the MADDPG algorithm for the UAVs to update their trajectory and transmission-scheduling strategies. Then, given the UAVs' upper-layer decisions, we further employ the DQN algorithm for each UAV to update the GUs' access-control strategies under its coverage. As illustrated in Figure 3, each UAV is viewed as an independent agent in the upper-layer learning framework. The centralized training phase of the MADDPG

algorithm can be performed by the RBS. After that, each UAV updates its trajectory and beamforming strategy in a distributed manner by individual actor networks based on local observations. When the UAVs move to their new trajectory points, each UAV can collect the status information of the underlying GUs without reporting such information to the RBS. Based on the UAVs' local observations, each UAV can further decide the GUs' access-control and mode-selection strategies by the lower-layer DQN method. Then, each GU can follow the UAV's decision to upload its data to the UAV.

The upper-layer and lower-layer state spaces in each time slot are denoted as $\mathcal{S}^o$ and $\mathcal{S}^c$, separately. Correspondingly, the upper-layer and lower-layer action spaces of each UAV are given by $\mathcal{A}^o$ and $\mathcal{A}^c$, respectively. The state spaces include the environmental information that can be used to learn the UAVs' upper-layer and lower-layer actions. The upper-layer state $\chi_i^o \in \mathcal{S}^o$ includes each UAV's channel information $\mathbf{g}_i$, energy status $\mathbf{E}_i^o$, and buffer size Q_i following the dynamics in (5). Thus, we denote it as $\chi_i^o = (\mathbf{g}_i, \mathbf{E}_i^o, Q_i)$. Similarly, the lower-layer state $\chi_i^c \in \mathcal{S}^c$ includes all information of the GUs in the set $\mathcal{M}_i^a$ under the UAV's coverage, including all GUs' energy statuses $\mathbf{E}_i^c$, the GU-UAV channels $\mathbf{h}_i$ for uplink information transmission, and the buffer size $\boldsymbol{\zeta}_i$ following the dynamics in (4). Thus, we denote it as $\chi_i^c = (\mathbf{h}_i, \mathbf{E}_i^c, \boldsymbol{\zeta}_i)$. The system reward is determined by the state $\{\mathcal{S}^o, \mathcal{S}^c\}$ and joint action by $\{\mathcal{A}^o, \mathcal{A}^c\}$. As such, we can define the hierarchical learning framework by the information tuple $(\{\mathcal{S}^o, \mathcal{S}^c\}, \{\mathcal{A}^o, \mathcal{A}^c\}, \{\mathcal{R}^o, \mathcal{R}^c\})$. Correspondingly, $\{\mathcal{R}^o, \mathcal{R}^c\}$ denotes the reward functions for the upper- and lower-layer agents. In the sequel, we explain the two parts of the algorithm design.

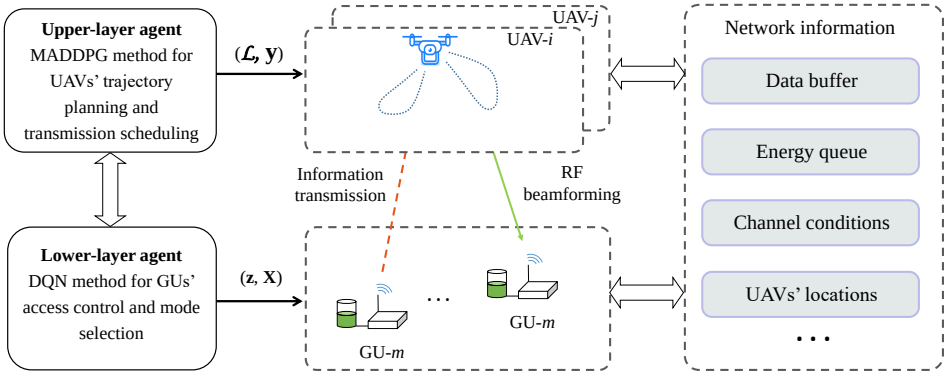


Figure 3. Illustration of the hierarchical framework.

5.2. Upper-Layer MADDPG for Trajectory Planning and Scheduling

UAV trajectory-planning and transmission-scheduling strategies can be updated by the MADDPG algorithm according to the UAVs' local states, including the GUs' data demands, the UAVs' energy status, and the channel conditions with the RBS. The RBS can collect all UAVs' state information and carry out centralized training in the offline phase. For each UAV, the MADDPG algorithm maintains individual actor and critic networks, which have to be trained jointly during the offline phase. After centralized training, the RBS disseminates the actor networks to individual UAVs and allows them to make trajectory-planning and scheduling decisions $\mathcal{A}^o$ in a distributed manner according to local observations. Then, each UAV hovers at a specific location in the next time slot to collect sensing information from a subset of GUs. The upper-layer action space can be expressed as $\mathcal{A}^o = \{\boldsymbol{\mathcal{L}}_i, \mathbf{y}_i\}_{i \in \mathcal{N}}$, which includes the UAV's trajectory-planning $\boldsymbol{\mathcal{L}}_i$ and transmission-scheduling $\mathbf{y}_i$ decisions in the next time slot. When the UAV-to-GU distance is less than a threshold, the GU's uplink signals can be successfully decoded by the UAV, and thus, the GU is considered to be covered by the UAV. It is clear that UAV-i's reward $\mathcal{R}_i^o$ firstly relates to the amount of sensing data received from the GUs under its coverage, which is determined by the lower-layer access-control decision. Let $\mathcal{R}_i^c$ denote the lower-layer sensing reward, which characterizes the amount of sensing data and the resource consumption during the UAV's sensing phase. The detailed expression of $\mathcal{R}_i^c$ is defined in

the next subsection. Secondly, UAV-i's reward $\mathcal{R}_i^o$ also includes the transmission reward, which depends on the UAV's buffer size and the distance to the RBS. A positive reward is accrued $y_i O_i$ when UAV-i is scheduled to forward its buffered information to the RBS. Additionally, UAV-i's reward $\mathcal{R}_i^o$ has to punish any potential collision with the other UAVs to ensure safety. Similar to (11) for the conventional MADDPG algorithm, the penalty term is defined as $r_p(t) = \sum_{j \in \mathcal{N}, j \neq i} \mathbf{I}(d_{i,j}(t) < d_{\min})$. As such, we can define the UAV's reward in the upper-layer learning framework as follows:

$$\mathcal{R}_i^o = \mathcal{R}_i^c + y_i O_i - r_p. \tag{14}$$

Let π_i^o denote UAV-i's trajectory-planning and transmission-scheduling policy and $\boldsymbol{\pi}_{-i}^o$ denote the other UAVs' joint policies in the upper-layer learning phase. The long-term expected reward of all UAVs in the upper-layer learning for trajectory and scheduling strategies can be defined as follows:

$$V^o(\pi_i^o, \boldsymbol{\pi}_{-i}^o) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}_i^o(\mathbf{s}_t^o, \mathbf{a}_t^o)\right], \tag{15}$$

where the UAVs' joint states and actions are denoted as $\mathbf{s}_t^o = [\boldsymbol{\chi}_1^o(t), \boldsymbol{\chi}_2^o(t), ..., \boldsymbol{\chi}_N^o(t)]$, and $\mathbf{a}_t^o = [\mathbf{a}_1^o(t), \mathbf{a}_2^o(t), ..., \mathbf{a}_N^o(t)]$, respectively. Each UAV's policy π_i^o determines its own action given different state, i.e., $\mathbf{a}_i^o(t) = \pi_i(\boldsymbol{\chi}_i^o(t))$. However, its Q-value estimation $Q_i^{\pi_i^o, \boldsymbol{\pi}_{-i}^o}(\mathbf{s}_t^o, \mathbf{a}_t^o)$ has to be trained in a centralized manner by the MADDPG algorithm deployed in the RBS, relying on the information collection from all UAVs. By jointly adapting all UAV actions, the value function in (15) can be improved gradually and stabilizes at the convergence. During online execution, each UAV-i follows its own policy π_i^o to generate localized trajectory-planning and transmission-scheduling action $\mathbf{a}_i^o(t) = (\mathcal{L}_i(t), \mathbf{y}_i(t))$ based on UAV-i's local observation $\boldsymbol{\chi}_i^o(t)$.

5.3. Lower-Layer DQN for GU Access Control and Mode Selection

Given the upper-layer's trajectory-planning decisions, each UAV is given a new hovering position for information sensing in the next time slot. Thus, in the next step, each UAV updates the access-control decision for the GUs under its coverage. UAV-i's lower-layer action can be defined as $\mathbf{a}_i^c(t) = (\mathbf{z}_i(t), \mathbf{X}_i(t)) \in \mathcal{A}^c$, including each GU's access-control and mode-selection decisions in the next time slot. Considering the combinatorial nature of the discrete action space $\mathcal{A}^c$, we can resort to the classic DQN method to update each UAV's lower-layer action $\mathbf{a}_i^c(t)$. According to the local information regarding the GUs, each UAV can adapt its access-control and mode-selection strategy for uplink data transmission to improve the total reward perceived by the UAV. To improve the sensing efficiency, we can define the UAV's instant reward $\mathcal{R}_i^c(t)$ in the lower-layer learning framework as a weighted combination of the sensing throughput and the energy consumption as follows:

$$\mathcal{R}_i^c(t) = \sum_{m \in \mathcal{M}_i^a} x_{m,i} s_{m,i} - \eta_1 E_m^h, \tag{16}$$

where η_1 is the trade-off parameter for the sensing throughput and the energy consumption. Thus, UAV-i's long-term reward in the lower-layer learning can be denoted as follows:

$$V_i^c(\pi_i^c) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}_i^c(\boldsymbol{\chi}_i^c(t), \mathbf{a}_i^c(t))\right],$$

which only relates to UAV-i's local observations $\boldsymbol{\chi}_i^c = (\mathbf{h}_i, \mathbf{E}_i^c, \boldsymbol{\zeta}_i)$ due to the spatial separation of different UAVs. Therefore, each UAV can individually adapt the GU access-control and mode-selection policy π_i^c to improve and stabilize the value function $V_i^c(\pi_i^c)$.

Given any state–action pair $(\mathbf{s}_t^c, \mathbf{a}_t^c)$, the DQN method deployed in each UAV-i estimates its value function $V_i^c(\pi_i^c)$ or the variant Q-function $Q_i^c(\boldsymbol{\chi}_i^c(t), \mathbf{a}_i^c(t)|\boldsymbol{\omega}_t)$ by two sets of

DNNs with weight parameters ω_t and ω'_t, respectively. To stabilize the learning, the weight parameter ω'_t of the target Q network is copied from the online Q network ω_t regularly every few steps. Hence, the target Q-value can be estimated as follows:

$$y_i^c(t) = \mathcal{R}_i^c(t) + \gamma Q_i^c(\chi_i^c(t+1), \mathbf{a}_i^c(t+1)|\omega'_t), \tag{17}$$

where the new action $\mathbf{a}_i^c(t+1)$ is obtained from the online Q network with the parameter ω_t given the state transition to $\chi_i^c(t+1)$, i.e., $\mathbf{a}_i^c(t+1) \triangleq \arg\max_{\mathbf{a}_i^c \in \mathcal{A}_i^c} Q^c(\chi_i^c(t+1), \mathbf{a}_i^c|\omega_t)$. Then, the update to the DNN parameter ω_t is performed by the gradient descent method to minimize the mean square error between $Q_i^c(\chi_i^c(t), \mathbf{a}_i^c(t)|\omega_t)$ and the target value $y_i^c(t)$ in (17), similar to (13). The size of the state and action spaces in each learning layer affects the computational complexity of the algorithm, which depends on the selection of parameters such as size, depth, learning rate, and discount factor. The deep neural networks of the lower-layer DQN method include three fully connected layers and two relu layers. The upper-layer MADDPG includes two sets of DNNs to approximate the Q network and the policy network, respectively. Each DNN in the MADDPG follows the same structure as that of the DQN method. The process is shown in Algorithm 2.

Algorithm 2 Hierarchical learning for multi-UAV trajectory planning, transmission scheduling, and access control

1: Initialize the observations of UAVs and GUs
2: Collect state information $\{\mathcal{S}^o, \mathcal{S}^c\}$
3: %**Upper-layer MADDPG for trajectory learning**
4: **for** each UAV $i \in \mathcal{N}$ **do**
5: Collect UAVs' state information s_i^o
6: Execute the upper-layer action a_i^o
7: %**Lower-layer DQN for transmission learning**
8: Collect GUs' state information $\mathcal{S}_i^c$
9: Execute the lower-layer action $\mathcal{A}_i^c$
10: **end for**
11: Update the joint action $\{\mathcal{A}^o, [\mathcal{A}_i^c]_{i\in\mathcal{N}}\}$
12: Observe total reward function $\{\mathcal{R}^o, [\mathcal{R}_i^c]_{i\in\mathcal{N}}\}$
13: Update all networks
14: Loop back to Step (3)

6. Numerical Results

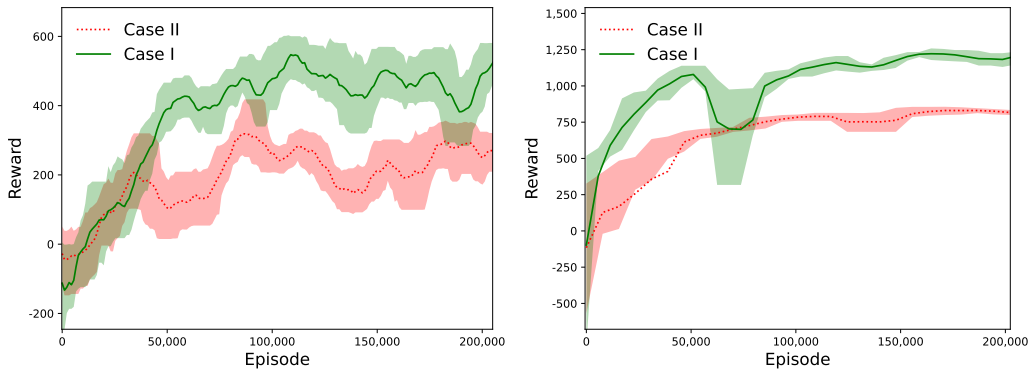
In this section, we evaluate the performance of the MADDPG and the hierarchical learning algorithms. Without loss of generality, we focus on a UAV-assisted wireless sensing network with one RBS and three UAVs assisting with information collection from a group of GUs randomly distributed in a two-dimensional coordinate system scaled to the range $[-1, 1]$. All GUs are far away from the RBS and there are no direct links between the RBS and GUs. More-detailed parameters are listed in Table 1.

Table 1. Parameter settings in the numerical simulations.

Parameter	Setting
Training cycles per episode	30
Path-loss coefficient α	2
Range of GU's data size ζ_m	$[5, 15]$ Mbits
Maximum UAV speed $v_{\max}$	25 m/s
Noise power δ	-90 dBm
ϵ-greedy parameter	0.05
Actor's learning rate	10^{-3}
Critic's learning rate	10^{-4}
Batch size	32
Reward discount	0.95
Memory capacity	2000
Target replace iter	100

6.1. Convergence and Reward Performance

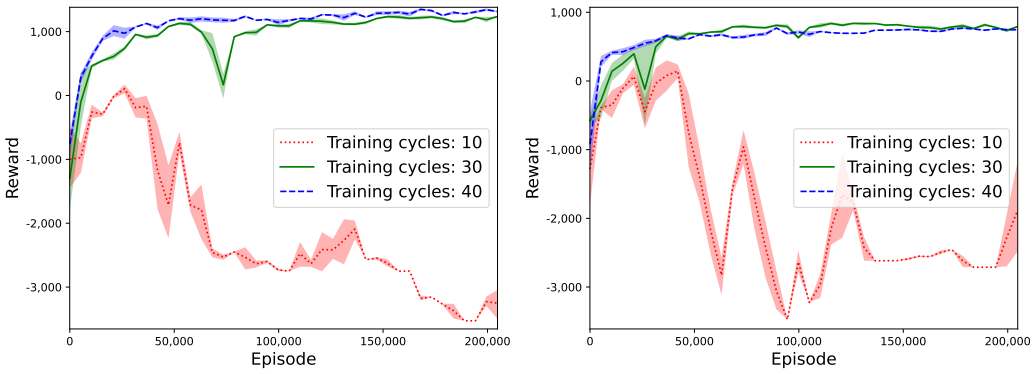
Firstly, we evaluate the learning performance for UAV trajectory planning in the upper-layer learning phase considering two different cases. For Case I, we assume that all UAVs are initially deployed at random locations to serve the GUs. In Case II, the UAVs are assumed to take off from the same dispatch point. The reward dynamics in the conventional MADDPG and the proposed hierarchical learning method for the two cases are compared in Figure 4. By interacting with the environment and adapting UAV trajectories, the reward values in both methods increase and eventually stabilized after a number of iterations, which verifies the effectiveness and convergence of the proposed learning method. An interesting observation is that the reward in Case I is generally higher than that achieved in Case II. This implies that the UAVs' initial dispatch locations are important to the overall sensing performance with the same data traffic distribution of the GUs. When the UAVs are scattered over the service coverage area, it can be faster and more efficient for the UAVs to find preferable sensing locations and trajectories to avoid service overlap and resource conflicts. When all UAVs start from the same location, there always exists some service overlap in the early stage of their trajectories. This implies inefficient cooperation among different UAVs and leads to reduced reward performance.



(**a**) MADDPG method (**b**) Hierarchical learning method

Figure 4. The reward dynamics of two algorithms for two cases.

The convergence properties with different training cycles are shown in Figure 5. It reveals that a shorter training step has difficultly achieving convergence, as shown by the red dotted lines. The convergence results with training cycles of 30 and 40 are very similar. Hence, we set the training cycle to 30 in our simulations. We also test different combinations of learning parameters, including the learning rate, mini-batch size, replay buffer size, and discount factor, to help select the best hyperparameters for our experiments.



(**a**) Convergence in Case I (**b**) Convergence in Case II

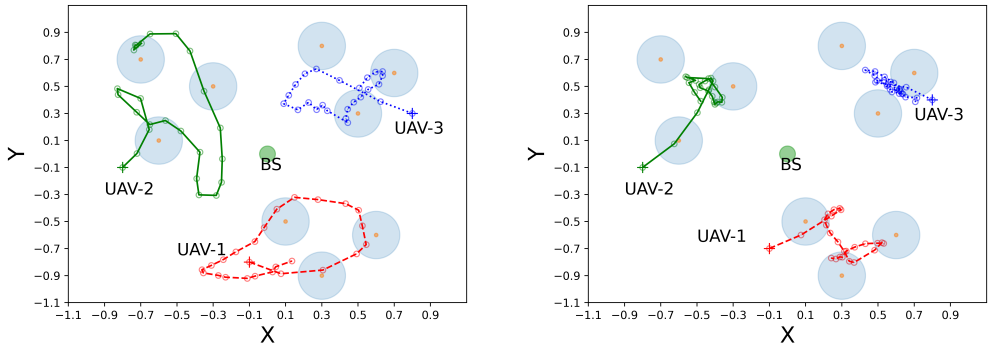
Figure 5. Performance comparison with different training cycles.

Compared with the conventional MADDPG, the reward curve of the hierarchical learning method is more stable and is smoother, as shown in Figure 4b. The hierarchical learning method achieves faster convergence and a much higher reward and stabilizes after 50 k learning episodes, while the MADDPG method still shows obvious fluctuations after 200 k learning episodes. A possible explanation of this observation is that the conventional MADDPG method adapts the high-dimensional control variables simultaneously, including UAV trajectory-planning and transmission-scheduling and GU access-control and mode-selection strategies, while the hierarchical learning method updates a smaller size of decision variables in each learning episode with reduced action space. Note that the UAVs' space separation limits their interference with each other. As such, the hierarchical learning structure can avoid inefficient action combinations from the UAVs and the GUs, therefore reducing the overall action space and improving learning efficiency. Another advantage lies in that the hierarchical learning structure only requires the RBS to have limited communications with the UAVs. GU status information is not necessarily reported to the RBS for efficient trajectory planning and transmission scheduling. This avoids excessive communication overhead and latency in online learning.

6.2. Trajectory Planning in Two Cases

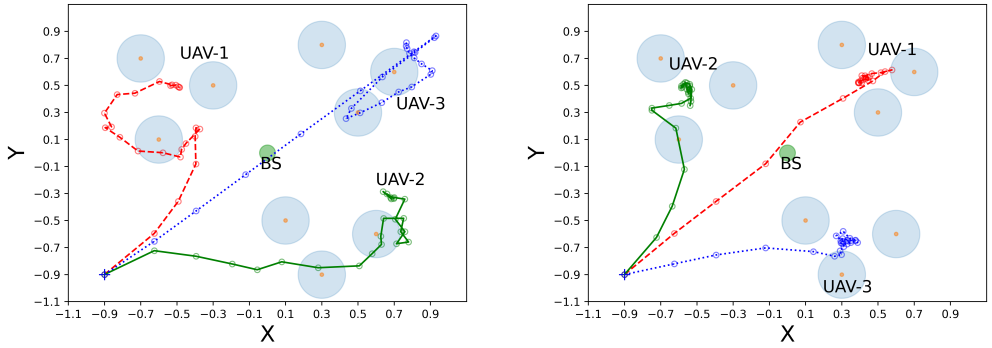
In Figures 6 and 7, we compare UAV trajectory planning in 2D coordinate for two cases with different algorithms. The colored lines in these figures represent UAV trajectories, and the hollow circles represent UAV hovering points on the trajectories during different time slots. We observe that, after training, the UAVs can fly to different service regions without interfering with each other in both the MADDPG and the hierarchical learning algorithms. This shows the task collaboration of different UAVs to cover a large service area. As shown in Figure 6, though UAV trajectories are different with the two planning algorithms, each UAV intends to serve the closest group of GUs starting from the initial location. The hierarchical learning method can be more efficient, as UAV trajectories are confined to small service regions, as shown in Figure 6b, while UAV trajectories in the MADDPG method cover a larger area, as shown in Figure 6a. Similar observations are revealed in Case II, where all UAVs plan their trajectories from the same starting point. Both trajectory-planning algorithms ensure that the UAVs quickly reach their service regions to efficiently explore their task collaboration in a large-scale sensing network. For Case II, the trajectory comparison between two planning algorithms also verifies that the proposed hierarchical learning method achieves a more compact trajectory for each UAV compared to that of the conventional MADDPG method. This corroborates the more stable and faster learning performance shown in Figure 4. A possible explanation to this observation is that the conventional MADDPG needs to collect the status information from both the UAVs and GUs when making trajectory-planning decisions. The GUs' random task arrivals and channel fluctuations may disturb UAV trajectories and thus create instability during the

learning process. On the contrary, the hierarchical learning method only focuses on UAV status information and reduces the action space in the upper-layer trajectory planning. The GUs' dynamic information is evaluated by individual UAVs and is used to assess the quality of the upper-layer trajectory planning.



(**a**) MADDPG method (**b**) Hierarchical learning method

Figure 6. Case I: trajectory planning from different starting points.



(**a**) MADDPG method (**b**) Hierarchical learning method

Figure 7. Case II: trajectory planning from the same starting point.

6.3. Access Control and Buffer Dynamics

In this part, we evaluate the GU access-control strategies in the hierarchical learning algorithm. Since the access-control strategy is updated by the lower-layer DQN method at each UAV, we can observe the dynamics of the reward function $\mathcal{R}_i^c(t)$ in different time slots, as shown in Figure 8. Taking UAV-i as an example, we show the reward curves in three consecutive time slots for the UAV's data sensing. During these time slots, the distances between UAV-i and the GUs are decreasing. It is clear that UAV-i can achieve a larger reward as it approaches the GUs gradually. One possible explanation is that both the GUs' energy harvesting capabilities and the transmission rates for backscatter communications can be enhanced as the channel conditions between UAV-i and the GUs under its coverage improve. In each time slot, we can observe that UAV-i's access-control strategy results in a gradually increasing reward function $\mathcal{R}_i^c(t)$ until convergence after 30k learning episodes. Even if there is a performance drop in the reward curve, the UAV's learning can quickly resume higher reward performance by adapting its access-control strategy, as shown in the third time slot in Figure 8.

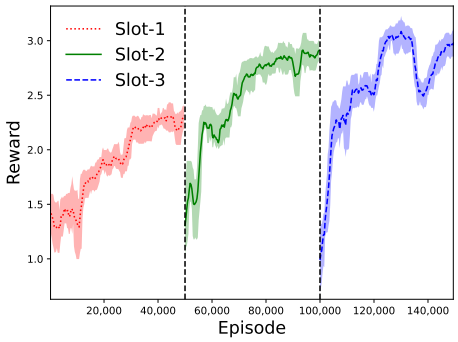
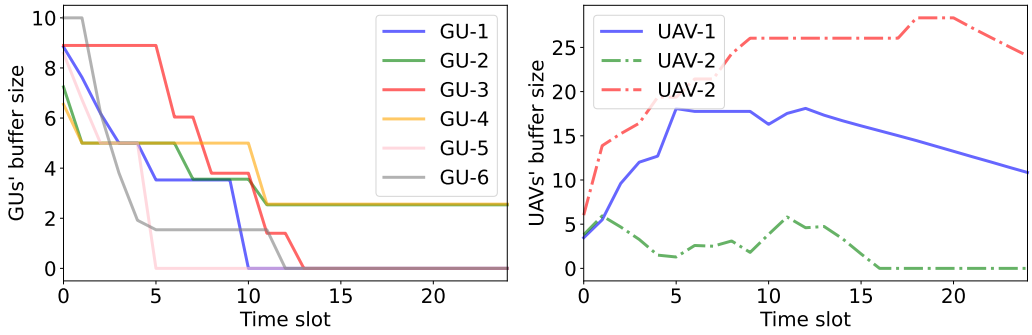


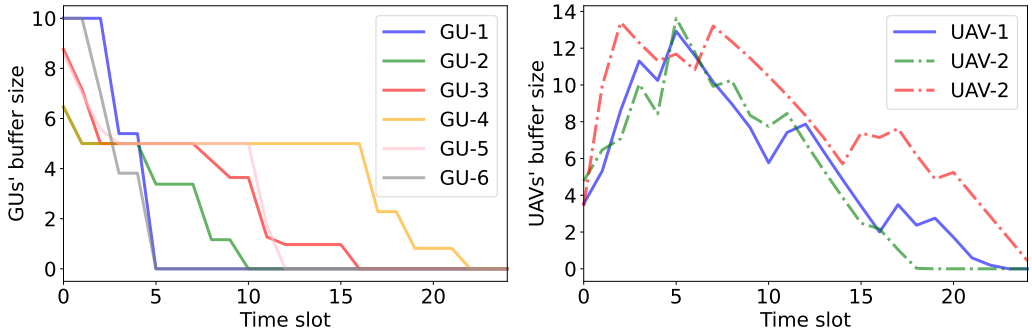
Figure 8. Reward dynamics in the lower-layer DQN algorithm.

We further verify the performance of the UAVs' access-control strategies by examining the GUs' and UAVs' buffer dynamics, as shown in (4) and (5), respectively. A preferable access-control strategy ensures a stable buffer size and fairness among different GUs. For performance comparison, we introduce a single-agent independent DDPG (denoted as iDDPG) that regards each UAV as an independent agent. It allows each UAV to learn its own strategy independently based on its local observations. We apply iDDPG, MADDPG, and the proposed hierarchical learning algorithms to adapt the UAVs' access-control strategies. The UAVs' and the GUs' buffer dynamics with different algorithms are shown in Figures 9–11. In the simulation, we assume that all GUs constantly generate data traffic and the UAVs help forward GU data to the RBS. If the UAVs complete the data collection in advance, a new round of data collection can be carried out. The black dotted lines in Figures 10 and 11 represent the beginning of a new round of data collection.



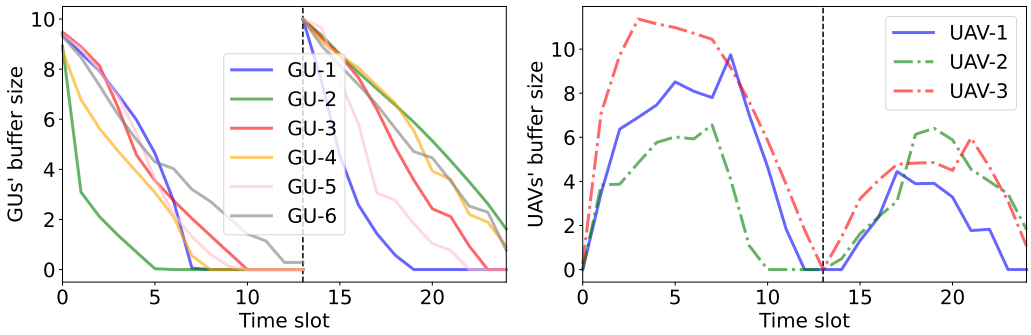
(**a**) GU buffer dynamics (**b**) UAV buffer dynamics

Figure 9. The buffer dynamics in the iDDPG algorithm.



(**a**) GU buffer dynamics (**b**) UAV buffer dynamics

Figure 10. The buffer dynamics in the MADDPG algorithm.

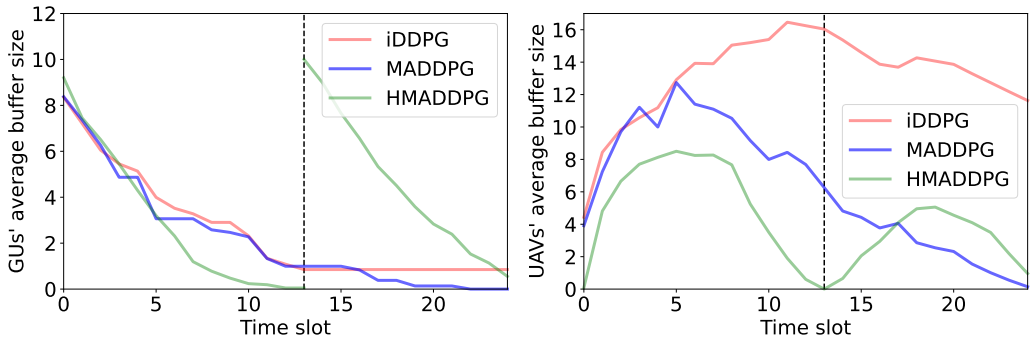


(**a**) GU buffer dynamics (**b**) UAV buffer dynamics

Figure 11. The buffer dynamics in the hierarchical learning algorithm.

Figure 9 shows the dynamics of the GU and UAV buffer states over different time slots by the iDDPG algorithm. In Figure 9a, we can see that some GU data traffic cannot be completely collected by the UAVs and forwarded to the RBS in a timely manner. Hence, the buffer sizes drop slowly and still remain at non-negative values after 25 time slots. Additionally, as shown in Figure 9b, the UAV buffer sizes are unbalanced. This implies that the iDDPG algorithm cannot fully explore the UAVs' task cooperation to maximize the overall energy efficiency. In the MADDPG algorithm, as shown in Figure 10, each UAV has its own service region and collects the GU data traffic more efficiently. By the UAVs' cooperative operation, the GUs can deplete their data buffers faster and resume the next round of data collection, as shown in Figure 10a. Compared with the iDDPG algorithm, the UAVs' data buffers are more balanced in the MADDPG algorithm. This is because the UAVs have different service regions and thus can avoid interfere with each other, as shown in Figure 10b. In the hierarchical learning algorithm, we find that the UAVs can complete data transmission faster, as their data buffers turn into zero and then start a new round of data collection, as shown in Figure 11. Moreover, the data collected by each UAV is well-balanced by the UAVs' collaborative trajectory planning. This reveals that the hierarchical learning algorithm has higher energy efficiency compared to iDDPG and MADDPG.

Figure 12 compares the GUs' and the UAVs' average buffer sizes in the iDDPG, MADDPG, and hierarchical learning (denoted as the HMADDPG) methods. The average buffer size slowly decreases in the iDDPG algorithm. The reason is that the UAVs cannot obtain all of the other UAVs' status information when making trajectory-planning decisions, which may lead to suboptimal deployment locations for the UAVs and degrade the overall energy efficiency of the system. We also observe that the HMADDPG method achieves a faster decrease in the GUs' average buffer size compared with the other methods, as shown in Figure 12a. The UAVs' average buffer size also goes to zero at a much faster rate, as shown in Figure 12b. This implies that the HMADDPG method allows the UAVs to collect more GU sensing data compared to the other baselines by smartly adapting UAV trajectory and access-control strategies.



(**a**) GU buffer dynamics (**b**) UAV buffer dynamics

Figure 12. Average buffer size for three learning algorithms.

7. Conclusions and Future Work

In this paper, we proposed a hierarchical learning algorithm to maximize the sensing capacity of a multi-UAV-assisted sensing network by adapting the GUs' access-control and mode-selection strategies as well as the UAVs' transmission-scheduling and trajectory-planning strategies. Leveraging the distributed nature of the multi-UAV-assisted network, we proposed a hierarchical learning framework that decomposes the control variables into two layers. The upper-layer MADDPG algorithm is employed to adapt the UAV trajectory-planning and scheduling strategies based on UAV status information, while the lower-layer DQN algorithm is proposed to update the GU access-control and mode-selection strategies within each individual UAV's service coverage area. Our numerical results show that the hierarchical learning algorithm can efficiently exploit UAV task cooperation and also improve overall learning efficiency. The distributed and hierarchical learning methods can improve data transmission performance in future UAV-assisted wireless networks. This allows UAVs to quickly adapt to the time-varying channel environment in a large-scale wireless network. However, practically, the hierarchical DRL learning scheme may still require a long time to train the lower-layer DQN for each upper-layer decision epoch. This results in an excessive run-time of the learning algorithm. In the future, we can consider improving the learning efficiency and accelerating the convergence speed of the lower-layer DQN method by integrating model-based local information.

Author Contributions: Conceptualization, X.L., C.Z., C.L., J.X., and S.G.; software, C.C.; validation, X.L., C.C., and S.G.; investigation, C.L., C.L., and J.X.; writing—original draft preparation, C.C., S.G., and J.X.; writing—review and editing, X.L., C.Z., C.L., J.X., and S.G.; supervision, C.Z., C.L., and J.X.; project administration, C.Z. and C.L.; funding acquisition, C.L.; X.L., and C.C. contributed equally to this paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

Unmanned Aerial Vehicle	UAV
Internet of Things	IoT
Ground User	GU
Remote Base Station	RBS
Block Coordinate Descent	BCD
Multi-agent Proximal Policy Optimization	MAPPO
Multi-agent Deep Deterministic Policy Gradient	MADDPG
Federated MADDPG	F-MADDPG
Federated Averaging	FA
Hierarchical Multi-Agent DRL	H-MADRL
Quality-of-Service	QoS
Mixed-Integer Nonlinear Programming	MINLP
Line-of-Sight	LOS
Markov Decision Process	MDP
Deep Neural Network	DNN
Independent DDPG	IDDPG
Non-Orthogonal Multiple Access	NOMA
Rate-Splitting Multiple Access	RSMA

References

1. Lagkas, T.; Argyriou, V.; Bibi, S.; Sarigiannidis, P. UAV IoT framework views and challenges: Towards protecting drones as "Things". *Sensors* **2018**, *18*, 4015. [CrossRef] [PubMed]
2. Gupta, L.; Jain, R.; Vaszkun, G. Survey of important issues in UAV communication networks. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 1123–1152. [CrossRef]
3. Han, R.; Bai, L.; Wen, Y.; Liu, J.; Choi, J.; Zhang, W. UAV-aided backscatter communications: Performance analysis and trajectory optimization. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 3129–3143. [CrossRef]
4. Yang, G.; Dai, R.; Liang, Y.C. Energy-efficient UAV backscatter communication with joint trajectory design and resource optimization. *IEEE Trans. Wirel. Commun.* **2020**, *20*, 926–941. [CrossRef]
5. Zhao, N.; Lu, W.; Sheng, M.; Chen, Y.; Tang, J.; Yu, F.R.; Wong, K.K. UAV-assisted emergency networks in disasters. *IEEE Trans. Wirel. Commun.* **2019**, *26*, 45–51. [CrossRef]
6. Boccardo, P.; Chiabrando, F.; Dutto, F.; Giulio Tonolo, F.; Lingua, A. UAV deployment exercise for mapping purposes: Evaluation of emergency response applications. *Sensors* **2015**, *15*, 15717–15737. [CrossRef]
7. Arafat, M.Y.; Moh, S. Localization and clustering based on swarm intelligence in UAV networks for emergency communications. *IEEE Internet Things J.* **2019**, *6*, 8958–8976. [CrossRef]
8. Hayat, S.; Yanmaz, E.; Muzaffar, R. Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2624–2661. [CrossRef]
9. Ding, G.; Wu, Q.; Zhang, L.; Lin, Y.; Tsiftsis, T.A.; Yao, Y.D. An amateur drone surveillance system based on the cognitive Internet of Things. *IEEE Commun. Mag.* **2018**, *56*, 29–35. [CrossRef]
10. Zhao, C.; Liu, J.; Sheng, M.; Teng, W.; Zheng, Y.; Li, J. Multi-UAV trajectory planning for energy-efficient content coverage: A decentralized learning-based approach. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 3193–3207. [CrossRef]
11. Tran, D.H.; Nguyen, V.D.; Chatzinotas, S.; Vu, T.X.; Ottersten, B. UAV relay-assisted emergency communications in IoT networks: Resource allocation and trajectory optimization. *IEEE Trans. Wirel. Commun.* **2021**, *21*, 1621–1637. [CrossRef]
12. You, C.; Zhang, R. 3D trajectory optimization in Rician fading for UAV-enabled data harvesting. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 3192–3207. [CrossRef]
13. Zhang, X.; Zhang, H.; Du, W.; Long, K.; Nallanathan, A. IRS empowered UAV wireless communication with resource allocation, reflecting design and trajectory optimization. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 7867–7880. [CrossRef]
14. Lakew, D.S.; Tran, A.T.; Dao, N.N.; Cho, S. Intelligent offloading and resource allocation in heterogeneous aerial access IoT networks. *IEEE Internet Things J.* **2022**, *10*, 5704–5718. [CrossRef]
15. Kang, H.; Chang, X.; Mišić, J.; Mišić, V.B.; Fan, J.; Liu, Y. Cooperative UAV Resource Allocation and Task Offloading in Hierarchical Aerial Computing Systems: A MAPPO Based Approach. *IEEE Internet Things J.* **2023**. [CrossRef]
16. Wang, Y.; Gao, Z.; Zhang, J.; Cao, X.; Zheng, D.; Gao, Y.; Ng, D.W.K.; Di Renzo, M. Trajectory Design for UAV-Based Internet of Things Data Collection: A Deep Reinforcement Learning Approach. *IEEE Internet Things J.* **2021**, *9*, 3899–3912. [CrossRef]
17. Wang, M.; Long, Y.; Gong, S.; Xu, J. Adaptive Network Formation and Trajectory Optimization for Multi-UAV-Assisted Wireless Data Offloading. In Proceedings of the 2021 IEEE 23rd International Conference on High Performance Computing & Communications, Haikou, China, 20–22 December 2021; pp. 961–967. [CrossRef]
18. Wu, S.; Xu, W.; Wang, F.; Li, G.; Pan, M. Distributed federated deep reinforcement learning based trajectory optimization for air-ground cooperative emergency networks. *IEEE Trans. Veh. Technol.* **2022**, *71*, 9107–9112. [CrossRef]

19. Qian, L.P.; Zhang, H.; Wang, Q.; Wu, Y.; Lin, B. Joint Multi-Domain Resource Allocation and Trajectory Optimization in UAV-Assisted Maritime IoT Networks. *IEEE Internet Things J.* **2022**, *10*, 539–552. [CrossRef]
20. Zhou, H.; Long, Y.; Gong, S.; Zhu, K.; Hoang, D.T.; Niyato, D. Hierarchical Multi-Agent Deep Reinforcement Learning for Energy-Efficient Hybrid Computation Offloading. *IEEE Trans. Veh. Technol.* **2022**, *72*, 986–1001. [CrossRef]
21. Gong, S.; Cui, L.; Gu, B.; Lyu, B.; Hoang, D.T.; Niyato, D. Hierarchical Deep Reinforcement Learning for Age-of-Information Minimization in IRS-aided and Wireless-powered Wireless Networks. *IEEE Trans. Wirel. Commun.* **2023**. [CrossRef]
22. Gong, S.; Wang, M.; Gu, B.; Zhang, W.; Hoang, D.T.; Niyato, D. Bayesian Optimization Enhanced Deep Reinforcement Learning for Trajectory Planning and Network Formation in Multi-UAV Networks. *IEEE Trans. Veh. Technol.* **2023**. [CrossRef]
23. Zheng, J.; Chen, R.; Yang, T.; Liu, X.; Liu, H.; Su, T.; Wan, L. An efficient strategy for accurate detection and localization of UAV swarms. *IEEE Internet Things J.* **2021**, *8*, 15372–15381. [CrossRef]
24. Mou, Z.; Zhang, Y.; Gao, F.; Wang, H.; Zhang, T.; Han, Z. Deep reinforcement learning based three-dimensional area coverage with UAV swarm. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 3160–3176. [CrossRef]
25. Ye, Z.; Wang, K.; Chen, Y.; Jiang, X.; Song, G. Multi-UAV Navigation for Partially Observable Communication Coverage by Graph Reinforcement Learning. *IEEE Trans. Mobil. Comput.* **2022**, *8*, 15372–15381. [CrossRef]
26. Jung, S.; Yun, W.J.; Shin, M.; Kim, J.; Kim, J.H. Orchestrated scheduling and multi-agent deep reinforcement learning for cloud-assisted multi-UAV charging systems. *IEEE Trans. Veh. Technol.* **2021**, *70*, 5362–5377. [CrossRef]
27. Abd-Elmagid, M.A.; Dhillon, H.S. Average peak age-of-information minimization in UAV-assisted IoT networks. *IEEE Trans. Veh. Technol.* **2018**, *68*, 2003–2008. [CrossRef]
28. Du, Y.; Wang, K.; Yang, K.; Zhang, G. Energy-efficient resource allocation in UAV based MEC system for IoT devices. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–6. [CrossRef]
29. Luong, P.; Gagnon, F.; Tran, L.N.; Labeau, F. Deep reinforcement learning-based resource allocation in cooperative UAV-assisted wireless networks. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 7610–7625. [CrossRef]
30. Singh, S.K.; Agrawal, K.; Singh, K.; Chen, Y.M.; Li, C.P. Ergodic Capacity and Placement Optimization for RSMA-Enabled UAV-Assisted Communication. *IEEE Syst. J.* **2022**. [CrossRef]
31. Singh, S.K.; Agrawal, K.; Singh, K.; Chen, Y.M.; Li, C.P. Performance Analysis and Optimization of RSMA Enabled UAV-Aided IBL and FBL Communication with Imperfect SIC and CSI. *IEEE Trans. Wirel. Commun.* **2022**. [CrossRef]
32. Sohail, M.F.; Leow, C.Y.; Won, S. Non-orthogonal multiple access for unmanned aerial vehicle assisted communication. *IEEE Access* **2018**, *6*, 22716–22727. [CrossRef]
33. Hosseini, M.; Ghazizadeh, R. Stackelberg game-based deployment design and radio resource allocation in coordinated UAVs-assisted vehicular communication networks. *IEEE Trans. Veh. Technol.* **2022**, *72*, 1196–1210. [CrossRef]
34. Liang, Y.C.; Zhang, Q.; Wang, J.; Long, R.; Zhou, H.; Yang, G. Backscatter communication assisted by reconfigurable intelligent surfaces. *IEEE Trans. Knowl. Data Eng.* **2022**, *28*, 2296–2319. [CrossRef]
35. Gong, S.; Gao, L.; Xu, J.; Guo, Y.; Hoang, D.T.; Niyato, D. Exploiting backscatter-aided relay communications with hybrid access model in device-to-device networks. *IEEE Trans. Cogn. Commun. Netw.* **2019**, *5*, 835–848. [CrossRef]
36. Lowe, R.; Wu, Y.I.; Tamar, A.; Harb, J.; Pieter, A.; Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30. [CrossRef]
37. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2016**. [CrossRef]