Deep Reinforcement Learning Approach for Joint Trajectory Design in Multi-UAV IoT Networks

Shu Xu, *Student Member, IEEE*,
Xiangyu Zhang, *Student Member, IEEE*,
Chunguo Li, *Senior Member, IEEE*,
Dongming Wang, *Member, IEEE*,
and Luxi Yang, *Senior Member, IEEE*

***Abstract*—In this paper, we investigate an unmanned aerial vehicle (UAV) communication system, where the trajectories of multi-UAVs are designed for the data collection mission of IoT nodes. We aim at minimizing the mission time with constraints of UAV's maximum speed and acceleration, the collision avoidance, and communication interference among UAVs. We propose a three-step approach to solve this problem, which is based on the K-means algorithm, and Deep Reinforcement Learning (DRL) with a distributed manner and a centralized manner. The mutual influences like collision avoidance and interference among UAVs are explicitly expressed in our algorithm. Numerical results show the advantage of our proposed approach.**

***Index Terms*—Multi-UAV, trajectory design, multi-agent Deep Reinforcement Learning.**

I. Introduction

Unmanned aerial vehicles (UAVs) have been widely adopted in the field of military, civilian, and commercial applications [1]. Working in dangerous geographical areas or complex communication environments to collect data from IoT nodes makes them indispensable tools in communication systems [2]. With the advantages of high mobility and flexible deployment, UAVs are likely to have better communication channels, therefore achieving better performance than traditional ground Base Stations (BSs). Thus, trajectory planning has become the vital problem that influences the performance of UAVs.

However, the trajectory planning problem in UAV communications also faces several challenges. Firstly, accurate models are not easily established in practice. Secondly, the optimization problem is highly non-convex when considering trajectory planning under the constraint of communication quality indicators. Furthermore, as the number of UAVs increases, this problem becomes even more complicated. The reasons can be boardly concluded as a) collision avoidance becomes the issue, b) the transmitting interference among UAVs must be taken into consideration when two or more of them communicating with users at the same time, and c) multi-UAV system apparently brings efficiency promotion if full collaboration can be achieved among UAVs. Previous works [3]–[5] modeled multi-UAV trajectory planning as a mixed-integer non-convex optimization problem, and it was optimized by applying successive convex optimization techniques. However, all of them need global precise models, and the heavy calculation stuck the implementation of UAV communications in practice. Meanwhile, they had the shortage that the total number of steps in UAVs' trajectories must be given before the problem was formulated. In [6], [7], a Q-learning based approach towards the deployment and movement problem of UAVs was proposed to maximize the total communication rate of users under the condition that the minimum rate requirement was satisfied. Additionally, a multi-agent Deep Reinforcement Learning (DRL) based resource allocation algorithm was proposed for multi-UAV networks in [8]. However, the cooperation among UAVs could not be specified with their algorithm. The authors in [9] studied the problem of trajectory design for a group of cooperative UAVs, where the Value-Decomposition (V-D) algorithm was proposed to maximize coverage of the dynamic requests of the ground users. The Meta-training method was applied to optimize the initializations in a V-D solution.

In this paper, we consider a multi-UAV uplink data collection scenario aiming at receiving all the data sent from IoT nodes. The main contributions can be summarized as follows.

- Different from the task modeled as dispatching DBSs to each ground user in a discrete form [9], we solve a continuous trajectory design problem for multi-UAV.
- We formulate the problem as a Markov Decision Process (MDP) and solve it by utilizing the DRL method, which has low computational complexity. Regarded as a data-driven approach that does not need a precise model, this model-free method saved the cost of global modeling.
- We propose a three-step approach to solve the problem, which consists of the K-means algorithm for task allocation, the distributed and the centralized design of UAVs' trajectories based on DRL. We design the Q-network in DRL specifically for our model, where the cooperation information can be learned when conflicting situations occur, which guides the training of the Q-network.
- We design the Q-network of action-value function in DRL with a combination of distributed and centralized manner, which explicitly separates the value for individual actions and the group state for UAVs. We compare our algorithm with the existing fully distributed method Independent Agent Q-Learning (IAQL) and fully centralized method V-D algorithm [10]. Numerical results have verified the effectiveness of our proposed approach.

II. System Model and Problem Formulation

A. System Description

As illustrated in Fig. 1, we consider a multi-UAV uplink wireless communication network in a particular area, which consists of K IoT nodes. The target task in this paper is to collect data from these IoT nodes with the aid of UAVs. Specifically, M UAVs are dispatched by the control center at the same time to support the communication service. We assume that all the UAVs are the same type, and share the same frequency band when communicating with IoT nodes. Each IoT node is only scheduled for transmission by the UAV after being woken up by a pilot sequence. Additionally, assuming that each IoT node can only be served by a single UAV, and each UAV can only serve a single IoT node simultaneously.

Manuscript received January 31, 2021; revised June 17, 2021, October 17, 2021, and January 7, 2022; accepted January 12, 2022. Date of publication January 21, 2022; date of current version March 15, 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFB1807201, in part by the National Natural Science Foundation of China under Grants U1936201, 62171119, and 61971128, and in part by the Key Research and Development Plan of Jiangsu Province under Grant BE2021013-3. The review of this article was coordinated by Dr. Ahmed Hamdi Sakr. *(Corresponding author: Chunguo Li.)*

The authors are with the National Mobile Communications Research Laboratory, School of Infomation Science and Engineering, Southeast University, Nanjing 210096, China (e-mail: shuxu@seu.edu.cn; xy_zhang@seu.edu.cn; chunguoli@seu.edu.cn; wangdm@seu.edu.cn; lxyang@seu.edu.cn).

Digital Object Identifier 10.1109/TVT.2022.3144277

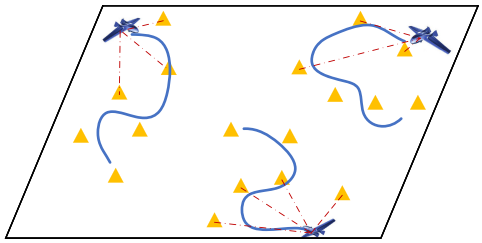

Fig. 1. Multi-UAV data collection scenario.

B. UAV Mobility Model

Let T_m denote the completion time of the task for UAV m, where $m \in \{1, 2, \ldots, M\}$. Then, we discretize the time duration T_m with fixed time intervals δ_t into $N_m = \frac{T_m}{\delta_t}$ time slots. Thus, the total completion time for the whole task can be expressed as $\max_m \{N_m\}$.

For ease of expression, we formulate the model in a three-dimensional (3D) Cartesian coordinate system. We assume that all the UAVs fly at a constant altitude H_U, and the horizontal trajectory of UAV m at time n is denoted by $\mathbf{q}_m[n] = [x_m[n], y_m[n]]^T \in \mathbb{R}^{2\times 1}$, where $n = 0, 1, \ldots, N_m$, and $\mathbb{R}^{2\times 1}$ denotes the two-dimensional real-valued vector space. We define $\mathbf{a}_m[n] \in \mathbb{R}^{2\times 1}$ and $\mathbf{v}_m[n] \in \mathbb{R}^{2\times 1}$ as the acceleration and the velocity of UAV m, respectively, which must satisfy the following constraints

$$\mathbf{v}_m[n+1] = \mathbf{v}_m[n] + \mathbf{a}_m[n]\,\delta_t, \tag{1}$$

$$\mathbf{q}_m[n+1] = \mathbf{q}_m[n] + \mathbf{v}_m[n]\,\delta_t + \frac{1}{2}\mathbf{a}_m[n]\,{\delta_t}^2, \tag{2}$$

$$\mathbf{q}_m[0] = \mathbf{q}_{m,0}, \tag{3}$$

$$||\mathbf{q}_i[n] - \mathbf{q}_j[n]|| \geq d_{\min}, \forall i, j \in \{1, 2, .., M\}, i \neq j, \tag{4}$$

where constraint (3) indicates that all the UAVs are launched at certain locations, and $d_{\min}$ in constraint (4) is the minimum distance among the UAVs for collision avoidance. In this paper, we choose fixed-wing UAVs to perform the task mentioned above. Thus, the acceleration and the velocity of this kind of UAVs should follow that

$$||\mathbf{a}_m[n]|| \leq a_{\max}, \tag{5}$$

$$v_{\min} \leq ||\mathbf{v}_m[n]|| \leq v_{\max}, \tag{6}$$

where constraint (5) and (6) denote the maximum acceleration, and the maximum/minimum velocity constraints for fixed-wing UAVs, respectively.

The horizontal coordinate of each IoT node is denoted by $\omega_k = [x_k, y_k]^T$, where $k \in \{1, 2, \ldots, K\}$. Hence, the distance between the UAV m and the IoT node k at time n can be expressed as

$$d_{m,k}[n] = \sqrt{{H_U}^2 + ||\mathbf{q}_m[n] - \omega_k||^2}. \tag{7}$$

C. Channel Model

Consider the transmission scenario in an urban area, where a Probabilistic LoS Channel Model is applied [1]. The expected channel power gain from IoT node k to UAV m is given by

$$g_{m,k}[n] = [P_{LoS} + (1 - P_{LoS})\,\kappa]^{-1}\beta_0 d_{m,k}^{-\alpha}[n], \tag{8}$$

where the factor κ is the attenuation coefficient due to the NLoS link, the factor $\beta_0 = (\frac{\lambda}{4\pi})^2$ is the channel power gain at the reference distance of 1 m, λ is the carrier wavelength, and P_{LoS} denotes the probability of having LoS link, which is the function of the elevation angle θ as

$$P_{LoS} = \frac{1}{1 + a\exp\left(-b\left(\sin^{-1}\left(\frac{H_U}{d_{m,k}[n]}\right) - a\right)\right)}, \tag{9}$$

where a and b are the modeling parameters.

We suppose that all the IoT nodes have the same constant transmission power P_t. So the expected SINR between UAV m and IoT node k can be expressed as

$$\gamma_{m,k}[n] = \frac{P_t g_{m,k}[n]}{P_{Iter,m}[n] + \sigma^2}, \tag{10}$$

where σ^2 denotes the power of the Additive White Gaussian Noise. Assume that each IoT node should satisfy the minimum SINR threshold γ_{th} when establishing a communication link with UAV, we define the communication indicator function as

$$I_{m,k}[n] = \begin{cases} 1, \ if \ \begin{array}{l}\gamma_{m',k}[n] \geq \gamma_{th} \ \ and \\ m = \arg\max_{m'}\{\gamma_{m',k}[n]\}\end{array} \\ 0, \ otherwise \end{cases}, \tag{11}$$

where $m = \arg\max_{m'}\{\gamma_{m',k}[n]\}$ denotes that UAV m chooses to serve the IoT nodes with highest SINR. Thus, the instantaneous interference from other transmitting IoT nodes $P_{Iter}[n]$ can be expressed as

$$P_{Iter,m}[n] = \sum_{k=1}^{K}\sum_{m'=1, m'\neq m}^{M} I_{m',k}[n]\,P_t g_{m,k}[n]. \tag{12}$$

If IoT node k is scheduled for communication with UAV m, the achievable communication throughput is

$$r_{m,k}[n] = B\log_2\left(1 + \frac{P_t g_{m,k}[n]}{P_{Iter,m}[n] + \sigma^2}\right), \tag{13}$$

where B is the total bandwidth.

D. Problem Formulation

Our task is to minimize the total time steps for the multi-UAV system accomplishing the data collection mission, which can be formulated as the optimization problem $P1$:

$$(P1): \min_{\{\mathbf{a}_m\}} \max_m \{N_m\} \tag{14}$$

$$s.t. \sum_{m=1}^{M}\sum_{n=0}^{N_m} I_{m,k}[n]\,r_{m,k}[n] = G_k, \forall k,$$

$$(1) \sim (13). \tag{15}$$

where constraint (15) denotes that all the data must be collected by UAVs before completing the task, G_k is the total amount of data from IoT node k.

Obviously, the optimization problem $P1$ is non-convex due to the objective function, which is highly related to the current transmission quality of the link. In addition, the value of N_m is always varying, which results in the varying dimension of the variable sequence $\{\mathbf{a}_m[n]\}_{n=1}^{N_m}$. In this case, it makes the existing method based on the optimization approach infeasible to solve this problem. Fortunately, DRL can successfully tackle it because the variation sequence can be described by MDP, which is suitable for applying DRL to obtain the solution. In the following, we propose our algorithm based on the DRL, which combines K-means algorithm for efficient preliminary training to solve the multi-UAV trajectory design problem.

III. Proposed Solution

In this section, a three-step approach is proposed. Firstly, an IoT nodes based K-means algorithm is proposed for task allocation of UAVs [6]. Secondly, a distributed multi-agent based DRL algorithm is proposed for the preliminary design of UAVs' trajectories. Thirdly, a centralized multi-agent based DRL algorithm is proposed for the joint trajectory design of drone swarms. We will examine our approach as follows.

A. K-Means Algorithm for Task Allocation

To avoid the collision among UAVs as much as possible and reduce the complexity in DRL, the first step of our approach is to allocate all the IoT nodes for each UAV by employing the K-means algorithm, and the classification results can be denoted as $\mathbf{C} = \{C_1, C_2, \ldots, C_M\}$. We use all the coordinates of IoT nodes $\{\omega_1, \omega_2, \ldots, \omega_K\}$ as the intra-cluster similarity measurement, and maximize the intra-cluster similarity by minimizing the squared error

$$E = \sum_{m=1}^{M} \sum_{\omega \in C_m} ||\omega - \mu_m||^2, \tag{16}$$

where the mean vector of each cluster $\mu_m = \frac{1}{|C_m|} \sum_{\omega \in C_m} \omega$, which is served by a single UAV. Next, we will regard each UAV as a DRL agent, and develop a mission strategy for it.

B. Multi-Agent MDP Formulation

The multi-agent DRL aims at solving a Markov Game, where all the agents iteratively interact with the environment. At each time slot, each agent observes a state $\mathbf{s}_{m,t} \in S$, and selects an action $a_{m,t} \in A$ based on the current state. Accordingly, each agent receives a immediate reward $R_{m,t}$ and transitions to a new state $\mathbf{s}_{m,t+1}$. The goal of each agent is to improve its policy based on its experience, so as to maximize the long-term cumulative reward [11]. Thus, to solve the optimization problem $P1$, each agent is referred to as the UAV and we formulate the trajectory design problem as a MDP for each agent. Mathematically, a MDP of agent m can be specified by a 4-tuple $< S_m, A_m, P_m, R_m >$, which is defined as follows.

1) State S_m: The state space consists of three parts: the legitimate location of UAV m $\mathbf{q}_m[n]$, the legitimate velocity of UAV m $\mathbf{v}_m[n]$, and the channel indicator function (11).

2) Action A_m: The action space is composed of all the available movements for UAV m, i.e. the acceleration $\mathbf{a}_m[n]$.

3) Transforming Probability P_m: It is the state transition probability, with $P_m(\mathbf{s}_m{}'|\mathbf{s}_m, \mathbf{a}_m)$ specifying the probability of agent m transforming from the current state $\mathbf{s}_m$ to the next state $\mathbf{s}_m{}'$ after applying action $\mathbf{a}_m$. To be specific, we design

$$P_m(\mathbf{s}_m{}'|\mathbf{s}_m, \mathbf{a}_m) = \begin{cases} 1, \ if \begin{matrix} (\mathbf{s}_m{}' = \mathbf{s}_{m,new} and \mathbf{s}_{m,new} \in S_m) \\ or \ (\mathbf{s}_m{}' = \mathbf{s}_m and \mathbf{s}_{m,new} \notin S_m) \end{matrix} \\ 0, \qquad otherwise \end{cases}, \tag{17}$$

where $\mathbf{s}_{m,new}$ is the next state theoretically, which may not always satisfy the constraints in practice.

4) Reward Function R_m: Since the goal of RL is to maximize the agent's future rewards received from the environment, the design of the reward function should be strongly correlated to the objective function and constraints in $P1$. The reward function is designed as follows.

- The agent receives a negative reward for speeding

$$R_{m,1}[n] = \begin{cases} -\alpha_1 \Delta v, & ||\mathbf{v}_m[n]|| > v_{\max} \\ 0, & otherwise \end{cases}, \tag{18}$$

where $\Delta v = ||\mathbf{v}_m[n]|| - v_{\max}$ is the overspeed part.

TABLE I
Scenarios With Randomly Distributed IoT Nodes

	Settings		Completion Time (s)		
Scenario	M	K	Our algorithm	V-D Network	IAQL
I	4	30	69	*	91
II	4	30	94.5	*	106
III	4	24	93	*	*
IV	3	20	71	110	78.5

- The agent receives a positive reward when completing the data collection of one IoT node

$$R_{m,2}[n] = \begin{cases} \alpha_2, & n = N_m \\ 0, & otherwise \end{cases}. \tag{19}$$

- When the relative distance between UAV i and j flows below the safety distance, both agent i and agent j receive a negative reward

$$R_{m,3}[n] = \begin{cases} 0, & if\ (4)\ is\ satisfied \\ -\alpha_3, & otherwise \end{cases}. \tag{20}$$

- When a UAV completes the data collection task for a single IoT node or achieves an SINR gain compared to the previous state, the agent receives a positive reward, while receiving punishment for SINR attenuation,

$$R_{m,4}[n] = \alpha_4 \left\{ \sum_{k=1}^{K} I_{m,k}[n] \, r_{m,k}[n] \right\} + \alpha_5. \tag{21}$$

Where $\alpha_i > 0, i = 1, 2, 3, 4$ are constant parameters of the reward and punishment. α_5 is proportional to the gain or attenuation of SINR, which can be denoted as $\alpha_5 = \beta[\max\{\gamma_m[n], \gamma_{th}\} - \max\{\gamma_m[n-1], \gamma_{th}\}]$, and $\beta > 0$ is the scaling factor. At each epoch, agent m receives the term of '-1' as a penalization to finish its data collection task as quickly as possible, as well as four types of reward mentioned above. Thus, the reward function $R_m[n]$ can be expressed as

$$R_m[n] = -1 + \sum_{k=1}^{4} R_{m,k}[n], \forall m, n. \tag{22}$$

In our simulation, we set that the velocity of the UAV will remain unchanged if a speed-up command is given at its maximum speed (i.e., the setting stated in (17)). Thus, if conditions of $||\mathbf{v}_m[n]|| > v_{\max}$ are satisfied, the agent will be penalized as $R_{m,1}[n]$ shown, while at the same time, perform nothing changed to the velocity of each UAV. Similar setting of $R_{m,3}[n]$ is made for safety consideration.

Remark 1: Based on our simulation settings, the designs of $R_{m,1}[n]$ and $R_{m,3}[n]$ will not affect the optimal solution in (14). $R_{m,2}[n]$ and $\alpha_4\{\sum_{k=1}^{K} I_{m,k}[n] r_{m,k}[n]\}$ are constants, under the condition that the UAV system completes the data collection task. No higher cumulative reward can be achieved under the condition of accomplishment failure. The design of α_5 in $R_{m,4}[n]$ is based on the reward shaping technique [12] to maintain its equivalence. Thus, the designed reward function does not change the optimal solution of the optimization problem, but greatly accelerates the training process of the agent's optimal policy. Because the use of the reward shaping technique is considered as a way to speed up learning and make the agent achieve a better knowledge of the environment.

C. DRL Based Multi-UAV Trajectory Design

In multi-agent based DRL framework, each agent records its own action-value function in memory D as

$$Q_m(\mathbf{s}_m, a, \Theta) = \mathbb{E}\left[\sum_{\tau=t}^{\infty} R_m[\tau] \mid \mathbf{s}_t = \mathbf{s}_m, a_t = a, \Theta \right], \tag{23}$$

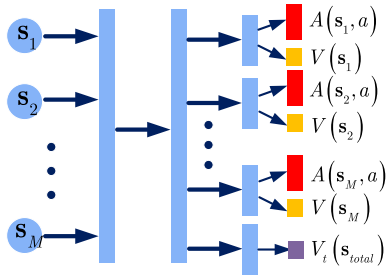

Fig. 2. The framework of multi-agent network.

which represents the expected long-term cumulative reward achievable under current policy characterized by the parameters of the Q-network Θ. The purpose of our algorithm is to learn a policy that obtains the maximum action-value function. In the following, we drop Θ for clear expression. The reward function received by every single UAV does not entirely depend upon its own state but is also correlated with other agents' actions and states. Meanwhile, it may affect the rewards received by other agents. According to this consideration, the action-value function of each single agent can be represented by Q-network as

$$Q_m(\mathbf{s}_m, a) = A(\mathbf{s}_m, a) + V(\mathbf{s}_m) + V_t(\mathbf{s}_{total}), \quad (24)$$

where $V(\mathbf{s}_m)$ and $V_t(\mathbf{s}_{total})$ are the State Value Functions only related to agents' states, and $A(\mathbf{s}_m, a)$ is the Advantage Function [13] related to both the state and the action agent m takes. We design the neural network of Q-network as shown in Fig. 2. This network consists of two streams of fully connected layers for agent m based on (24), which are the private part of each single agent $A(\mathbf{s}_m, a)$ and $V(\mathbf{s}_m)$, and the public part calculated for all the agents $V_t(\mathbf{s}_{total})$.

In this way, the network can be learned with a centralized strategy in order to characterize the condition of encountering the conflicting situation, such as the distance between two UAVs flowing below the safety distance, or the strong interference generated by other UAVs when communicating with IoT nodes simultaneously. Numerical results show that the two situations mentioned above may result in a low value of $V_t(\mathbf{s}_{total})$. Thus, we verify the validity of the public value function, and enhance the interpretability of the multi-agent network. Additionally, by introducing the distributed and centralized architecture, the network can learn the meaningless states like the condition that all the UAVs are flying away from target IoT nodes, without having to learn specifically the private state of each agent. In our model, the public network and private networks share a common fully connected network in the first two layers, which is useful for avoiding the lazy agent problem [10].

In Algorithm 1, each UAV acts as an agent, executing the action based on its current state and the Q-network issued by the control center. The Q-network is trained at the control center by feeding the 4-tuple transition of MDP as training samples provided by UAVs. The update of Q-network is separated into two steps, i.e., the second and the third step of our approach. The Q-network is trained by minimizing two different loss functions $L_{1,m}(\Theta_m)$ and $L_2(\Theta)$ via Stochastic Gradient Descent algorithm, respectively. Both of them are updated iteratively under the framework of Algorithm 1. For the purpose of pretraining the Q-networks and achieving a relatively feasible solution for each agent, the loss function $L_{1,m}(\Theta_m)$ in the second step is designed as

$$L_{1,m}(\Theta_m) = E_{s,a\sim\rho(\cdot)}\left[(y_m - Q_m(\mathbf{s}_m, a, \Theta_m))^2\right], \quad (25)$$

where $y_m = R_m + \gamma \max_{a' \in A_m(s')} Q_m(\mathbf{s}_m', a'; \Theta_m)$ is the target function for agent m, and $\rho(\mathbf{s}, a)$ is the probability distribution of selecting actions at state $\mathbf{s}$. In this step, both $A(\mathbf{s}_m, a)$ and $V(\mathbf{s}_m)$ will remain unchanged when the action-value networks of other agents are being updated, as can be observed from (24). In the second step, the policy cannot achieve a better performance in our limited training time. Because it neglects the interactive influences among UAVs. In this front, the third step is designed with a centralized framework to realize a joint optimization for agents' cooperative work. Thus, the loss function $L_2(\Theta)$ is designed as

$$L_2(\Theta) = E_{s,a\sim\rho(\cdot)}\left[\left(\sum_{m=1}^{M}(y_m - Q_m(\mathbf{s}_m, a, \Theta_m))\right)^2\right], \quad (26)$$

where the Q-networks of all agents are updated simultaneously with a joint reward $R_t = \sum_{m=1}^{M} R_m$. In this step, we aim to learn a joint action-value function

$$Q_t(\mathbf{s}, a, \Theta) = \sum_{m=1}^{M} Q_m(\mathbf{s}_m, a, \Theta_m). \quad (27)$$

The goal of each agent is not to maximize the action-value fucntion of itself, instead, they pursue for a cooperation performance. Thus, a decentralized policy can be arisen for each agent with centralized consideration of all agents' states.

Algorithm 1: Multi-UAV Trajectory Design With DRL.

Initialize: IoT nodes task allocation based on (16); Replay memory D to capacity N_D; Maximum training epochs N_e; Maximum steps of each epoch N. Randomly initialize the network weights Θ_m of action-value function $Q_m(\mathbf{s}_m, a; \Theta_m)$;
for *Epoch = 1, 2, ..., N_e* **do**
  Initialize the state of each agent $\mathbf{s}_m$;
  Initialize the time step $n \leftarrow 0$;
  repeat
    Select action $\mathbf{a}_m$ for each agent based on ϵ-greedy policy derived from Q_m to obtain maximum Q_m;
    Perform actions in emulator, observe reward r_m and the next state $\mathbf{s}_m'$;
    Store 4-tuple transition $(\mathbf{s}_m, \mathbf{a}_m, r_m, \mathbf{s}_m')$ in replay memory D;
    Sample a minibatch of transition records from D randomly;
    Update the action-value network based on the loss function of Equation (25) or (26);
    $n \leftarrow n + 1$
  until *n=N or mission complete*;
end

D. Analysis of DRL Implementation

The designed operational procedure can be divided into two phases, Training and Execution. In training phase, the control center iteratively updates the policy for each UAV based on the 4-tuple transition stored in replay memory D. This process is fit by the update of Q-network based on the loss function (25) and (26), i.e., the distributed and the centralized training, respectively. While IAQL is a fully distributed method that treats all other agents as agent m's environment, of which performance is limited as verified in subsequent simulation results. In execution phase, all the UAVs execute the action based on the decentralized policy instructed by the control center. Two phases are separately implemented

by the control center and UAVs. In terms of communication overheads, each UAV obtains its own execution policy from the control center and reports the state-action pairs in execution phase to the control center as training samples for training use. Compared to the fully centralized method V-D, our proposed algorithm has fewer communication overheads, for which all the other agents' states should be included in the communication overheads when calculating each agent's policy in V-D algorithm.

In training phase, computational complexity in each iteration is measured by the floating-point operations per second (FLOPs). It is mainly determined by the matrix multiplication of the Q-network, where the fully connected layer is designed as three hidden layers of $f_1 = 256$, $f_2 = 256$, and $f_3 = 64$. Thus, the total number of FLOPs during the inference is

$$FLOPs = 2\,M \cdot (f_1|\mathcal{S}_i| + f_3|\mathcal{A}_i| + f_2 f_3) + f_1 f_2. \tag{28}$$

where $|\mathcal{S}_i|$ is the dimension of state space (Q-network's input), and $|\mathcal{A}_i|$ is the number of optional actions (Q-network's output). Thus, computational complexity is approximately proportional to the number of UAVs M.

IV. Numerical Results

We performed simulations to evaluate the effectiveness of the data collection task with our proposed multi-UAV trajectory design algorithm. For the channel parameters, the attenuation coefficient $\kappa = 0.05$, the AWGN power assumed to be $\sigma^2 = -120dBmW$, the SINR threshold $\gamma_{th} = 0\,dB$, and the modeling parameters for P_{LoS} is $a = 9.53$ and $b = 0.41$, respectively. The maximum transmission power of IoT node is $P_t = 10dBmW$, and all the IoT nodes are assumed to transmit the data with the maximum power while connecting to UAVs. The maximum/minimum velocity and the maximum acceleration of UAVs are assumed as $v_{\max} = 30\,m/s$, $v_{\min} = 1\,m/s$ and $a_{\max} = 5\,m/s^2$, respectively.

We considered the scenario with the number of UAVs $M = 4$, and the number of IoT nodes $K = 30$ in the urban district of size $2\,km \times 2\,km$. For the first 1200 training epochs, we train the Q-network under loss function $L_{1,m}(\Theta)$, and for the 1201st to the 2000th training epoch, we apply $L_2(\Theta)$ as mentioned in Part III. Fig. 3(a) depicts the multi-UAV joint trajectory designed in this scenario based on our algorithm. We put the cumulative reward curves and our optimization objective (the completion time) in the same figure for intuitive comparison. In Fig. 3(b) and (c), the left and the right y axis represent the average cumulative rewards, and the average completion times to achieve the goal, respectively. Fig. 3(b) plots the comparison between our proposed algorithm and the scheme based on V-D Network. The result shows that the V-D scheme failed to converge to an optimal solution within limited training time compared to our algorithm, and accordingly failed to complete the mission within limited time $200\,s$. It was worth mentioning that a corresponding rise in the curves of reward could be observed with the decrease of the completion times, which means a better strategy is learned by the RL agents. Fig. 3(c) compares our proposed approach with the scheme based on IAQL for the training epoch 1201st to 2000th. We can see that a performance improvement is achieved for the third step of our approach compared to the scheme based on IAQL.

The completion time slot of our proposed approach is $69\,s$, and that of IAQL is $91\,s$, while the scheme based on V-D Network failed to complete the mission within limited time $200\,s$. Thus, according to all numerical results and analysis mentioned above, we claim that our proposed algorithm gives better performance than the two existing DRL algorithms in terms of the reward curves, as well as the optimization objective (the completion time) in the optimization problem.

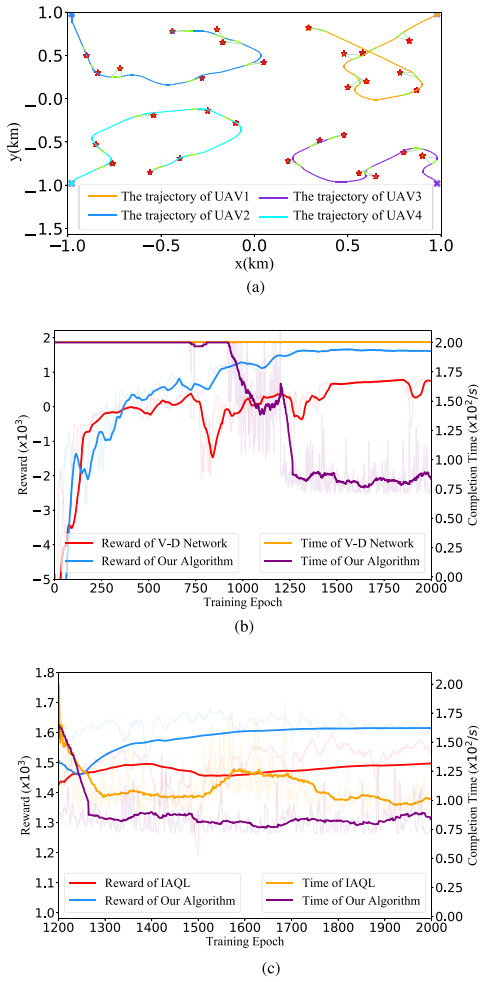

Fig. 3. The experimental results. (a) Multi-UAV Joint Trajectory Design. (b) Our Algorithm vs. V-D Network. (c) Our Algorithm vs. IAQL.

We also remove the K-means algorithm for task allocation, which results in Q-network convergence failure with a very low reward and failed in accomplishing the mission. Additionally, more scenarios with randomly distributed IoT nodes are considered. The completion times for the UAV system to achieve the goal are shown in TABLE I with different number of UAVs M, and different number of IoT nodes K, where the symbol of '*' denotes that the algorithm failed to train a feasible strategy to complete the data collection task in our given training epochs. Similar conclusions can be made that our proposed algorithm gives better performance than the scheme based on V-D Network and IAQL.

V. Conclusion

This paper studied multi-UAV trajectory design problem for uplink data collection task based on multi-agent DRL algorithm. The trajectories of UAVs were jointly optimized to minimize the total completion time with constraints of the maximum speed and acceleration, the collision avoidance, and communication interference among UAVs. With a three-step approach and the network of action-value function specifically designed for this generalized problem, we got the desired performance. Numerical results demonstrate that our proposed approach achieves superior performance over two existing multi-agent methods.

REFERENCES

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

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

[3] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for UAV-enabled multiple access," in *Proc. IEEE Glob. Commun. Conf.*, 2017, pp. 1–6.

[4] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 2109–2121, Mar. 2018.

[5] J. Zhang *et al.*, "Computation-efficient offloading and trajectory scheduling for multi-UAV assisted mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2114–2125, Feb. 2020.

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

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

[8] J. Cui, Y. Liu, and A. Nallanathan, "Multi-agent reinforcement learning-based resource allocation for UAV networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 729–743, Feb. 2020.

[9] Y. Hu *et al.* "Distributed multi-agent meta learning for trajectory design in wireless drone networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 10, pp. 3177–3192, Oct. 2020.

[10] P. Sunehag *et al.*, "Value-decomposition networks for cooperative multi-agent learning," 2017, *arXiv: 1706.05296*.

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

[12] A. Y. Ng, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Proc. 16th Int. Conf. Mach. Learn.*, Morgan Kaufmann, May 1999, pp. 278–287.

[13] Z. Wang *et al.*, "Dueling network architectures for deep reinforcement learning," in *Proc. 33rd Int. Conf. Int. Conf. Mach. Learn.*, 2016, pp. 1995–2003.