DRL-UTPS: DRL-Based Trajectory Planning for Unmanned Aerial Vehicles for Data Collection in Dynamic IoT Network

Run Liu, Zhenzhe Qu, Guosheng Huang, Mianxiong Dong, *Member, IEEE*, Tian Wang, Shaobo Zhang, and Anfeng Liu

***Abstract*—Using highly maneuverable Unmanned Aerial Vehicles (UAV) to collect data is a fast and efficient method that is widely studied. In most studies, they assume that the UAVs can obtain the location of the Cluster Head (CH) before take-off, allocate CHs, and optimize the trajectory in advance. However, in many real scenarios, many sensing devices are deployed in areas with no basic communication infrastructure or cannot communicate with the Internet due to emergencies such as disasters. In this kind of sensing network, the surviving devices often change, and the CHs cannot be known and allocated in advance, thus bringing new challenges to the efficient data collection of the networks by using UAVs. In this paper, a UAV path planning scheme for IoT networks based on reinforcement learning is proposed. It plans hover points for UAV by learning the historical location of CHs and maximizes the probability of meeting CHs and plans the shortest UAV path to visit all hover points by using the simulated annealing method. In addition, an algorithm to search for the location of CHs is proposed which is named Cluster-head Searching Algorithm with Autonomous Exploration Pattern (CHSA-AEP). By using CHSA-AEP, our scheme enables the UAV to respond to the position change of the CHs. Finally, we compare our scheme with other algorithms (area coverage algorithms and random algorithm). It is found that our proposed scheme is superior to other methods in energy efficiency and time utilization ratio.**

***Index Terms*—Data collection, deep reinforcement learning, UAV trajectory planning, unconnected network.**

Manuscript received 15 August 2022; revised 25 September 2022; accepted 6 October 2022. Date of publication 11 October 2022; date of current version 20 March 2023. This work was supported in part by the Natural Science Foundation of Hunan Province under Grant 2020JJ4237 and in part by the National Natural Science Foundation of China under Grant 62072475. *(Corresponding authors: Guosheng Huang; Anfeng Liu.)*

Run Liu, Zhenzhe Qu, and Anfeng Liu are with the School of Computer Science and Engineering, Central South University, Changsha 410083, China (e-mail: runliu@csu.edu.cn; zhenzheQu@csu.edu.cn; afengliu@csu.edu.cn).

Guosheng Huang is with the School of Information Science and Engineering, Hunan First Normal University, Changsha 410205, China (e-mail: huanggs@hnfnu.edu.cn).

Mianxiong Dong is with the Department of Information and Electronic Engineering, Muroran Institute of Technology, Muroran 050-8585, Japan (e-mail: mx.dong@csse.muroran-it.ac.jp).

Tian Wang is with the Department of Artificial Intelligence and Future Networks, Beijing Normal University and UIC, Zhuhai 519088, China (e-mail: tianwang@bnu.edu.cn).

Shaobo Zhang is with the School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, China (e-mail: shaobozhang@hnust.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIV.2022.3213703.

Digital Object Identifier 10.1109/TIV.2022.3213703

I. Introduction

ACCORDING to a white paper published by Cisco in 2018 [1], the number of sensing devices is expected to increase 2.4 times from 6.1 billion in 2018 to 14.7 billion in 2023. Massive sensing devices are deployed to sense rich data in various application scenarios, and the amount of sensed data is expected to reach 90 zettabytes by 2025 [2]. The vast amount of sensed data is available for a variety of applications [3], [4], including NoiseTube [5] and Ear-phone [6] for noise monitoring, SignalGuru [7] and CrowdAtlas [8]. Data collection is one of the key issues. In traditional Wireless Sensor Networks (WSN), sensing devices are deployed in areas where they need to be monitored. There are two main methods for data collection. One is that each device directly transmits its perceived data to a sink through multi-hop routing. The other form of data collection is that the sensing devices adopt clustering routing protocol and sensing devices are divided into clusters according to the geographical location information to sense data distributedly. Each cluster selects a Cluster Head (CH), and the CH is used to gather the data of member nodes. After the CH fuses the data, the amount of data is greatly reduced, and then routed to sink through multi-hop routing. In the WSN, a communication link to the Internet needs to be established. Therefore, the cost of deployment, time, efficiency and other aspects are difficult to meet rapid perception application requirements. In practical applications, it is often necessary to deploy sensing devices in areas without basic communication facilities, or in some scenarios, network communication links may be interrupted by disasters, such as fire, flood, war, etc. At this time, although many sensing devices can self-organize into multiple local networks, there is no communication link with the outside world. Therefore, how to collect data for such networks presents a huge challenge compared to traditional WSNs. UAV plays an important role in collecting data [9], [10], [11], with UAVs acting as flight sinks for data collection and forming UAV-IoT [12] and UAV-WSN [13] systems. In such a system, UAVs can fly into the sensing devices' area and traverse each of the sensing device to collect data. Obviously, the UAVs need to traverse every area of the network, so their flight path is very long. Therefore, clustering is a more effective method. In cluster-based networks, the UAVs can efficiently collect data by traversing each CH. Collecting data in such networks, and optimizing the path of UAV is an

TABLE I
LIST OF IMPORTANT ABBREVIATIONS

Abbr.	Definition
A2G	Air to ground
AI	Artificial Intelligence
BS	Base Station
CH	Cluster Head
DA	Data Access
DRL	Deep Reinforcement Learning
GAE	Generalized Advantage Estimation
IoT	Internet of Things
MBS	Mobile Base Station
PPO	Proximal Policy Optimization
TSP	Travelling Salesman Problem
WSN	Wireless Sensor Networks
MDP	Markov Decision Process
UAV	Unmanned Aerial Vehicles

important research issue. This issue is somewhat similar to the Travelling Salesman Problem (TSP) but more difficult. There are two parts of the research to collect data using UAVs. One is to choose which node to act as CH, and the other is how UAVs traverse these CHs. The number and the location of the selected CHs will affect the path of UAV and the overall costs of the system. These two issues influence each other and make it a challenge to solve. There are a lot of researches on this. Zhu et al. proposed a UAV path planning scheme for UAV-IoT [12] and UAV-WSN [13] respectively, which reduced total energy consumption of the system by jointly selecting CHs from a cluster-based IoT network [12] and WSN [13] and planning the UAV's trajectory [14].

However, the biggest assumption in previous studies is that UAVs know the deployment of the network in advance, so that UAVs can allocate which sensing devices act as CHs before take-off, and then the UAVs traverse all CHs along the planned trajectory for data collection. But such methods often fail in real scenarios. UAVs are often employed to collect data in areas where there are no communications infrastructures, or in networks where they cannot communicate with the Internet due to disasters. This is because networks that have a direct communication link to the Internet can send data directly to the platform. Its data collection will be more timely, convenient and cheaper which does not require UAVs for data collection. Some nodes of these isolated networks may die in a disaster environment, and the networks' connection status is constantly changing. Because the platform cannot communicate with the network, it cannot know the networks' situation, nor can it specify which sensing devices in the networks are CHs, nor can it plan the path of UAVs in advance. Moreover, even if UAVs know the network deployment, CHs of the network are dynamically replaced according to the cluster-head election protocol to ensure that the network will not die in advance. Therefore, the UAVs do not know the location of CHs and cannot carry out path planning before they are dispatched. Therefore, the previous assumptions that CHs can be selected and plan the path in advance can not be applied in practice. So, the research issue that UAVs cannot know the location of the CHs in advance for data collection is of more practical significance. For example, when disasters occur, communication facilities may be damaged. UAV can serve as Mobile Base Station (MBS) to provide secure data communication for the public (e.g., as China Mobile Communications Group Co., Ltd did in February 2021 for the flood in Henan province, China) [15] in which the location and the number of devices are not known.

How to make use of the high flexibility of UAVs so that the platform can assign UAVs to efficiently collect data without knowing the specifics of the networks' situation in advance is a challenging issue. These challenges are mainly due to the following reasons. In such a network, a large number of sensing devices are deployed in the areas to be monitored as needed. These sensing devices can self-organize into a network and automatically select devices as CHs. At the same time, due to economic costs and security reasons, some devices in the network have advanced hardware, more powerful functions and higher costs, and can act as CHs, but some devices simply do not act as CHs and do not have the ability to communicate with UAV. From a security design point of view, some devices are programmed to communicate only with neighbor devices and not with other devices such as UAVs for security and energy savings. Some devices can either act as CHs or submit data by communicating with the UAV when it is within range of its communication. In terms of the overall network, the network is deployed on demand and cannot communicate with the outside world, so how many sensing devices are there on the network? Who are the CHs? Where are they? The platform is not known until the UAV reaches the network. This kind of network results in the inapplicability of the strategy of selecting CHs in advance and assigning UAVs to collect data after optimizing the path.

To solve the above problems, we propose a novel Deep Reinforcement Learning (DRL) based path planning method for UAV to search CHs and dynamically adjust the trajectory of UAV during data collection. As a popular Artificial Intelligence (AI) technology, DRL has attracted more and more researchers' attention. DRL enables agents to take actions based on the current state with the greatest long-term benefit. In UAV path planning field, DRL is increasingly used for planning. When UAV needs to make decisions in the face of changing states in the environment, DRL is used to learn UAV motion decisions in various states, and various optimal decisions based on reward functions can be obtained. Therefore, when CHs in a cluster-based IoT network change according to the cluster-head election protocol, DRL can learn the data collection strategy of UAV in various CHs' states to obtain the flight trajectory of UAV. Under the above circumstances, we focus on optimizing the trajectory of UAV considering energy efficiency and time utilization ratio in the cluster-based IoT network for data collection. Among them, CHs are elected according to the cluster-head election protocol, that is, each cluster uses the dynamic routing protocol. The main contributions of the paper are summarized as follows:

1) First, we put forward a network model that conforms to the actual situation, which overcomes the hypothesis that the network model does not conform to the reality in previous studies. In many practical UAV-IoT systems, sensing devices cannot be obtained in advance, CHs cannot be

specified in advance, and the trajectory of the UAV cannot be planned in advance. The network model, communication model and UAV energy consumption model are modeled according to the practical situation.

2) Then, we propose a UAV path planning scheme based on DRL (DRL-UTPS). In this scheme, the UAV path planning problem is divided into two sub-problems: the UAV hovering point planning problem and the TSP problem. For the hovering point planning problem of UAV data collection, we modeled the hovering point optimization process by Markov Decision Process (MDP), and optimized the UAV's hovering points' position by using PPO. The UAV's hovering points were optimized by maximizing the number of times that the UAV could communicate with CHs and collect data when it reached the hovering points. For TSP problem, since it is NP-hard, we use the Simulated Annealing algorithm to quickly solve the shortest path that can traverse all hover points.
3) A cluster-head search algorithm, CHSA-AEP, is designed for IoT networks to solve the problem that when the UAV is at the hover point but the CH is not within the UAV's communication range.
4) Finally, DRL-UTPS proposed in this paper are compared with baselines GBA [16] and CPP-SDA [17]. Meanwhile, Random, which randomly selects hover points, is also compared. Experimental results show that DRL-UTPS is superior to other algorithms in energy efficiency and time efficiency.

The rest of the paper is organized as follows. The related works are given in Section II. The system model and problem statements are presented in Section III. In Section IV, our proposed DRL-UTPS is established. The experimental results are given in Section V. In Section VI, we discuss the limitations of DRL-UTPS and the potential directions of our works and finally we also conclude in Section VI.

II. Related Works

In this section, we present a brief review of existing works in the three related areas: cluster-based networks, deep reinforcement learning, and UAV trajectory planning. The important abbreviations of this paper are summarized in Table I.

A. Cluster-Based Network

Wireless Sensor Networks and IoT Networks compose many low-cost, low energy consumption devices that are not easy to charge. In such networks, how to save energy is very important. Hierarchical clustering technology is an energy-saving technology [18]. In hierarchical clustering technology, only CH can communicate with Base Station (BS), which reduces the communication energy consumption of nodes and thus improves the network life. In order to further reduce energy consumption of nodes and improve network life, various clustering protocols are proposed, such as LEACH [19], HEED [20], EDIT [21], etc., timely transform CHs in various ways to balance energy consumption of nodes and improve network life.

B. Deep Reinforcement Learning

There are a number of reinforcement learning approaches that are pushing the agenda for general AI applications. DQN [22], DDPG [23], A3C [24], PPO [25] and other methods have been applied in a variety of fields and achieved good results. In related research on UAV path planning, Zhang et al. proposed a DQN-based UAV path planning algorithm, enabling UAV to provide reliable and flexible emergency communication for disaster areas after disasters [26]. Li et al. proposed the UAV ground target tracking algorithm based on DDPG, which enables the UAV to effectively maintain target tracking and avoid obstacles [27]. Zhang et al. considering the problem of UAV-assisted data acquisition in wireless sensor networks, the transmission opportunities of ground sensor nodes and the flight trajectory of energy acquisition UAVs were jointly optimized, and a real-time decision-making framework using A3C algorithm was proposed [28]. Samir et al. considered UAV-assisted IoT networks, where low-resource IoT devices periodically sample a random process and need to upload the latest information to the BS. An online free model based DRL method is proposed. PPO algorithm was used to solve the problem of representation [29].

C. UAV Trajectory Planning

More and more scholars are paying attention to the use of UAVs to collect IoT network data [30]. Generally, UAV path planning solutions proposed in studies can be divided into two categories, one is the solution using traditional algorithms, and the other is the solution using AI algorithms.

The solutions using traditional algorithms usually obtain the flight trajectory of UAV by mathematical planning method or heuristic algorithms such as Simulated Annealing (SA) or Genetic Algorithm (GA) after determining various constraints of the research problem. In [31], the authors proposed a moving edge computing system based on UAV. Through joint optimization of unloading data bits, users and UAV computing frequency and UAV path, a solution to minimize power consumption is proposed, and the path of UAV is solved by successive convex approximation method. In [32], the authors proposed a UAV path trajectory planning scheme for fair data collection, which improves the flight speed of UAV and the fairness of UAV data collection through Dichotomy Algorithm and UAV Speed Planning method. In addition, Ant Colony Optimization (ACO) [33] and Simulated Annealing Optimization (SAO) [34] have also been applied to generate UAV paths for collecting data.

The most commonly used method in AI is DRL. In [35], the authors propose a deep learning framework to deal with the efficient collection of sensor data by multiple UAVs in multiple randomly deployed charging station environments. In [36], the authors design a fully distributed UAV control solution to plan the movement routes of a group of UAVs as Mobile Base Stations (MBSs) to provide long-term communication coverage for ground mobile users. By maximizing UAV average coverage score, maximizing geographic fairness of points of interest, and minimizing total energy consumption, a decentralized DRL-based framework is proposed to control each UAV in a distributed manner.

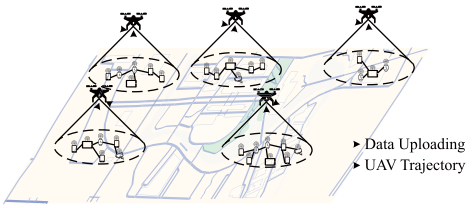

Fig. 1. Illustration of a typical UAV-enabled IoT network with network clustering.

However, at present, both traditional algorithms and AI algorithms of UAV path planning plan focus on optimizing the energy consumption of the UAV itself or the total energy consumption and data offloading efficiency of the UAV, but do not consider the changing of the CHs in cluster-based IoT network. Because these solutions do not take into account the variability of CHs in the network, they are not suitable for cluster-based IoT networks with changeable CHs.

III. System Model and Problem Formulation

This section will introduce the system architecture of the research object, establish a mathematical model, and finally formalize our research problem.

The system model is shown in Fig. 1. IoT devices are randomly deployed in specific areas within a range to sense the environment and collect data. In the case of a large number of IoT devices, in order to improve the efficiency of data collection, clustering technology can be used for IoT devices according to geographical location. That is $C = \{c_i | i = 1, 2, \ldots, m\}$, where c_i represents the i-th cluster. The data sensed by IoT devices in each cluster converge to CH. When the UAV arrives near the CH, the CH can connect to the UAV through the WiFi interface supporting IEEE 802.11 standard and transmit the collected data [37]. In this way, UAV and IoT devices together form a UAV-Enabled IoT network using clustering technology. Since the UAV needs to collect data from each cluster in a " moving-hover" mode, it needs to hover at a specific position when collecting data from a cluster. Therefore, the trajectory of UAV is composed of multiple hover points $\{h_0, h_1, \ldots, h_c, h_{|c|+1}\}$, which are connected in a certain sequence. The starting point for the h_0, end of $h_{|c|+1}$. Therefore, the UAV flight path planning problem can be regarded as the planning problem of multiple hover points.

A. System Model

1) Network Model: The network studied in this paper is a UAV-assisted IoT network that uses a dynamic routing protocol. Compared with static routing protocols, dynamic routing protocols can reasonably distribute network loads and evenly consume the energy of sensing devices, so that the network can cope with data collection and task offloading. Assume that the target network collected by the UAV is located in a rectangular region with side length L. Consider the set of IoT devices in the network as $K = \{1, 2, \ldots, k\}$, and the position of each device K is represented by $(x_k, y_k)_{k \in K}$. IoT devices are clustered geographically, and each cluster selects CH according to its own routing protocol. Since this paper focuses on UAV trajectory planning under dynamic routing protocol, CH elected by each cluster according to routing protocol change with time. So for cluster set C, we have $C = \{1, 2, \ldots, c\}$, and the position of the CH H_c^t is $(x_c^t, y_c^t)_{c \in C}$ where t represents t-th time slot.

In dynamic hierarchical routing protocols, the CH selection process takes the classical LEACH algorithm as an example. In LEACH, the CH selection process is divided into N rounds. In each round, each node decides whether to become the CH according to the threshold value. In each round, each node selects a random number between 0 and 1. If less than the threshold, the node becomes the CH of the current round. For node k, the threshold $T(k)$ is defined as [38]

$$T(k) = \begin{cases} \frac{p}{1 - p\left(r \bmod \frac{1}{p}\right)}, & \text{if } k \in G \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Where p is the expected percentage of CHs, r is the current number of rounds, and G is the set of nodes that were not selected as CHs in the previous round.

After the CH election, each cluster has a CH, and other nodes of the cluster are called member nodes. The CH are responsible for collecting data sent by member nodes. UAV are used to collect data from CH. With the help of UAV, IoT devices can access the Internet through aerial base station, and CH can transmit data to UAV so that the UAV can complete data collection.

2) Communication Model: In this paper, we use the Air-to-ground (A2G) model as the communication model between UAV and IoT devices [13]. For the A2G communication model, when the UAV communicates with H_i^t at hover point h_i, the probability of Line of Sight (LoS) and Non Line of Sight (NLoS) are denoted as $P_{i,LoS}$ and $P_{i,NLoS}$, respectively. $P_{i,LoS}$ can be calculated by

$$P_{i,LoS} = \frac{1}{1 + \alpha \exp\left[-\beta\left(\phi - \alpha\right)\right]}, \tag{2}$$

where α and β are determined by the environment. $\phi = \frac{180}{\pi} \times \sin^{-1}(\frac{H}{d_i})$ represents the degree of angle of attack. d_i represents the Euclidean distance between h_i and H_i^t. H indicates the hover height of the UAV. Therefore, the probability of NLoS is $P_{i,NLoS} = 1 - P_{i,LoS}$. Then the average path loss between H_i^t and the UAV is

$$\bar{P}_{i,L} = P_{i,LoS}\left(K_0 + \lambda_{LoS}\right) + P_{i,NLoS}\left(K_0 + \lambda_{NLoS}\right), \tag{3}$$

where λ_{LoS} and λ_{NLoS} represent the average path loss of LoS link and NLoS link respectively. F$K_0 = 10k \log_{10}(4\pi f_c \frac{H}{c})$, k is path loss index. c stands for the speed of light. f_c represents the carrier frequency. Then the average data transmission rate r_i^t between UAV and H_i^t can be calculated by Shannon formula

$$r_i^t = B \log_2\left(1 + \frac{P_{H_i^t}}{\bar{P}_{i,L} N_0}\right), \tag{4}$$

Where B for communication bandwidth, $P_{H_i^t}$ is transmission power, N_0 is noise power spectral density.

3) UAV Energy Consumption Model: The energy consumption of the UAV is divided into three parts: 1. The energy consumption of the UAV to maintain the flight height and overcome gravity. 2. Movement energy consumption of the UAV during flying from one hover point to another. 3. Communication energy consumption of the UAV when collecting data.

In order to maintain a certain height H during flight or hovering, UAV needs to overcome gravity and do work. The power P_h of this part is given by the following formula [13]

$$P_h = \sqrt{\frac{(m_{uav}g)^3}{2\pi r_p^2 n_p \rho}}, \tag{5}$$

where m_{uav} is the mass of UAV, g is the gravitational acceleration, r_p is the radius of propeller, n_p is the number of propeller, and ρ is the density of air. The horizontal moving flight power P_m of UAV is positively correlated with the moving speed of UAV, which is expressed by the formula

$$P_m = \frac{P_{max} - P_{idle}}{v_{max}} v_{uav} + P_{idle}, \tag{6}$$

where, v_{max} represents the maximum movement speed of UAV, P_{max} represents the flight power of UAV when it moves at full speed, and P_{idle} represents the power consumption of UAV under ideal conditions. Since the flight path of UAV is composed of multiple hover points $\{h_0, h_1, \ldots, h_c, h_{|c|+1}\}$, which are connected in a certain sequence. We consider that the UAV flies at H altitude, let the coordinate of hover point h_i be (x_i, y_i, H), and the coordinate of hover point h_j be (x_j, y_j, H), then the Euclidean distance between these two points is

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \tag{7}$$

let T be the total flight time of UAV, and its calculation method is

$$T = \frac{1}{v_{uav}} \sum_{i=0}^{c+1} \sum_{j=0, j\neq i}^{c+1} d_{ij} L_{ij}, \tag{8}$$

where L_{ij} represents whether the UAV flies to hover point h_j after reaching hover point h_i. If $L_{ij} = 1$, it indicates that the UAV flies from hover point h_i to hover point h_j; If $L_{ij} = 0$, it indicates that there is no adjacent access order between hover point h_i and hover point h_j. Then the total flight energy consumption of UAV E_f is

$$E_f = T\left(P_h + P_m\right), \tag{9}$$

this formula indicates that in the process of UAV flight, the flight energy consumption includes the energy consumption of UAV to maintain the height and the energy consumption of the UAV to travel.

The energy consumed in collecting the data of $H_i t$, the CH of the ith cluster, at the hover point h_i, is written as E_i. Assuming that the data volume of $H_i t$ is M_i^t, the calculation formula of E_i is

$$E_i = \frac{M_i^t}{r_i^t}\left(P_h + P_c\right), \tag{10}$$

where P_c represents the power consumption of communication between UAV and IoT devices, which is related to hardware devices. Then the energy consumed by the UAV in collecting data of all CHs in the network is

$$E = \sum_{i=1}^{c} E_i + E_f. \tag{11}$$

B. Problem Formulation

In this section, we will formalize the main research questions with the following background. IoT devices are randomly deployed in the target area D with side length L, and these devices are divided into clusters according to geographical location. Assuming that IoT devices in the network are divided into m clusters, clusters can be represented by $C = \{c_i | i = 1, 2, \ldots, m\}$. Each cluster dynamically selects the CH, that is, the dynamic cluster routing protocol is used. The CH collects the data sent by member nodes. At the time slot t, the CH set of the network is defined as $H_t = \{H_i^t | i = 1, 2, \ldots, m\}$, H_i^t means that at time slot t, the CH of the cluster i is H_i^t. The UAV starts from the starting point, moves according to the planned flight path, and is responsible for collecting data generated by all clusters in the network. Finally, it flies to the end and transmits the collected data back to the data center. Since the flight mode of UAV is "moving-hover" mode, that is, UAV keeps hovering at a high altitude when collecting data, but moves in other cases, which is a common UAV movement mode in other studies [39]. Therefore, the flight path of UAV should include the following two parts, one is the hovering position of UAV when collecting data, and the other is the sequence of UAV visiting these hovering points. In this way, a path can be formed by the position and access sequence, which is the flight path of UAV.

If we use Γ to represent the flight path of UAV, let P be the set of hover points and V be the order in which the UAV visits the hover points, we can get that $\Gamma = \langle P, V \rangle$. So a path of the UAV is related to the hover points of the UAV and the order in which these hover points are visited. Let the set of the hover points be $P = \{h_i | i = 1, 2, \ldots, n\}$ and it is easy to know that $|P| = |C| + 2 = m + 2$. That is, the number of hover points should be the number of clusters in the network plus the start and end points of the path.

However, since the position of the CH in the network is not fixed, the UAV cannot know the geographic location of the CH H_i^t of the target cluster c_i before reaching the hovering point (or during the flight). When the UAV reaches the hover point, it may happen that the CH is not within the reliable communication range of the UAV, which we call the non-ideal situation.

Fig. 2 is a schematic diagram of the UAV adjusting the hover point to collect data for clusters. First, the hovering points of the UAV in clusters c_n and c_{n+1} are h_n and h_{n+1} respectively, and the access order of h_n and h_{n+1} is $h_n \rightarrow h_{n+1}$, that is, $L_{n,n+1} = 1$. When the UAV reaches the hovering point h_n, it will wake up the IoT devices within the communication range of the UAV. At this time slot t, if the CH H_n^t of the cluster is within the communication range of the UAV, then the UAV will communicate with H_n^t using the TDMA to complete the data

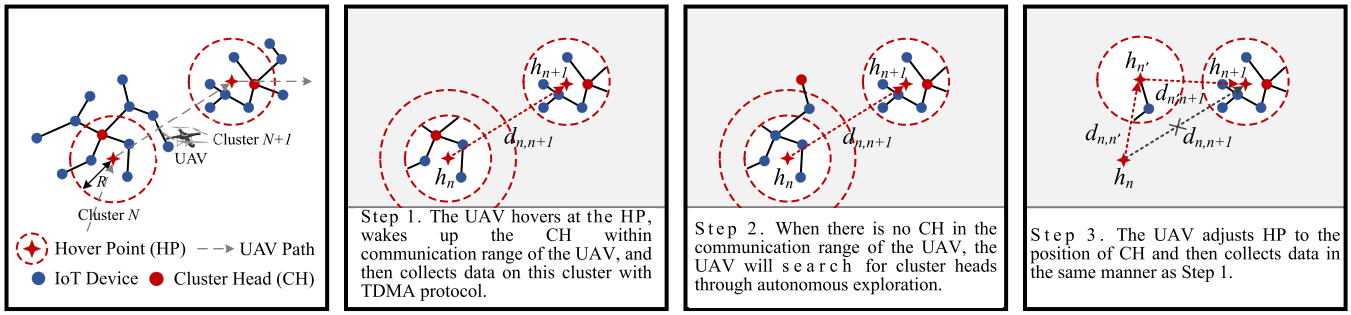

Fig. 2. Illustration of searching the CH.

collection of the cluster c_n, and then the UAV will communicate with H_n^t using the TDMA protocol. Then fly to cluster c_{n+1}, and collect the data of cluster c_{n+1} with the same steps. However, if the CH H_n^t of the cluster is not within the communication range of the UAV, the UAV will search for the location of the CH until the CH is within the communication range of the UAV. After the UAV collects the data of the cluster c_n, the UAV will head to c_{n+1}. In this process, the distance traveled by the UAV is $\tilde{d}_{n,n+1}$ and $\tilde{d}_{n,n+1} > d_{n,n+1}$. Therefore, compared with the ideal situation, the UAV consumes more energy in the non-ideal situation. The occurrence of the non-ideal situation is related to the position of the hovering point of the UAV, so the main problem of this paper is to optimize the UAV. The hovering point of the aircraft, in the case of meeting the CH as much as possible, plan a most energy-efficient flight path, we define this problem as problem $\mathbf{P}$. We use formal language reduction to problem $\mathbf{P}$.

$$\mathbf{P}: \min_{\Gamma=<P,V>} \sum_{i=1}^{|C|} E_i + E_f \tag{12}$$

$$\text{s.t. C1:} \sum_{i=0, i\neq j}^{|P|-1} L_{ij} = 1 \tag{13}$$

$$\text{C2:} \sum_{j=0, j\neq i}^{|P|-1} L_{ij} = 1 \tag{14}$$

$$\text{C3:} \sum_{h_i \in SHP} \sum_{h_j \in \overline{SHP}} L_{ij} \geqslant 1 \tag{15}$$

$$\text{C4:} \quad L_{ij} \in \{0, 1\} \tag{16}$$

$$\text{C5:} \quad |C| + 2 \leqslant |P| \leqslant 2\,|C| + 2 \tag{17}$$

$$\text{C6:} \quad \forall h_i \in D \tag{18}$$

In problem $\mathbf{P}$, the objective function is to minimize the total energy consumption of the UAV. C1 and C2 ensure that the out-degree and in-degree of each hover point are 1. C3 is to eliminate sub-loops in the path, that is, the path of the drone should form a closed loop containing all nodes. C4 guarantees that the connectivity between two points conforms to duality. C5 indicates that the number of hover points can only be between $|C| + 2$ and $2|C| + 2$. The best case is that every time the drone collects cluster data, the CH is in the communication of the drone. Within the range, the drone does not need to adjust the hovering point, and the number of hovering points is the number of clusters plus 2; in the worst case, each time the cluster data is collected, the CH is not within the communication range of the drone, then the drone needs to adjust the hover point every time, and the number of clusters with twice the number of hover points is plus 2. C6 ensures that the hover point can only be within the target area D.

IV. DRL-Based UAV Trajectory Planning Solution (DRL-UTPS) for Date Collection in IoT Network

In this section, we propose a UAV data collection path planning algorithm based on reinforcement learning method to solve the flight trajectory with the least energy consumption when the CH changes with time. The UAV learns a strategy for controlling and optimizing the hovering points, and then solves the TSP problem through SA to obtain the flight trajectory with the shortest distance.

We formulate our problem as a Markov Decision Process (MDP), which can be expressed as a 5-tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, R, P, \gamma \rangle$ where $\mathcal{S}$ is the state set, $\mathcal{A}$ is the action set, R is the value function, P is the state transition function, γ is the discount rate, $\gamma \in (0, 1]$ Divide the Markov decision process into multiple time slots $t \in \{1, 2, \ldots, T\}$, then when the agent takes the action $a_t \in \mathcal{A}$ when the state of the time slot t is $s_t \in \mathcal{S}$ the agent will get the reward $r_t = R(s_t, a_t)$, and then the state will be transformed into $s_{t+1} \in \mathcal{S}$ according to the state transition function $P(s_{t+1}|s_t, a_t)$. Repeat the above process to get a series of decision trajectories

$$\tau = \langle s_0, a_0, r_1, s_1, a_1, r_2, \ldots, s_{T-1}, a_{T-1}, r_T \rangle$$

The goal of the reinforcement learning method is to find the strategy with the maximum cumulative reward. In order to apply the reinforcement learning method to plan the flight trajectory for the UAV, we need to model the hover point optimization process of the UAV by MDP, and then use the reinforcement learning method to find The optimization strategy of MDP, then MDP is modeled as follows.

1) State($\mathcal{S}$): In the time slot t, the state of the drone can be determined by the position of the uncorrected hover point $P_t = \{h_1^t, h_2^t, \ldots, h_{|C|}^t\}$ and the position of the CH

in the network $H_t = \{ch_1^t, ch_2^t, \ldots, ch_{|C|}^t\}$ is determined, that is, $s_t = \langle P_t, H_t \rangle$. where $h_i^t = [x_i^t, y_i^t]$represents the collection of the first i by the drone in time slot t Hover point of the data of a cluster, $ch_i^t = [x_i^t, y_i^t]$ represents the position of i clusters in time slot t. So State can be defined as $\mathcal{S} \triangleq \langle s_t \rangle = \{P_t, H_t\}$.

2) Action($\mathcal{A}$): We use reinforcement learning to optimize the hover point of the drone to plan the optimal flight trajectory for the drone. The hovering point of the drone should be within the target area D with a side length of L, then you can use α_i^t, β_i^trepresents the scaling factor of the drone hovering point h_i^t in the x axis direction and y axis direction relative to the side length L of the target area, that is, $h_i^t = (\alpha_i^t L, \beta_i^t L)$. So we define $\mathcal{A} \triangleq \langle a_t \rangle = \{\alpha_1^t, \beta_1^t, \alpha_2^t, \beta_2^t, \ldots, \alpha_{|C|}^t, \beta_{|C|}^t\}$. The size of the action space is $2|C|$.

3) Reward($R(\cdot, \cdot)$): The reward function should include the objective function 12. In addition, it should also consider whether there is a target CH within the communication range of the hovering point of the UAV. If there is no CH, the distance between the hovering point and the CH should be used to guide the optimization direction.. So the reward function is defined as $R(s_t, a_t) = \mu_1 \sum_{i=1}^{|C|} Hit(h_i^t) - \sum_{i=1}^{|C|} E_i m - \frac{P_h + P_m}{v_{uav}} \sum_{i=0}^{|P|+1} \sum_{j=0, j \neq i}^{|P|+1} \tilde{d}_{ij} L_{\text{ij}}$, where μ_1 is the reward coefficient of whether there is a target CH within the communication range of the hovering point, and the coefficient is a positive number. The first item of the reward function represents the reward value of how many times the hovering point of the UAV communicates with the corresponding CH, the second item is the communication energy consumption between the UAV and the device, and the third item is the mobile flight of the UAV Energy consumption, the energy consumption of moving flight includes the energy consumption of flying between clusters, and the energy consumption of the drone flying to the adjusted hovering point.

Algorithm 1: DRL-UTPS

Input: Learning rate, environment parameters (network model, communication model, energy consumption model of UAV).
Output: Hovering point of UAV, access sequence.

Randomly initialize network parameters θ, initialize policy $\pi_\theta old$, and $\theta old \leftarrow \theta$.
for *episode = 0,1,2,...* **do**
  for t *= 0,1,2,...,*T **do**
    Observe the state of hovering point P_t and CH H_t. Union them as s_t.
    Get a_t based on policy $\pi_{\theta old}$.
    Execute a_t to get the hovering point.
    Use Simulated Annealing (SA) to get the access sequence V.
    for i *= 1,2,...,*$|C|$ **do**
      UAV obtain the location of clster head ch_i^{t+1} of cluster c_i.
      if *CH* ch_i^{t+1} *is out of the communication range of UAV when UAV at the hovering point* h_i **then**
        Adjust hovering point h_i to $h_{i'}$.
        Update $P = P \cup \{h_{i'}\}$.
      end
    end
    Get new state s_{t+1}.
    Obtain the reward $R(s_t, a_t)$.
    Store $\langle s_t, a_t, R(s_t, a_t), s_{t+1} \rangle$ in D for trainning.
    Compute advantage estimation.
  end
  for $epoch$ *= 0,1,2,...* **do**
    Update actor network by using clipped loss function
    $\theta = \arg\max_\theta \frac{1}{|D|T} \sum_{\tau \in D} \sum_{t=0}^{T}$
    $\min\left(\eta_\theta A_{\pi'}(s_t, a_t), \text{clip}\left(\eta_\theta, 1-\delta, 1+\delta\right) A_{\pi'}(s_t, a_t)\right)$
    Update critic network by using mean-squared error function
    $\phi = \arg\min_\theta \frac{1}{|D|T} \sum_{\tau \in D} \sum_{t=0}^{T} \left(V_\phi(s_t) - \hat{R}\right)^2$
  end
end

A. Reinforcement Learning - PPO

Proximal Policy Optimization (PPO) is a state-of-the-art reinforcement learning method. Due to the high performance and ease of use of PPO, OpenAI uses PPO as the default reinforcement learning algorithm. Next we will illustrate how DRL-UTPS uses PPO for UAV trajectory optimization. We first give the state action value function $Q_\pi(s_t, a_t)$, the state value function $V_\pi(s_t)$ and the advantage function $A_\pi(s_t, a_t)$.

$$Q_\pi(s_t, a_t) = E_{s_{t+1}, a_{t+1}, \cdots}\left[\sum_{l=0}^{\infty} \gamma^t r_{t+l}\right], \tag{19}$$

$$V_\pi(s_t) = E_{a_t, s_{t+1}, \cdots}\left[\sum_{l=0}^{\infty} \gamma^t r_{t+l}\right], \tag{20}$$

$$A_\pi(s_t, a_t) = Q_\pi(s_t, a_t) - V_\pi(s_t), \tag{21}$$

PPO considers the influence of bias and variance on the model, and uses generalized advantage estimation (GAE) to balance the bias and variance of the estimated value. The GAE estimator is the following formula

$$A_t = \sum_{l=0}^{\infty} (\gamma\phi)^l \kappa_{t+l}^V, \tag{22}$$

where ϕ is used to balance bias and variance, and κ_{t+l}^V can be calculated by

$$\kappa_t^V = r_t + \gamma V_{\pi'}(s_{t+1}) - V_{\pi'}(s_t), \tag{23}$$

We define the updated policy function $\pi_\theta(a_t|s_t)$ and the pre-updated policy function $\pi_{\theta'}(a_t|s_t)$ is η_θ, that is

$$\eta_\theta = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta'}(a_t|s_t)}, \tag{24}$$

Then the loss function of PPO is defined as

$$J_{ppo}(\theta) = E\left[\min\left(\eta_\theta A_{\pi'}(s_t, a_t), \text{clip}\left(\eta_\theta, 1-\delta, 1+\delta\right) A_{\pi'}(s_t, a_t)\right)\right]. \tag{25}$$

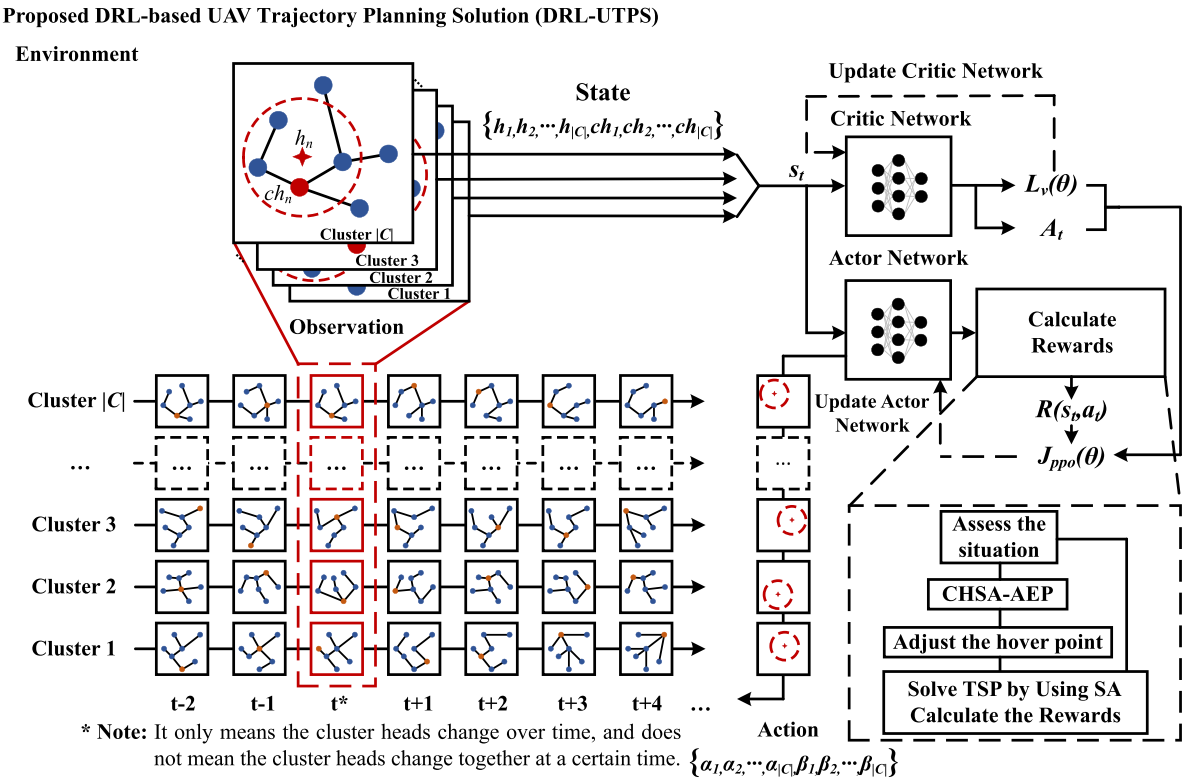

Fig. 3. Illustration of DRL-UTPS.

B. Framework of DRL-UTPS

The overall structure of DRL-UTPS is shown in Fig. 3. Since the scheme is based on the PPO algorithm, it is model-free. Model-free means that DRL-UTPS needs to continuously interact with the environment before it can finally learn an appropriate policy [40]. Then the process of DRL-UTPS is shown in Algorithm 1.

First initialize the parameters of the actor network and critic network. Then, for each round of interaction with the environment, the UAV will observe the hover point P_t given by the decision and the position H_t of each CH, and combine them as s_t. Based on the policy π_θ, according to the state s_t, the UAV will get a a_t, and from this a_t, the UAV will get a new hover point. The optimal access order is calculated by the SA method for these hovering points, thus obtaining a UAV path according to the current s_t. When the UAV accesses all the clusters in the network according to this access path, the UAV will observe the position of each CH. If the CH is not within the communication range of the hovering point where the UAV is located, no one will The machine will adjust the hover point to reach the exact CH position, and this adjusted position will be included in the hover point set. When the UAV has visited all the clusters, that is, after completing a round of data collection tasks, the UAV will get a new state s_{t+1}. The new state s_{t+1} is the unadjusted hover point obtained by the UAV through a_t and the hover point adjusted according to the situation after reaching the top of the cluster, and the arrival between these hover points composed in sequence. Then get $R(s_t, a_t)$ through the reward function, store this tuple in D for training, and finally calculate the dominance estimate. Finally, the parameters of the network actor and critic are optimized by the Clipped loss function and the mean-squared error function respectively. When the hover point is planned, the UAV needs to arrive at the position of the hover point to collect the data on the CH of the corresponding cluster.We designed CHSA-AEP to search for the CH. The CHSA-RP pseudo code of the algorithm is shown in Algorithm 2. After adjust the hover points, it is necessary to optimize the access sequence of the UAV according to the position of the hover points. This problem can be regarded as a TSP problem. In order to quickly obtain satisfactory optimization results, the simulated annealing algorithm used in this paper is used. See Algorithm 3 for pseudo code of simulated annealing to solve the TSP problem.

V. Performance and Analysis

In this section, we will implement DRL-UTPS algorithm and compare it with the baseline of UAV data acquisition path planning algorithm, which are GBA and CPP-SDA respectively. DRL-UTPS collects data by planning UAV hovering points.

- DRL-UTPS (CHSA-AEP): The hover points of the UAV optimized by PPO are obtained. When the UAV reaches one of the hover points but fails to communicate with CH, the UAV will use CHSA-AEP to search for the CH. Otherwise, the UAV can communicate directly with the CH and complete data collection.

Since a lot of researches are to find the CH by covering the region of interest, we take typical region covering algorithms as

Algorithm 2: Cluster-head Searching Algorithm with Autonomous Exploration Pattern (CHSA-AEP)

Input: the location of hovering point P_t, CH H_t and the communications range of the UAV R.
Output: Adjusted hovering point.

for i = 1,2,...,$|C|$ **do**
  $d \leftarrow 0$.
  if $\sqrt{(x_{h_i} - x_{ch_i})^2 + (y_{h_i} - y_{ch_i})^2} < R$ **then**
    Collect the data of CH ch_i.
  else
    while ch_i *is not in the communication range of the UAV* **do**
      $d \leftarrow d + R$.
      The UAV moves clockwise or counterclockwise with (x_{h_i}, y_{h_i}) as the center and d as the radius.
    end
    The UAV adjusts the hover point to the CH.
  end
end

Algorithm 3: Solve TSP by Using Simulated Annealing Algorithm.

Input: the location of hovering point P_t, iteration parameter I_0.
Output: The order V in which the UAV visits these hover points.

Initialize the iteration boundary I_{min} and initial order V.
$I \leftarrow I_0$.
while $I > I_{min}$ **do**
  Perturb V to produce a new solution V_{new}.
  Calculate the difference between the length of UAV path obtained in the order of V_{new} and that obtained in the order of V, and write it as dD.
  if $dD > 0$ **then**
    if $exp(dD/I) > rand(0, 1)$ **then**
      $V \leftarrow V_{new}$.
    end
    $V \leftarrow V_{new}$.
  else
    $I \leftarrow rI$.
  end
end

the comparison algorithm. In the area coverage problem, GBA and CPP-SDA are representative UAV path planning algorithms:

- GBA [16]: This method divides the large network into smaller grids, establishes the task environment in the grids, and divides the whole region through the grid clustering method, so as to obtain a path that meets the CPP problem. This method is a representative scheme of path planning based on grid.
- CPP-SDA [17]: The core problem solved by this method is the coverage path planning problem. By dividing the problem into two steps: the optimization of the access order problem and the optimization of the route direction problem, it proposes a method to solve the CPP in the disjoint area. The method is a representative scheme based on the back and Forth pattern.

In addition, Random algorithm is added as a comparison.

- Random (CHSA-AEP): This method randomly selects a device in each cluster as the hover point. When the UAV reaches the hovering point and there are CHs within the communication range to respond, data will be collected; otherwise, CHSA-AEP method will be adopted to search for CHs and adjust the hover point to collect data.

TABLE II
THE VALUE OF THE SIMULATION PARAMETERS

Notation	Value	Unit	Ref.
L	1000	m	[-]
$\|C\|$	8, 10, 12, 14	-	[-]
$\|N\|$	25, 50, 75, 100	-	[-]
R	10, 15, 20, 25, 30	m	[-]
B	1	MHz	[31]
N_0	-174	dBm/Hz	[31]
$P_{H_i^t}$	21	dBm/Hz	[31]
$P_{i,LoS}$	1	dB	[31]
$P_{i,NLoS}$	20	dB	[31]
α, β	0.03, 10	-	[31]
H	50	m	[14]
v_{uav}	15	m/s	[31]
m_{uav}	0.5	kg	[31]
r_p	0.2	m	[31]
n_p	4	-	[31]
P_{max}	5	W	[31]
P_{idle}	0	W	[31]
P_c	0.0126	W	[31]

Next, we first give experimental parameters, and then analyze the convergence of DRL-UTPS. Then, DRL-UTPS proposed in this paper is compared with other popular algorithms GBA, CPP-SDA and Random method. By adjusting the network scale, the number of regions of interest and the communication radius of UAV, the differences of each method in UAV energy consumption and time utilization ratio are compared.

A. Experiment Settings

The size of the IoT network used in the simulation experiment is $1000m \times 1000m$. There are 8, 10, 12, 14, 16 clusters in the network, and each cluster has 25, 50, 75, and 100 IoT devices, which are randomly deployed in the network. In addition, the dynamic hierarchical routing protocol used by each cluster is the LEACH [38] protocol. See Table II for other major parameters. The operating system environment of the simulation experiment is Windows 10, the CPU is Intel Core i5-6400 T, the main frequency is 2.20 GHz, and the programming language version is Python 3.6 with Keras 2.6.

B. Convergence

After DRL-UTPS is implemented, we first test the convergence of the proposed model. Episode was set as 800000, and the model of the number of different clusters, the number of devices in each cluster and the communication radius of UAV was trained respectively. "C10N50R20" means that there are 10 clusters in the network, and each cluster has 50 IoT devices, the communication radius of UAV is 20 m, and other legends follow the same pattern. Then the functional relationship between episode and reward is shown in Fig. 4. Here an episode means that the UAV collects data of all clusters in a network six times in total.

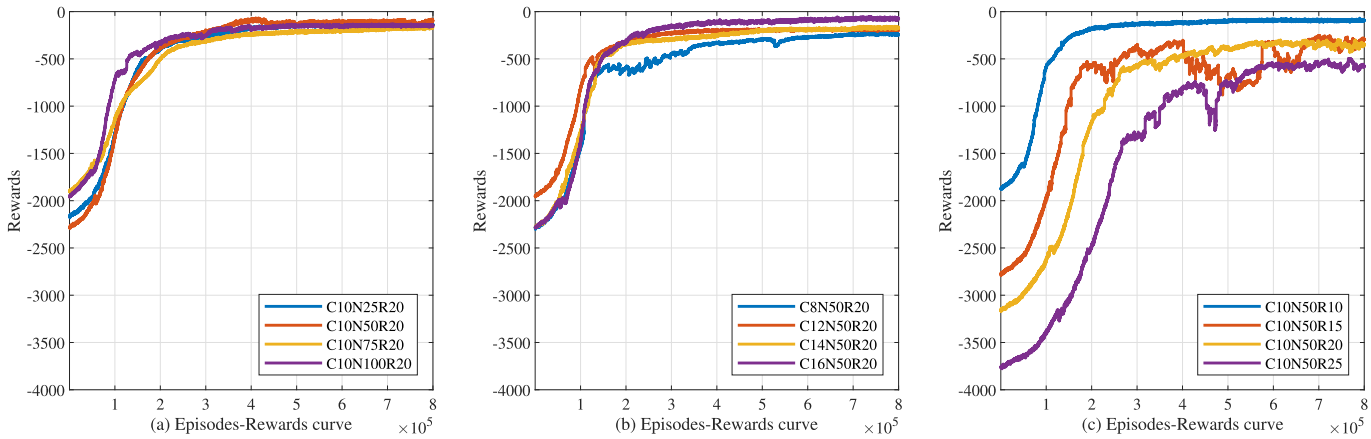

Fig. 4. Learning curve with episode = 800000.

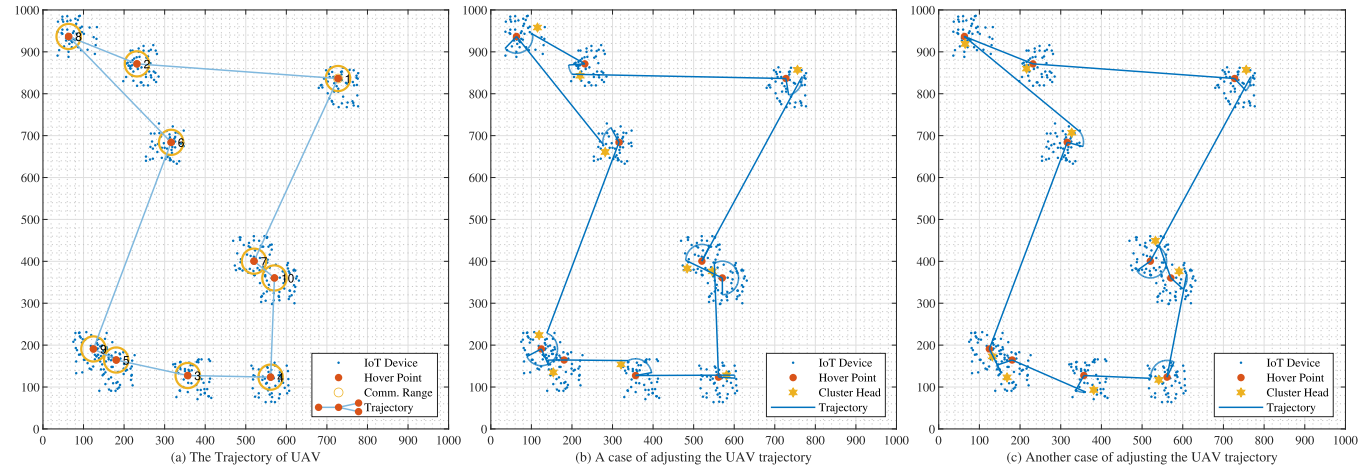

Fig. 5. The dynamic flight trajectory of the UAV.

As can be seen from Fig. 4, reward of most models increased rapidly before 300,000 episodes. After 300,000 episodes, the value of rewards becomes stable and oscillates around a certain value until the training ends. These results indicate that the DRL-UTPS model can converge successfully after training, and can make a decision with a high reward in a given system state. It also means that with trained models, UAV can collect data in an efficient and energy efficient way and return to the origin after collecting the data.

C. Dynamic Adjustment of the UAV Trajectory

Fig. 5 is a schematic diagram of the flight trajectory of the UAV in data collection. Fig. 5(a) shows the general flight trajectory of the UAV. Fig. 5(b) and Fig. 5(c) simultaneously show the changes in UAV flight trajectory when facing the following scenarios: 1. The UAV directly communicates with the CH; 2. The UAV needs to use CHSA-AEP to search the CH. The horizontal comparison between Fig. 5(b) and Fig. 5(c) shows that when the CHs in the network change, the flight trajectory of the UAV will also change accordingly. And the horizontal comparison also shows that the UAV can effectively find the CH through CHSA-AEP without bringing a long flight distance. This illustrates the effectiveness of DRL-UTPS (CHSA-AEP). In the next few sections, we will show the efficiency of DRL-UTPS.

D. The Gap of the Hover Points Selection

In this section, we compare DRL-UTPS with the Random algorithm, and analyze the gap between the two algorithms in choosing the position of the hover points.

From Fig. 5(a), We can see the trajectory of the UAV and the hover points selected by DRL-UTPS. Fig. 6(c) and Fig. 6(d) show the difference in data access (DA) area between DRL-UTPS and Random algorithms. Here, DA area represents the area that UAV needs to fly to complete the data collection task of a cluster. In heatmaps Fig. 6(c) and Fig. 6(d), the brighter the color, the higher the frequency of UAV visits to the area. Therefore, it can be seen from Fig. 6(c) and Fig. 6(d) that the DA area of DRL-UTPS is more concentrated and smaller, while the DA area of Random algorithm is more scattered and larger. It

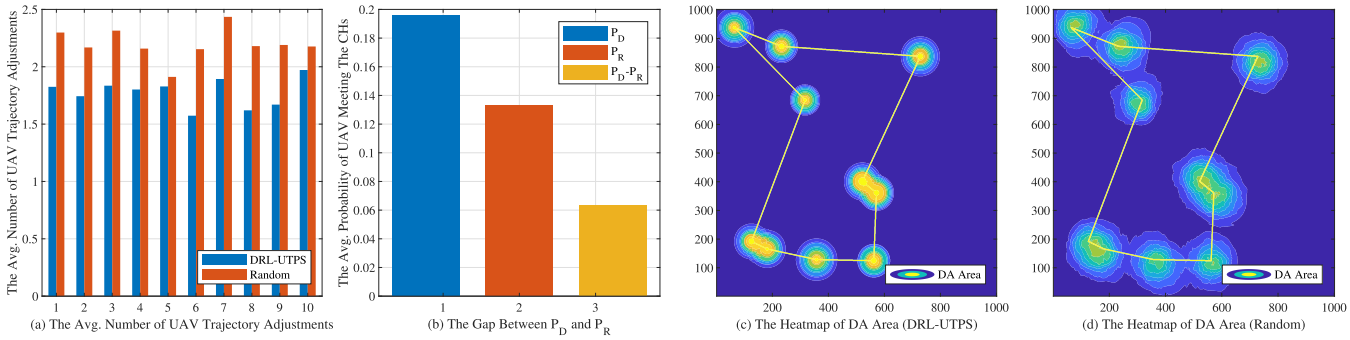

Fig. 6. Simulation results of DRL-UTPS and Random algorithm.

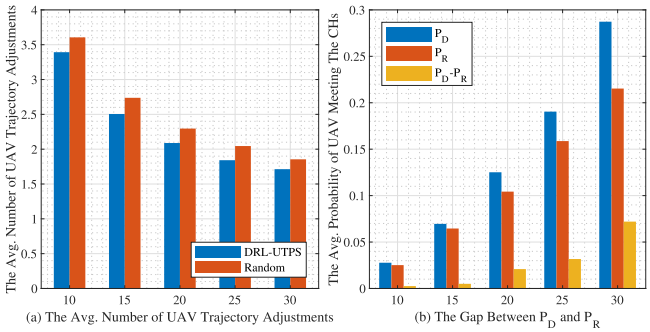

Fig. 7. Simulation results of DRL-UTPS and Random algorithm.

shows that compared with the Random algorithm, DRL-UTPS can reduce the range of the UAV to explore the CH. In the case that the CH cannot be met at the first time, the UAV can explore a smaller area to find the CH.

From Fig. 6(a), we can also see that compared with the Random algorithm, the average adjustment times of the UAV trajectory of DRL-UTPS is lower than that of the Random algorithm, which also shows that DRL-UTPS is easier to meet CHs than Random. From Fig. 6(b), we can see that DRL-UTPS has a greater probability of direct communication with CH at the hover point than Random algorithm. From Fig. 7, the simulation compares the performance differences of the probability of direct communication with CH at the hover point of the two algorithms with different communication radii. In each communication radius condition, the number of devices per cluster was tested separately from 100 to 250 in steps of 25. Under the conditions of a certain communication radius and a certain number of devices, 1000 random simulation tests are carried out, and the average value of the results is calculated. Then the test result corresponding to each radius in the figure is obtained by taking the average of $7 \times 1,000 = 7,000$ random test results. Then a total of 35,000 random simulation experiments were carried out. The experimental results show that under the conditions of different communication radii, the average probability of direct communication with CH at the hover point of DRL-UTPS is relatively higher than that of the Random algorithm, and the average number of UAV flight trajectory adjustments is lower.

E. Time Utilization Ratio

In this section, we compare the time utilization ratio between DRL-UTPS, GBA, CPP-SDA and Random algorithms. Time utilization ratio represents the proportion of the time the UAV spends collecting data in the total working time. The lower the time utilization ratio is, the higher the proportion of time spent by UAV in flight.

We analyze the difference of time utilization ratio of different algorithms by changing the number of clusters in the network. Experimental results are shown in Fig. 8(a). The number of IoT devices in each cluster was fixed at $|K| = 50$. The communication radius of UAV was fixed as $R = 20m$. UAV searched CHs using CHSA-AEP algorithm. It can be found that as the number of clusters in the network changes, the time utilization ratio of DRL-UTPS is higher than other algorithms, and the GBA, CPP-SDA has the worst performance. The increase of the number of clusters in the network does not have a very obvious impact on the time utilization of UAV. The difference in the time utilization of various algorithms is largely caused by the performance difference of the strategies themselves. It can be seen that DRL-UTPS can increase the time utilization ratio by 41.2% on average compared with GBA and CPP-SDA and 28.1% on average compared with Random algorithm.

Then, we analyze the influence of UAV communication radius on time utilization ratio. Experimental results are shown in Fig. 8(b). The number of IoT devices in each cluster was fixed at $|K| = 50$. The number of clusters was fixed for $|C| = 10$. UAV searched CHs using CHSA-AEP algorithm. With the increase of UAV communication radius, the time utilization of each algorithm increases to a certain extent. This is also because the increase of UAV radius improves the coverage efficiency of area coverage algorithm and the probability of DRL-UTPS and Random algorithm communicating with CH at hovering point. It can be seen that DRL-UTPS can increase the time utilization ratio by 32.3% on average compared with GBA and CPP-SDA and 30.8% on average compared with Random algorithm.

Finally, we analyze the difference in time utilization ratio of each algorithm by changing the number of IoT devices in each cluster. Experimental results are shown in Fig. 8(c). The number of clusters was fixed to $|C| = 10$. The communication radius of UAV was fixed to $R = 20m$. UAV searched CHs using CHSA-AEP algorithm. It can be found that compared with GBA, CPP-SDA and Random algorithm, DRL-UTPS has the

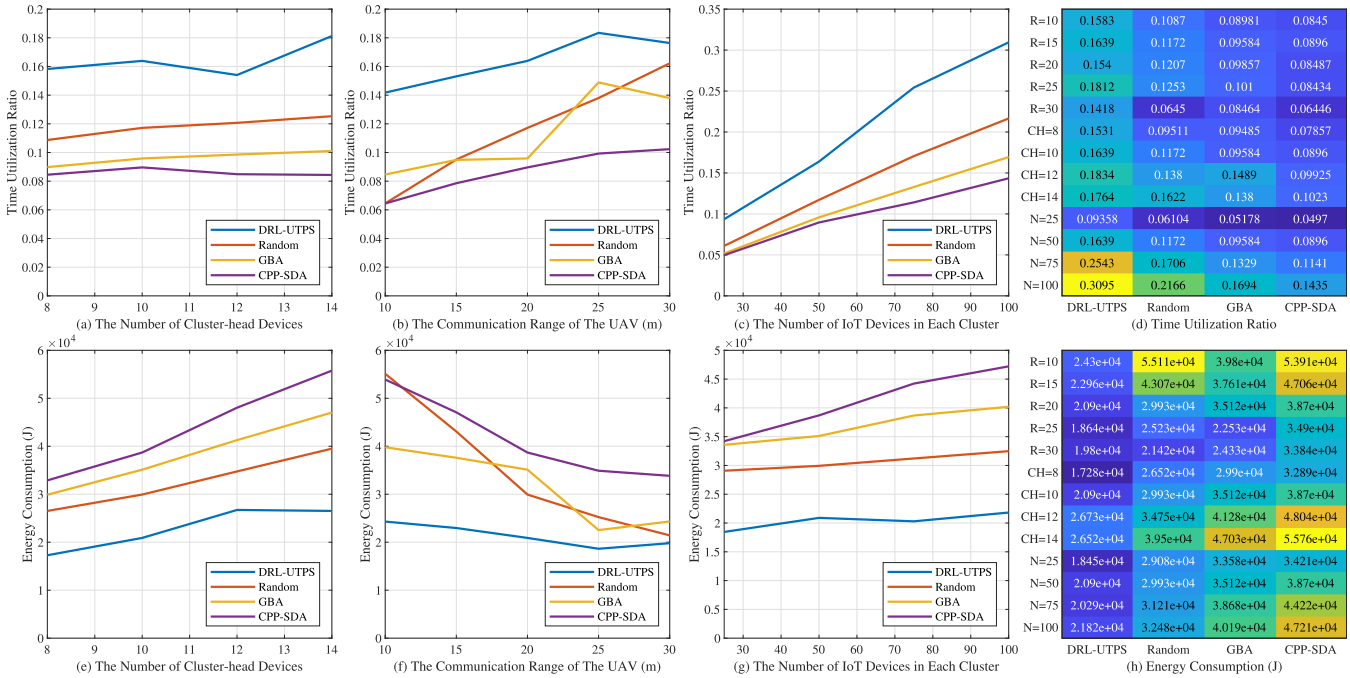

Fig. 8. Experimental results of UAV energy consumption and UAV time utilization ratio.

highest time utilization ratio. With the increase of the number of IoT devices in each cluster, the time utilization ratio of each algorithm also increases. Compared with BGA and CPP-SDA, DRL-UTPS and Random have better performance. Among them, DRL-UTPS can select a more suitable hover point for UAV, which improves the possibility of communication with CHs, thus improving the time utilization ratio. It can be seen that DRL-UTPS can improve the time utilization ratio by 44.8% on average compared with GBA and CPP-SDA and 31.5% on average compared with Random algorithm.

From the performance heat map of each algorithm in Fig. 8(d), it can be seen that the performance of DRL-UTPS is better than that of GBA and CPP-SDA and Random algorithm in each measured dimension.

F. Energy Consumption

In this section, we compare the differences of UAV energy consumption between DRL-UTPS, GBA, CPP-SDA and Random algorithms.

We first analyze the difference in UAV energy consumption of each algorithm by changing the number of clusters in the network. Experimental results are shown in Fig. 8(e). The number of IoT devices in each cluster was fixed to $|K| = 50$. The communication radius of UAV was fixed to $R = 20m$. UAV searched CHs using CHSA-AEP algorithm. From Fig. 8(e), it can be found that compared with GBA, CPP-SDA and Random algorithm, DRL-UTPS has lower energy consumption. The energy consumption of GBA is similar to CPP-SDA. Although Random algorithm has some advantages in performance compared with the path planning algorithm that only uses area coverage strategy, it is still inferior to DRL-UTPS. This is because DRL-UTPS can learn better hover points of UAV. Compared with randomly selecting hover points, UAV can communicate with CHs with a high probability, thus reducing the flight energy consumption of UAV. It can be seen that DRL-UTPS can reduce energy consumption by 40.6% on average compared with GBA and CP-SDA and 33.1% on average relative to Random algorithm.

Then, we analyze the influence of UAV communication radius on UAV energy consumption. Experimental results are shown in Fig. 8(f). The number of IoT devices in each cluster was fixed to $|K| = 50$. The number of clusters was fixed to $|C| = 10$. UAV searched CHs using CHSA-AEP algorithm. It can be found that as the communication radius of UAV increases, the differences between algorithms gradually become smaller. This reflects that with the increase of UAV communication radius, the probability of UAV to establish communication with CH at hover point increases, thus reducing UAV flight energy consumption. However, it can also be found that the performance of DRL-UTPS is superior to other algorithms. Because DRL-UTPS can learn more appropriate hover points for UAV, it is more likely to communicate directly with CHs at the hover points, thus further reducing UAV energy consumption. It can be seen that DRL-UTPS can reduce the energy consumption by 44.2% on average compared with GBA and CPP-SDA and 30.2% on average relative to Random algorithm.

Finally, we analyze the difference of UAV energy consumption of different algorithms by changing the number of IoT devices in each cluster. Experimental results are shown in Fig. 8(g). The number of clusters was fixed to $|C| = 10$. The communication radius of UAV was fixed to $R = 20m$. UAV searched CHs using CHSA-AEP algorithm. From Fig. 8(g), the same conclusion can still be obtained as the previous analysis, that is, DRL-UTPS has better performance than using only the area coverage algorithm, and the hover point obtained by DRL-UTPS is also better than that obtained by Random algorithm. It can be

seen that DRL-UTPS can reduce the energy consumption by 47.5% on average relative to GBA and CPP-SDA and 33.6% on average relative to Random algorithm.

From the performance heat map of each algorithm in Fig. 8(h), it can be seen that the performance of DRL-UTPS is better than that of GBA and CPP-SDA and Random algorithm in each measured dimension.

VI. Discussions and Conclusion

A. Limitations of DRL-UTPS

1) **Lack of data protection.** With the application of technologies such as digital twin [41], data collected by IoT devices may be sensitive, and data protection technology is not considered when UAVs collect data. Attackers may deploy malicious UAVs to collect sensitive data from IoT devices. We can consider using Block-chain technology [42] to enhance data protection.
2) **The influence of obstacles and terrain on the movement of the UAV is not considered.** In the real scenarios, the environment faced by the UAV is complex. In terms of different terrain and obstacles in the environment, the UAV needs to make correct responses to various environments and obstacles to avoid dangers.

B. Potential Direction

1) Studies reinforcement learning strategies for data collection in mobile IoT scenarios. By improving the network architecture, the motion patterns of moving devices can be learned and predicted, and on this basis, the possible distribution of moving CHs can be learned, so as to collect data.
2) A 3D simulation environment can be established and obstacles can be introduced to enable the UAV to collect IoT device data in an environment more consistent with the real scenarios.

C. Conclusion

It is a hot research topic to collect data generated by IoT networks with high mobility of UAVs. This paper proposes an efficient and energy-saving path planning scheme for UAV to collect data, DRL-UTPS, to solve the problem that UAV cannot know the real-time location of CHs in advance when communication is blocked or poor. DRL-UTPS learns the historical location information of CHs in IoT networks through DRL and adjusts the hover points of UAV, so that UAV can establish communication with CHs directly with a high probability after reaching the hover points. When the UAV reaches the hover point, if it cannot establish communication with the CH, DRL-UTPS will search the CH with CHSA-AEP until the CH is found, and then collect data. Through a lot of simulation experiments, compared with the area coverage algorithm GBA and CPP-SDA, due to the DRL-UTPS allows UAV to establish communication with CHs directly with a high probability after reaching the hover points, reduces the flight distance and reduces the energy consumption, and increases the time utilization ratio. In addition, compared with Random algorithm, which selects a device in the cluster as the hover point in a random way, DRL-UTPS can train a more appropriate hover point through historical data, which makes DRL-UTPS have better energy consumption and time utilization ratio. In terms of UAV energy consumption, DRL-UTPS can reduce flight energy consumption by 51.9% and 15.7% on average compared with area coverage algorithm (GBA, CPP-SDA) and Random algorithm. Compared with GBA, CPP-SDA and Random algorithm, DRL-UTPS can increase the time utilization ratio by 44.6% and 14.3% on average.

Acknowledgment

The first author Run Liu would like to thank Siyi Lu at the School of Computer Science, Central South University, for valuable discussion.

References

[1] U. Cisco, "Cisco annual internet report (2018–2023) white paper," 2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html
[2] D. Reinsel, J. Gantz, and J. Rydning, "Data age 2025: The digitization of the world from edge to core," 2018. [Online]. Available: https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf
[3] F. Zhu, Y. Lv, Y. Chen, X. Wang, G. Xiong, and F.-Y. Wang, "Parallel transportation systems: Toward Iot-enabled smart urban traffic control and management," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 10, pp. 4063–4071, Oct. 2020.
[4] R. Ke, Y. Zhuang, Z. Pu, and Y. Wang, "A smart, efficient, and reliable parking surveillance system with edge artificial intelligence on IoT devices," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 4962–4974, Aug. 2021.
[5] N. Maisonneuve, M. Stevens, M. E. Niessen, and L. Steels, "Noisetube: Measuring and mapping noise pollution with mobile phones," in *Information Technologies in Environmental Engineering*. Berlin, Germany: Springer, 2009, pp. 215–228.
[6] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Ear-phone: An end-to-end participatory urban noise mapping system," in *Proc. 9th IEEE/ACM Int. Conf. Inf. Process. Sensor Netw.*, 2010, pp. 105–116.
[7] E. Koukoumidis, M. Martonosi, and L.-S. Peh, "Leveraging smartphone cameras for collaborative road advisories," *IEEE Trans. Mobile Comput.*, vol. 11, no. 5, pp. 707–723, May 2012.
[8] J. E., M. Li, and J. Huang, "Crowdatlas: Estimating crowd distribution within the urban rail transit system," in *Proc. IEEE 37th Int. Conf. Data Eng.*, 2021, pp. 2219–2224.
[9] S. P. Bharati, Y. Wu, Y. Sui, C. Padgett, and G. Wang, "Real-time obstacle detection and tracking for sense-and-avoid mechanism in UAVs," *IEEE Trans. Intell. Veh.*, vol. 3, no. 2, pp. 185–197, Jun. 2018.
[10] A. Bansal, N. Agrawal, and K. Singh, "Rate-splitting multiple access for UAV-based RIS-enabled interference-limited vehicular communication system," *IEEE Trans. Intell. Veh.*, early access, Apr. 19, 2022, doi: 10.1109/TIV.2022.3168159.
[11] W. Yao et al., "Evolutionary utility prediction matrix-basedmission planning for unmanned aerial vehicles in complex urban environments," *IEEE Trans. Intell. Veh.*, early access, Jul. 20, 2022, doi: 10.1109/TIV.2022.3192525.
[12] B. Zhu, E. Bedeer, H. H. Nguyen, R. Barton, and J. Henry, "Joint cluster head selection and trajectory planning in UAV-aided IoT networks by reinforcement learning with sequential model," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 12071–12084, Jul. 2022.
[13] B. Zhu, E. Bedeer, H. H. Nguyen, and R. Barton, "UAV trajectory planning in wireless sensor networks for energy consumption minimization by deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9540–9554, Sep. 2021.
[14] R. Liu, A. Liu, Z. Qu, and N. N. Xiong, "An UAV-enabled intelligent connected transportation system with 6G communications for internet of vehicles," *IEEE Trans. Intell. Transp. Syst.*, early access, Oct. 28, 2021 doi: 10.1109/TITS.2021.3122567.

[15] W. Jin, J. Yang, Y. Fang, and W. Feng, "Research on application and deployment of uav in emergency response," in *Proc. IEEE 10th Int. Conf. Electron. Inf. Emerg. Commun.*, 2020, pp. 277–280.
[16] W. Khiati, Y. Moumen, A. El Habchi, I. Zerrouk, J. Berrich, and T. Bouchentouf, "Grid based approach (GBA): A new approach based on the grid-clustering algorithm to solve a CPP type problem for air surveillance using UAVs," in *Proc. IEEE 4th Int. Conf. On Intell. Comput. Data Sci.*, 2020, pp. 1–5.
[17] J. I. Vasquez-Gomez, J.-C. Herrera-Lozada, and M. Olguin-Carbajal, "Coverage path planning for surveying disjoint areas," in *Proc. IEEE Int. Conf. Unmanned Aircr. Syst.*, 2018, pp. 899–904.
[18] M. Wang and J. Zeng, "Hierarchical clustering nodes collaborative scheduling in wireless sensor network," *IEEE Sensors J.*, vol. 22, no. 2, pp. 1786–1798, Jan. 2022.
[19] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. IEEE 33rd Annu. Hawaii Int. Conf. Syst. Sci.*, 2000, pp. 4–7.
[20] O. Younis and S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans. Mobile Comput.*, vol. 3, no. 4, pp. 366–379, Oct.–Dec. 2004.
[21] A. Thakkar and K. Kotecha, "Cluster head election for energy and delay constraint applications of wireless sensor network," *IEEE Sensors J.*, vol. 14, no. 8, pp. 2658–2664, Aug. 2014.
[22] V. Mnih et al., "Playing Atari with deep reinforcement learning," 2013. [Online]. Available: https://arxiv.org/abs/arXiv:1312.5602
[23] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," 2015. [Online]. Available: https://arxiv.org/abs/arXiv:1509.02971
[24] V. Mnih et al., "Asynchronous methods for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1928–1937.
[25] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017. [Online]. Available: https://arxiv.org/abs/arXiv:1707.06347
[26] T. Zhang, J. Lei, Y. Liu, C. Feng, and A. Nallanathan, "Trajectory optimization for UAV emergency communication with limited user equipment energy: A safe-DQN approach," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 3, pp. 1236–1247, Sep. 2021.
[27] B. Li and Y. Wu, "Path planning for UAV ground target tracking via deep reinforcement learning," *IEEE Access*, vol. 8, pp. 29064–29074, 2020.
[28] N. Zhang, J. Liu, L. Xie, and P. Tong, "A deep reinforcement learning approach to energy-harvesting UAV-aided data collection," in *Proc. IEEE Int. Conf. Wireless Commun. Signal Process.*, 2020, pp. 93–98.
[29] M. Samir, C. Assi, S. Sharafeddine, and A. Ghrayeb, "Online altitude control and scheduling policy for minimizing AoI in UAV-assisted IoT wireless networks," *IEEE Trans. Mobile Comput.*, vol. 21, no. 7, pp. 2493–2505, Jul. 2022.
[30] P. Zhong, B. Chen, S. Lu, X. Meng, and Y. Liang, "Information-driven fast marching autonomous exploration with aerial robots," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 810–817, Apr. 2022.
[31] F. Zhou, Y. Wu, H. Sun, and Z. Chu, "UAV-enabled mobile edge computing: Offloading optimization and trajectory design," in *Proc. IEEE Int. Conf. Commun.*, 2018, pp. 1–6.
[32] X. Li, J. Tan, A. Liu, P. Vijayakumar, N. Kumar, and M. Alazab, "A novel UAV-enabled data collection scheme for intelligent transportation system through UAV speed control," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2100–2110, Apr. 2021.
[33] A. A. Al-Habob, O. A. Dobre, S. Muhaidat, and H. V. Poor, "Energy-efficient data dissemination using a UAV: An ant colony approach," *IEEE Wireless Commun. Lett.*, vol. 10, no. 1, pp. 16–20, Jan. 2020.
[34] H. Daryanavard and A. Harifi, "UAV path planning for data gathering of IoT nodes: Ant colony or simulated annealing optimization," in *Proc. 3rd Int. Conf. Internet Things Appl.*, 2019, pp. 1–4.
[35] C. H. Liu, C. Piao, and J. Tang, "Energy-efficient UAV crowdsensing with multiple charging stations by deep learning," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 199–208.
[36] C. H. Liu, X. Ma, X. Gao, and J. Tang, "Distributed energy-efficient multi-UAV navigation for long-term communication coverage by deep reinforcement learning," *IEEE Trans. Mobile Comput.*, vol. 19, no. 6, pp. 1274–1285, Jun. 2020.
[37] Y. Yan, B. Zhang, C. Li, and C. Su, "Cooperative caching and fetching in D2D communications-A fully decentralized multi-agent reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 16095–16109, Dec. 2020.
[38] J.-S. Leu, T.-H. Chiang, M.-C. Yu, and K.-W. Su, "Energy efficient clustering scheme for prolonging the lifetime of wireless sensor network with isolated nodes," *IEEE Commun. Lett.*, vol. 19, no. 2, pp. 259–262, Feb. 2014.
[39] H. Guo and J. Liu, "UAV-enhanced intelligent offloading for Internet of Things at the edge," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2737–2746, Apr. 2020.
[40] A. Al-Hilo, M. Samir, C. Assi, S. Sharafeddine, and D. Ebrahimi, "UAV-assisted content delivery in intelligent transportation systems-joint trajectory planning and cache management," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5155–5167, Aug. 2021.
[41] Z. Wang, K. Han, and P. Tiwari, "Digital twin-assisted cooperative driving at non-signalized intersections," *IEEE Trans. Intell. Veh.*, vol. 7, no. 2, pp. 198–209, Jun. 2022.
[42] P. K. Sharma and J. H. Park, "Blockchain-based secure mist computing network architecture for intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5168–5177, Aug. 2021.

Run Liu received the B.E. degree from the Internet of Things Engineering, School of Computer Science and Engineering, Central South University, Changsha, China, in 2022. He is currently working toward the master's degree in computer applied technology with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China. His research interests include Internet of Things, wireless networks, wireless communications, edge computing and deep reinforcement learning.

Zhenzhe Qu received the master's degree from the School of Software, Central South University, Changsha, China, in 2019. He is currently working toward the Ph.D. degree with the School of Computer Science and Engineering, Central South University. His research focuses on edge computing.

Guosheng Huang received the M.S. and Ph.D. degrees in computer science from Central South University, Changsha, China, 2001 and 2010 respectively. He was a Visiting Scholar with Sun Yat-Sen University, Guangzhou, China, From 2017 to 2018. He is currently an Associate Professor of the School of Information Science and Engineering, Hunan First Normal University, Changsha. His research interests include MIMO techniques, wireless sensor network, and mobile computing.

Mianxiong Dong (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science and engineering from The University of Aizu, Aizuwakamatsu, Japan. He is currently the Vice President and a youngest ever Professor of the Muroran Institute of Technology, Muroran, Japan. He was a JSPS Research Fellow with the School of Computer Science and Engineering, The University of Aizu, and was a Visiting Scholar with BBCR Group with the University of Waterloo, Waterloo, ON, Canada, supported by JSPS Excellent Young Researcher Overseas Visit Program from April 2010 to August 2011. Dr. Dong was selected as a Foreigner Research Fellow (a total of three recipients all over Japan) by NEC C&C Foundation in 2011. He was the recipient of IEEE TCSC Early Career Award 2016, IEEE SCSTC Outstanding Young Researcher Award 2017, The 12th IEEE ComSoc Asia-Pacific Young Researcher Award 2017, Funai Research Award 2018 and NISTEP Researcher 2018 (one of only 11 people in Japan) in recognition of significant contributions in science and technology. He is Clarivate Analytics 2019 Highly Cited Researcher (Web of Science) and Foreign Fellow of EAJ.

Tian Wang received the B.Sc. and M.Sc. degrees in computer science from Central South University, Changsha, China, in 2004 and 2007, and the Ph.D. degree from the City University of Hong Kong, Hong Kong, in 2011. He is currently a Professor with the Artificial Intelligence and Future Networks, Beijing Normal University, Beijing, China, and UIC, China. His research interests include the Internet of Things, edge computing, and mobile computing.

Shaobo Zhang received the B.Sc. and M.Sc. degrees in computer science from the Hunan University of Science and Technology, Xiangtan, China, in 2003 and 2009 respectively, and the Ph.D. degree in computer science from Central South University, Changsha, China, in 2017. He is currently an Associate Professor with the School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China. His research interests include privacy and security issues in social networks and cloud computing.

Anfeng Liu received the M.Sc. and Ph.D. degrees in computer science from Central South University, Changsha, China, in 2002 and 2005, respectively. He is currently a Professor of the School of Information Science and Engineering, Central South University. His major research interests include wireless sensor networks, Internet of Things, information security, edge computing and crowdsensing.