Ding T, Liu N, Yan ZM *et al.* An efficient reinforcement learning game framework for UAV-enabled wireless sensor network data collection. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 37(6): 1356–1368 Nov. 2022. DOI 10.1007/s11390-022-2419-8

An Efficient Reinforcement Learning Game Framework for UAV-Enabled Wireless Sensor Network Data Collection

Tong Ding (丁　桐), Ning Liu* (刘　宁), *Member*, *CCF*, Zhong-Min Yan (闫中敏), *Member*, *CCF*
Lei Liu* (刘　磊), *Member*, *CCF*, and Li-Zhen Cui (崔立真), *Distinguished Member*, *CCF*

School of Software, Shandong University, Jinan 250101, China

E-mail: tongding@mail.sdu.edu.cn; {liun21cs, yzm, l.liu, clz}@sdu.edu.cn

Received April 15, 2022; accepted November 18, 2022.

Abstract With the developing demands of massive-data services, the applications that rely on big geographic data play crucial roles in academic and industrial communities. Unmanned aerial vehicles (UAVs), combining with terrestrial wireless sensor networks (WSN), can provide sustainable solutions for data harvesting. The rising demands for efficient data collection in a larger open area have been posed in the literature, which requires efficient UAV trajectory planning with lower energy consumption methods. Currently, there are amounts of inextricable solutions of UAV planning for a larger open area, and one of the most practical techniques in previous studies is deep reinforcement learning (DRL). However, the overestimated problem in limited-experience DRL quickly throws the UAV path planning process into a locally optimized condition. Moreover, using the central nodes of the sub-WSNs as the sink nodes or navigation points for UAVs to visit may lead to extra collection costs. This paper develops a data-driven DRL-based game framework with two partners to fulfill the above demands. A cluster head processor (CHP) is employed to determine the sink nodes, and a navigation order processor (NOP) is established to plan the path. CHP and NOP receive information from each other and provide optimized solutions after the Nash equilibrium. The numerical results show that the proposed game framework could offer UAVs low-cost data collection trajectories, which can save at least 17.58% of energy consumption compared with the baseline methods.

Keywords wireless sensor network, efficient data collection, deep reinforcement learning, game theory

1 Introduction

With big geographic data playing a crucial role in current life, efficient geo-data collection has become an urgent request for academic research [1, 2]. The request demands collecting geo-data from broader and more complex areas to carry out the tasks of sensing, exploring, and monitoring harsh fields, which is still challenging [3, 4]. One of the most crucial factors is the cost of data collection [5, 6]. Unmanned aerial vehicles (UAVs), with their flexibility, could be integrated with the ground wireless sensor networks (WSN) as low-cost enablers for collecting data automatically [7]. However, the broad geographic space with a significant amount of data far from sustainable supplements challenges UAVs-enabled tasks. UAVs may execute missions with unreasonable trajectories, raising energy consumption and increasing data collection costs. Naturally, planning trajectories with low-energy costs for the UAV has become one of the most critical problems in geographic data harvesting.

The current literature shows that automatic data collection has been one of the most crucial techniques in UAVs-enabled geographic space scenarios [8]. Vast geographic data are delivered among parties, posing urgent concerns of costs and efficiency for data collection. Previous studies have suggested that the optimization of trajectory planning for UAVs can significantly reduce the cost of data collection tasks [9]. To reduce the

Regular Paper

This work was (partially) supported by the National Natural Science Foundation of China under Grant No. 61972230 and the Natural Science Foundation of Shandong Province of China under Grant No. ZR2021LZH006.

*Corresponding Author (Ning Liu contributed the key ideas and participated in polishing the full paper, and Lei Liu provided funds for this research and offered the experimental equipment.)

data collection costs, quantities of heuristic methods are proposed, achieving comparable success with approximate optimal results [10]. However, these heuristic algorithms acquire limited performance and can hardly provide practical solutions [11, 12]. Hence, many data-driven algorithms were developed to plan the energy-conservation paths for UAVs [13]. For example, Zhao *et al.* [14] considered a model-free strategy to obtain optimized paths from data rather than dynamic knowledge. Bono *et al.* [15] regarded the UAV planning problem as a particular orienteering problem with spatiotemporal dependencies, which could provide the optimal solutions for small-scale situations. However, these data-driven methods are task-specific and data-dependent, which cannot be applied to larger geographic spaces with limited data. Recently, some studies have focused on finding solutions with limited data. One of the most typical methods is offline deep reinforcement learning (DRL), by which agents can learn strategies from ready-made experiences [16]. These developed DRL methods have been proven to provide sustainable solutions for ordering navigation [17–19]. For example, Challita *et al.* [20] developed an anti-disturbing scheme to process trajectories for UAVs, which employs the echo state network (ESN) cells to minimize the time-relevant utility. Xie *et al.* [21] made efforts on the 3-dimensional path processing for UAVs and employed the dueling network to find locally optimized paths. Mukherjee *et al.* [22] considered the task of allocating path processing for UAVs, which can reduce the computing complexity of mission execution with large-scale data. Zhu *et al.* [23] combined the pointer network and the A* algorithm to obtain efficient trajectories, which enables the UAV to collect the data from the clustered sensors. Zhang *et al.* [24] developed a DRL approach based on the twin delayed deep deterministic policy gradients (TD3) algorithm, which enables UAVs to perform navigation process tasks with unpredictable and dynamic multi-obstacle settings. Shi *et al.* [25] developed a novel coverage issue in battery-free wireless sensor networks (BF-WSNs) to maximize the coverage quality. Zhang *et al.* [26] presented an intelligent blockchain-assisted massive Internet of Things (IoT) data collection (MIDC) architecture to enhance the efficiency and security considerations of vast data collection for large-scale heterogeneous WSNs.

Nevertheless, current DRL-based data-driven methods still suffer from overestimating during the optimization process. Consequently, the current DRL-based methods easily throw the process of path planning into a locally optimized condition. Due to the high-dimension search space of the UAV planning in open areas, the neglection of consideration for different initial states may lead to overestimating. For example, Fig.1 shows that the clustered ground WSNs use the central nodes as the sink nodes and navigation potions for the UAV, which may lead to extra energy consumption. Specifically, the UAV collects the data from a WSN that only needs to visit the sink nodes. The selected sink nodes of sub-WSNs will determine the UAV's navigation points, and the cost of data collection is mainly related to the visiting order of these navigation points. Fig.1(a) shows that choosing the central nodes of clusters as sink nodes may lead to extra energy consumption of UAVs. Under some circumstances, using limbic nodes as sink nodes may be the optimal choice for UAV trajectory planning, as shown in Fig.1(b). Hence, choosing suitable sink nodes as the navigation while

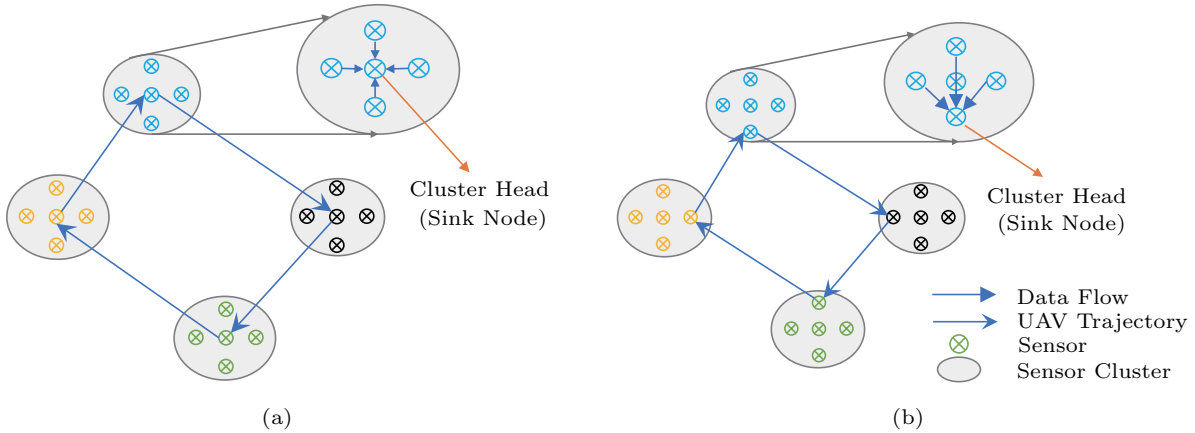

Fig.1. Example of the UAV trajectory planning with different sink nodes. (a) Trajectory planning by using central points of clusters as the sink nodes. (b) Trajectory planning by using limbic points of clusters as the sink nodes.

planning the energy-conserved path for the UAV is still challenging.

Previous studies have shown that the game theory could help the neural network break the local optimization, and each game participant could obtain more information from the data distribution to achieve a better convergence [27]. Thus, this paper employs the game framework to develop a DRL-based approach that consists of two partners: a cluster head processor (CHP) and a navigation order processor (NOP). During the game process, the CHP receives the navigation order list from the NOP and uses this order to evaluate the utility for all cluster heads. The NOP accepts the cluster head list from the CHP and then plans the data-collection path for the UAV. This process repeats until two partners reach a Nash Equilibrium. The developed game framework provides two partners with a stably convergent impact factor, which can help the agents release from the overestimated situation within the training process. According to the numerical results, the NOP adjusts the convergent trend continually and strides over the local optimized station with a higher probability. At the same time, compared with using the central cluster nodes directly, choosing cluster heads with the CHP as navigation could also reduce the energy consumption of UAVs. Our contributions can be concluded as follows.

- This paper formulates a UAV-enabled data-collection optimization problem for the clustered aerial-terrestrial WSN network, which is used to optimize the cost of geo-data harvesting.
- This paper constructs the process of cluster finding and navigation ordering with the Markov decision process (MDP) separately. The constructed MDPs are used to find the cluster heads and low-cost UAV's data-collecting trajectories.
- This paper develops a game-based training framework to assist the data-driven DRL in releasing from the overestimating situation, which consists of two DRL-based partners: CHP and NOP. The two partners receive the results from each other and generate optimized solutions after the Nash equilibrium.
- This paper conducts extensive numerical experiments to validate the advancement of the developed method. The numerical results illustrate that the CHP can find better sink nodes for clustered WSNs, and the NOP can provide convergent solutions for planning UAV trajectories. It is proved that the developed game-frame DRL method can overcome the local optimization with more possibilities during the training phases and can acquire more energy-conserved solutions.

The rest of this paper is organized as follows. Section 2 introduces the related work about the DRL-based methods and game theories. The background and problem formulation is presented in Section 3, which includes the model preliminaries, mission background, data collection cost model, and problem definition. Section 4 presents the constructions of the CHP and the NOP in detail and describes the developed game framework for cluster heads choosing and UAV path planning optimization. Section 5 describes the numerical results and shows the advancement of the developed method. Finally, this paper is concluded in Section 6.

2 Related Work

2.1 Reinforcement Learning Approaches

The reinforcement learning (RL) methods iterate the learning process under the MDP framework and have been widely used in different industrial scenarios [28, 29]. The typical RL-based method is q-learning [30], which establishes a table to evaluate the actions according to the time difference (TD) error. Combined with the deep learning, Deep q-learning Networks (DQNs) predict the values with neural networks to fit the improving capacity of the state space [31]. The action space of q-learning and DQN is discrete, which is not acceptable for continuous actions. Hence, the deep deterministic policy gradients (DDPG) based methods [32] are established for continuous actions control. The DDPG-based methods obey the actor-critic (AC) [33] framework and use the TD error to update the neural network. The flourishing achievements of the AC-based framework have also pioneered DRL in recent years, such as advantage actor-critic (A2C) [34] and asynchronous advantage actor-critic (A3C) [35]. A2C uses the advantage function to replace the original reward in the critic network, which can be utilized to be the metric to evaluate the value of choosing actions and average actions. A3C is an asynchronous method to obtain the training experience from environments. Each worker of A3C employs the parameters from the global network to interact with the environment. All workers give their gradients to update the global network. Furthermore, some DRL-based methods also inspire our work. Schaul *et al.* [36] improved the experience playback mechanism to focus on the samples' priority. Wang *et al.* [37] developed a dueling neural network structure, which can evaluate both actions and status.

2.2 Stackelberg Game Methods

With the development of the game theory, the Stackelberg game has been employed in different scenarios. For example, Su *et al.* [38] formulated a jamming counter measure Stackelberg game to describe the system's jamming power control dynamic and developed an iterative measure algorithm to acquire the Stackelberg equilibrium. Bansal and Sikdar [39] developed a Stackelberg game based approach to construct a security service pricing model for UAV swarms, which can maximize the profit of providing security services. Shi *et al.* [40] conducted a problem of cooperative low probability of intercept (LPI) in a multi-static radar system, and they developed a Stackelberg game based method to get the optimal global solution. A general Stackelberg game system includes a leader (or leaders) and a follower (or followers), which could be used in satellite communications [41] and anti-tracking jammer [42]. Note that our work is mainly related to a typical Stackelberg model with one leader and multiple followers [27].

3 Background and Problem Formulation

In this section, we introduce model preliminaries, mission background, and data collection cost model at first. Then, we formulate the problem that needs to be optimized.

3.1 Model Preliminaries

3.1.1 MDP and RL Approaches

In this paper, the CHP and the NOP are modeled with two MDPs with finite time steps. The MDP is a stochastic sequential decision-making approach that consists of an agent set I, a state space S, an action space A, and a reward function $R : S \times A \to \mathcal{R}$. At each time step, the agent observes the state $s_t \in S$ of the environment and chooses an action $a_t \in A$ based on the policy $\pi(s)$. The environment changes its state from s_t to s_{t+1} and outputs a reward r_t according to the state transfer function $T : S \times A \times S \to [0, 1]$. The agent adjusts its action-choosing policy $\pi(s)$ according to the reward r_t. This process executes continuously until the optimal policy π^* is constructed.

This paper uses the value-based RL methods DQN [31] and Dueling network [37] to obtain the optimized solutions with the modeled MDP. DQN is an off-policy learning scheme that is used to obtain a greedy policy $\pi(s) = \text{argmax}_a Q_\pi(s, a)$, where $Q_\pi(s, a)$ indicates the Q value utilized to evaluate the actions. The Q value can be calculated by the equation $Q_\theta(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim E}(r(s_t, a_t) + \gamma Q_\theta(s_{t+1}, a_{t+1}))$, where $Q_\theta(s, a)$ is a neural network parameterized by θ, E denotes the environment, r denotes the reward, and γ is the discount factor. To overcome the overestimations of previous DQN methods [31], the Dueling network is developed [37].

3.1.2 Stackelberg Game Process

The Stackelberg game is a two-stage dynamic game with complete information. We describe the Stackelberg game process for UAV path planning from Wang *et al.* [27]. Specifically, this paper conducts the Stackelberg game with multiple followers to design the interactive framework for the CHP and the NOP. At the beginning of the training phases, the leader adopts a strategy $\pi \in \Pi$, and n followers produce reactions after observing the leader's strategy. Once the leader's strategy π is determined, an n-player simultaneous game will be constructed and each participant should minimize or maximize its utility function. The i-th follower's utility function can be concluded as $f_i(x_i, x_{-i}, \pi)$, where $x_i \in X_i$ is the i-th follower's own action, $x_{-i} \in X_{-i}$ indicates the joint action of other followers, X_i is the i-th follower's strategy space, and $\pi \in \Pi$ represents the leader's strategy. It is assumed that there is no secret among followers, i.e., all the followers have access to the other followers' objective functions and strategy space. After all followers obtain the unique equilibrium $x^*_{1,\ldots,n}$, the leader will optimize the objective to get the new policy π. This process continues until the Stackelberg leader chooses an optimal π^* to maximize its utility, and the game process reaches a Nash equilibrium.

3.2 Mission Background

This paper considers a terrestrial WSN deployed into an open area to execute the missions. The WSN consists of a group of sensors $B = \{b_i \mid i = 1, ..., k\}$ which are classified into K clusters $C = \{c_j \mid j = 1, ..., K\}$. A sub-WSN connects the sensors in each cluster, and all sensors generate the data continuously according to different mission requirements. It is assumed that the sensors in the same sub-WSN can transmit their data to others, while sensors in different sub-WSNs cannot share the information. Thus, there is a sink node to gather all data of sensors in a sub-WSN, and the vehicle collects the data by visiting all sub-WSNs' sink nodes rather than all

sensors. This work employs a solo UAV to harvest and deliver the collected data from WSN to the base station. The UAV keeps a fixed velocity v_{uav} and height h during the mission execution. We hypothesize that the movement of the UAV will not be influenced by temperature, humidity, air density, or other environmental impacts. Under these considerations, this paper aims to find optimized solutions for low-cost data collection paths and suitable sink nodes for all clusters.

3.3 Data Collection Cost Model

This work requires the UAV to collect all data from the WSN with low-level energy consumption. The previous studies make assumptions that the data collection cost depends on the utility of UAVs [23, 43, 44]. Therefore, we use the energy-conservation utility to evaluate the cost, which means a path with high utility is the trajectory with a low data collection cost, i.e., the energy-conservation utility is inversely proportional to the UAV's energy consumption. In this paper, we will not evaluate the data transmitting energy due to the transmission relying on the facility carried by the UAV. Consequently, the UAV's energy consumption is shown as (1).

$$E_{\text{total}} = E_{\text{flight}} + E_{\text{hover}}, \tag{1}$$

where E_{flight} is the flight energy, and E_{hover} is the hovering energy of the employed UAV. In fact, E_{flight} can be affected by three factors, including flight time T_{flight}, moving power P_{move} and hovering power P_{hover}. Following the previous studies [23, 43], we define the flight energy as (2).

$$E_{\text{flight}} = T_{\text{flight}} \left(P_{\text{hover}} + P_{\text{move}}\right), \tag{2}$$

where T_{flight} is the total flight time of the UAV. The total flight time T_{flight} depends on the UAV's velocity and the distance has passed [23, 43], which is shown in (3).

$$T_{\text{flight}} = \frac{1}{v_{\text{UAV}}} \sum_{i=1}^{K} \sum_{\substack{j=1 \\ j \neq i}}^{K} d_{c_i,c_j} L_{c_i,c_j}, \quad \forall c_i, c_j \in C, \tag{3}$$

where v_{UAV} is the velocity of UAV, which is a fixed parameter according to the hypothesis that the UAV keeps a fixed velocity during executing missions. $C = \{c_1, c_2, ..., c_K\}$ is the set of cluster heads and d_{c_i,c_j} indicates the distance between the cluster heads c_i and c_j. $\boldsymbol{L} \in R^{|C|\times|C|}$ is an indicator matrix. If $L_{c_i,c_j} = 1$, it means that the UAV would fly from c_i to c_j with the current trajectory; otherwise $L_{c_i,c_j} = 0$. P_{hover} and P_{move} are the hovering and moving power of UAV [23, 43], which are described with (4) and (5) respectively.

$$P_{\text{hover}} = \sqrt{\frac{(mg)^3}{2\pi r_p^2 n_p \rho}}, \tag{4}$$

$$P_{\text{move}} = \frac{P_{\max} - P_{\text{idle}}}{v_{\max}} v_{\text{UAV}} + P_{\text{idle}}, \tag{5}$$

where m is the mass of the UAV with the carried data-transmitting facility, g is a fixed factor and represents the earth gravity, r_p is the radius of the UAV's propellers, n_p is the number of propellers, and ρ is the air density. $v_{\max}$ is the maximum speed of the UAV and P_{idle} is the power of the UAV with the idle status. All of these parameters are fixed.

Moreover, the hovering energy consumption E_{hover} relies on the hovering time T_{hover} and power P_{hover}, which is shown in (6) [23, 43].

$$E_{\text{hover}} = T_{\text{hover}} \times P_{\text{hover}} = \sum_{i=1}^{K} \frac{c_i^{\text{data}}}{r_{\text{trans}}} \times P_{\text{hover}}, \tag{6}$$

where c_i^{data} is the data volume of the i-th cluster, and r_{trans} is the data transmission rate. The mission's sustainably execution relies on energy conservation within the aerial-terrestrial data collection task. Hence, this paper uses UT_{UAV} to define the energy-conservation utility, which is shown in (7).

$$UT_{\text{UAV}} = \frac{1}{E_{\text{total}}} = \frac{1}{E_{\text{flight}} + E_{\text{hover}}}. \tag{7}$$

(7) indicates that the utility of a mission execution is inversely proportional to the energy consumption.

3.4 Problem Definition

This work requires the UAV to collect all data from the sensors with a high energy-conservation utility (defined in (7)), and this utility is related to the energy consumption. The UAV which flies from one cluster head to another would consume energy. Thus, different visiting locations and orders require distinct energy costs. Under these considerations, this work aims to maximize UT_{UAV} of the UAV data collection mission, which is shown in (8).

$$\begin{aligned} &\max UT_{\text{UAV}} \\ &= \min(E_{\text{total}}) = \min(E_{\text{flight}} + E_{\text{hover}}) \\ &= \min(T_{\text{flight}} \left(P_{\text{hover}} + P_{\text{move}}\right) + E_{\text{hover}}). \end{aligned} \tag{8}$$

It is also assumed that the amount of harvesting data at each cluster head is the same, wherefore E_{hover}

is also fixed. Hence, this problem can be simplified with (9).

$$\begin{aligned} &\max UT_{\mathrm{UAV}} \\ =\ &\min T_{\mathrm{flight}} \\ =\ &\min \frac{1}{v_{\mathrm{UAV}}} \sum_{i=1}^{K} \sum_{\substack{j=1 \\ j\neq i}}^{K} d_{c_i,c_j} L_{c_i,c_j}, \forall c_i, c_j \in C, \end{aligned} \tag{9}$$

where v_{UAV} is a fixed number. The final optimization problem is shown in (10).

$$\begin{aligned} &\max UT_{\mathrm{UAV}} \\ =\ &\min \sum_{i=1}^{K} \sum_{\substack{j=1 \\ j\neq i}}^{K} d_{c_i,c_j} L_{c_i,c_j}, \quad \forall c_i, c_j \in C. \end{aligned} \tag{10}$$

From (10), we can see that the objective function depends on two parts: the visitation order list $d_{c_i,c_j} L_{c_i,c_j}$ and the cluster heads set C. This paper will optimize these two parts with a game framework to get optimal solutions for the objective function.

4 DRL-Based Game Framework for Cluster Head Determination and Path Processing

This section describes a data-driven DRL-based game framework composed of two participants: the CHP and the NOP. We first formulate the CHP and the NOP separately with the MDPs. Then, we introduce the developed Stackelberg game-based method and describe how the two participants interact with each other.

4.1 Sink Nodes Determination with Cluster Head Processor

CHP is one of the crucial parts of the developed game framework, which receives the UAV path information from the NOP and chooses sink nodes for all clusters. We formulate the cluster heads finding process with MDP, where there are K agents that stand for the choosers to find the optimal cluster heads. Given one chooser in the time step t, we explain the state, action, and reward function established in the optimizing process as follows.

1) *State.* To describe the chosen result of the ground WSN's cluster heads, we employ the local state S_{L}^t and the global state S_{G}^t to represent the state of the CHP. The agent of the i-th cluster has a local indicative state $S_{L_i}^t \in S_{\mathrm{L}}^t$, which denotes its cluster head chosen result. The local indicative state $S_{L_i}^t$ is described by a one-hot vector to indicate which node is selected as the cluster head. In particular, if the chooser of the i-th cluster selects the j-th node to be the cluster head, we set the j-th value of $S_{L_i}^t$ to 1, and the other values to 0. Consequently, the global state of the CHP is represented by $S_{\mathrm{G}}^t = (S_{L_1}^t, S_{L_2}^t, ..., S_{L_K}^t)$, which indicates the heads chosen result of all clusters.

2) *Action.* The action of the i-th head chooser is denoted as $a_{c_i}^t = f_{\pi_{c_i}}(S_{L_i}^t)$. $a_{c_i}^t$ is conducted by the selecting policy $f_{\pi_{c_i}}$ and indicates which node will be set as the new cluster head. In this work, we employ a one-hot vector for representing $a_{c_i}^t$ to perform the cluster head chosen action. For example, if the j-th value of $a_{c_i}^t$ is 1, it indicates that the j-th node in c_i will be set as the cluster head. Thus, the joint action of the CHP in the time step t is $a_{\mathrm{C}}^t = (a_{c_1}^t, a_{c_2}^t, ..., a_{c_K}^t)$.

3) *Reward.* This work employs the reward function $r_{\mathrm{C}}^t = R_{\mathrm{C}}(j^t, a_{\mathrm{C}}^t)$ for CHP, where j^t is the path acquired from the NOP. The reward r_{C}^t depends on the utility that we consider in (7), which is shown in (11).

$$\begin{aligned} r_{\mathrm{C}}^t &= R_{\mathrm{C}}(j^t, a_{\mathrm{C}}^t) \\ &= UT_{\mathrm{UAV}}(d^t, L^t) \\ &= \frac{P}{v_{\mathrm{UAV}}} \sum_{i=1}^{K} \sum_{\substack{j=1 \\ j\neq i}}^{K} d_{c_i,c_j}^t L_{c_i,c_j}^t, \end{aligned} \tag{11}$$

where $P = P_{\mathrm{flight}} + P_{\mathrm{hover}}$, L_{c_i,c_j}^t relies on the path j^t obtained from the NOP, and d_{c_i,c_j}^t depends on the locations of all cluster heads decided by a_{C}^t. This reward function encourages the CHP to select cluster heads with high energy-conservation utility according to the path information from the NOP.

4.2 UAV Path Planning with Navigation Order Processor

According to the energy-conservation utility definition and the optimization problem defined in Subsection 3.4, the UAV needs to find an optimal path with lower energy consumption. During the mission of the UAV-enabled data delivery, the current position of the UAV is only related to its prior location. Thus, we construct the NOP with MDP to find the optimal solutions. There is an agent in the NOP that represents the controller to instruct the flight of the UAV. Given a controller in the time step t, we explain the state, action, and reward function for planning the energy-conserved path as follows.

1) *State.* The agent of the NOP has an indicative state s_{N}^t to present the UAV's location. The indicative

state s_{N}^t is represented by a K-dimensional one-hot vector to indicate which cluster the UAV stays in the time step t. For example, if the UAV is located in the cluster c_i in the time step t, the i-th value of s_{N}^t is set to 1. Otherwise, it is set to 0. Hence, S_{N}^t reflects the position situation of the UAV and can be used to form the path $(S_{\mathrm{N}}^1, S_{\mathrm{N}}^2, ..., S_{\mathrm{N}}^K)$.

2) *Action.* The action of the NOP is set as $a_{\mathrm{N}}^t = f_{\pi_{\mathrm{N}}}(s_{\mathrm{N}}^t)$. It is performed by the navigation ordering policy π_{N} and indicates where the UAV will fly to in the following time step. We utilize a K-dimensional one-hot vector to describe a_{N}^t. If the i-th value of a_{N}^t is 1, it indicates that the UAV will fly to the i-th cluster c_i in the next time step.

3) *Reward.* This work employs the temporal reward for the NOP, which can be represented as $r_{\mathrm{N}}^t = R_{\mathrm{N}}(S_{\mathrm{G}}^t, S_{\mathrm{N}}^t, a_{\mathrm{N}}^t)$ shown in (12).

$$\begin{aligned} r_{\mathrm{N}}^t &= R_{\mathrm{N}}(S_{\mathrm{G}}^t, S_{\mathrm{N}}^t, a_{\mathrm{N}}^t) \\ &= \frac{P_{\mathrm{flight}}}{\sqrt{\left(x_{c_t} - x_{c_{t+1}}\right)^2 + \left(y_{c_t} - y_{c_{t+1}}\right)^2}}, \end{aligned} \tag{12}$$

where c_t and c_{t+1} indicate the clusters which the UAV visits in the time step t, and x_{c_t}, y_{c_t}, $x_{c_{t+1}}$, $y_{c_{t+1}}$ are their heads' coordinates respectively. These coordinates rely on the cluster heads list decided by S_{G}^t of the CHP. x_{c_t} and y_{c_t} are the coordinates of the UAV's current location, which depend on S_{N}^t. $x_{c_{t+1}}$ and $y_{c_{t+1}}$ are the coordinates of the navigation point that the UAV will fly to in the next time step, which depend on a_{N}^t. This function encourages to perform the actions that ensure the agent to get more rewards from lower energy consumption.

4.3 Stackelberg Game Framework for CHP and NOP

Subsection 4.1 and Subsection 4.2 introduce the MDPs of CHP and NOP, which are the two participants of the developed game framework. In this subsection, we present the details of how to construct the Stackelberg game framework in aerial-terrestrial WSN data collection tasks. The Stackelberg game framework provides an interaction scheme for connecting the CHP and the NOP, which aims at finding the optimal energy conservation path to reduce the data collection cost.

1) *Nash Equilibrium Between the CHP and the NOP.* In this work, we consider that the policy $\pi_{\mathrm{C}}^* = \{\pi_{c_1}^*, \pi_{c_2}^*, ..., \pi_{c_K}^*\}$ of the CHP and π_{N}^* of the NOP achieve the Nash equilibrium if no partner has obvious changes from the current status at the same time. It means that the CHP optimizes its object function in which the NOP's current optimal policy is regarded as the condition defined in (13).

$$\forall i : f_{c_i}\left(\pi_{c_i}^*, \pi_{\mathrm{N}}^*\right) > f_{c_i}\left(\pi_{c_i}, \pi_{\mathrm{N}}^*\right), \quad \forall \pi_{c_i} \in \Pi_{\mathrm{C}}, \tag{13}$$

where f_{c_i} is the utility function of the i-th cluster, and Π_{C} is the strategy space of the CHP. The NOP optimizes its object function based on the optimal policy of the CHP defined in (14).

$$f_{\mathrm{N}}\left(\pi_{\mathrm{N}}^*, \pi_{\mathrm{C}}^*\right) > f_{\mathrm{N}}\left(\pi_{\mathrm{N}}, \pi_{\mathrm{C}}^*\right), \quad \forall \pi_{\mathrm{N}} \in \Pi_{\mathrm{N}}, \tag{14}$$

where f_{N} is the utility function, and Π_{N} is the strategy space of the NOP. In this work, if the utility changes of each partner fluctuate within a small and acceptable range, it can be considered that the game process achieves the Nash equilibrium.

2) *Game Framework.* This work develops a game-based DRL framework that contains two parts: the CHP and the NOP, which influence each other during the training phases. The interactive learning framework based on the Stackelberg game theory is shown in Fig.2. It presents that the CHP evaluates the energy consumption by obtaining the flight order list generated from the NOP to find the optimal sink nodes. The NOP regards the flight order list obtained from the CHP as the constant when planning the optimal path. After the stable convergence of the CHP, the NOP will evaluate the cost of data collection with fixed coordinates of the sink nodes set, which is outputted from the CHP. After the game process achieves the Nash equilibrium, the CHP will obtain the optimized result of cluster heads, and the NOP will output the energy-conserved path for the UAV.

3) *Training Process.* In the beginning, both partners of the game framework stay at their initial states, and none of them can produce convergent solutions. The game framework first constructs the neural networks for the CHP as the action executors for all clusters and randomly generates the path as the input of the CHP. The actions executor of the NOP is also constructed with a neural network. Then, the CHP updates its parameters until convergence (i.e., the optimizing result changes are not obvious after several epochs). After the convergence, the network of each cluster will produce a set with sink nodes as the navigation. By receiving the sink nodes set from the CHP, the NOP updates its parameters of the executor and produces the clusters' visitation order list. The CHP accepts the order list and starts new episodes for training. The policy of choosing actions employed in the training phases is the annealed

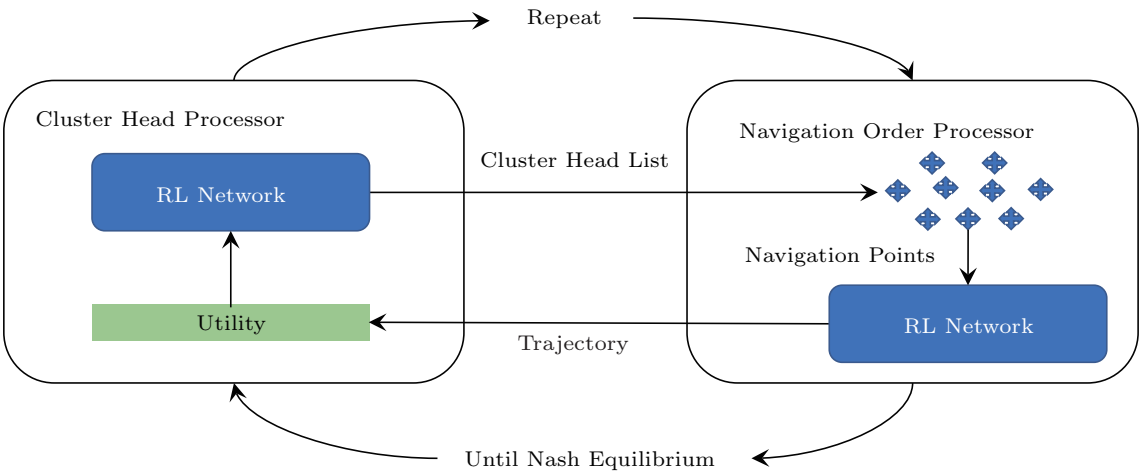

Fig.2. Game framework of the CHP and the NOP. The CHP is used to produce the sink nodes set, which depends on the input of the navigation order list generated by the NOP. The NOP evaluates the visiting order with fixed coordinates of sink nodes outputted from the CHP. This process is executed continuously until the Nash equilibrium.

ϵ-greedy policy. An illustrative example of presenting the possibility of random actions is shown in Fig.3.

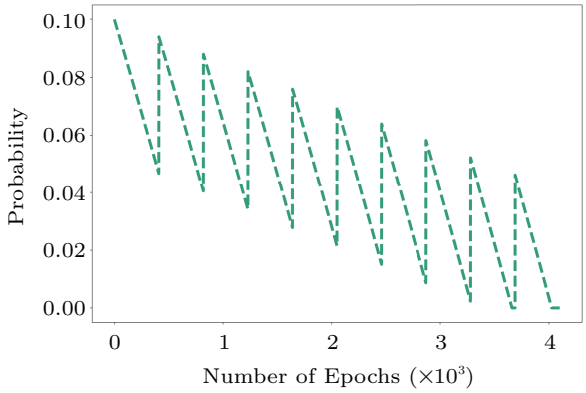

Fig.3. Example of annealed epsilon with 4 000 epochs. The annealed ϵ-greedy policy means that the agent randomly chooses actions with the probabilistic ϵ_t at each time step, and ϵ_t diminishes gradually. After each episode ϵ_t will be enhanced to a new value $\epsilon_{t+1} < \epsilon_{t+1}$.

5 Simulation and Experimental Results

This section first describes the experimental settings we employed, including the baseline approaches, parameter settings and the policy for choosing actions. Then, this section introduces the numerical results to analyze the advancement of our method.

5.1 Experimental Settings

1) *Baseline Approaches.* In this paper, we employ the central cluster heads of WSN as navigation with two typical DRL methods, DQN [31] and Dueling network [37], as the baseline path planning approaches. DQN is a typical value-based DRL method that employs the deep neural network as the action controller, which relies on the TD error obtained from the offline datasets to update the parameters. The Dueling network is an updated value-based DRL approach, and its improved network structure can evaluate both actions and status. Specifically, we use the central node in a cluster as the sink node and perform the UAV planning, named DQN and Dueling network for simple illustration, respectively. At the same time, G-F DQN is the developed DQN with the game framework, and G-F Dueling is the developed Dueling network with the game framework.

2) *Parameter Settings.* This work conducts the data collection tasks in an open area with the size of 7×20 $(\times 10^5)$ m^2, and there are 200 sensors deployed in this open area. These sensors have been classified into 20 clusters, each containing 10 sensors separately. An illustrative example is shown in Fig.4. The experimental parameters employed in this paper are shown in Table 1, which includes the settings of G-F DQN and G-F Dueling. ϵ is the initial probability that the agent chooses actions from the network. ϵ^{dis} is the discount factor of ϵ. MC is the memory capacity that is the maximum amount of data in the experience pool. NI is the frequent interval for updating the target net. γ presents the learning discount factor. LR indicates the learning rate. BS is the batch size of data used to train the model in each learning step.

3) *Action Selecting Policy.* We use the annealed ϵ-greedy as the policy for all experiments to choose actions for the CHP and the NOP. Fig. 3 describes an

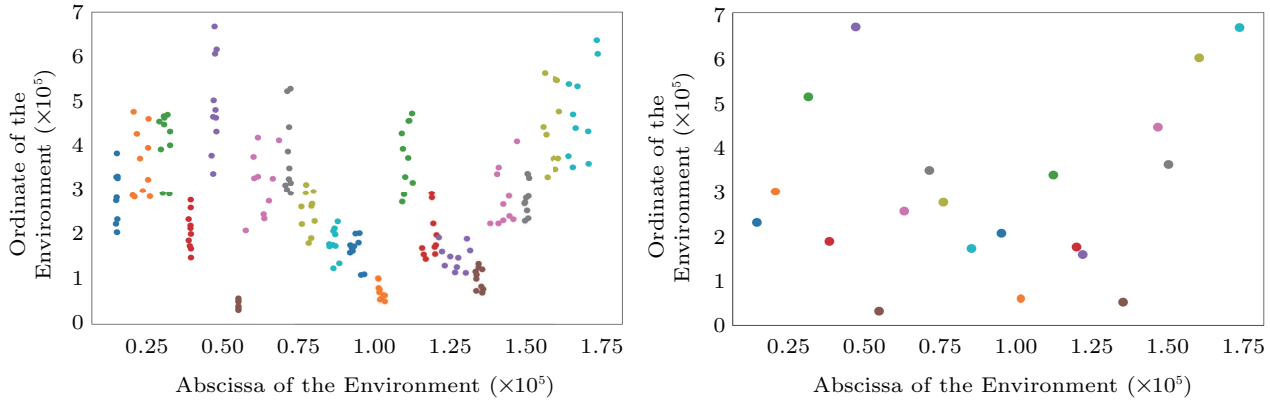

Fig.4. Example of initial sensors distribution and the cluster heads. (a) Distribution of the initial sensors. (b) Sink heads chosen result.

Table 1. Parameter Settings of G-F DQN and G-F Dueling

Parameter	ϵ	$\epsilon^{dis}_{\text{CHP}}$	$\epsilon^{dis}_{\text{NOP}}$	MC_{CHP}	MC_{NOP}	NI	γ	LR_{CHP}	LR_{NOP} ($\times 10^{-3}$)	BS_{CHP}	BS_{NOP}
G-F DQN	0.9	1.000 027	1.000 14	1 000	3 000	500	0.9	0.001	4, 4, 0.04	256	512
G-F Dueling	0.9	1.000 027	1.000 14	1 000	3 000	500	0.9	0.001	10, 2, 0.4	256	512

illustrative example of epsilon changing. As shown in Fig. 3, the initial value of the epsilon is 0.1. It means that the agent chooses actions randomly with the probability of 0.1 and chooses actions that rely on the neural network with the possibility of 0.9. The value decreases with a linear function, and after a fixed amount of epochs, the epsilon will be initiated with a lower value. ϵ decreases to 0 after 4 000 epochs, which means the executor acquires all actions according to the neural network's output at the 4 000 epochs. This mechanism ensures that the agents can break the local optimization by more chances and stay at a stable learning status at the end of the training.

5.2 Numerical Results

In this subsection, we employ extensive experiments to validate the improvement of the developed game-frame DRL method. Specifically, the compared algorithms with different learning rates are used to prove the advancement and generalization of the proposed DRL game-frame method.

1) *Effect of the CHP.* Fig.5 records the energy consumption changes with the iteration of the CHP. It shows that energy consumption is reduced with the enhancement of training epochs. In the first stage, the algorithm chooses the random path for the CHP, whereof the energy consumption remains at a high level. After the NOP finds a better path, the energy consumption reduces sharply and decreases continuously in the following phases. It is proved that the interaction with the NOP could enable the CHP to find better sink nodes for all clusters.

2) *Data Collection Cost Analysis on G-F DQN.* The results shown in Fig.6(a), Fig.6(b), and Fig.6(c) present the energy-conservation performance comparison for the developed G-F DQN with DQN. As we can see, G-F DQN presents the best performance in all situations compared with DQN. More specifically, the energy of both G-F DQN and DQN reduces with the increasing number of epochs, but the G-F DQN presents a better convergence. Training with larger learning rates enables the algorithms to converge faster, and G-F DQN remains at a lower energy consumption level all the time. To present more convincing details for proving the performance-enhancing of the developed framework, we employ the comparison of the average cases and the best cases shown in Table 2. Table 2 shows that compared with benchmark methods, the energy conservation based on our method can conserve at least 17.58% of energy and can save 38.01% of energy in the best case.

3) *Data Collection Cost Analysis on G-F Dueling.* In this subsection, we employ three types of experiments with different learning rates to validate the advancement of the developed G-F Dueling. Fig. 6(d), Fig. 6(e), and Fig. 6(f) present the numerical results that the G-F Dueling can provide solutions with more energy-conservation than the Dueling network. The numerical results show that training with a larger learning rate enables the algorithm to converge faster so that the agent can quickly reach the local optimization. On the

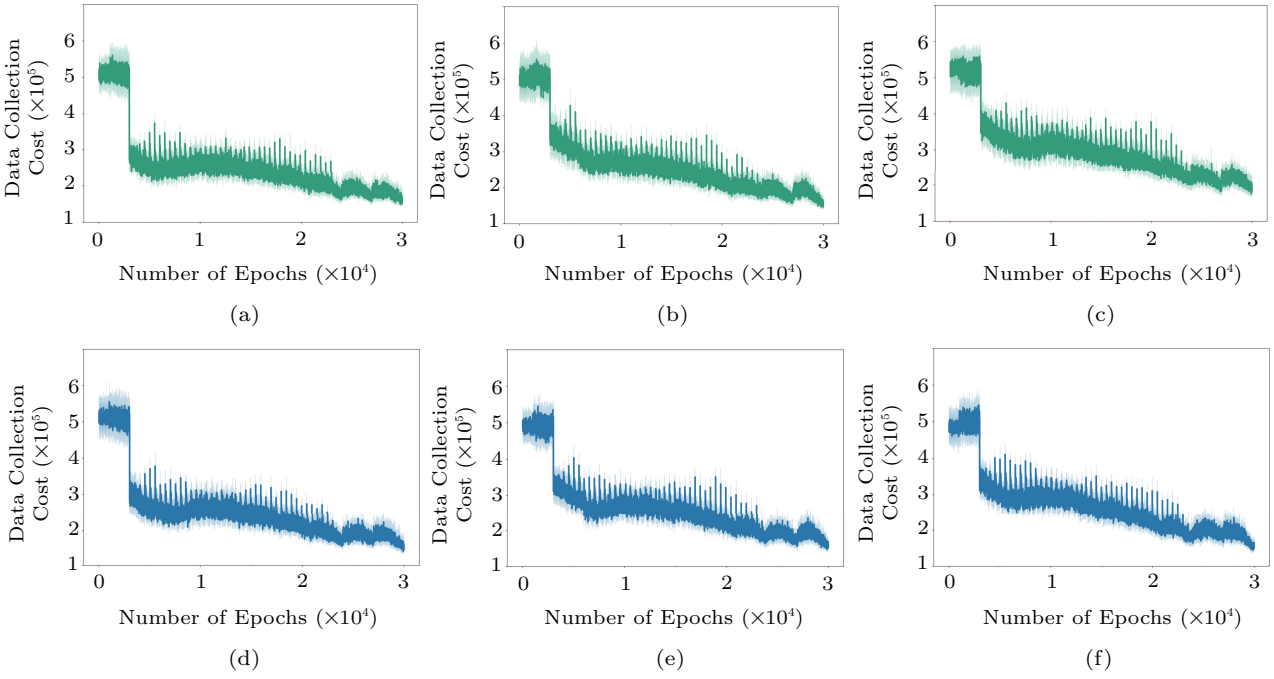

Fig.5. Data collection cost with different learning rates of the CHP. (a) G-F DQN with LR_{CHP} of 0.001 and LR_{NOP} of 0.004. (b) G-F DQN with LR_{CHP} of 0.001 and LR_{NOP} of 0.000 4. (c) G-F DQN with LR_{CHP} of 0.001 and LR_{NOP} of 0.000 04. (d) G-F Dueling with LR_{CHP} of 0.001 and LR_{NOP} of 0.01. (e) G-F Dueling with LR_{CHP} of 0.001 and LR_{NOP} of 0.002. (f) G-F Dueling with LR_{CHP} of 0.001 and LR_{NOP} of 0.000 4.

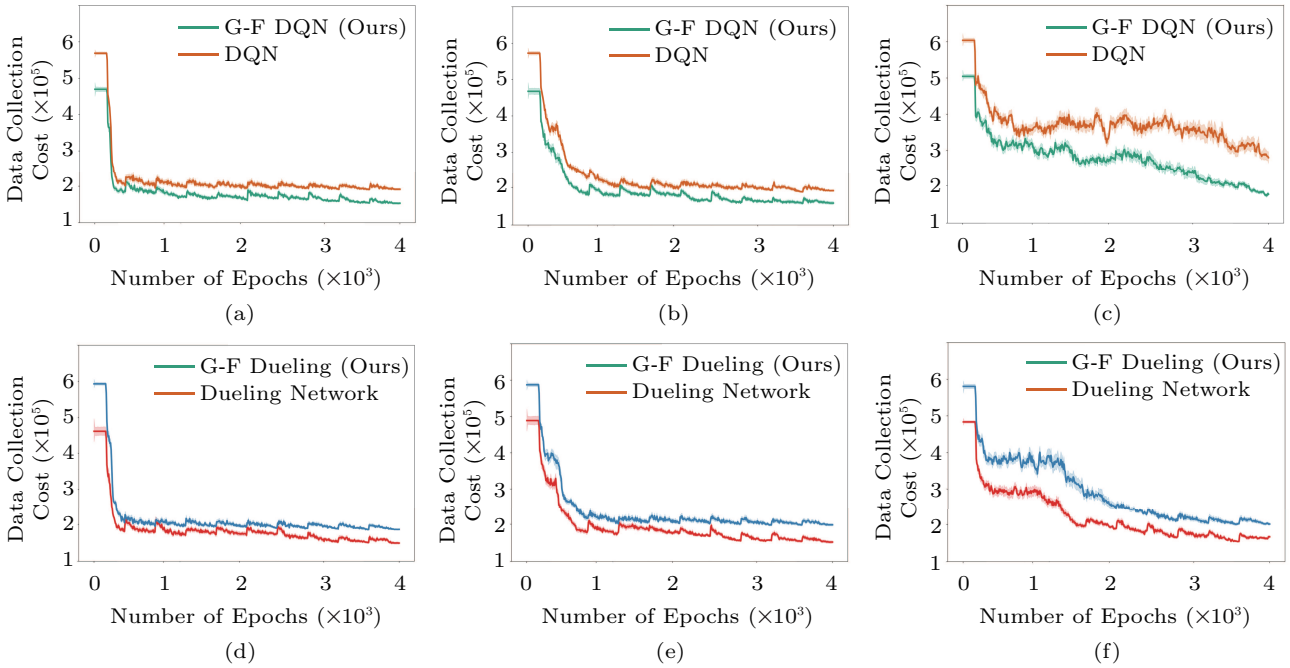

Fig.6. Data collection cost comparison with different learning rates of the NOP. (a) G-F DQN with LR_{CHP} of 0.001 and LR_{NOP} of 0.004. (b) G-F DQN with LR_{CHP} of 0.001 and LR_{NOP} of 0.000 4. (c) G-F DQN with LR_{CHP} of 0.001 and LR_{NOP} of 0.000 04. (d) G-F Dueling with LR_{CHP} of 0.001 and LR_{NOP} of 0.01. (e) G-F Dueling with LR_{CHP} of 0.001 and LR_{NOP} of 0.002. (f) G-F Dueling with LR_{CHP} of 0.001 and LR_{NOP} of 0.000 4.

contrary, a smaller learning rate experiment shows a fluctuating trend and can hardly achieve a stable status. For both situations, G-F Dueling shows the best convergence of energy conservation. Table 3 presents

Table 2. Energy Conservation Comparison of G-F DQN

Learning Rate	Average Energy Consumption ($\times 10^5$)			Best Energy Consumption ($\times 10^5$)		
	DQN	G-F DQN (Ours)	Energy Conserving (%)	DQN	G-F DQN (Ours)	Energy Conserving (%)
0.004	1.89 ± 0.09	1.53 ± 0.09	20.03	1.68 ± 0.04	1.23 ± 0.03	22.69
0.000 4	1.91 ± 0.08	1.58 ± 0.13	17.58	1.69 ± 0.06	1.31 ± 0.04	22.18
0.000 04	2.87 ± 0.41	1.78 ± 0.14	38.01	2.25 ± 0.14	1.55 ± 0.06	31.10

Table 3. Energy Consumption Comparison of G-F Dueling

Learning Rate	Average Energy Consumption ($\times 10^5$)			Best Energy Consumption ($\times 10^5$)		
	Dueling Network	G-F Dueling (Ours)	Energy Conserving (%)	Dueling Network	G-F Dueling (Ours)	Energy Conserving (%)
0.01	1.86 ± 0.19	1.48 ± 0.09	20.43	1.71 ± 0.01	1.28 ± 0.04	24.82
0.002	1.98 ± 0.19	1.51 ± 0.11	23.81	1.76 ± 0.04	1.31 ± 0.06	25.61
0.000 4	2.02 ± 0.11	1.63 ± 0.09	19.11	1.78 ± 0.06	1.33 ± 0.05	25.38

the average and the best energy-conservation comparison results. It can be seen from Table 3 that better solutions in all cases come from the experiment with G-F Dueling. Specifically, the G-F Dueling can conserve energy of 19.11%–25.61% more than the Dueling network.

6 Conclusions

This paper developed a DRL-based game framework with two parts, the CHP and the NOP, to find the sink nodes of the clustered WSN, and optimized trajectories for UAV-enabled data delivery. The CHP was utilized to determine the cluster heads instead of directly using the central nodes as the navigation points. The NOP was employed to find the optimized trajectories for the UAV. Moreover, the developed game framework provides an interactive pattern for the CHP and the NOP and gets the optimized solution when the partners achieve the Nash equilibrium. The numerical results showed that the developed method could provide lower-cost data collection solutions. More specifically, compared with the benchmark methods, the DRL methods with a game framework could enable the UAV to save more than 17.58% of energy and save 38% of energy in the best cases. It is suggested that the developed game framework, with its advancement, could allow a UAV to reduce the overestimation and find a better energy conservation path for open area data delivery. In the future, we will further analyze the feasibility of the DRL-based game framework with three partners, including sensor clustering, sink node finding, and path processing.

References

[1] Chen Q, Zhu H, Yang L, Chen X Q, Pollin S, Vinogradov E. Edge computing assisted autonomous flight for UAV: Synergies between vision and communications. *IEEE Communications Magazine*, 2021, 59(1): 28-33. DOI: 10.1109/MCOM.001.2000501.

[2] Liu D X, Xu Y H, Wang J L, Chen J, Yao K L, Wu Q H, Anpalagan A. Opportunistic UAV utilization in wireless networks: Motivations, applications, and challenges. *IEEE Communications Magazine*, 2020, 58(5): 62-68. DOI: 10.1109/MCOM.001.1900687.

[3] Ma M, Yang Y Y, Zhao M. Tour planning for mobile data-gathering mechanisms in wireless sensor networks. *IEEE Trans. Vehicular Technology*, 2013, 62(4): 1472-1483. DOI: 10.1109/TVT.2012.2229309.

[4] Zhan C, Zeng Y, Zhang R. Energy-efficient data collection in UAV enabled wireless sensor network. *IEEE Wireless Communications Letters*, 2018, 7(3): 328-331. DOI: 10.1109/LWC.2017.2776922.

[5] Chai C L, Liu J B, Tang N, Li G L, Luo Y Y. Selective data acquisition in the wild for model charging. *Proceedings of the VLDB Endowment*, 2022, 15(7): 1466-1478. DOI: 10.14778/3523210.3523223.

[6] Chai C L, Cao L, Li G L, Li J, Luo Y Y, Madden S. Human-in-the-loop outlier detection. In *Proc. the 2020 ACM SIGMOD Int. Conf. Management of Data*, June 2020, pp.19-33. DOI: 10.1145/3318464.3389772.

[7] Dong M X, Ota K, Lin M, Tang Z Y, Du S G, Zhu H J. UAV-assisted data gathering in wireless sensor networks. *The Journal of Supercomputing*, 2014, 70(3): 1142-1155. DOI: 10.1007/s11227-014-1161-6.

[8] Zhan C, Zeng Y. Aerial–ground cost tradeoff for multi-UAV-enabled data collection in wireless sensor networks. *IEEE Trans. Communications*, 2020, 68(3): 1937-1950. DOI: 10.1109/TCOMM.2019.2962479.

[9] Asadi K, Kalkunte Suresh A, Ender A, Gotad S, Maniyar S, Anand S, Noghabaei M, Han K, Lobaton E, Wu T F. An integrated UGV-UAV system for construction site data collection. *Automation in Construction*, 2020, 112: Article No. 103068. DOI: 10.1016/j.autcon.2019.103068.

[10] Chai C L, Li G L, Li J, Deng D, Feng J H. Cost-effective crowdsourced entity resolution: A partial-order approach. In *Proc. the 2016 International Conference on Management of Data*, June 2016, pp.969-984. DOI: 10.1145/2882903.2915252.

[11] Li G L, Chai C L, Fan J, Weng X P, Li J, Zheng Y D, Li Y B, Yu X, Zhang X H, Yuan H T. CDB: Optimizing queries with crowd-based selections and joins. In *Proc. the 2017 International Conference on Management of Data*, May 2017, pp.1463-1478. DOI: 10.1145/3035918.3064036.

[12] Chai C L, Fan J, Li G L. Incentive-based entity collection using crowdsourcing. In *Proc. the 34th International Conference on Data Engineering*, April 2018, pp.341-352. DOI: 10.1109/ICDE.2018.00039.

[13] Baek J, Han S I, Han Y. Energy-efficient UAV routing for wireless sensor networks. *IEEE Trans. Vehicular Technology*, 2020, 69(2): 1741-1750. DOI: 10.1109/TVT.2019.2959808.

[14] Zhao S L, Wang X K, Kong W W, Zhang D B, Shen L C. A novel data-driven control for fixed-wing UAV path following. In *Proc. the 2015 IEEE International Conference on Information and Automation*, Apr. 2015, pp.3051-3056. DOI: 10.1109/ICInfA.2015.7279812.

[15] Rossello N B, Carpio R F, Gasparri A, Garone E. Information-driven path planning for UAV with limited autonomy in large-scale field monitoring. *IEEE Trans. Automation Science and Engineering*, 2022, 19(3): 2450-2460. DOI: 10.1109/TASE.2021.3085365.

[16] Hydher H, Jayakody D N K, Hemachandra K T, Samarasinghe T. Intelligent UAV deployment for a disaster-resilient wireless network. *Sensors*, 2020, 20(21): Article No. 6140. DOI: 10.3390/s20216140.

[17] Chen W C, Zhao S J, Zhang R Q, Chen Y, Yang L Q. UAV-assisted data collection with nonorthogonal multiple access. *IEEE Internet of Things Journal*, 2021, 8(1): 501-511. DOI: 10.1109/JIOT.2020.3005271.

[18] Xiong Z H, Zhang Y, Lim W Y B, Kang J W, Niyato D, Leung C, Miao C Y. UAV-assisted wireless energy and data transfer with deep reinforcement learning. *IEEE Trans. Cognitive Communications and Networking*, 2021, 7(1): 85-99. DOI: 10.1109/TCCN.2020.3027696.

[19] Duo B, Wu Q Q, Yuan X J, Zhang R. Anti-jamming 3D trajectory design for UAV-enabled wireless sensor networks under probabilistic LoS channel. *IEEE Trans. Vehicular Technology*, 2020, 69(12): 16288-16293. DOI: 10.1109/TVT.2020.3040334.

[20] Challita U, Saad W, Bettstetter C. Interference management for cellular-connected UAVs: A deep reinforcement learning approach. *IEEE Trans. Wireless Communications*, 2019, 18(4): 2125-2140. DOI: 10.1109/TWC.2019.2900035.

[21] Xie H, Yang D C, Xiao L, Lyu J B. Connectivity-aware 3D UAV path design with deep reinforcement learning. *IEEE Trans. Vehicular Technology*, 2021, 70(12): 13022-13034. DOI: 10.1109/TVT.2021.3121747.

[22] Mukherjee A, Misra S, Chandra V S P, Obaidat M S. Resource-optimized multiarmed bandit-based offload path selection in edge UAV swarms. *IEEE Internet of Things Journal*, 2019, 6(3): 4889-4896. DOI: 10.1109/JIOT.2018.2879459.

[23] Zhu B T, Bedeer E, Nguyen H H, Barton R, Henry J. UAV trajectory planning in wireless sensor networks for energy consumption minimization by deep reinforcement learning. *IEEE Trans. Vehicular Technology*, 2021, 70(9): 9540-9554. DOI: 10.1109/TVT.2021.3102161.

[24] Zhang S T, Li Y B, Dong Q H. Autonomous navigation of UAV in multi-obstacle environments based on a deep reinforcement learning approach. *Applied Soft Computing*, 2022, 115: 108194. DOI: 10.1016/j.asoc.2021.108194.

[25] Shi T, Li J Z, Gao H, Cai Z P. A novel framework for the coverage problem in battery-free wireless sensor networks. *IEEE Trans. Mobile Computing*, 2022, 21(3): 783-798. DOI: 10.1109/TMC.2020.3019470.

[26] Zhang L P, Li F Q, Wang P C, Su R, Chi Z Z. A blockchain-assisted massive IoT data collection intelligent framework. *IEEE Internet of Things Journal*, 2021, 9(16): 14708-14722. DOI: 10.1109/JIOT.2021.3049674.

[27] Wang K, Xu L, Perrault A, Reiter M K, Tambe M. Coordinating followers to reach better equilibria: End-to-end gradient descent for stackelberg games. In *Proc. the 36th AAAI Conf. Artificial Intelligence*, 2022, pp.5219-5227. DOI: 10.1609/aaai.v36i5.20457.

[28] van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double Q-learning. In *Proc. the 30th AAAI Conf. Artificial Intelligence*, February 2016, pp.2094-2100. DOI: 10.1609/aaai.v30i1.10295.

[29] Liu J B, Chai C L, Luo Y Y, Lou Y, Feng J H, Tang N. Feature augmentation with reinforcement learning. In *Proc. the 38th Int. Conf. Data Engineering*, May 2022, pp.3360-3372. DOI: 10.1109/ICDE53745.2022.00317.

[30] Watkins C J C H, Dayan P. *Q*-learning. *Machine Learning*, 1992, 8(3): 279-292. DOI: 10.1007/BF00992698.

[31] Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, Riedmiller M. Playing Atari with deep reinforcement learning. arXiv: 1312.56022013, 2013. https://arxiv.org/abs/1312.5602, Nov. 2022.

[32] Lillicrap T P, Hunt J J, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D. Continuous control with deep reinforcement learning. In *Proc. the 4th International Conference on Learning Representations*, May 2016.

[33] Barto A G, Sutton R S, Anderson C W. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Trans. Systems, Man, and Cybernetics*, 1983, SMC-13(5): 834-846. DOI: 10.1109/TSMC.1983.6313077.

[34] Bian W W, Wei J, Huang K H, Wang J X, Lv X, Yuan W N. Intelligent decision algorithm of target compound interception based on A2C-PPO. In *Proc. the 2021 International. Conference on Cyber-Physical Social Intelligence*, December 2021. DOI: 10.1109/ICCSI53130.2021.9736236.

[35] Mnih V, Badia A P, Mirza M, Graves A, Harley T, Lillicrap T P, Silver D, Kavukcuoglu K. Asynchronous methods for deep reinforcement learning. In *Proc. the 33rd International Conference on Machine Learning*, Jun 2016, pp.1928-1937.

[36] Schaul T, Quan J, Antonoglou I, Silver D. Prioritized experience replay. In *Proc. the 4th International Conference on Learning Representations*, May 2016.

[37] Wang Z Y, Schaul T, Hessel M, Van Hasselt H, Lanctot M, De Freitas N. Dueling network architectures for deep reinforcement learning. In *Proc. the 33rd International Conference on Machine Learning*, June 2016, pp.1995-2003.DOI: 10.5555/3045390.3045601.

[38] Su Z, Qi N, Yan Y J, Du Z Y, Chen J X, Feng Z B, Wu Q H. Guarding legal communication with smart jammer: Stackelberg game based power control analysis. *China Communications*, 2021, 18(4): 126-136. DOI: 10.23919/JCC.2021.04.010.

[39] Bansal G, Sikdar B. Security service pricing model for UAV swarms: A stackelberg game approach. In *Proc. the 2021 IEEE Conference on Computer Communications Workshops*, May 2021, pp.126-136. DOI: 10.1109/INFOCOMWKSHPS51825.2021.9484577.

[40] Shi C G, Qiu W, Wang F, Salous S, Zhou J J. Cooperative LPI performance optimization for multistatic radar system: A stackelberg game. In *Proc. the 2019 International. Applied Computational Electromagnetics Society Symposium*, Aug. 2019. DOI: 10.23919/ACES48530.2019.9060749.

[41] Su J T, Yang S S, Xu H T, Zhou X W. A stackelberg differential game based bandwidth allocation in satellite communication network. *China Communications*, 2018, 15(8): 205-214. DOI: 10.1109/CC.2018.8438284.

[42] Zhang X B, Wang H, Xu Y F, Feng Z B, Zhang Y P. Put others before itself: A multi-leader one-follower anti-jamming stackelberg game against tracking jammer. *China Communications*, 2021, 18(11): 168-181. DOI: 10.23919/JCC.2021.11.012.

[43] Ghorbel M B, Rodríguez-Duarte D, Ghazzai H, Hossain M J, Menouar H. Joint position and travel path optimization for energy efficient wireless data gathering using unmanned aerial vehicles. *IEEE Trans. Vehicular Technology*, 2019, 68(3): 2165-2175. DOI: 10.1109/TVT.2019.2893374.

[44] Hulens D, Verbeke J, Goedemé T. How to choose the best embedded processing platform for on-board UAV image processing? In *Proc. the 10th International Conference on Computer Vision Theory and Applications*, Mar. 2015, pp.377-386. DOI: 10.5220/0005359403770386.

Tong Ding received his B.S. degree from Nanjing University of Posts and Telecommunications, Nanjing, in 2016, and his M.S. degree from China University of Petroleum, Qingdao, in 2020, both in software engineering. Currently, he is working towards his Ph.D. degree with the School of Software at Shandong University, Jinan. His research interests include UAV path processing, reinforcement learning, and federated learning.

Ning Liu received his Ph.D. degree in computer science and technology from Tsinghua University, Beijing, in 2021. He is currently an assistant professor in the School of Software, Shandong University, Jinan. His research interests mainly include data mining and knowledge-driven applications, especially textural data and sequential data mining. He now is a member of CCF.

Zhong-Min Yan received her M.S. degree from the College of Computer Science and Technology, Shandong University, Jinan, in 2001, and her Ph.D. degree from Shandong University, Jinan, in 2010, both in computer software and theory. She is currently with the School of Software, Shandong University, Jinan. Her research interests include big data analytics. She has published over 30 research papers in international conferences and journals.

Lei Liu is a full professor in the School of Software, Shandong University, Jinan. He obtained his M.S. degree in software engineering and Ph.D. degree in computer science and technology in 2005 and 2010 respectively, from Bradford University, UK. Dr. Liu has published over 70 research papers in international conferences and journals. His research interests include network performance engineering, 5G technology, quality of service, IoT and UAVs.

Li-Zhen Cui is a professor in the School of Software, Shandong University, Jinan. He received his M.S. and Ph.D. degrees in computer science and technology from Shandong University, Jinan, in 2002, and 2005, respectively. His research interests include data mining, trustworthy artificial intelligence and smart healthcare. He has published in TKDE, TPAMI, TPDS, Scientific Data, KDD, SIGIR, WWW, CVPR, ICDE, IJCAI, AAAI, CIKM, and other venues.