# On Collaborative Multi-UAV Trajectory Planning for Data Collection

Shahnila Rahim, Limei Peng, Shihyu Chang, and Pin-Han Ho

***Abstract*—This paper investigates the scenario of the Internet of things (IoT) data collection via multiple unmanned aerial vehicles (UAVs), where a novel collaborative multi-agent trajectory planning and data collection (CMA-TD) algorithm is introduced for online obtaining the trajectories of the multiple UAVs without any prior knowledge of the sensor locations. We first provide two integer linear programs (ILPs) for the considered system by taking the coverage and the total power usage as the optimization targets. As a complement to the ILPs and to avoid intractable computation, the proposed CMA-TD algorithm can effectively solve the formulated problem via a deep reinforcement learning (DRL) process on a double deep Q-learning network (DDQN). Extensive simulations are conducted to verify the performance of the proposed CMA-TD algorithm and compare it with a couple of state-of-the-art counterparts in terms of the amount of served IoT nodes, energy consumption, and utilization rates.**

***Index Terms*—Collaborative UAVs, data collection, deep reinforcement learning, energy efficiency, IoT coverage, trajectory planning.**

## I. Introduction

DATA collection is one of the major applications of Internet of thing (IoT) systems, and its design is subject to many challenges, particularly in the event that each thing is under stringent capacity constraints related to power consumption, computation, and communication ranges. It has been an emerging application by using UAVs as aerial access points for data collection where the terrestrial telecommunication infrastructure, such as mobile data services, is unavailable [1]–[3]. There are numerous advantages to using UAVs for IoT data collection in the aspects of cost-effectiveness, adaptation to the environment, ad hoc network access of UAVs, as well as the possibility of a line-of-sight (LoS) transmission link thanks to the high altitude of the UAVs [4]–[6].

Despite the numerous advantages of employing UAVs in IoT data collection, a number of issues arise and need to be addressed before the considered scenario can be readily launched. The most critical one is to cope with the excessive energy consumption by the UAVs to support their cruising/hovering in the course of data collection [7], which could seriously impair the operational efficiency and applicability. An energy-efficient resource allocation strategy has been considered essential in achieving satisfactory performance for UAV-based data collection [8]. In [9], a multi-dimension search space was proposed to enhance the energy efficiency of a single battery-power-limited UAV in data collection scenarios.

### A. Literature Review

Extensive research has been conducted on the topic of UAV-assisted data collection for IoT systems. In [15], the authors optimized the data offloading with minimum power usage by formulating and solving a non-convex problem. The authors in [17] and [18] considered using multiple UAVs for data collection, and they attempted to achieve an optimized trajectory design via clustering and cluster-heads formation to minimize the total power consumption, where the clustering is performed based on prior knowledge of the environment. In [19], the authors examined the power efficiency of a single UAV where hovering points were predefined and considered the fly-hover-communication design to optimize hovering points and flight duration.

Until now, very few studies have been reported to deal exclusively with the UAV trajectory design problem by considering the partially observed networking environment and unexpected mobility/locations of the IoT devices. In [20], the authors introduced a model-based deep reinforcement learning (DRL) UAV path planning algorithm for data collection, where a device localization mechanism was used by dividing the ground nodes into either known or unknown locations. Nonetheless, they made the assumption that the UAVs are given predetermined targets and the IoT nodes are static with complete location information. In [10], the authors proposed a novel approach named meta-TD3 that integrates DRL with meta-learning to control a UAV for tracking uncertain moving targets in various scenarios. Moreover in [12] and [13], the authors focused on single-UAV trajectory design and considered radio resource allocation in their optimization.

It is clear that launching multiple UAVs can achieve much better efficiency and economic effectiveness compared to its single-UAV counterpart, where multiple UAVs are managed to cover the given geographic area and collect information, e.g., taking photos, from different perspectives simultaneously.

Manuscript received March 9, 2023 revised June 8, 2023; approved for publication by Abbas Jamalipour, Division 2 Editor, July 10, 2023.

This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Korean Government (Grant No.: 2020R1I1A3072688) and the Brain Pool program funded by the Ministry of Science and ICT through the National Research Foundation of Korea (Grant number: NRF-2022H1D3A2A01063679).

S. Rahim and L. Peng are with the School of Computer Science and Engineering, Kyungpook National University, Daegu, South Korea. email: {shahnila.rahim, auroraplm}@knu.ac.kr.

S. Chang is with the Department of Applied Data Science, San Jose State University, San Jose, USA. email: shihyu.chang@sjsu.edu.

P.-H. Ho is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. email: p4ho@uwaterloo.ca.

L. Peng is the corresponding author.

Digital Object Identifier: 10.23919/JCN.2023.000031



1229-2370/23/$10.00 © 2023 KICS

Table I
COMPARISON BETWEEN THE PROPOSED SOLUTION AND EXISTING WORKS.

| Ref | Collaborative UAVs | Unknown environment | Dynamic environment | Adaptive beamwidth | Energy efficiency | Trajectory Optimization | RL | IoT nodes coverage maximization |
|---|---|---|---|---|---|---|---|---|
| [10] | | ✓ | ✓ | | | | ✓ | |
| [9] | | | | | ✓ | ✓ | ✓ | |
| [6] | | | | | | | | |
| [11] | ✓ | | | | | ✓ | ✓ | |
| [12] | | | ✓ | | ✓ | ✓ | ✓ | |
| [13] | | | | | ✓ | ✓ | | ✓ |
| [14] | | | | | ✓ | | ✓ | ✓ |
| [7] | | | | | ✓ | ✓ | | |
| [15] | | | | | ✓ | ✓ | | |
| [16] | | | | | ✓ | ✓ | | |
| Proposed work | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

In spite of its apparent advantage, the multi-UAV scenario is subject to a number of challenges that need to be addressed before making such a scenario in reality [21]. The authors in [2] investigated an online multi-UAV age-of-information-aware planning process where the mobility of IoT devices is randomly generated. Nonetheless, it was stated that the majority of the proposed policies related to the UAV-enabled optimizations are just for a single UAV while the collaboration among the multiple UAVs is not fully explored and thus cannot take full advantage of the multi-UAV collaboration [2], [22]. A multi-UAV-assisted wireless system was studied in [16] with the aim of maximizing the minimum throughput amongst all the ground nodes. In [23], the mobility and deployment of multi-UAVs were studied to collect data from IoT nodes such that the transmitted energy of the IoT node is minimized. However, the collaboration among the UAVs and overall system utilization by using multiple UAVs were not discussed. Authors in [11] optimize resource allocation and trajectories with multiple UAVs using multi-agent RL and a distributed learning framework to enhance the overall fairness and system throughput.

It is notable that all the above studies designed the UAV trajectory in an offline manner, which may not be feasible due to the ideal assumptions that all the environmental conditions and parameters, such as the locations of the sensors and the amount of data to be collected, are available. Instead, an intelligent solution without prior knowledge of the predetermined targets is desired. As such, people have resorted to solutions based on reinforcement learning (RL), which does not require historical data for training, while serving as an excellent complement to the conventional offline optimization solutions thanks to its superb ability to learn unknown environments in a trial-and-error manner [24]–[26]. Nevertheless, model-free RL algorithms, e.g., Q-learning, generally consider a large number of states and actions and thus require a large amount of memory to obtain the optimal policy. To reduce the computational complexity, a combination of neural networks and RL, namely DRL, is exceptionally suitable for high-dimensional problems with complex state space and time-varying environments, where a deep neural network (DNN) is used to guide decision-making for satisfying performance with even zero domain knowledge [14], [27].

With DRL, the authors in [14] proposed a trajectory planning algorithm for a single UAV on an IoT data harvesting mission maximizing energy efficiency. The solution proposed there was to try to maximize the fairness of communication coverage. In [8], the authors designed a UAV-aided IoT system relying on the shortest flight route of the UAV while maximizing the volume of data collected from the IoT nodes. After that, they applied a DRL-based method for optimal path discovery and throughput maximization in a particular coverage region.

### *B. Motivation and Contributions*

From the existing literature, we find some serious issues not well addressed. Firstly, the previously reported studies are mostly based on an impractical assumption that the environmental information is fully observable by each UAV. Secondly, the previously reported studies do not allow UAVs to learn online. Even though some existing works considered artificial intelligence, but may fail to incorporate the unknown environments in the trajectory planning and IoT node communication with minimum energy consumption.

Thus, the paper attempts to resolve the above-mentioned two aspects by investigating a novel DRL-based scheme, namely the CMA-TD algorithm, which is characterized by employing a double deep Q-learning network (DDQN) for online training. The contributions of this paper are summarized as follows:

- We formulate two optimization problems into integer linear programs (ILP) for maximizing the coverage of IoT nodes and minimizing the required total energy by considering a wireless communication channel and a set of UAV parameters.
- To avoid intractable computation complexity in solving the ILPs, the proposed DRL-based CMA-TD algorithm serves as an effective online framework for tackling dynamic multi-UAV environments.
- Extensive simulation is launched to evaluate the performance of the proposed DRL-based CMA-TD algorithm in terms of the number of successfully served IoT nodes, energy consumption, and utilization rate. We also compare

Table II
LIST OF PARAMETER NOTATIONS.

| Parameters | Values |
|---|---|
| $\mathcal{I}$ | Set of IoT nodes |
| $m \times n$ | Number of unit cells |
| $H$ | Flying altitude of UAV $(m)$ |
| $T_{\max}$ | Maximum flying time |
| $E_{j,\max}$ | Maximum power of the UAV $j$ $(W)$ |
| $\mathcal{U}$ | Set of UAVs |
| $SP_j$ | Starting position of UAV $j$ |
| $P_f$ | Final position of UAVs |
| $Q_j^t = (x_j^t, y_j^t, H)$ | Coordinates of UAV $j$ at time $t$, |
| $G$ | Antenna gain $(dB)$ |
| $E_{comm}$ | Communication related energy $(W)$ |
| $E_{prop}$ | Propulsion energy $(W)$ |
| $E_{hover}$ | Hovering energy $(W)$ |
| $E_{DC}$ | Power consumed by data collection $(W)$ |
| $\xi_0$ | Blade profile |
| $\xi_1$ | Induced power of UAV |
| $v_0$ | Rotor induced velocity $(m/s)$ |
| $\mu_{tip}$ | Tip of the rotor blade $(m/s)$ |
| $z_0$ | Fuselage drag ratio $(m^2)$ |
| $\tau$ | Rotor solidity |
| $\kappa$ | Air density $(kg/m^3)$ |
| $A$ | Rotor disc area $(m^2)$ |
| $B$ | Bandwidth |
| $E_t$ | Total power consumption $(W)$ |
| $d_{\min}$ | Minimum distance between UAVs $(m)$ |
| $\alpha$ | Path-loss exponent |
| $\sigma$ | Noise variance |
| $Y_{i,j} \in \{0, 1\}$ | Binary variable, 1 if UAV $j$ can successfully serve IoT node $i$, and 0 otherwise. |

the proposed scheme with two previously reported online learning algorithms, namely distributed multi-agent Q learning (DMA-QL) [28] and deep Q network-based trajectory and data collection optimization (DQN-TDCO) [8] in designing the multi-UAVs trajectories. Our simulation results demonstrate that the proposed DRL-based CMA-TD algorithm outperforms its counterparts under various dynamic environments and wireless channels.

### C. Organization

The rest of this paper is organized as follows. Section II presents the system model which includes the network architecture, energy consumption model, and reinforcement learning fundamentals employed in this study. Section III presents the problem formulations in ILP and a DRL process for the multi-UAV data collection scenario considered in this study. Section IV presents the proposed DRL-based CMA-TD algorithm. Simulation results are presented in Section V. Section VI concludes the paper.

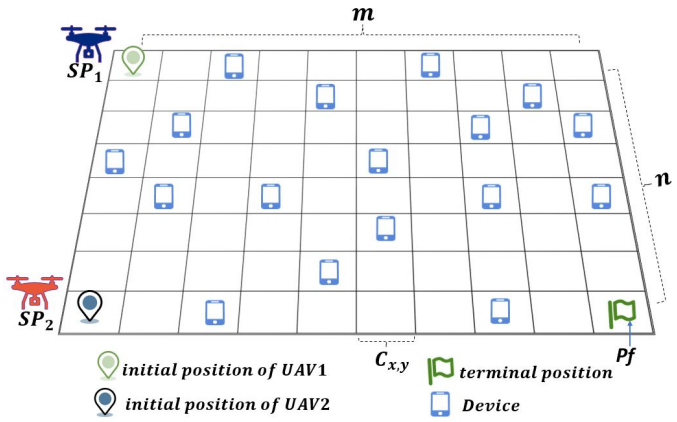



Fig. 1. System architecture.

## II. SYSTEM MODEL

### A. Network Architecture

We consider a UAV-assisted IoT network with a set of IoT nodes $\mathcal{I} = \{1, \cdots, I\}$, distributed randomly in the area of interest (AoI). A set of UAVs $\mathcal{U} = \{1, \cdots, U\}$ are launched for data collection from the IoT nodes without the assistance of any terrestrial communication infrastructure. Let the AoI be divided into $m \times n$ equal-sized square grid cells where $m, n \in \mathbb{N}$. Let the coordinates of the grid cells be denoted by $c_{x,y}$, where $c_{x,y} = \{c_{1,1}, c_{1,2}, \cdots, c_{m,n}\}$. Without loss of generality, let the UAVs be flying over the AoI at altitude $H$ and attempting to maximize the number of served IoT nodes within the given serving time period $ST_{\max}$. We assume the UAVs do not have any prior knowledge regarding the locations of the IoT nodes, while they can communicate with each other and share their coordinates to avoid collisions.

Let the UAVs start their mission at the initial starting positions, denoted as $SP_j$, where $j \in \mathcal{U}$. The current location of UAV $j$, at time step $t$ is defined as $Q_j^t = (x_j^t, y_j^t, H)$, where $t \in \mathcal{T} = \{0, 1, 2, \cdots, T\}$, where $T$ is the final time step and $H$ is the altitude of UAV $j$. Let the resting location of the UAVs be denoted as $Pf$ as shown in Fig. 1.

Fig. 2 illustrates that the collaborative UAVs start their mission and plan their trajectories for coverage maximization while avoiding collision with each other. We assume that each UAV is equipped with a directional antenna of adjustable beamwidth for data collection, and another set of antennas for inter-UAV communication.

For simplicity, we assume that the azimuth and elevation half-power beamwidths of the UAVs antenna are equal, where both angles are measured as $2\vartheta$ in radian for $\vartheta \in (0, \frac{\pi}{2})$ [29]. Furthermore, the corresponding antenna gain in direction $(\theta, \delta)$ is approximately modeled as

$$G(\theta, \delta) = \begin{cases} \frac{G_0}{\vartheta^2}, & -\vartheta \leq \theta \leq \vartheta, -\vartheta \leq \delta \leq \vartheta, \\ g \approx 0, & otherwise. \end{cases}, \quad (1)$$

where $G_0 = (30000/2^2) \times (\pi/180)^2 \approx 2.2846$; $\theta$ and $\delta$ are azimuth and elevation angle, respectively [29]. Note that, in practice $g$ satisfies $0 < g << G_0/\vartheta^2$, and for simplicity we assume $g = 0$. The beamwidth angle is adjusted according to

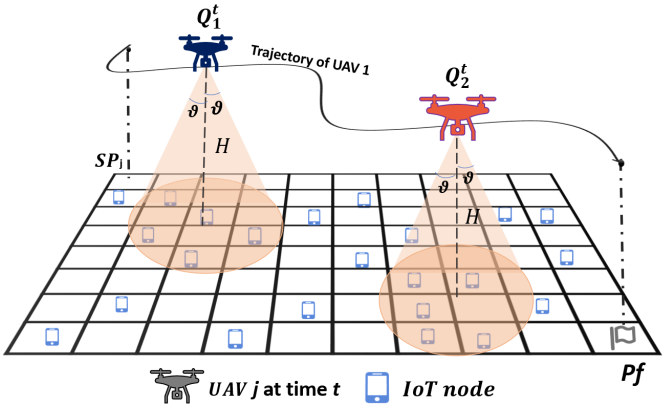



Fig. 2. An illustration of collaborative UAV-enabled wireless communication with dynamic beamwidth.

the number of IoT nodes detected. As exemplified in Fig. 2, UAV $j$ at time $t$ has detected a number of $N_{j,t}$ IoT nodes. To strengthen the received signal, UAV $j$ changes the antenna angle to narrow the lobe until at least $N_{j,t}/2$ IoT nodes are in the range.

With a non-observable environment, the UAVs have no prior knowledge of the IoT node locations. Moreover, when data of a node is collected by one of the UAVs, the node is then marked as collected to avoid double collection by another UAV. In a large area of certain parameters and grid sizes, we are interested in the minimal number of collaborative UAVs required to accomplish a given target of data collection provided with minimum total energy consumption.

### B. Reinforcement Learning

This study applies a reinforcement learning method to solve the proposed CMA-TD problem which is modeled as a Markov decision process (MDP), denoted by a 4-element tuple, i.e., $M = (S, A, R, P_a)$. Here, $A$, $S$, and $R$ represent the action, the state, and the reward function, respectively; $P_a$ is the probability of transiting from state $s$ to state $s'$. A policy $\pi$ in RL is a mapping from state $s$ to action $a$. However, the policy controls the agent's action and, consequently, the rewards it obtains. The agent in the MDP learns from scratch in a trial-and-error manner by taking time-discrete actions to interact with the environment [22], [30]. Specifically, the state observed by the agent in each time slot $t$, denoted by $s_t \in S$, takes action $a_t \in A$ and gets a negative or positive reward $r_t \in R$. As the process iterates, the agent propagates to the new state $s_{t+1}$ according to the policy $\pi$. The reward at time step $t$, i.e., $R_t$, is expressed as follows,

$$R_t = \sum_{t=1}^{T} \gamma^{t-1} r_t, \tag{2}$$

where $\gamma$ is the discount factor ranging from 0 to 1, and a larger $\gamma$ value indicates the significance of future rewards. $T$ is the final time step. Specifically, the agent seeks behavior policy $\pi$ that can maximize the cumulative expected reward, also known as the Q function, given as:

$$Q_\pi(s_t, a_t) = \operatorname{argmax} \mathbb{E}[R_t | s_t, a_t]. \tag{3}$$

This study employs the double deep Q-network (DDQN) technique [31] with the target value given by:

$$Y_t^{DD} = r_{t+1} + \gamma Q_{\tilde{\theta}}(s_{t+1}, \underset{a_t+1}{\operatorname{argmax}}\, Q_\theta(s_{t+1}, a_{t+1})), \tag{4}$$

and the corresponding loss function is expressed as:

$$L^{DD}(\theta) = \mathbb{E}[(Q_\theta(s_t, a_t)) - Y_t^{DD})^2]. \tag{5}$$

With DDQN, two parameters $\theta$ and $\tilde{\theta}$ are introduced, which are used to suppress any possible overestimation of the action values and estimate the value of that action, respectively. When calculating $L^{DD}(\theta)$, the target value is taken and thus the back-propagating gradient is stopped before $Y_t^{DD}$.

### C. Energy Consumption Model

There are two main parts for the UAV power consumption: (i) Propulsion power, and (ii) communication-related power. The communication-related power is utilized when the UAVs send, process, and receive signals, denoted by $E_{comm}(j)$ for UAV $j$. The propulsion power is consumed while hovering and flying, which is formulated as a function of its speed [19]:

$$E_{prop}(j) = \xi_0 \left(1 + \frac{3v^2}{\mu_{tip}^2}\right) + \xi_1 \left(\sqrt{1 + \frac{v^4}{4v_0^2}} - \frac{v^2}{2v_0^2}\right)^{\frac{1}{2}} + \left(\frac{1}{2} z_0 \tau \kappa A v^3\right), \tag{6}$$

where $\xi_0$ and $\xi_1$ are constant parameters that denote blade profile and induced power, respectively. While the UAV is at its hovering state, $\mu_{tip}$ represents the tip of the rotor blade and we use $v_0$ to represent the induced rotor velocity during hovering. Furthermore, $\kappa$, $\tau$, $z_0$, and $A$ are parameters representing air density, rotor disc area, rotor solidity, and fuselage drag ratio, respectively.

To calculate the energy consumed by UAV $j$ during hovering, we take $v = 0$ in (6), then the hovering energy can be expressed as:

$$E_{hover}(j) = \xi_0 + \xi_1. \tag{7}$$

When a UAV collects data from an IoT node, it hovers above the IoT nodes and consumes communication-related power. Thus, the total power consumed during data collection by UAV $j$ is denoted by $E_{DC}(j)$ and given as:

$$E_{DC}(j) = E_{hover}(j) + E_{comm}(j), \tag{8}$$

Whereas, the total power consumed by UAV $j$ at time $t$ can be calculated as:

$$E_j^t = E_{DC}^t(j) + E_{prop}^t(j). \tag{9}$$

During the flying period, the trajectory of UAV $j$ is expressed by a series of $k_j$ cells that are visited by the UAV, denoted as $\hat{i}^j = [\hat{i}_1^j, \hat{i}_2^j, ..., \hat{i}_{k_j}^j]$, is the vector which includes the cells UAV $j$ has visited and served the IoT nodes and $k_j$ represents the last cell visited by UAV $j$. In each move, a UAV can take either one of the discrete four directions

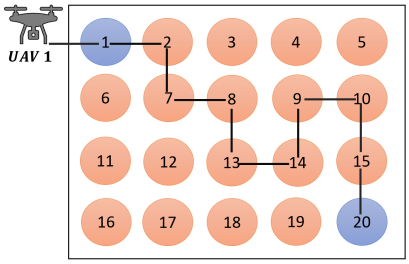



Fig. 3. Trajectory of UAV.

from its current position, i.e., north, south, east, and west. Fig. 3 shows an example of the trajectory of UAV 1 in the AoI that is expressed as the cells visited during the mission, i.e., $\hat{i}^1 = [1, 2, 7, 8, 13, 14, 9, 10, 15, 20]$. The total energy consumption by UAV $j$ in the mission defined by $\hat{i}^j$ is composed of the components due to UAV cruising, hovering, and data collection given in (9), which is denoted as $\tilde{\mathcal{F}}(\hat{i}^j)$:

$$\tilde{\mathcal{F}}(\hat{i}^j) = \sum_{t \in \mathcal{T}} (E^t_{DC}(j) + E^t_{prop}(j)). \tag{10}$$

### D. Channel Model and Data Collection Rate

A UAV operated at a sufficiently high altitude tends to establish LoS links with the ground IoT nodes. However, it also experiences small-scale fading caused by the presence of rich scattering in the environment [32]. For the up-link channel between UAV and IoT node, we use the rician fading channel [33]. The channel model between UAV $j \in \mathcal{U}$ and IoT node $i \in \mathcal{I}$ at time $t \in \mathcal{T}$ can be expressed as:

$$h^t_{j,i} = \sqrt{\beta_{j,i}} g_{j,i}, \tag{11}$$

where $g_{j,i}$ is the small-scale fading co-efficient and $\beta_{j,i}$ is the average channel power gain accounting for signal attenuation, including both shadowing and path loss, which can be expressed as:

$$\beta_{j,i} = \beta_0 d^{-\alpha}_{j,i}, \tag{12}$$

where $d_{j,i}$ denotes the horizontal distance between UAV $j$ and IoT node $i$ at height $H$. $\beta_0$ is the average channel power gain at the reference distance of $d_0 = 1$ m and $\alpha$ is the path loss exponent that usually has a value between 2 and 6 [33].

The small-scale fading of the LoS path can be modeled by rician fading as:

$$g_{j,i} = \sqrt{\frac{\kappa_{j,i}}{\kappa_{j,i}+1}} g + \sqrt{\frac{1}{\kappa_{j,i}+1}} \tilde{g}, \tag{13}$$

where $g$ represents the deterministic LoS channel component with $|g| = 1$. The variable $\tilde{g}$ denotes the randomly scattered component, which follows a zero-mean unit-variance symmetric complex gaussian distribution random variable [32]. The parameter $\kappa_{j,i}$ represents the rician factor of the channel between IoT node $i$ and UAV $j$.

In the considered scenario, UAV $j$ has detected $N$ IoT nodes for communication at time $t$ using the directional antenna by adjusting the beam width. When each of the $N$ IoT nodes is detected, a data communication link is established and UAV $j$ starts to collect data from the $N$ IoT nodes. Let the achievable rate between the IoT node $i$ and UAV $j$ be expressed as:

$$K^t_{j,i} = B \log \left(1 + \frac{\rho_t G_t |h^t_{j,i}|^2}{\sigma^2}\right), \tag{14}$$

where $B$ is the bandwidth of the channel, $\rho_t$ is the transmitted power of the IoT node at time $t$, $G_t$ is the antenna power gain of the ground (IoT node) to the flying UAV link at time $t$, and $\sigma$ is the noise variance. We define $\Gamma$ as $\rho_t G_t |h^t_{j,i}|^2 / \sigma^2$, which is signal-to-noise-ratio.

## III. CMA-TD Problem Formulation

We formulate the proposed CMA-TD problem into two ILPs and a DRL process, respectively, which are presented in this section.

### A. ILP-based Formulations

The CMA-TD problem is first formulated into two ILPs with respective targets. The first is to maximize the total number of served IoT nodes within the flight time $T$. To serve IoT node $i$, its data $D_i$ should be completely collected by UAV $j$ within a given time constraint. Let $Y_{i,j} \in [0, 1]$, $\forall i \in \mathcal{I}$, $j \in \mathcal{U}$, be a binary variable that is equal to 1 if UAV $j$ can successfully serve IoT node $i$, and 0 otherwise. The formulated optimization problem for maximizing the overall served IoT nodes is expressed as:

$$\max_{Y} \quad \sum_{j \in \mathcal{U}} \sum_{i \in \mathcal{I}} Y_{i,j} \tag{15a}$$

$$\text{s.t.} \quad ST_i \leq ST_{\max}, \forall i \in \mathcal{I}, \tag{15b}$$

$$Q^0_j = SP_j, Q_j = Pf, \forall j \in \mathcal{U}, \tag{15c}$$

$$\sum_{j \in \mathcal{U}} \sum_{i \in \mathcal{I}} Y_{i,j} = 1, \tag{15d}$$

$$Y_{i,j} \in \{0, 1\}, \forall i \in \mathcal{I}, \forall j \in \mathcal{U}, \tag{15e}$$

$$\tilde{\mathcal{F}}(\hat{i}^j) \leq E_{j,\max}, \forall j \in \mathcal{U}, \forall i \in \mathcal{I} \tag{15f}$$

Constraint (15b) guarantees that each served IoT node uploads the data at a given serving time $ST_{\max}$. (15c) defines the initial and final position of UAV $j$. In (15d), $Y_{i,j}$ is set to 1 if IoT node $i$ is assigned to UAV $j$, and 0 otherwise. (15e) guarantees $Y_{i,j}$ can be either 0 or 1 at time $t$. (15f) ensures that the energy consumed by UAV $j$ when following a trajectory $\hat{i}^j$ must be less than the maximum energy of UAV $j$.

The second ILP formulation aims to minimize the total power consumed by collaborative UAVs which is given as

follows:

$$\min_{Y} \quad \sum_{j\in\mathcal{U}}\sum_{t\in\mathcal{T}} E_j^t \tag{16a}$$

$$\text{s.t.} \quad E_j^t \le E_{j,\max}, \forall j \in \mathcal{U}, \forall t \in \mathcal{T}, \tag{16b}$$

$$Q_j^t \le \psi, \forall j \in \mathcal{U}, \forall t \in \mathcal{T}, \tag{16c}$$

$$Q_j^0 = SP_j, Q_j = Pf, \forall j \in \mathcal{U}, \tag{16d}$$

$$\tilde{\mathcal{F}}(\hat{i}^j) \le E_{j,\max}, \forall j \in \mathcal{U}, \forall i \in \mathcal{I}, \tag{16e}$$

$$\sum_{j\in\mathcal{U}}\sum_{i\in\mathcal{I}} Y_{i,j} \ge M_j. \tag{16f}$$

Constraints (16b) and (16c) guarantee that UAV $j$ energy consumption at time $t$ should be less than the maximum power and fly inside the AoI, i.e., $\psi$. (16d) defines the initial and final position of UAV $j$. (16e) stipends that the energy consumed by UAV $j$ for following a sequence to detect and serve IoT nodes must be less than the maximum energy of UAV $j$. (16f) ensures that UAV $j$ must cover at least a number of $M_j$ IoT nodes.

### *B. DRL-based Formulation*

Solving the above ILPs could lead to serious scalability issues and hardly be feasible in a large network environment. Accordingly, we are motivated to resort to a DRL-based approach, namely the CMA-TD algorithm, that is expected to not only achieve efficient path planning and energy management but also better scale with the problem size. Given multiple identical UAVs collaboratively working as agents with a similar set of states and actions, the goal of the proposed DRL-based CMA-TD algorithm is to jointly optimize the number of served IoT nodes as well as the overall power consumption.

The state and action of the proposed DRL-based CMA-TD problem are given as follows:

1) **State space**: The state $s_t$ at time $t$ is a five-element tuple given as follows: $s_t = (Q_j^t, \eta_j^t, \varphi_j^t, \sigma_j^t, \chi_j^t)$,
   - $\boldsymbol{Q_j^t}$ is the cell that UAV $j$ is located at time slot $t$.
   - $\boldsymbol{\eta_j^t}$ is the set of cells in the range of UAV $j$ at time $t$.
   - $\boldsymbol{\varphi_j^t}$ is the remaining power of UAV $j$ at time $t$.
   - $\boldsymbol{\sigma_j^t}$ is the remaining time for completing the data collection of IoT node $i$ at time $t$
   - $\boldsymbol{\chi_j^t}$ is the remaining data to be collected of IoT node $i$ at time $t$

   Note that the value $\eta_j^t$ is a variable subject to beam width parameters.

2) **Action space**: An agent may take one of the five moving actions at each state, denoted as $A = \{+x, +y, -x, -y, 0\}$ to represent the action, where $-y$, $+y$, $-x$, or $+x$ indicates that UAV $j$ makes a change of its states by moving downwards, upwards, right, or left, respectively. Contrarily, 0 represents that UAV $j$ is hovering for data collection.

3) **Reward function**: The purpose of the proposed DRL-based CMA-TD problem is to maximize the expected reward by UAV $j$ on completing a single mission from its initial to the final position. The trajectory reward is defined as

$$r_j^t = \begin{cases} +z, & \text{if } Q_j^t = P_f, \\ -1, & \text{otherwise,} \end{cases} \tag{17}$$

where a positive reward $z$ is received when UAV $j$ reaches the final destination, otherwise negative 1 penalty is received for taking a step and not completing the mission. Further, with (15), (16), and (17), we design a reward function by using parameters $\zeta$ and $\xi$, which encourages the UAVs to maximize the reward by serving the IoT nodes with minimal energy consumption. The combined reward of UAV $j$ at time $t$ is defined as

$$R_j^t = \zeta \frac{Y_{i,j}}{E_j^t} + K_{j,i}^t + \xi r_j^t, \tag{18}$$

where $Y_{i,j}$ and $E_j^t$ are defined according to (15) and (16), respectively. The total reward after the mission completion of UAV $j$ can be formulated as below:

$$RE_j = \zeta \frac{Y}{E} + K + \xi r, \tag{19}$$

where $Y = \sum_{j=1}^{U}\sum_{i=1}^{I} Y_{i,j}$ and $E = \sum_{t=0}^{T}\sum_{j=1}^{U} E_j^t$, $K = \sum_{t=0}^{T} K_{j,i}^t$ and $r = \sum_{t=1}^{T} r_j^t$.
The total reward of the episode including all UAVs can be calculated as:

$$R_{total} = \sum_{j=1}^{U} RE_j. \tag{20}$$

## IV. Proposed CMA-TD Algorithm

To solve the DRL-based formulation, we introduce a novel CMA-TD algorithm given in Algorithm 1. The input of CMA-TD are the state of UAVs, replay memory $M_r$, as well as other parameters including learning rate $\alpha$, discount factor $\gamma$, and epsilon probability $\epsilon$. The output is the optimal policy $\pi^*$. In the training phase, we first initialize the evaluation and target network, and other parameters (lines 3–4). In line 5, an action space is generated according to Section (III-B). In each training episode, a UAV flies around the AoI to serve the IoT nodes and reach the resting place. Particularly, the environment is reset at the beginning of each episode (line 8). At each time step $t$, each UAV makes its own observation of the environment and takes action randomly with probability $\epsilon$ using the $\epsilon$-greedy policy, and otherwise selects an action with the maximum Q-value (lines 10–12) that can be obtained by (3).

After the execution of the selected action, the UAV receives a reward $R_j^t$ from the environment according to (18), observes the new state, and adjusts the beamwidth according to Section II-A (lines 16–18). In line 19, the transition tuples, i.e., $(s_t, a_t, r_t, s_{t+1})$, are stored in a shared replay memory $M_r$. To train the evaluation network $\theta$, a mini-batch of tuples $B_m$ can

**Algorithm 1:** Collaborative multi-agent for trajectory planning and data collection

1 **Input:** replay memory $M_r$, epsilon probability $\epsilon \in$ [0,1], states, learning rate $\alpha \in$ [0,1], discount factor $\gamma \in$ [0,1]
2 **Output:** The optimal policy $\pi^*$
3 Initialize current network parameter $\theta$ and the target network $\tilde{\theta}$;
4 Initialize current network $Q(s_t, a_t, \theta)$ with weights $\theta$ and the target network $Q(s_t, a_t, \tilde{\theta})$ with weights $\tilde{\theta}$;
5 $\mathcal{A} \leftarrow$ sampleActionSpace();
6 **while** ⌉ $\leq$ *Total_Episodes* **do**
7 $t \leftarrow 0$;
8 $\delta \leftarrow$ resetEnvironment();
9 **while** *($Q_t \neq Pf$)* **do**
10 $s_t \leftarrow$ observeState($\delta$);
11 $c \leftarrow$ randomSample([0,1]);
12 Select action:
$$\begin{cases} a_t \leftarrow randomAction(\mathcal{A}), & \text{if } c \leq \epsilon \\ a_t \leftarrow argmaxQ(s_t, a_t, \theta), & \text{Otherwise.} \end{cases}$$
13 **if** *UAV $j$ collects data from IoT node $i$* **then**
14 mark $i$ as collected
15 **end if**
16 $R_j^t \leftarrow$ obtainReward($a_t$);
17 $s_j^{t+1} \leftarrow$ observeNewState($a_t$);
18 Observe $s_t$ and adjust beamwidth;
19 Store the transition tuple $(s_t, a_t, r_t, s_{t+1})$ in common $M_r$;
20 Sample mini batch of $B_m$ tuples;
21 **if** ($s_{t+1}$ = $Pf$) **then** ;
22 Calculate target;
23 $Y_t = R_{t+1}$;
24 else;
25 $$Y_t^{DD} = R_{t+1} + \gamma Q_{\tilde{\theta}}(s_{t+1}, \underset{a_t+1}{\operatorname{argmax}} Q_\theta(s_{t+1}, a_{t+1})),$$
26 Perform the gradient decent step;
27 Calculate the loss $L^{DD}(\theta) = \mathbb{E}[(Q_\theta(s_t, a_t)) - Y_t^{DD})^2]$;
28 Soft update of target parameters,
29 $\tilde{\theta} = (1 - \tilde{x})\tilde{\theta} + \tilde{x}\theta$ (update factor $\tilde{x}$ = [0,1]);
30 ⌉ = ⌉ + 1
31 **end while**
32 **end while**

be randomly sampled from replay memory $M_r$ (line 20). $\theta$ is updated by stochastic gradient descent (back-propagation) on the sampled mini-batch, and the loss is calculated before updating (lines 21–29). The main improvement made by DDQN [31] is made in the sense that the action values may get overestimated due to approximating the value of the expected maximum value-action of the next state (see Section II-B), where the overestimation of action values can be decreased by choosing the best action using $\theta$ but estimating the value of that action using $\tilde{\theta}$. When calculating $L^{DD}(\theta)$ the target value is taken; thus, the back-propagating gradient is stopped before $Y_t^{DD}$. Finally, the episode ends when the UAVs arrive at their destination. Line 7 to line 29 is repeated for $Total\_Episodes$. After training, a policy is obtained with a well-trained DNN that UAV can navigate in a real-time environment.

The computational complexity of the proposed technique can be analyzed by considering various factors and components involved. Let $m \times n$ be denoted as the grid size. We assume a neural network with $P$ parameters, $T_{ep}$ training episodes, and a replay memory size of $M_r$.

During the training phase, the input complexity can be expressed as $O(m \times n)$, representing the number of grid elements. The complexity of the network architecture is $O(P)$, reflecting the number of learnable parameters. The training iterations contribute a complexity of $O(T_{ep})$. Therefore, the overall complexity of the training phase is denoted as $O((m \times n) * P * T_{ep})$. In the Q-learning update step, the action space complexity is $O(1)$, assuming a constant number of actions denoted as $A$. The Q-value update has a constant complexity of $O(1)$. The replay memory complexity is $O(M_r)$, representing the size of the memory. Thus, the overall complexity of the Q-learning update step can be expressed as $O(A * M_r)$.

Considering the above, the total complexity of the approach can be summarized as $O((m \times n) * P * T_{ep}) + O(A * M_r)$. This highlights the significant computational requirements of the training process, as it depends on the grid size, number of parameters in the neural network, training iterations, action space, and replay memory size.

## V. Performance Evaluation

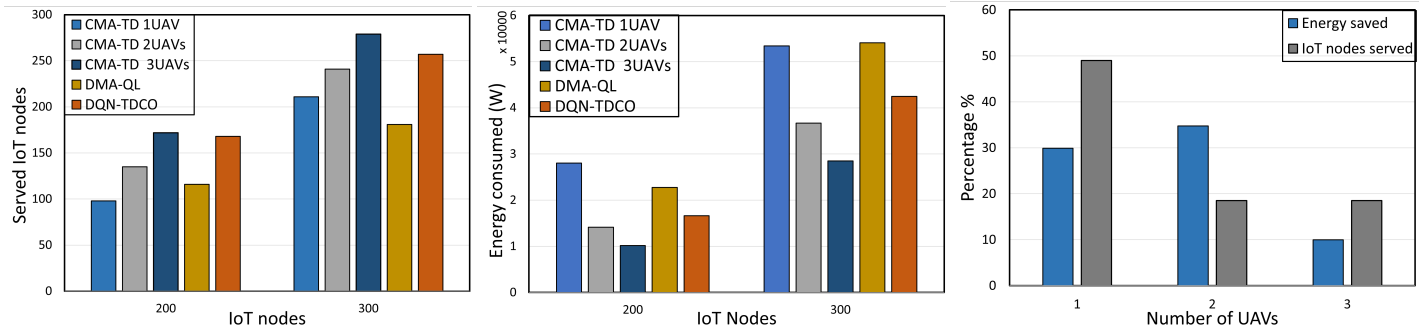
Extensive simulation is conducted to evaluate the performance of the proposed DRL-based CMA-TD algorithm. This section discusses the simulation settings and presents the results and analysis.

### A. Simulation Setup

We evaluate the performance of the proposed DRL-based CMA-TD algorithm on the number of required collaborative UAVs under various several scenarios. The DRL model is trained using the parameters listed in Table III.

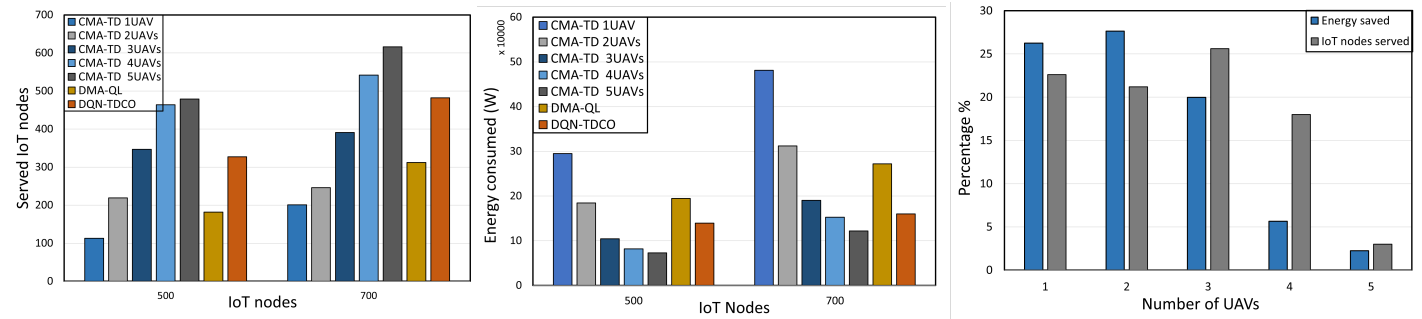
We consider a square area of 40 km, which is divided into three different grid sizes. The details of the three scenarios are presented below:

1) ***Scenario 1:*** The square area grid size is 500 × 500 cells with 80 m size per cell and IoT nodes i.e., $I = 200$ and $I = 300$, are randomly distributed in which 5% of IoT nodes are moving. Whereas, 1 to 3 UAVs are deployed to analyze the optimal number of UAVs.
2) ***Scenario 2:*** The square area grid size is 1000 × 1000 cells with 40 m size per cell and IoT nodes i.e., $I = 500$ and $I = 700$ are randomly distributed in which 8% of IoT nodes are moving. Moreover, 1 to 5 UAVs are deployed to analyze the optimal number of UAVs.



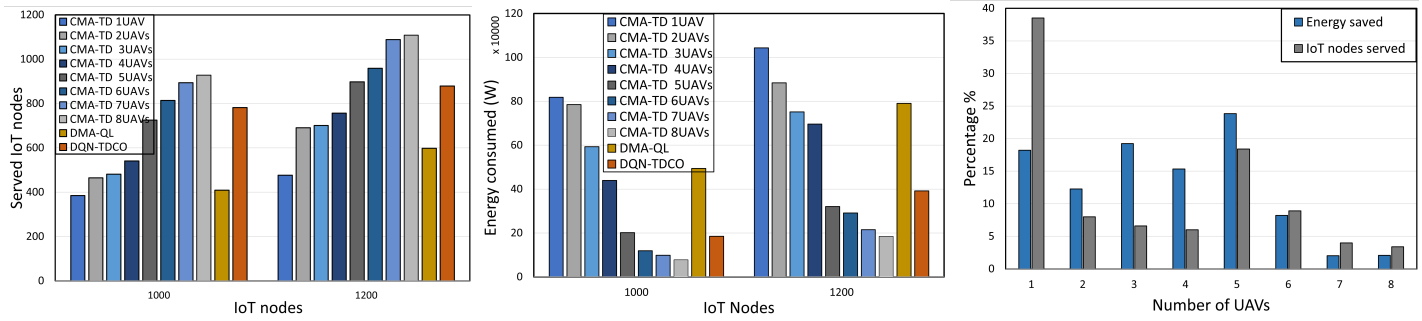

(a) Successfully served IoT nodes. (b) Energy consumed by UAVs. (c) Overall utilization rate of the system.

Fig. 4. Comparison of the proposed DRL-based CMA-TD algorithm with existing methods in scenario 1 a) a total number of IoT nodes served successfully, b) energy consumed while serving and flying with a varying number of ground nodes, and c) overall utilization of the proposed DRL-based CMA-TD algorithm with 300 IoT nodes.




(a) Successfully served IoT nodes (b) Energy consumed by UAVs (c) Overall utilization rate of the system

Fig. 5. Comparison of the proposed DRL-based CMA-TD algorithm with existing methods in scenario 2 a) a total number of IoT nodes served successfully, b) energy consumed while serving and flying with a varying number of ground nodes, and c) overall utilization of the proposed DRL-based CMA-TD algorithm with 500 IoT nodes.




(a) Successfully served IoT nodes (b) Energy consumed by UAVs (c) Overall utilization rate of the system

Fig. 6. Comparison of the proposed DRL-based CMA-TD algorithm with existing methods in scenario 3 a) a total number of IoT nodes served successfully, b) energy consumed while serving and flying with different numbers of ground nodes, and c) Overall utilization of the proposed DRL-based CMA-TD algorithm with 1200 IoT nodes.

3) ***Scenario 3:*** The square area grid size is 2000 $\times$ 2000 cells with 20 m size per cell and IoT nodes i.e., $I = 1000$ and $I = 1200$, are randomly distributed in which 10% of IoT nodes are moving. Additionally, 1 to 8 UAVs are deployed to analyze the optimal number of UAVs.

A four-layer neural network architecture with 256, 300, 218, and 118 neurons in each layer, respectively, is employed to constitute our DRL-based model. The experimental parameters of DRL were determined through a trial-and-error approach, where different parameter values were tested and evaluated iteratively to find the optimal settings. The mini-batch size, the discount factor $\gamma$, and the learning rate $\alpha$ are taken as 128, 0.85, and 0.01, respectively. Initially, $\epsilon$ is set to 0.9 and is decayed by a factor of 0.855 until it reaches 0.05. For the comparison purpose, we consider two related works as the best representation of the state-of-the-art, namely DMA-QL [28] and DQN-TDCO [8]. The former is based on a multi-agent distributed Q-learning algorithm to optimize energy efficiency and outage of ground users, while the later employs a plain DRL process for trajectory design and data collection optimization in a UAV-based IoT network.

Table III
SIMULATION PARAMETERS.

| Parameters | Values |
|---|---|
| v | 25 $m/s$ |
| $B$ | 1 MHz [33] |
| $H$ | 100 m [29] |
| $v_0$ | 7.2 [33] |
| $\xi_0$ | 79.9 W |
| $\xi_1$ | 88.6 W |
| $\mu_{tip}$ | 200 $m/s$ [33] |
| $z_0$ | 0.3 $m^2$ [33] |
| $\tau$ | 0.05 [33] |
| $\kappa$, | 1.225 $kg/m^3$ [33] |
| $A$ | 0.79 $m^2$ [33] |
| $\zeta$ | 0.6 |
| $\xi$ | 0.4 |
| $\vartheta$ | 80°– 140° |
| $\gamma$ | 0.8 |
| $\alpha$ | 0.01 |
| $\epsilon$ | 0.9 |
| Optimizer | Adam |
| Mini-batch size | 128 |

### B. Results and Analyses

We analyze the performance of the proposed CMA-TD algorithm for trajectory design in terms of the number of served IoT nodes. We consider the effect of the number of collaborative UAVs and the total energy consumption.

The simulation results for scenario 1 with different numbers of IoT nodes on the ground and different numbers of UAVs in the air are given in Fig. 4, while DMA-QL and DQN-TDCO employ 3 UAVs, respectively. It is clear that the proposed DRL-based CMA-TD algorithm consistently outperforms DMA-QL and DQN-TDCO in terms of the number of served IoT nodes. As shown in Fig. 4(a), the number of served IoT nodes increases with the increment of the number of UAVs. Compared to DMA-QL and DQN-TDCO, the proposed DRL-based CMA-TD algorithm is improved by at least 35% in the coverage of IoT nodes, while consuming much less total energy when the number of UAVs increases, as shown in Fig. 4(b). We can see that when the total number of UAVs is three (3), the proposed DRL-based CMA-TD algorithm can save at least 30% total energy compared to existing works. In Fig. 4(c), we show the benefits of using multiple UAVs by examining how much energy is saved and how many more IoT nodes can be served by adding a UAV.

The simulation results for scenario 2 with different numbers of IoT nodes on the ground and different numbers of UAVs in the air are given in Fig. 5 where DMA-QL and DQN-TDCO employed 4 UAVs. It is evident that the proposed DRL-based CMA-TD algorithm can achieve better performance than that of the counterparts DMA-QL and DQN-TDCO in terms of serving maximum IoT nodes. As illustrated by Fig. 5(a), the number of served IoT nodes increases with the increment of the number of UAVs. Compared to DMA-QL and DQN-TDCO, the proposed method can serve at least twice the total number of IoT nodes. When the number of

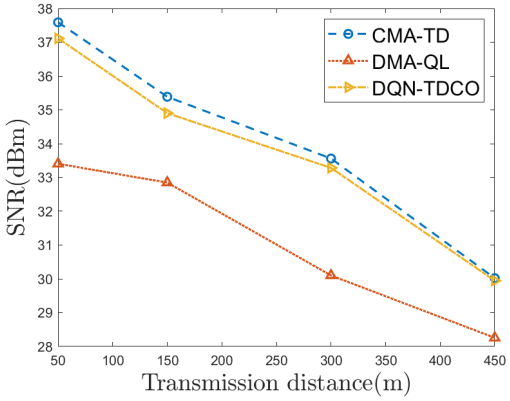



Fig. 7. The effect of transmission distance to SNR.

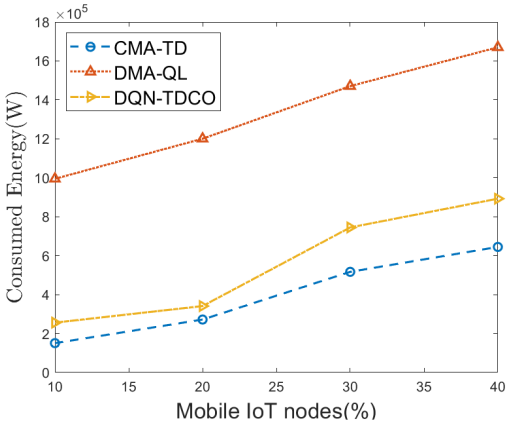
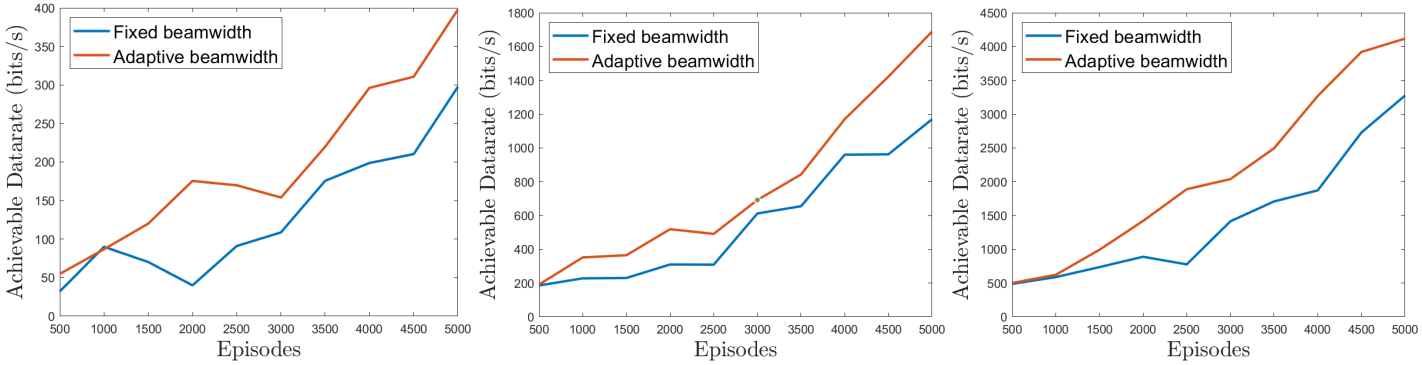



Fig. 8. The effect of IoT nodes mobility to consumed energy.

UAVs is increased to five, the proposed DRL-based CMA-TD algorithm consumes only 40% of the total energy compared to the DMA-QL method and only 70% of the total energy compared to the DQN-TDCO method when there are 700 IoT nodes. When the IoT nodes are 500, the proposed DRL-based CMA-TD algorithm only consumes at most 40% of the total energy compared to existing DRL-based methods, as shown in Fig. 5(b). In Fig. 5(c), it is observed that both the saved energy and the increment of served IoT nodes can reach at least 20% with the increment of the number of UAVs when the total number of UAVs is less than 4.

The simulation results for scenario 3 with various numbers of ground IoT nodes and UAVs in the air are given in Fig. 6, while 6 UAVs are employed in the cases of DMA-QL and DQN-TDCO. The proposed DRL-based CMA-TD algorithm is still able to cover significantly more IoT nodes compared to that of DMA-QL and DQN-TDCO, respectively. The number of served IoT nodes increases with the increment of the number of UAVs, as indicated by Fig. 6(a). Notably, the number of IoT nodes served by the proposed CMA-TD algorithm is at least 100% and 20% more compared to that by DMA-QL and DQN-TDCO, respectively. When the number of UAVs is increased up to seven and the number of IoT nodes is 1000, the proposed CMA-TD algorithm consumes only 15% and 40% of the total energy compared to that by DMA-QL and DQN-TDCO, respectively. When the number of IoT nodes is further increased up to 1200, CMA-TD consumes only 20%




(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

Fig. 9. The effect of adaptive beamwidth to the achieved data rate.

Table IV
TRAVELING COST (M).

| | Scenario 1 | Scenario 2 | Scenario 3 |
|---|---|---|---|
| **DMA-QL [28]** | 327494 | 649773 | 1043421 |
| **DQN-TDCO [8]** | 214372 | 303998 | 850066 |
| **DRL-based CMA-TD** | 38122 | 78950 | 98806 |

Table V
COMPARATIVE ANALYSIS OF ENERGY CONSUMPTION (W): CMA-TD VERSUS ILP SCHEME.

| | 10×10 Cells | 15×15 Cells |
|---|---|---|
| ILP model | 2698.22 | 3045.47 |
| DRL-based CMA-TD | 2831.75 | 3336.15 |

and 50% of the total energy compared to that by DMA-QL and DQN-TDCO, respectively, as shown in Fig. 6(b). In Fig. 6(c), we can observe that both the saved energy and the increment of served IoT nodes can be improved with the increment of the number of UAVs; and there is no significant performance enhancement by further increasing the number of UAVs when 6 collaborative UAVs are already in place.

Fig. 7 shows the SNR with respect to the transmission distance, defined as the distance between the IoT node on the ground and the UAVs in the air. From Fig. 7, we observe that our method is more robust than the considered counterparts due to the higher SNR possessed by the proposed CMA-TD algorithm.

In Fig. 8, we analyze the impact of the proportion of the number of mobile IoT nodes to the consumed energy. We can see that CMA-TD significantly outperforms the other DRL-based counterparts. It is worth noting that the mobility increment introduces an increment of total energy consumption, which is attested by the doubled energy consumption in the presence of increased mobility of the IoT nodes from 10% to 30%.

Fig. 9 shows the achievable data rate by the fixed beamwidth and adaptive beamwidth schemes at the UAVs where the proposed CMA-TD algorithm is deployed. The antenna angle varies from degree 80 to 140, which will affect the data rate when serving the IoT nodes. Figs. 9(a), 9(b), and 9(c) show the achievable rates for scenario 1 (200 IoT nodes and 3 UAVs), scenario 2 (500 IoT nodes and 4 UAVs) and scenario 3 (1000 IoT nodes and 6 UAVs), respectively. It is observed that the adaptive beamwidth scheme can provide significantly higher data rates in all three scenarios than the other. The main reason is that the adaptive beamwidth scheme can change its beamwidth angle during the mission. Such adaptation to the IoT environment is at the expense of higher computation and hardware complexity.

Table IV compares the traveling cost of each scheme, which is defined as the total distance traveled by all UAVs. This table shows that launching a larger number of UAVs helps to reduce the traveling cost, where the proposed CMA-TD algorithm has taken significantly less cost than that by DMA-QL [28] and DQN-TDCO [8], thanks to the use of DDQN that facilitates much faster learning and thus better performance. We have also seen that all the schemes take higher traveling costs in scenario 3 than that in the others because the number of cells is the largest among all scenarios, i.e., larger cell sizes and more steps to finish the assigned mission.

In this study, we compared the energy efficiency of our proposed CMA-TD scheme with that of the ILP model. Table V presents the results obtained from two scenarios in an area of 200 m. In the first scenario, the area is divided into 10×10 cells with 50 IoT nodes, while the second scenario considers 15×15 cells with 100 IoT nodes. In the 10×10 cell scenario, the ILP resulted in an energy consumption of 2698.22 W, whereas the CMA-TD scheme consumed 2831.75 W. Similarly, in the 15×15 cell scenario, the ILP model showed an energy consumption of 3045.47 W, while the CMA-TD scheme consumed 3336.15 W. The results confirm that the proposed CMA-TD has the ability of achieving close performance to an optimal one obtained by the ILP model.

## VI. CONCLUSIONS

In this paper, we introduced a novel collaborative multi-agent trajectory planning and data collection (CMA-TD) algorithm to solve the data collection problem for multiple UAVs. Notably, the proposed CMA-TD algorithm leverages

the double deep Q-learning (DDQN) architecture, aiming at an effective real-time learning and decision-making process without any prior knowledge of the network environment, where energy efficiency and coverage of the multiple UAVs can be maximized via collaboratively sharing of the respectively observed information. The extensive simulation results verified the significant benefits of applying the proposed DRL-based CMA-TD algorithm against a couple of state-of-the-art representative counterparts in all the considered scenarios regarding SNR, achieved data rate, and traveling cost. In future research, we intend to investigate the deployment of a centralized system, such as aerial computing, to facilitate seamless communication and efficient coordination among UAVs in the 3D environment.

## REFERENCES


[1] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Drone small cells in the clouds: Design, deployment and performance analysis," in *Proc. IEEE GLOBECOM*, 2015.

[2] O. S. Oubbati *et al.*, "Multi-UAV-enabled AoI-aware WPCN: A multi-agent reinforcement learning strategy," in *Proc. IEEE INFOCOM*, 2021.

[3] H. Huang and A. V. Savkin, "Towards the Internet of flying robots: A survey," *Sensors*, vol. 18, no. 11, p. 4038, 2018.

[4] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and trade-offs," *IEEE Trans. Wireless Commun.*, vol. 15, no. 6, pp. 3949–3963, 2016.

[5] D. Popescu, C. Dragana, F. Stoican, L. Ichim, and G. Stamatescu, "A collaborative UAV-WSN network for monitoring large areas," *Sensors*, vol. 18, no. 12, p. 4202, 2018.

[6] A. Islam, A. Al Amin, and S. Y. Shin, "FBI: A federated learning-based blockchain-embedded data accumulation scheme using drones for Internet of things," *IEEE Wireless Commun. Lett.*, vol. 11, no. 5, pp. 972–976, 2022.

[7] C. Zhan and Y. Zeng, "Aerial-ground cost trade-off for multi-UAV-enabled data collection in wireless sensor networks," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1937–1950, 2019.

[8] K. K. Nguyen, T. Q. Duong, T. Do-Duy, H. Claussen, and L. Hanzo, "3D UAV trajectory and data collection optimisation via deep reinforcement learning," *IEEE Trans. Commun.*, vol. 70, no. 4, pp. 2358–2371, 2022.

[9] S. Y. Shin *et al.*, "Energy-efficient multidimensional trajectory of UAV-aided iot networks with reinforcement learning," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 19 214–19 226, 2022.

[10] B. Li, Z. Gan, D. Chen, and D. Sergey Aleksandrovich, "UAV maneuvering target tracking in uncertain environments based on deep reinforcement learning and meta-learning," *Remote Sensing*, vol. 12, no. 22, p. 3789, 2020.

[11] S. Yin and F. R. Yu, "Resource allocation and trajectory design in UAV-aided cellular networks based on multiagent reinforcement learning," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2933–2943, 2021.

[12] 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, 2020.

[13] M. Samir, S. Sharafeddine, C. M. Assi, T. M. Nguyen, and A. Ghrayeb, "UAV trajectory planning for data collection from time-constrained IoT devices," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 34–46, 2019.

[14] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, "Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach," *IEEE J. Sel. Areas Commun*, vol. 36, no. 9, pp. 2059–2070, 2018.

[15] M. Li *et al.*, "Energy-efficient UAV-assisted mobile edge computing: Resource allocation and trajectory optimization," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3424–3438, 2020.

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

[17] O. Ghdiri, W. Jaafar, S. Alfattani, J. B. Abderrazak, and H. Yanikomeroglu, "Energy-efficient multi-UAV data collection for IoT networks with time deadlines," in *Proc. IEEE GLOBECOM*, 2020.

[18] 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, 2021.

[19] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, 2019.

[20] O. Esrafilian, H. Bayerlein, and D. Gesbert, "Model-aided deep reinforcement learning for sample-efficient UAV trajectory design in IoT networks," *preprint arXiv:2104.10403*, 2021.

[21] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in uav communication networks," *IEEE Commun, Surveys Tut.*, vol. 18, no. 2, pp. 1123–1152, 2015.

[22] Y. Wang *et al.*, "Trajectory design for UAV-based Internet of things data collection: A deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3899–3912, 2021.

[23] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile unmanned aerial vehicles for energy-efficient Internet of things communications," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7574–7589, 2017.

[24] Y. Li, A. H. Aghvami, and D. Dong, "Path planning for cellular-connected UAV: A DRL solution with quantum-inspired experience replay," *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 7897–7912, 2022.

[25] S. Yin, S. Zhao, Y. Zhao, and F. R. Yu, "Intelligent trajectory design in UAV-aided communications with reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8227–8231, 2019.

[26] S. Rahim, M. M. Razaq, S. Y. Chang, and L. Peng, "A reinforcement learning-based path planning for collaborative UAVs," in *Proc. ACM/SIGAPP SAC*, 2022.

[27] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[28] S. Lee, H. Yu, and H. Lee, "Multi-agent Q-learning-based multi-UAV wireless networks for maximizing energy efficiency: Deployment and power control strategy design," *IEEE Internet Things J.*, vol. 9, no. 9, pp. 6434–6442, 2021.

[29] H. He, S. Zhang, Y. Zeng, and R. Zhang, "Joint altitude and beamwidth optimization for UAV-enabled multi-user communications," *IEEE Commun. Lett.*, vol. 22, no. 2, pp. 344–347, 2017.

[30] X. Zhu *et al.*, "Path planning of multi-UAVs based on deep Q-network for energy-efficient data collection in UAVs-assisted IoT," *Veh. Commun.*, p. 100491, 2022.

[31] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. AAAI*, 2016.

[32] C. You and R. Zhang, "3D trajectory optimization in Rician fading for UAV-enabled data harvesting," *IEEE Trans. Wireless Commun.*, vol. 18, no. 6, pp. 3192–3207, 2019.

[33] S. S. Khodaparast, X. Lu, P. Wang, and U. T. Nguyen, "Deep reinforcement learning based energy efficient multi-UAV data collection for IoT networks," *IEEE Open J. Veh. Technol.*, vol. 2, pp. 249–260, 2021.


**Shahnila Rahim** received her M.S. in Computer Science degree from the National University of Computer and Emerging Sciences (NUCES), Islamabad, Pakistan. She is currently pursuing a Ph.D. degree at the School of Computer Science and Engineering, Kyungpook National University (KNU), Daegu, Republic of Korea. Her research interests include machine learning, deep learning, B5G communication network, and the Internet of things (IoT).

**Limei Peng** is currently an Associate Professor at the School of Computer Science and Engineering, Kyungpook National University (KNU), Daegu, Republic of Korea. Her research interests include cloud computing, fog computing, data center networks, Internet of things (IoT) / Internet of vehicles (IoV), and 5G communications networks.

**Shihyu Chang** received a BSEE degree from National Taiwan University, Taiwan, in 1998, and Ph. D. degree in Electrical Engineering and Computer Engineering from the University of Michigan, Ann Arbor, in 2006. From August 2006 to February 2016, he was the Faculty in the Department of Computer Engineering, National Tsing Hua University, Hsinchu, Taiwan. From July to August 2007, Dr. Chang had been a visiting Assistant Professor at Television and Networks Transmission Group, Communications Research Centre, Ottawa, Canada. From June 2018, he began to provide lectures about machine learning, data science, and AI at San Jose State University, San Jose, CA, USA. Besides academic positions, Dr. Chang also provides consulting work as an AI technical lead focusing on applying machine learning techniques to automate office work. Dr. Chang has published more than 90 peer-refereed technical journals and conference articles in electrical and computer engineering. His research interests include the areas of wireless networks, wireless communications, and signal processing. He currently serves as the Technical Committee, Symposium Chair, Track Chair, or Reviewer in networking, signal processing, communications, and computers.

**Pin-Han Ho** is currently a Full Professor in the Department of Electrical and Computer Engineering, University of Waterloo. He is the author/co-author of over 400 refereed technical papers, several book chapters, and the co-author of two books on Internet and optical network survivability. His current research interests cover a wide range of topics in broadband wired and wireless communication networks, including wireless transmission techniques, mobile system design and optimization, and network dimensioning and resource allocation. He is in the rank of IEEE Fellow and a Professional Engineer Ontario (PEO).