MAPPO-Based Multi-UAV Trajectory Optimization for AoI Minimization with Obstacle and Energy Awareness

Shilpi Kumari, Dev Gupta, Gaurav Rawat, Ajay Pratap
Department of CSE, Indian Institute of Technology (BHU), Varanasi
Email: {shilpikumari.rs.cse21, dev.gupta.cse22, gaurav.rawat.cse22, ajay.cse}@iitbhu.ac.in

***Abstract*—Future wireless networks stand to gain significantly from integrating Unmanned Aerial Vehicles (UAVs) for on-demand data collection, particularly in defense scenarios. While existing UAV systems emphasize energy conservation, they often overlook data freshness, a critical factor in military missions across difficult terrains. This work presents a novel solution that jointly considers Age of Information (AoI) and energy efficiency in Multi-UAV assisted IoT networks tailored for such environments. By employing clustering and Deep Reinforcement Learning (DRL) for UAV position optimization, we effectively reduce both average AoI and energy usage. Our Multi-Agent Proximal Policy Optimization UAV Trajectory Planning (MAPPO-UTP) framework leverages Deep Neural Networks (DNNs) for informed and adaptive trajectory decisions.**

***Index Terms*—UAV, Multi-Agent, IoT, AoI, Defense, Clustering, DRL.**

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are transforming the landscape of wireless communication, particularly in defense and mission-critical applications where timely data acquisition is essential. In such settings, Internet of Things Devices (IoTDs) are deployed to monitor and collect critical environmental and operational data. However, the vast deployment of these devices and their location in hostile or inaccessible terrains present challenges for timely data retrieval and task execution. Ensuring that the collected information remains fresh and actionable is vital for mission success. To this end, data freshness—quantified through the Age of Information (AoI)—becomes a crucial performance metric. In rugged military terrains, the distance between IoTDs and central Base Stations (BS) often results in significant communication delays, potentially compromising situational awareness and response effectiveness.Delayed or outdated data can severely hinder operational efficiency and responsiveness, especially in high-stakes defense missions requiring continuous situational awareness. Traditional ground-based networks often fall short in providing the needed coverage and adaptability in such environments.

To overcome these limitations, we propose a model that leverages UAVs not only as mobile relays but also as intelligent agents capable of adaptive trajectory planning and task offloading.UAVs have gained prominence as agile, mobile platforms capable of bridging the communication gap between IoTDs and central processing stations. Their ability to establish reliable Line-of-Sight (LoS) communication links and access otherwise unreachable areas offers a practical solution for data collection in harsh conditions. However, effective utilization of UAVs in such roles necessitates intelligent coordination, energy-aware planning, and data freshness optimization. Motivated by these needs, this work focuses on enhancing the performance and reliability of UAV-assisted IoT networks by exploring advanced task offloading and path planning strategies.

II. RELATED WORKS

Numerous works have aimed to enhance data collection efficiency and minimize the Age of Information (AoI) [1], [2]. Conventional techniques such as convex optimization, genetic algorithms, and Dynamic Programming (DP) have been extensively employed to find optimal solutions within communication networks [1], [2]. Nevertheless, these methods typically depend on accurate modeling and complete global knowledge, which can be challenging to obtain in practical military deployments, especially in rugged and mountainous terrains.

In addition to multi-UAV coordination strategies, several works have focused on task offloading and path planning using a single UAV. These models typically assume a single aerial agent responsible for collecting data from all IoTDs and offloading it to the base station [3]. While these systems simplify control and deployment, they often suffer from scalability issues and increased latency, particularly in large-scale or geographically complex environments. Despite achieving reasonable performance in smaller or less dynamic scenarios, single-UAV systems face limitations in meeting strict AoI and energy efficiency requirements in mission-critical settings, such as defense operations in hilly terrain. As a result, there is a growing interest in exploring multi-UAV frameworks to enhance robustness, reduce latency, and better adapt to dynamic environments.A comparative analysis of closely related works in the literature, as shown in Table I.

This paper focuses on the challenge of utility maximization for data collection in clustered IoT networks deployed across hilly terrains. Given the limited energy capacities of UAVs, it becomes crucial to design their flight paths in a way that maximizes overall utility while effectively navigating around natural obstacles. The complexity of this scenario makes exact optimization intractable; hence, we propose a sub-optimal yet efficient solution based on Deep Reinforcement Learning (DRL). Unlike traditional single-agent approaches, our model leverages a MAPPO framework that enables multiple UAVs to collaboratively coordinate their routes and tasks. We summarize the core contributions of this paper as follows:

- A novel formulation of the UAV utility maximization problem in challenging terrain environments.
- A multi-agent DRL-based framework for scalable and adaptive path and task allocation.
- Comprehensive simulation results validating the superior performance and practicality of our proposed method.

TABLE I: Summary of related works

Works	AoI	IoTD Clustering	Energy	Collision detection	Multi-Agent
[4] [5]	✓	×	✓	×	×
[6]	✓	×	✓	×	✓
[7]	×	✓	✓	×	✓
[8]	×	✓	✓	×	×
[9]	×	×	×	×	✓
[10]	×	×	×	×	×
[11]	✓	✓	✓	×	×
[3]	✓	✓	✓	✓	×
Ours	✓	✓	✓	✓	✓

The remainder of the paper is structured as follows: Section III presents the system model while Section IV presents the problem formulation and provides a proof of its NP-Hardness. Section V proposes a solution. Section VI provides a performance evaluation to validate the efficacy of the proposed system. Finally, Section VII presents conclusions and future research directions.

III. System Model

We examine a UAV-assisted IoT network as illustrated in (Fig 1), in which IoTDs need to upload their data to BS, for surveillance and security purposes in the hilly border area. Because IoTDs are energy-constrained and lack dedicated bandwidth, they cannot directly send data to the BS. A UAV u is thus dispatched to act as a mobile intermediary, collecting and uploading their data to the BS..Let $\mathcal{N} = \{1, \dots, n, \dots, N\}$ be the IoTDs set, with $I_n(t)$ denoting the data size of IoTD $n \in \mathcal{N}$ at time t. To facilitate the data collection of UAV u, all IoTDs are divided into M clusters, where $M < N$. Let $\mathcal{M} = \{1, \dots, m, \dots, M\}$ be the collection of clusters. Each cluster $m \in \mathcal{M}$ consists of one Cluster Head (CH) denoted as c_m, which collects data from each IoTD in its cluster and transfers it to UAV u.

We introduce a binary-variable $X_{n,m}(t)$ to check the association of IoTD n with cluster m at time t as:

$$X_{n,m}(t) = \begin{cases} 1, & \text{if IoTD } n \text{ belongs to cluster } m, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

We have assumed the IoTDs to be stationary in this paper so $X_{n,m}(t)$ will not change with time(is not a function of time)

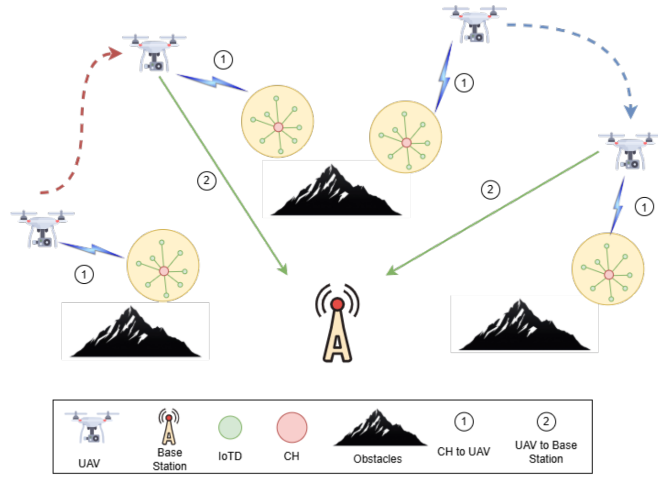

Fig. 1: System architecture.

but in future it can be used to accomodate dynamic clustering if the IoTDs are not stationary.

Let $\ell_u(t) = [x_u(t), y_u(t), h_u(t)]$, $\ell_n(t) = [x_n(t), y_n(t), 0]$, and $\ell_m(t) = [x_m(t), y_m(t), 0]$ represent the 3-D coordinates of UAV u, IoTD $n \in \mathcal{N}$, and CH $c_m \in \mathcal{M}$, respectively.In our work we have assumed $h_u(t)$ to be constant, i.e., we have assumed that the UAV flies at a constant height. The UAV hovers in the air during data collection to optimize data transmission. We introduce a variable $Y_{u,m}(t)$ to indicate whether UAV u is within the communication range of CH c_m at time t as follows:

$$Y_{u,m}(t) = \begin{cases} 1, & \text{provided UAV } u \text{ is within coverage area of CH } c_m, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

A. IoTDs to CH transmission model

When UAV u gathers data directly from each IoTD, it must fly over individual IoTD locations, operating at peak power to maximize coverage of IoTDs within a defense monitoring zone. Simultaneously, IoTDs constantly search for the UAV to transmit data, causing high energy usage for both UAV and IoTDs. To reduce this, IoTDs are organized into clusters, each managed by a CH. Hence, the data collection latency for CH c_m from IoTDs in its assigned cluster is given by:

$$T_m^{up}(t) = \max_{n \in \mathcal{N}} \left\{ X_{m,n}(t) \frac{I_n(t)}{R_{m,n}(t)} \right\} \tag{3}$$

where $R_{m,n}(t)$ is the data rate between IoTD n and CH c_m. Similarly, the energy consumption between IoTD n and CH c_m can be defined as:

$$E_{m,n}^{up}(t) = (P_n + P_m^r) \left(X_{m,n}(t) \frac{I_n(t)}{R_{m,n}(t)} \right) \tag{4}$$

where P_n is the transmitting power of IoTD n, and P_m^r is the receiving power of CH c_m.

B. CH to UAV transmission model

Each cluster head (CH) c_m continuously listens for the presence of UAV u. Once the UAV is detected within range, the CH initiates data transmission. The latency associated with this transmission from CH c_m to UAV u at time frame t is given by:

$$T_{m,u}^{up}(t) = \frac{\sum_{n=1}^{N} I_n X_{n,m}(t) Y_{u,m}(t)}{R_{m,u}(t)} \tag{5}$$

where $R_{m,u}(t)$ denotes the data rate between CH c_m and UAV u. Correspondingly, the energy consumed by CH c_m during the transmission process can be expressed as:

$$E_{m,u}^{up}(t) = P_m T_{m,u}^{up}(t) \tag{6}$$

where P_m represents the transmission power of CH c_m.

C. UAV traversal and transmission model

The UAV flies over each CH to gather data. We assume the presence of O fixed obstacles such as hills and trees within the area. Let $\mathcal{O} = 1, 2, \ldots, o, \ldots, O$ represent the set of these obstacles, where each obstacle has 3D coordinates denoted by $\ell_o = [x_o, y_o, h_o]$. To prevent collisions, the UAV maintains a minimum safe distance d_s from all obstacles. The potential collision state between UAV u and any obstacle o can be represented by:

$$C_{u,o}(t) = \begin{cases} 1, & \text{if } d_{u,o}(t) \text{ is less than } d_s, \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

The number of collisions at time t for a UAV u can be given by :

$$Z_u(t) = \sum_{o=1}^{O} C_{u,o}(t) \tag{8}$$

The traversal time of UAV u to reach over CH c_m for data collection can be calculated as:

$$T_{m,u}^{trav}(t) = \frac{d_{m,u}}{v_u(t)} \tag{9}$$

Here $d_{m,u}$ denotes the distance travelled by the UAV to reach CH m at time t and $v_u(t)$ represents the UAV's velocity at that time.The traversal energy of UAV u to reach over CH c_m for data collection can be given by:

$$E_{m,u}^{trav}(t) = T_{m,u}^{trav}(t) P_u^{trav} \tag{10}$$

where P_u^{trav} represents the power consumed by the UAV during traversal. The UAV must remain stationary above each CH c_m until it has received the complete data transmission. Therefore, the energy consumed by UAV u due to hovering can be expressed as:

$$E_{m,u}^{hov}(t) = T_{m,u}^{up}(t) P_u^h \tag{11}$$

Here, P_u^h denotes the hovering power of UAV u. It is assumed that the data gathered from each CH is offloaded to the BS before the UAV reaches its next hovering position. The energy consumed by UAV u to transmit the data from CH c_m can be computed as:

$$E_{u,BS}^{up}(t) = \frac{I_u(t)}{R_{u,BS}(t)} P_u^{trans} \tag{12}$$

Here, P_u^{trans} denotes the transmission power of the UAV, and $R_{u,BS}(t)$ represents the rate of data exchange between the UAV and the BS. The term $I_u(t)$ refers to the amount of data stored on the UAV after receiving it from CH c_m, and is given by:

$$I_u(t) = \sum_{n=1}^{N} I_n(t) X_{m,n}(t) Y_{u,m}(t) \tag{13}$$

D. AoI

We define *AoI* as a crucial performance measure that captures the time duration since the latest received data was initially generated in the defense surveillance region. The *AoI* for cluster head m at the t^{th} time frame is evaluated as:

$$A_m(t) = \begin{cases} T_{m,u}^{trav}(t) + T_{m,u}^{up}(t) & \text{if } Y_{u,m}(t) = 1 \\ \\ A_m(t-1) + T_{m',u}^{trav}(t) & \text{if } m' \neq m \\ +T_{m',u}^{up}(t) & Y_{u,m'}(t) = 1. \end{cases} \tag{14}$$

The average AoI of all CHs at t^{th} time frame is given as:

$$A(t) = \frac{1}{M} \sum_{m=1}^{M} A_m(t) \tag{15}$$

IV. Problem Formulation and NP Hardness Proof

This study focuses on reducing the weighted sum of AoI and energy usage by both UAVs and IoTDs, while ensuring collision avoidance in the mountainous military terrain. Consequently, the utility function is formulated as:

$$U = \frac{1}{\mathcal{T}} \sum_{t=1}^{\mathcal{T}} \Bigg[\sum_{u=1}^{\upsilon} \Big(w_1 E[-A_u(t)] + w_2 E[-E_u(t)] + w_4 E[-Z_u(t)] \Big) + w_3 E[-E_{IoTD}(t)] \Bigg] \tag{16}$$

where $\mathcal{T}$ is the number of time frames. w_1, w_2, w_3, and w_4 are weight factors for average AoI, UAV energy, IoTD energy, and collision penalty, respectively, and υ is the total number of UAVs. $E_u(t)$ is given by:

$$E_u(t) = \sum_{m=1}^{M} \Big(E_{m,u}^{trav}(t) + E_{m,u}^{hov}(t) + E_{u,BS}^{up}(t) \Big) Y_{u,m}(t) \tag{17}$$

Similary, $E_{IoTD}(t)$ can be given as :

$$E_{IoTD}(t) = \sum_{m=1}^{M} \sum_{n=1}^{N} E_{m,n}^{up}(t) X_{m,n}(t) + \sum_{m=1}^{M} E_{m,u}^{up}(t) Y_{u,m}(t) \tag{18}$$

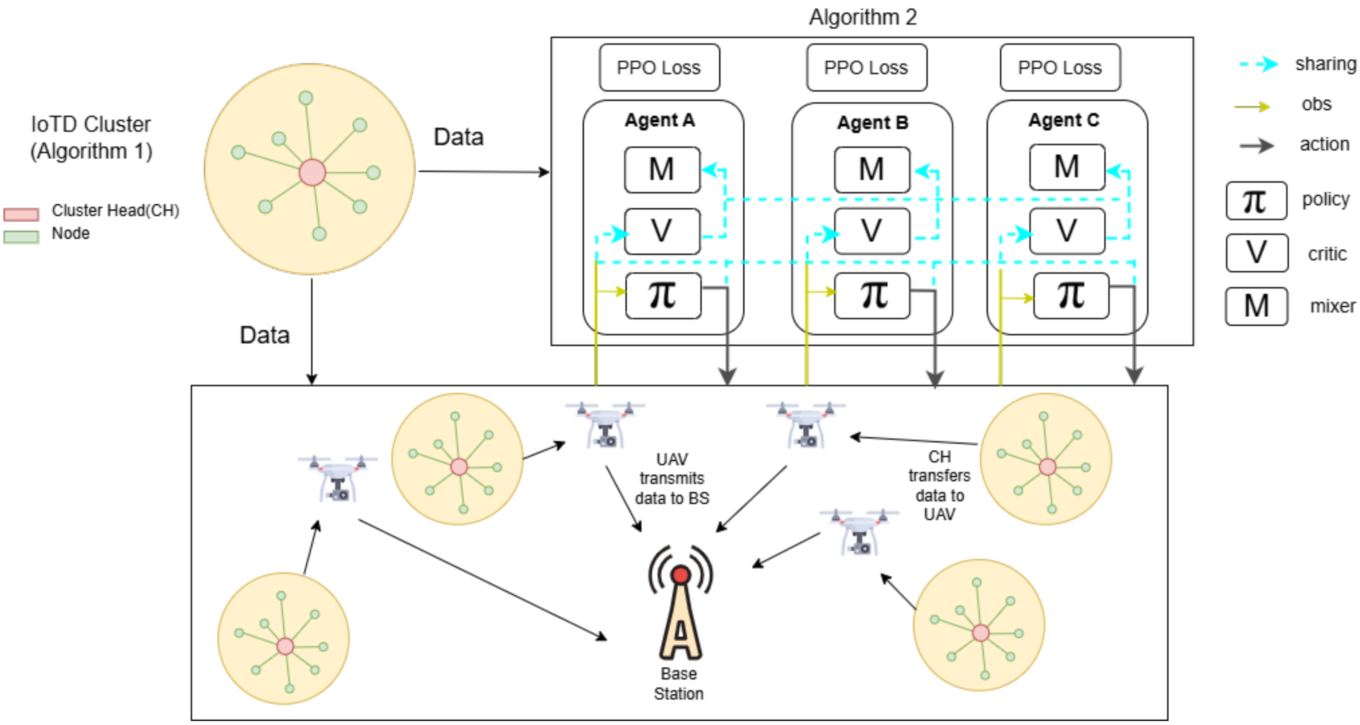

Fig. 2: Multi-Agent Proximal Policy Optimization

A. Problem Formulation

Our problem function is formulated as :

$$\textbf{P}: \quad \max \quad U \tag{19}$$

$$\sum_{t=1}^{\mathcal{T}} E_u(t) \leq E_u^{max}, \tag{19a}$$

$$v_u(t) \leq v_u^{max}, \forall t \in \mathcal{T}, \tag{19b}$$

$$y_{min} \leq y_u(t) \leq y_{max}, x_{min} \leq x_u(t) \leq x_{max}, \tag{19c}$$

$$d_{u,o}(t) \geq d_s, \forall o \in \mathcal{O} \tag{19d}$$

$$0 \leq w_1 + w_2 + w_3 + w_4 \leq 1, \tag{19e}$$

$$C_{max}^n(t) = h_u(t)\tan(\phi_n)., \tag{19f}$$

$$\|p_n(t) - p_j(t)\| \geq [C_{max}^n(t) + C_{max}^j(t)], \forall n, j, n \neq j., \tag{19g}$$

$$\|\ell_n(t) - \ell_j(t)\| \geq D_{min}, \forall n, j, n \neq j. \tag{19h}$$

Equation (19a) guarantees that the total energy consumed by UAV u remains within its maximum energy capacity E_u^{max}. Equation (19b) imposes a constraint that the velocity of the UAV at any time t must remain below the maximum permissible velocity v_u^{max}.Equation (19c) confines the UAV's location to remain within the specified boundaries of the service area, given by x_{min}, x_{max}, y_{min}, and y_{max}. According to Equation (19d), the UAV must maintain a safe minimum distance d_s from any obstacle $o \in \mathcal{O}$. Equation (19e) ensures that the combined weight factors remain within the [0, 1] range. Equation (19f) derives the maximum horizontal coverage radius $C_{max}^n(t)$ of UAV u at time t, based on its altitude (h_u) and the maximum elevation angle ϕ_u. In Equation (19g), $v_u(t)$ represents the UAV's 2D coordinates as $p_u(t) = [x_u(t), y_u(t)]^T$. This equation enforces the non-overlapping constraint to ensure that coverage areas of any two UAVs do not intersect. Lastly, Equation (19h) introduces the collision avoidance constraint, mandating that the separation between any two UAVs must be at least D_{min} to prevent collisions.

B. NP-Hardness Proof

The UAV path planning problem for IoT data collection with energy and AoI minimization is NP-hard.

To prove that this problem is NP-hard, we demonstrate that the Traveling Salesman Problem (TSP), which is known to be NP-hard, can be reduced to this problem in polynomial time.

Consider a special case of the UAV path planning problem with the following parameters:

- Set $\alpha = 1$ (weight for energy consumption)
- Set $\beta = 0$ (weight for AoI)
- Define energy consumption $E_u(p)$ as directly proportional to the Euclidean distance traveled along path p
- Require that the UAV must return to its starting position after visiting all IoT devices

Under these conditions, the problem reduces to:

$$p^* = \arg\min_{p \in P}\{E_u(p)\} = \arg\min_{p \in P}\{d(p)\} \tag{20}$$

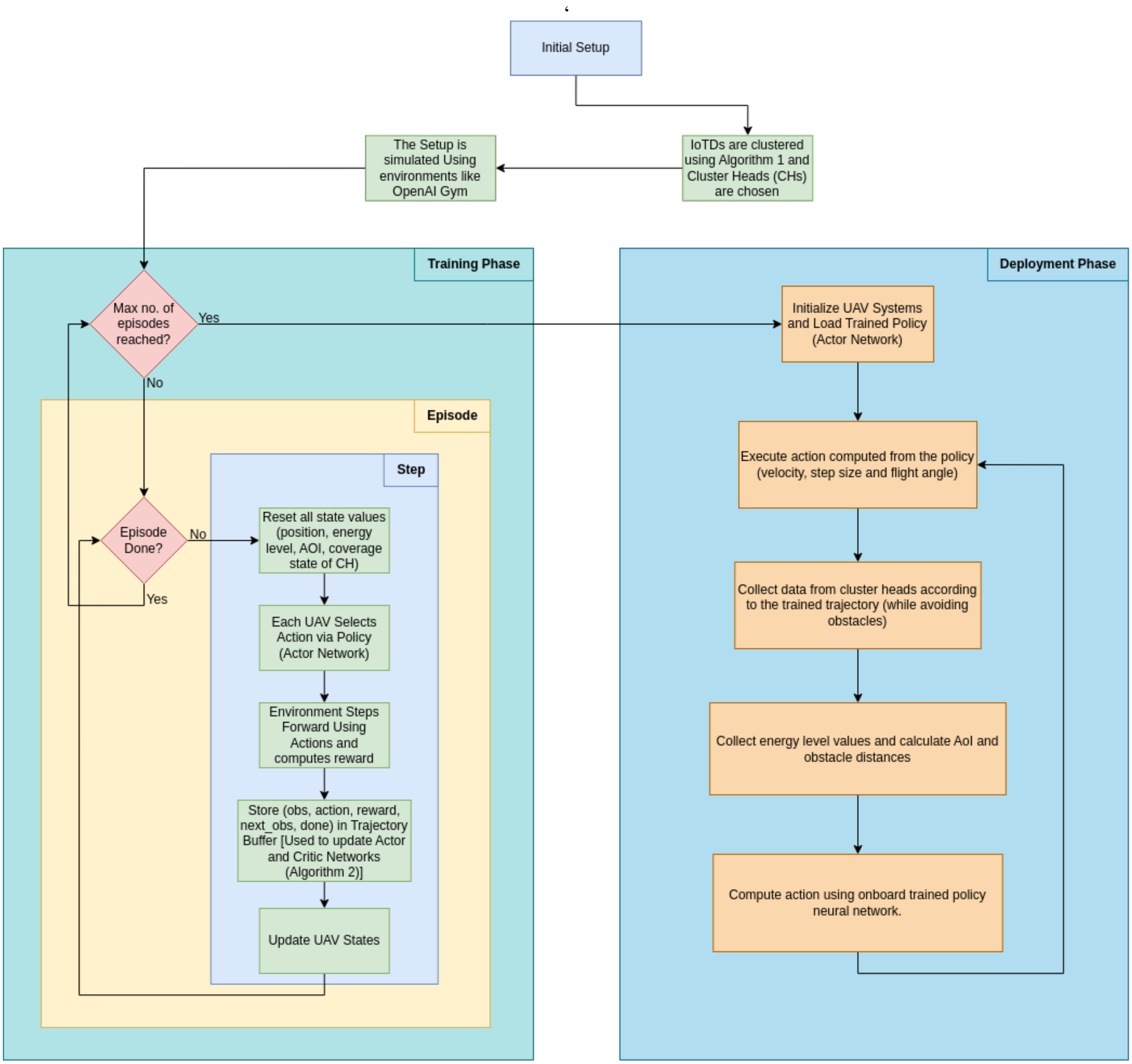

Fig. 3: Execution Flowchart

where $d(p)$ represents the total Euclidean distance of path p.

This is precisely the definition of the Euclidean Traveling Salesman Problem: finding the shortest path that visits each point exactly once and returns to the starting point. Since TSP is NP-hard, and this problem contains TSP as a special case, the UAV path planning problem is at least as hard as TSP, making it NP-hard.

Moreover, the addition of AoI considerations ($\beta > 0$) introduces time-dependent costs and further increases the problem complexity, as it creates a multi-objective optimization problem with time-sensitive constraints. This makes the problem a variant of the TSP with time windows, which is known to be strongly NP-hard.

C. Complexity Analysis

The UAV path planning problem exhibits several characteristics that contribute to its computational complexity:

1) **Combinatorial Explosion:** For N IoT devices, there are $(N!)$ possible visitation sequences, making exhaustive search intractable for large instances.
2) **Dynamic Costs:** The AoI increases with time, creating a time-dependent cost function where the cost of visiting a node depends on when it is visited.

3) **Multi-objective Optimization:** The problem involves balancing two potentially conflicting objectives: minimizing energy consumption and minimizing AoI.
4) **Spatial-Temporal Coupling:** Decisions about the spatial path directly impact the temporal aspects (AoI), creating complex interdependencies.

Algorithm 1: Clustering Algorithm

Input: The positions of the $\mathcal{N}$ nodes $p_{n_1}, \ldots, p_{n_N}$ within the region, the cluster radius R, UAV flight altitude h, flight speed v_u, and additional system parameters.

Output: A reduced set $\mathcal{M}$ containing information of cluster heads and corresponding nodes.

Initialization: Initialize related parameters;
for *i = 1 to N* **do**
  $temp_{C_i} \leftarrow$ cluster formation results;
  $temp_{CH_i} \leftarrow$ group of cluster heads;
  $temp_{C\_size_i} \leftarrow$ cluster size;
$M \leftarrow \min \text{size}(CH)$;
$C \leftarrow temp_C,$;
$CH \leftarrow temp_{CH}$;
$C_s ize \leftarrow temp_{C\_size}$;
Use the coordinates of the cluster head (CH) and data collector (DC) to solve subproblem 1, then optimize the UAV trajectory V, and determine the optimal solution for subproblem 2.;
for *i = 1 to C do* **do**
  for *k = 1 to C_size(j) do* **do**
    Determine the optimal sequence for collecting data from member nodes within a cluster to solve subproblem 3 effectively. ;

V. Proposed Solution

We solve the formulated problem outlined in Eq. (19) in three steps.

A. IoTD-CH clustering

First, we use a k-means-based clustering algorithm to find CHs (Algorithm 1) to segregate IoTDs into clusters and assign cluster heads to each cluster(Fig. 3).

B. MAPPO

Multi-Agent Proximal Policy Optimization (MAPPO) is a reinforcement learning algorithm designed for scenarios where multiple agents interact in a shared environment, each optimizing its own behavior while contributing to a common goal. MAPPO extends the single-agent PPO framework by incorporating centralized training with decentralized execution: during training, each agent's local policy benefits from access to global state information (via a shared critic), while at execution time, each agent acts using only its local observations. This design makes MAPPO robust in partially observable and cooperative settings, and it supports continuous action spaces, making it highly suitable for real-world robotic control and multi-UAV trajectory optimization.

Algorithm 2: MAPPO for UAV Cluster Head Data Collection

Input: Coordinates of IoT devices $\mathcal{N} = \{p_{n_1}, \ldots, p_{n_N}\}$, cluster heads $\mathcal{M} = \{p_{m_1}, \ldots, p_{m_K}\}$, UAVs $\mathcal{U} = \{p_{u_1}, \ldots, p_{u_U}\}$, obstacles $\mathcal{O} = \{p_{o_1}, \ldots, p_{o_O}\}$, actor-critic network parameters θ and ϕ

Output: Optimized policy networks π_θ for each UAV for visiting all cluster heads

Initialize actor networks π_θ and critic networks V_ϕ for each UAV;
for *each episode* **do**
  Reset environment and set $\mathcal{V} \leftarrow \emptyset$; `// Visited cluster heads`
  while $\mathcal{V} \neq \mathcal{M}$ ***and*** $t <$ *max_steps* **do**
    for *each UAV* $u_i \in \mathcal{U}$ **do**
      Observe $o_t^{(i)}$ (includes nearby obstacles, unvisited cluster heads);
      Select action $a_t^{(i)} \sim \pi_\theta(a_t^{(i)} | o_t^{(i)})$; `// Movement direction`
    Execute joint action $\mathbf{a}_t$ and move UAVs accordingly;
    Update $\mathcal{V}$ with newly visited cluster heads;
    Compute reward r_t based on visits, collisions, and energy consumption;
    Store transition $(s_t, \mathbf{o}_t, \mathbf{a}_t, r_t, s_{t+1}, \mathbf{o}_{t+1})$;
  Compute advantage estimates $A_t^{(i)}$ using GAE for each UAV;
  for K *epochs* **do**
    for *each mini-batch from collected data* **do**
      Update θ by maximizing clipped PPO objective:;

$$L(\theta) = \mathbb{E}\left[\min\left(r_\theta A, \, \text{clip}(r_\theta, 1-\epsilon, 1+\epsilon) \cdot A\right)\right]$$

      Update ϕ by minimizing value loss;

In the context of our paper, MAPPO provides an ideal framework due to the complex, dynamic, and decentralized nature of the environment. Here, multiple UAV agents must simultaneously optimize their trajectories to minimize the AoI while considering physical obstacles, energy constraints, and the networked structure of CHs that aggregate IoTD data. Each UAV, as a MAPPO agent, receives local observations (e.g., nearby CHs, its current AoI levels, distance to obstacles, remaining energy) and computes an action (trajectory update). During training, the centralized critic uses the global state

TABLE II: Simulation parameters

Parameter	Value	Parameter	Value
v_u	10 m/s	E_u^{max}	1000 J [4]
P_n	0.1 Watts [13]	I_n	5-10 KB
P^{trav}	[10-35] J/sec [14]	N	100
Max. steps	50	Batch size	256
r_m	500	p	100

(e.g., AoI across all CHs, energy status of all UAVs) to guide the learning of the local policies, ensuring coordinated and globally optimal behaviors.

MAPPO was chosen for our research because it efficiently handles the cooperative yet decentralized UAV coordination challenge, where each UAV must make local decisions that are still aligned with a global objective — minimizing network-wide AoI while avoiding collisions and conserving energy. Its policy-clipping and shared value function help ensure training stability and scalability across multiple agents, making it well-suited for real-time, large-scale UAV networks in smart IoT-based sensing systems.

C. Identification of optimal collection trajectory using MAPPO

In the third phase, we strategically use a Multi-Agent PPO-based policy [12] to find an optimal trajectory of the UAVs based on the locations of CHs In this setup, the UAVs are independent agents that learn and determine their next hovering location based on current environmental conditions and obstacles on the way. MAPPO employs a stochastic policy, ensuring automatic exploration. It collects a mini-batch of experiences from interactions with the environment to update the policy. After updating, a new batch is collected with the updated policy, classifying PPO as an on-policy algorithm. Its optimization strategy, which limits the policy update step, often results in more stable training and prevents drastic changes, making it well-suited for training systems like UAVs.

D. MDP Formulation

We consider the above-defined system model as an environment. Considering that each UAV's action may influence the environmental state, total utility is determined by the current state of the environment and the action of the UAV. Hence, the utility maximization problem can be re-formulated as a multi-agent MDP $\langle S, a, R, \epsilon, \gamma \rangle$ where S is state set of agent u, a is action space of agent u, ϵ denotes the state transition probability, R is reward function of agent u, and $\gamma \in [0,1]$ represents discount factor. A detailed description is provided in the following subsections.

1) State Space: The state of the environment encompasses various aspects, such as the position of the UAV, its energy levels, the average AoI of all CHs, and coverage status of the CHs. Mathematically, the state space can be represented as:

$$s(t) = \{\ell_u(t), \mathcal{E}_u(t), A(t), \mathcal{C}\} \tag{21}$$

TABLE III: Symbol description

Symb.	Description
$\mathcal{N}$	Set of IoTDs
$\mathcal{M}$	Set of cluster heads
$X_{n,m}(t)$	Variable to check association of IoTD n with cluster m at time t
$Y_{u,m}(t)$	Variable used to determine if UAV u is within the communication range of cluster head c_m at time t
$I_n(t)$	Data size of IoTD at t
r_o	Average radius of obstacle
$\mathcal{O}$	Coordinates of obstacles
h_u	Height of UAV (assumed to be constant)
$G(T^0_{m,t})$	The power gain of the channel between the m^{th} IOTD and the UAV
P_n	Transmitting power of the n^{th} IoTD
P_r^m	Receiving power of CH c_m
B	Available channel bandwidth
$R_{P_t,m}$	Data transmission rate
σ^2	Additive white Gaussian noise
$d_{m,u}$	Length of the flight segment from the last UAV position to the current CH
$R_{m,n}$	Transmission rate between IoTD and CH
$E_{m,n}(t)$	Energy consumption between IoTD n and CH c_m
β_0	Reference channel gain of distance
$d_{a,b}$	Distance between a and b
$A(\tau)$	AoI at time τ
$R_{m,u}(t)$	Data rate between CH c_m and UAV u
P_0	Blade profile
P_u^h	Hovering Power of UAV u
r_w	Wake radius
P_{wake}	Power consumed by the wake sensor equipped on the CH as the wake sensor is always active
d_s	Minimum safe distance from an obstacle
$C^t_{u,o}$	Collision state of the UAV u with obstacle o
$Z_u(t)$	Number of collisions at time t for UAV u
$v_u(t)$	Velocity of the UAV u
$I_u(t)$	Amount of data stored on the UAV u after receiving it from CH c_m
p_{n_i}	Position of 'i'th IoTD
p_{m_i}	Position of 'i'th CH
p_{u_i}	Position of 'i'th UAV
p_{o_i}	Position of 'i'th Obstacle

where $\mathcal{C} = \{C_1(t), \ldots, C_m(t), \ldots, C_M(t)\}$ denotes the coverage condition of all clusters. Binary variable $C_m(t) = 1$ means that CH m has been served, and vice versa.

2) Action Space: The action space of the UAV is:

$$a(t) = \{v_u(t), l_u(t), \theta_u(t)\} \tag{22}$$

where $l_u(t)$ represents the flight distance, and $\theta_u(t)$ denotes the flight angle of the UAV.

3) Reward: The agent's reward is influenced by the average Age of Information (AoI) of the cluster heads, the energy consumption of IoTDs, and the energy usage of the UAV, and

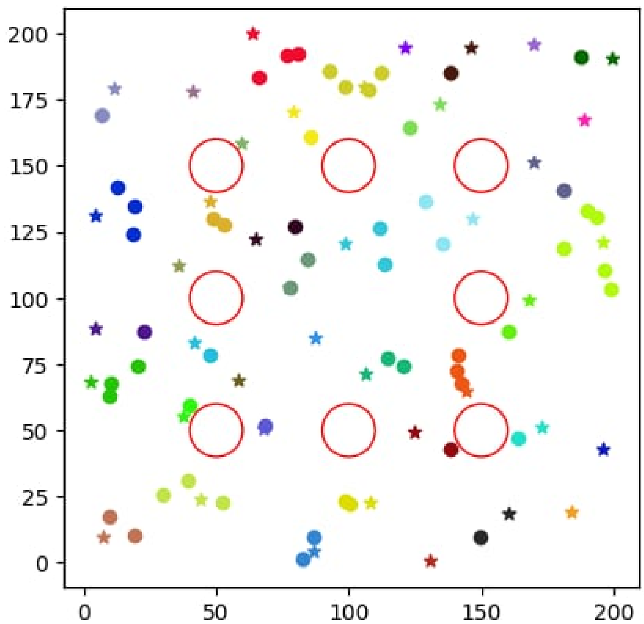

Fig. 4: Cluster heads.

can be expressed as:

$$r(t) = -w_1 A(t) - w_2 \sum_{u=1}^{v} E_u(t) - w_3 E_{IoTDs}(t) - w_4 \sum_{u=1}^{v} Z_u(t) + r_m - p \tag{23}$$

where r_m is the positive reward that UAV u receives for covering CH m, and p is the penalty received by the UAV for each uncovered CH.

VI. Performance Study

In this section, the training performance of the proposed algorithm is analyzed. Simulation parameters used in the experiment are given in Table II.

A. Comparing K-means with other clustering schemes

In this section, we compare the performance of our K-means clustering with Hierarchical clustering and DBSCAN. Hierarchical clustering builds a hierarchy of clusters by either merging smaller clusters (agglomerative) or splitting larger ones (divisive), based on distances between points, without needing a preset number of clusters. In our comparison, we have used agglomerative hierarchical clustering. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) groups points that are closely packed together based on distance and density, while marking isolated points in low-density regions as noise. Fig. 5 shows the variation of utility for each of the above clustering schemes. Here, K-means performs better than the other two clustering schemes. This is due to the fact that K-means works best on globular, convex, spherical clusters, which is the type of clusters we

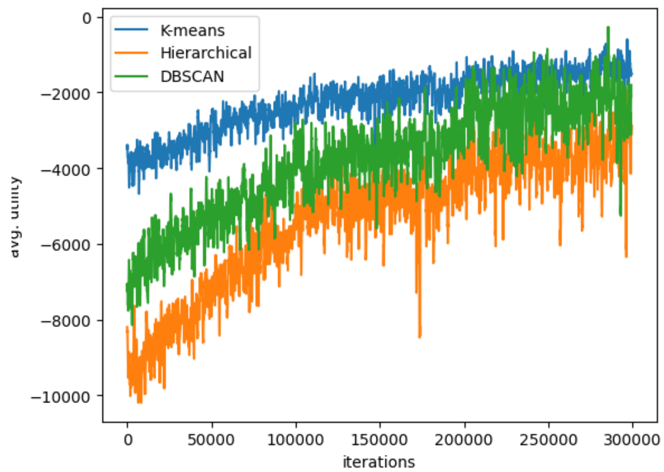

Fig. 5: Utility vs clustering scheme.

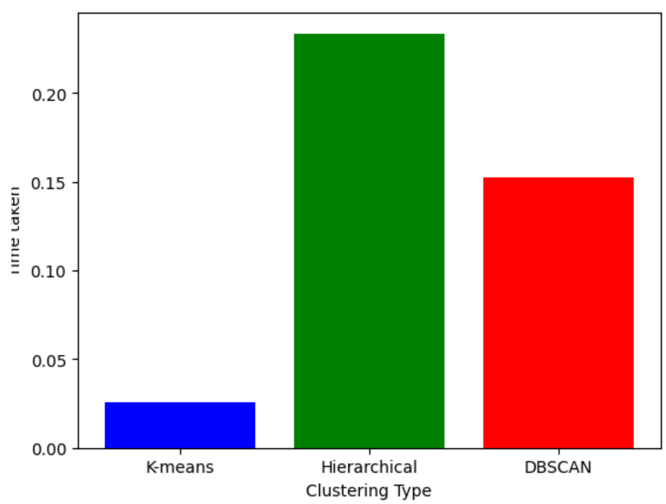

Fig. 6: Time taken vs clustering scheme.

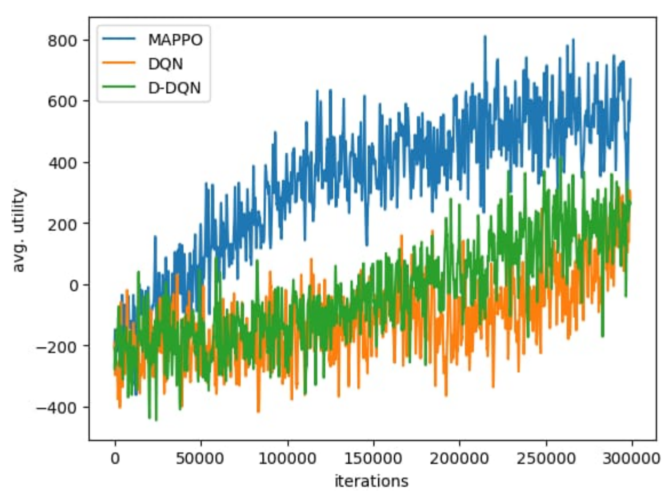

Fig. 7: Avg. Utility vs iterations.

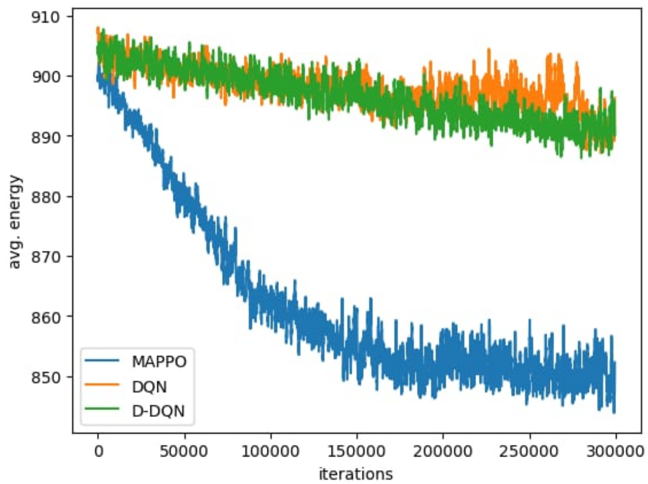

Fig. 8: Avg. Energy vs iterations.

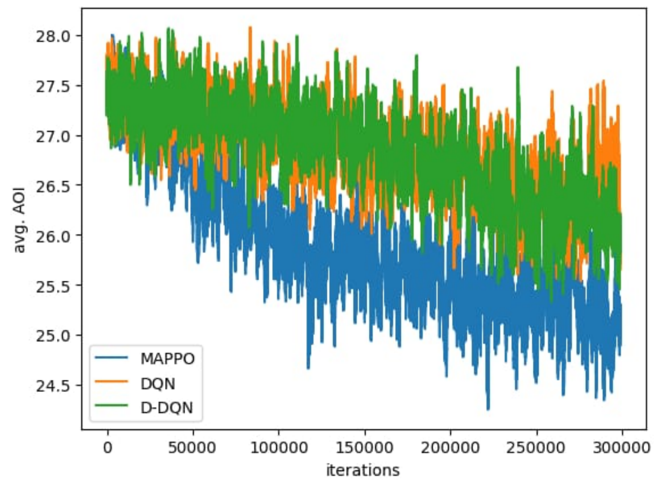

Fig. 9: Avg. AoI vs iterations.

are concerned with. Moreover, according to Fig. 6, K-means clustering takes significantly lesser time than the other clustering schemes. This is due to the fact K-means clustering has far less time complexity $(O(n))$ than DBSCAN $(O(nlogn))$ and Hierarchical clustering $(O(n^2))$.

B. Comparing MAPPO with other policies

We compare our work with the DRL-based Deep Q-Network (DQN) and Double Deep Q-Network (D-DQN) policies over 300,000 iterations using the same parameters to maximize utility. DQN and D-DQN are both value-based reinforcement learning algorithms that approximate Q-values using neural networks, contrasting with direct policy learning methods. As off-policy algorithms, both can learn from experiences collected using different policies than the one being optimized. Exploration is typically implemented through epsilon-greedy strategies, with random actions selected at a gradually decreasing probability to balance exploration (exploring new actions) and exploitation (choosing optimal actions). DQN achieves stability through two key innovations: experience replay buffers that break correlations in sequential data by randomly sampling stored transitions, and target networks that are periodically updated to mitigate moving target issues. D-DQN (Double DQN) extends this architecture by addressing DQN's tendency to overestimate Q-values, which can lead to suboptimal policies. It employs two separate networks—one to select actions and another to evaluate them—effectively decoupling action selection from evaluation. Both algorithms face challenges with continuous action spaces and struggle to scale efficiently to high-dimensional problems compared to policy gradient methods, explaining their lower performance in complex multi-UAV coordination tasks as shown in the comparative graphs.

We evaluated our approach by comparing its performance against DQN and D-DQN across multiple key metrics during the training process. Fig. (4) shows the spatial distribution of IoT devices (colored dots) grouped into clusters using K-means clustering, with each color representing a different cluster. Cluster heads are indicated by stars within each group. The red circles mark the locations of obstacles present in the environment.

Fig. (7) showcases the difference between the algorithms through average utility measurements. Our algorithm(Blue Line) demonstrates remarkable improvement, beginning at negative values around -200 and climbing to consistently positive values between 400-600 by training completion, with peaks reaching 800. This represents a transformative improvement in system effectiveness. In stark contrast, both DQN and D-DQN struggle significantly, starting around -400 and barely reaching positive territory by the end of training, with maximum values only around 200.

Similarly, Fig. (8) illustrates average energy consumption across the same training period, revealing our algorithm's superior energy efficiency compared to the alternatives. While all three algorithms begin with similar energy consumption levels, MAPPO demonstrates a substantial reduction by the end of training.

Fig. (9) demonstrates the average Age of Information (AoI) metric. Our algorithm significantly outperforms both DQN and D-DQN by consistently achieving lower AoI values throughout the training process.A lower AoI indicates superior information freshness in the UAV network. This is critical for real-time applications where timely data is essential.

The empirical results presented above validate the effectiveness of our approach to the utility maximization problem. Our comparative performance analysis demonstrates the algorithm's superior capability in optimizing multi-agent utility functions across diverse scenarios.

VII. Conclusion and Future Work

In this paper, we proposed a MAPPO-based framework for efficient task offloading and path planning in UAV-assisted IoT networks, specifically tailored for mission-critical scenarios such as defense operations. By jointly considering energy efficiency and the Age of Information (AoI), our model demonstrated significant improvements in both data freshness and energy conservation. The integration of K-means clustering and deep reinforcement learning enabled UAVs to effectively collect and relay data from IoTDs through designated cluster heads, ensuring reliable communication even in challenging terrains.

While our approach yields promising results, there are several avenues for future enhancement. Currently, both UAV-to-cluster allocation and UAV path planning are addressed under the same MAPPO framework. To improve computational efficiency and scalability, a promising direction would be to decouple UAV-to-cluster allocation as a separate subproblem. This would allow for more dynamic reassignment mechanisms in scenarios where a UAV fails or underperforms—ensuring continuous data collection by reassigning its cluster to another available UAV.

Moreover, our model assumes that UAVs operate at a fixed altitude throughout the mission. In practical deployments, allowing UAVs to adjust their altitude dynamically in response to obstacles, energy constraints, or communication quality could further optimize performance. Future work will explore the incorporation of variable-height UAV trajectory planning to enhance adaptability and mission robustness in complex environments.

References

[1] C. Liu, Y. Guo, N. Li, and X. Song, "Aoi-minimal task assignment and trajectory optimization in multi-uav-assisted iot networks," *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21 777–21 791, 2022.

[2] J. Baek, S. I. Han, and Y. Han, "Energy-efficient uav routing for wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1741–1750, 2019.

[3] S. Kumari, E. Sodhi, D. Gupta, and A. Pratap, "Aoi-aware deep reinforcement learning based uav path planning for defence applications," in *2024 IEEE Space, Aerospace and Defence Conference (SPACE)*, 2024, pp. 230–234.

[4] M. Sun, X. Xu, X. Qin, and P. Zhang, "Aoi-energy-aware uav-assisted data collection for iot networks: A deep reinforcement learning method," *IEEE Internet of Things Journal*, vol. 8, no. 24, pp. 17 275–17 289, 2021.

[5] M. Yi, X. Wang, J. Liu, Y. Zhang, and B. Bai, "Deep reinforcement learning for fresh data collection in uav-assisted iot networks," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2020, pp. 716–721.

[6] J. Xu, K. Ota, and M. Dong, "Big data on the fly: Uav-mounted mobile edge computing for disaster management," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2620–2630, 2020.

[7] C. Zhan and Y. Zeng, "Completion time minimization for multi-uav-enabled data collection," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4859–4872, 2019.

[8] R. Liu, Z. Qu, G. Huang, M. Dong, T. Wang, S. Zhang, and A. Liu, "Drl-utps: Drl-based trajectory planning for unmanned aerial vehicles for data collection in dynamic iot network," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1204–1218, 2022.

[9] P. Wan, G. Xu, J. Chen, and Y. Zhou, "Deep reinforcement learning enabled multi-uav scheduling for disaster data collection with time-varying value," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[10] J. Hu, X. Yang, W. Wang, P. Wei, L. Ying, and Y. Liu, "Obstacle avoidance for uas in continuous action space using deep reinforcement learning," *IEEE Access*, vol. 10, pp. 90 623–90 634, 2022.

[11] X. Zhou, Q. Zhu *et al.*, "Optimization algorithm for aoi-based uav-assisted data collection," *International Journal of Distributed Sensor Networks*, vol. 2024, 2024.

[12] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[13] L. Xu, M. Chen, M. Chen, Z. Yang, C. Chaccour, W. Saad, and C. S. Hong, "Joint location, bandwidth and power optimization for thz-enabled uav communications," *IEEE Communications Letters*, vol. 25, no. 6, pp. 1984–1988, 2021.

[14] W. Y. B. Lim, J. Huang, Z. Xiong, J. Kang, D. Niyato, X.-S. Hua, C. Leung, and C. Miao, "Towards federated learning in uav-enabled internet of vehicles: A multi-dimensional contract-matching approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5140–5154, 2021.