Available Online: 13 April 2026 DOI: 10.54254/2977-3903/2026.32787

Multi-UAV collaborative inspection method based on path overlap avoidance and differentiated quality control

Xingchao Qiu[1*], Xingyu He[2]

[1]School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai, China
[2]College of Publishing, University of Shanghai for Science and Technology, Shanghai, China

*Corresponding Author. Email: qiuchaocaho@gmail.com

Abstract. Existing studies on multi-UAV inspection methods largely overlook path overlap avoidance among UAVs and differentiated quality control of inspection tasks. As a result, such systems are unable to ensure both cooperative efficiency and the inspection quality of critical areas in practical operations. To address these limitations, this paper proposes a framework that integrates a path overlap avoidance mechanism with a task importance grading strategy. The framework adopts a hybrid decision-making architecture combining centralized task allocation with distributed reinforcement learning control. Specifically, a central controller performs global task assignment based on overall task importance, real-time UAV status, and available bandwidth. Meanwhile, individual Unmanned Aerial Vehicles (UAVs) employ reinforcement learning to autonomously conduct local trajectory planning and imaging decisions. Through the path overlap avoidance mechanism, redundant inspection of the same area is effectively prevented, while bandwidth is dynamically requested or released according to task importance. Simulation results demonstrate that, compared with existing approaches, the proposed method improves inspection efficiency by approximately 24% and enhances overall inspection quality by 20%.

Keywords: multi-UAV cooperative inspection, path overlap avoidance mechanism, task importance grading, reinforcement learning

1. Introduction

In recent years, with the acceleration of industrial intelligent transformation and the maturation of autonomous unmanned system technologies, the application of Unmanned Aerial Vehicles (UAVs) in industrial inspection has continued to expand. Owing to their flexible deployment and high maneuverability, single UAVs have been successfully applied in many standardized scenarios. However, when confronted with increasingly complex and large-scale modern industrial infrastructures, the limitations of single-UAV operations in terms of coverage capability, operational efficiency, and task robustness have become increasingly evident. Against this backdrop, multi-UAV cooperative inspection systems—capable of significantly improving inspection efficiency through parallel operations—have emerged as a focal research direction in both academia and industry [1].

The core objective of multi-UAV cooperative inspection is to achieve efficient, complete, and high-quality visual data acquisition of target areas under constraints of limited time and resources. To this end, existing studies have primarily focused on optimizing coverage efficiency and system energy consumption. From a technological development perspective, relevant approaches are gradually shifting from traditional optimization methods toward more flexible and adaptive reinforcement learning techniques [2].

With respect to traditional optimization methods, prior research has predominantly employed swarm intelligence algorithms such as genetic algorithms, particle swarm optimization, and ant colony optimization for solution optimization [3, 4]. These methods can achieve high coverage efficiency and path quality in scenarios where environmental information is complete and task constraints are fixed. Their strengths lie in clear physical interpretability and well-defined modeling structures. However, such approaches typically rely on accurate environmental modeling and global information synchronization. In dynamically changing inspection environments, they are prone to model mismatch and incur high computational costs for re-planning, making it difficult to meet the requirements for real-time responsiveness and adaptive adjustment.

With breakthroughs in deep reinforcement learning for sequential decision-making problems, data-driven multi-UAV cooperative inspection methods have emerged as a promising research direction. In this paradigm, each UAV is modeled as an intelligent agent that learns cooperative strategies through continuous interaction with the environment, without requiring precise prior mathematical modeling. Researchers have explored the use of deep Q-networks, actor–critic frameworks, and their multi-agent extensions (e.g., MADDPG, QMIX) to address problems such as cooperative path planning, dynamic task allocation, and collision avoidance [5, 6]. Reinforcement learning methods exhibit strong robustness and generalization potential in partially observable and communication-constrained environments, making them particularly suitable for complex inspection tasks where environmental models are unknown or difficult to construct accurately. Nevertheless, these methods still face challenges including instability in multi-agent training, low sample efficiency, and limited interpretability of learned policies, as well as a significant gap between simulation and real-world deployment.

Despite the notable progress achieved by the aforementioned studies in improving operational efficiency, a closer examination reveals that two critical issues remain insufficiently addressed. First, existing models generally lack explicit modeling and proactive suppression mechanisms for path overlap among multiple UAVs. In practical cooperative operations, due to the absence of global coordination or limitations in local perception, multiple UAVs are prone to redundant coverage in key areas or path conflicts in confined spaces. This not only leads to significant resource waste [7], but also directly constrains the realization of theoretical coverage efficiency. Second, most current studies treat inspection regions as homogeneous, failing to account for the differentiated quality requirements across different areas in industrial scenarios. In real-world inspection tasks, critical components often demand higher imaging precision and more reliable data transmission. However, due to the absence of task importance labeling, existing approaches are unable to dynamically allocate key resources—such as communication bandwidth—according to task priority [8, 9].

To systematically address these issues, this paper proposes a multi-UAV cooperative inspection framework that integrates a path overlap avoidance mechanism with task importance labeling. The main contributions of this study are as follows:

1) To address the widespread problem of path overlap in multi-UAV cooperative operations, a monitoring mechanism based on real-time geometric analysis of trajectories is designed. Through coordinated path adjustment, the proposed approach proactively avoids inspection path overlap at the system level, thereby significantly improving effective operational efficiency;

2) Breaking away from the traditional homogeneous task processing paradigm, this study explicitly incorporates regional importance grading and differentiated resource requirements into the optimization model

for the first time. An importance-aware joint optimization framework for imaging and transmission is developed, enabling intelligent and dynamic allocation of key resources—such as communication bandwidth—based on task priority.

3) The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the overall system modeling. Section 4 details the proposed utility-based centralized task allocation method. Section 5 introduces the reinforcement learning-based joint optimization algorithm. Section 6 provides numerical simulation results and analysis. Finally, the paper concludes with a summary of findings.

2. Related work

2.1. Multi-UAV cooperative inspection based on traditional methods

Multi-UAV cooperative inspection is a key research direction in the field of unmanned systems. In the development of cooperative inspection methods, approaches based on traditional optimization models and rule-based strategies have formed the early main technical pathways [10]. These methods typically model the problem as classical combinatorial optimization tasks, such as multi-traveling salesman problems or coverage path planning, and solve them using centralized or distributed strategies [11].

In centralized optimization, studies generally assume the presence of a global central node and apply heuristic methods, such as swarm intelligence algorithms, for solution search. For instance, Xu et al. [12] proposed a particle swarm optimization algorithm for multi-UAV cooperative path planning that accounts for communication constraints. This approach optimized path costs under static constraints; however, in dynamic scenarios, re-planning introduces significant computational overhead. While such methods can guarantee theoretical solution quality when accurate global information is available, their computational complexity grows exponentially with problem size and is highly dependent on stable and reliable global communication.

In distributed cooperation, research emphasizes self-organization through local rules. With the emergence of novel models such as Graph Neural Networks (GNNs), some studies have attempted to combine them with traditional optimization to enhance distributed cooperative performance. For example, Wang et al. [13] proposed a method that extracts topological features using GNNs and combines them with proximal policy optimization for cooperative path planning, improving the system's collaborative capability in partially observable environments. These approaches enhance scalability and robustness; however, because they still rely on accurate environment modeling, sudden changes in task patterns or strong disturbances can lead to reduced coordination efficiency or failure to resolve conflicts.

Although traditional methods offer advantages in modeling rigor and interpretability, their performance heavily depends on the accuracy of the preset model and the structured nature of the environment. In dynamic and complex industrial inspection scenarios, these methods exhibit clear limitations in real-time adaptability and cooperative flexibility, particularly in responding to changes in communication constraints or the insertion of unexpected tasks [14, 15].

2.2. Multi-UAV cooperative inspection based on reinforcement learning

With the rise of Deep Reinforcement Learning (DRL) techniques, researchers have explored multi-agent DRL for adaptive cooperative control of UAVs. For example, Westheider et al. [16] proposed an adaptive path planning method based on multi-agent DRL, which significantly improved multi-UAV task planning capability in 3D environments through the introduction of collaborative learning mechanisms and effectively addressed coordination among UAVs. Similarly, Shen et al. [17] developed the DNQMIX approach based on deep reinforcement learning for multi-UAV target search. By employing a centralized training and distributed

execution mechanism, this method enhanced coverage and target capture efficiency, demonstrating the effectiveness of deep reinforcement learning in complex cooperative UAV tasks. Additionally, Xu et al. [18] introduced a distributed actor–critic network structure for multi-UAV cooperative pursuit planning, achieving higher success rates and faster capture times, highlighting the potential of multi-agent reinforcement learning in complex dynamic decision-making.

Despite these advances in path planning, existing studies still show insufficient attention to two critical issues: first, there is a lack of dedicated mechanisms to handle path overlap and local conflicts among UAVs; second, current research does not adequately address differentiated inspection quality. In practical scenarios, different inspection targets have varying quality requirements, but existing methods treat them as homogeneous, limiting the ability to implement fine-grained quality control. Therefore, designing a joint decision-making method that integrates path overlap avoidance and task importance grading for bandwidth-constrained multi-UAV cooperative inspection tasks is of significant research value.

3. System model

3.1. Task scenario description

This chapter formalizes the multi-UAV cooperative inspection system and clearly defines the core optimization problems to be addressed. The system consists of a heterogeneous UAV fleet and a set of inspection tasks. The UAV fleet includes two types of roles:

Inspection UAVs responsible for capturing image data. Suppose there are (i) such UAVs, denoted as:

$$U = \{U_1, U_2, U_3 \ldots U_i\} \tag{1}$$

A data-collection UAV that receives data and serves as the central coordinator, denoted as U_0 .

UAVs operate in three-dimensional space, and their trajectories can be represented as discrete time-sequenced position series, where k is the maximum number of steps for task execution, as shown in Equation (2). The position of UAV U_i at time t is denoted by $p_i(t)$, determined by the current coordinates $(x_i(t), y_i(t), z_i(t))$:

$$p_i(t) = (x_i(t), y_i(t), z_i(t)), t = 1,2, \ldots, k \tag{2}$$

The energy consumption of an imaging $U_i(i > 0)$ is proportional to its flight distance, hovering time, and number of imaging actions. Let $E_i(t)$ denote the remaining energy of UAV U_i at time t , and $n_i^{cap}(t)$ denote the number of imaging actions executed at that time. Then, the energy update can be expressed as:

$$E_i(t + \Delta t) = E_i(t) - (||p_i(t + \Delta t) - p_i(\mathrm{t})|| + \Delta t + n_i^{cap}(t)) \tag{3}$$

Inspection tasks are discretized into a set of task points requiring high-quality imaging, such as industrial equipment or building surfaces:

$$T = \{T_1, T_2, T_3 \ldots T_j\} \tag{4}$$

Each task point T_j represents a specific piece of equipment or a building, characterized by its center position $O_j = (x_j, y_j, z_j)$, inspection radius r_j , volume information v_j , and a number of inspection subregions.

3.2. Spherical mapping and task representation

In cooperative UAV inspection, task points are often complex three-dimensional structures, such as equipment surfaces, 3D buildings, or power towers.

To unify the description of UAV inspection targets and camera orientations, this paper introduces a spherical task representation model. This model maps inspection targets from 3D Euclidean space to a spherical coordinate domain, and further projects them onto a 2D plane to enable unified representation and hierarchical allocation. The approach is as follows: for each target, a sphere is constructed with the target's center as the origin. The surface regions of the target are mapped onto this sphere, where key areas and normal areas are labeled. The labeled spherical surface is then unfolded into a 2D plane. Local inspection importance is encoded as weights on the discretized surface elements (Fig. 1).

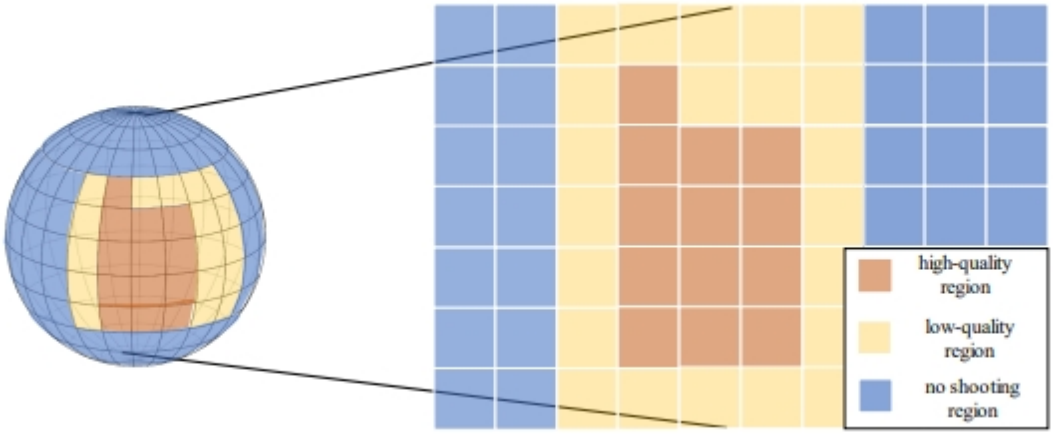

Figure 1. Spherical segmentation model

3.2.1. Spherical segmentation definition

For a task point T_j, let its geometric center be O_j and establish a local spherical coordinate system with O_j as the origin. To discretize the sphere, it is divided into $m*n$ small surface elements. Let $\mathscr{C}_j$ denote the set of all elements under task T_j. The UAV U_i at time t at position $p_i(t)$ with camera orientation $_i$ defines a capture projection function:

$$F(p_i(t),\ _i) = c_{d,b} \tag{5}$$

When UAV U_i is at position $p_i(t)$ and captures images along a viewing direction $_i$, the corresponding task elements on the spherical surface within its field of view are denoted as $c_{d,b}\,(0 < d \le m, 0 < b \le n)$. Each surface element $c_{d,b}$ has a binary state: $c_{d,b} = 1$ indicates that the element has been imaged, while $c_{d,b} = 0$ indicates it has not yet been imaged.

To emphasize the inspection priority of different local regions within a task point, an importance weight is assigned to each surface element $c_{d,b}$:

$$\omega_{d,b} = \begin{cases} \omega_{major}, & if\ c_{d,b}\ belongs\ to\ a\ key\ region \\ \omega_{normal}, & if\ c_{d,b}\ belongs\ to\ a\ normal\ region \\ 0, & if\ c_{d,b}\ does\ not\ require\ imaging \end{cases} \tag{6}$$

In the proposed inspection framework, the importance of each task point is modeled as the weight attribute of spatial surface elements. As expressed in Equation (6), each discretized surface element $c_{d,b}$ of a three-dimensional target is assigned a corresponding importance weight $\omega_{i,j}$ based on the structural criticality of the equipment. The weight of key regions, ω_{major}, is significantly higher than that of general regions, ω_{normal}, i.e., $\omega_{major} > \omega_{normal} > 0$. Under this weighting scheme, during task assignment and reinforcement learning decision-making, only surface elements with $\omega_{d,b} > 0$ are considered for task allocation and state updates. This enables priority-based coverage under resource-constrained conditions, ensuring that critical regions receive higher-probability and higher-quality imaging.

The advantage of representing tasks on a spherical surface lies in its ability to convert any complex 3D target into a continuous spatial set, allowing task assignment and reinforcement learning decisions to be conducted in a finite state space.

3.2.2. UAV imaging model

The imaging decision of UAV U_i at time t, denoted as $n_i^{cap}(t)$, updates the states of surface elements within its current field of view. The update rule is defined as:

$$c_{d,b}(t+1) = \begin{cases} 1, & if\ c_{d,b} = F(p_i(t),\ _i)\ and\ c_{d,b}(t) = 0 \\ c_{d,b}, & otherwise \end{cases} \tag{7}$$

This update rule ensures dynamic state management of task point elements: elements that are already imaged or do not require imaging are no longer assigned, effectively reducing task redundancy and energy consumption.

3.2.3. Relationship between image quality and bandwidth

Let the raw image quality $Q_i^{shot}(t)$ captured by inspection UAV $U_i(i > 0)$ at time t be denoted as:

$$Q_i^{\text{shot}}(t) = f(p_i(t), \theta_i(t), \phi_i(t)) \tag{8}$$

where $\theta_i(t), \phi_i(t)$ represent the elevation and azimuth angles of UAV U_i relative to the surface element $c_{d,b}$ at time t. That is, the clarity of the captured image is determined by the relative distance and viewing angle between the UAV and the target surface element.

In practical scenarios, limited communication bandwidth may lead to image quality degradation during transmission. Under bandwidth constraints, the captured images are compressed, and the effective image quality after compression $Q_i^{eff}(t)$ is expressed as:

$$Q_i^{eff}(t) = Q_i^{shot}(t) \cdot g\left(\frac{band_i(t)}{band_{req}}\right) \tag{9}$$

where $band_i(t)$ denotes the uplink bandwidth allocated to UAV U_i, and $band_{req}$ represents the minimum bandwidth required for high-quality image transmission. The function $g(\cdot)$ is a bandwidth factor, defined as the ratio of the currently allocated bandwidth $band_i(t)$ to the required minimum $band_{req}$. Specifically, $g(\cdot) = 1$ when $band_i(t) \geq band_{req}$; otherwise, $g(\cdot) < 1$, quantifying the reduction in image quality caused by insufficient bandwidth.

4. Inspection task allocation based on centralized optimization model

To address task conflicts, resource competition, and global efficiency optimization in multi-UAV cooperative inspection, this chapter proposes a centralized task allocation model based on global state awareness and constrained optimization. As shown in Fig. 2, the model uses the data-collection UAV U_0 as the central controller. It periodically collects global state information and solves a multi-constraint combinatorial optimization problem to assign appropriate inspection task points to each image-capturing UAV, achieving system-level coordination of task importance, UAV status, and communication resources.

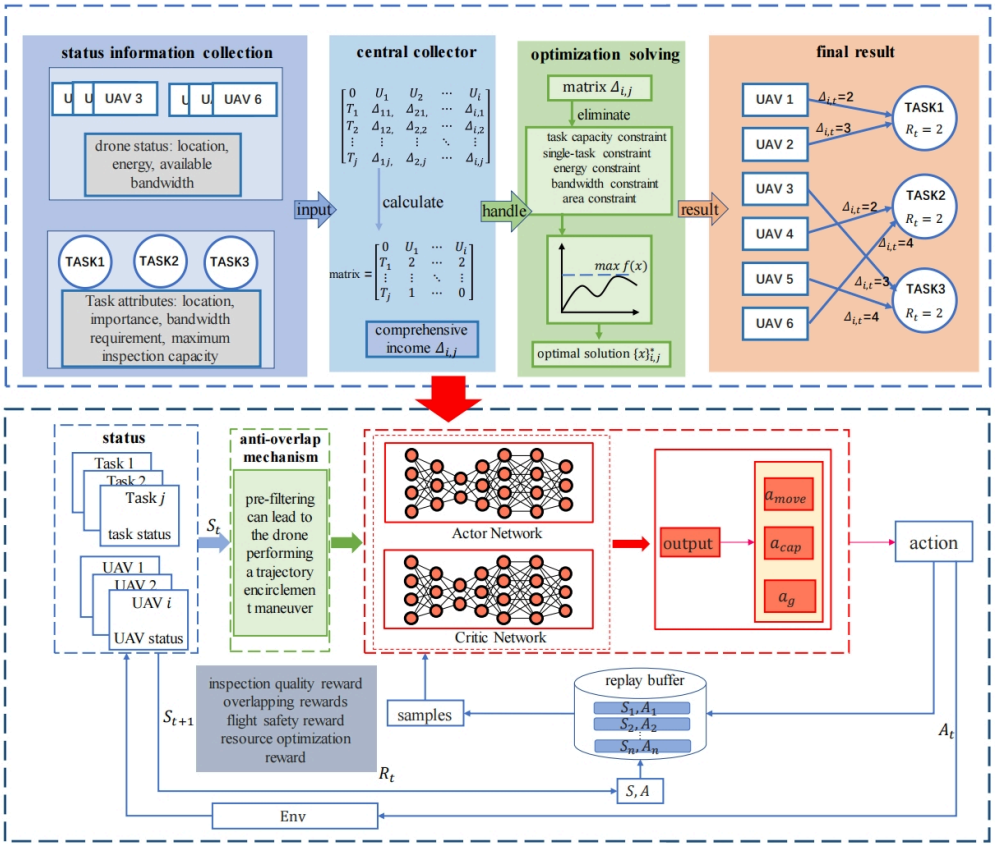

Figure 2. System framework

Given the significant heterogeneity of inspection tasks (e.g., varying spatial locations, criticality, and bandwidth requirements) and limited UAV resources (e.g., energy constraints, limited concurrency per task), a utility-based centralized allocation mechanism is employed for task assignment.

4.1. Model constraints

The task allocation problem is formulated as a multi-constraint combinatorial optimization problem. The objective function is defined as:

$$max f(x) = \sum_{U_i \in U} \sum_{T_j \in T} (x_{i,j} \cdot \Delta_{i,j}) \tag{10}$$

s.t.(11)-(14)

$$\sum_{i\in U} x_{i,j} \le R_j, \forall T_j \in T \tag{11}$$

$$\sum_{j\in T} x_{i,j} \le 1, \forall U_i \in U \tag{12}$$

$$x_{i,j} = 1 \to E_i \ge E_{min} + \gamma \cdot d_{i,j} \tag{13}$$

$$\forall T_j \in T_{critical}, \sum_{i:R_i \ge R_{req}} x_{i,j} \ge 1 or \sum_{U_i \in U} x_{i,j} = 0 \tag{14}$$

Here, U represents the set of inspection UAVs, T denotes the set of inspection tasks (a subset of surface elements $\mathscr{C}$ with $\omega_{d,b} > 0$ in the spherical mapping model), and $x_{i,j} \in \{0,1\}$ indicates whether UAV U_i is assigned to task T_j. $\Delta_{i,j}$ represents the expected inspection utility of UAV U_i performing task T_j, accounting for task importance, distance cost, UAV energy, and bandwidth availability.

Each task point allows a different number of inspection UAVs depending on its spherical-mapped volume and importance. Specifically, the maximum number of concurrent UAVs assigned to task T_j is denoted as R_j, formulated as Equation (11)

To avoid a single UAV performing multiple tasks simultaneously, each UAV is allowed to select at most one task point at any given time: Equation (12)

UAVs are required to maintain sufficient residual energy to safely return to base. Therefore, when selecting a task point, the UAV must consider its remaining energy, formulated as Equation (13) where E_i is the residual energy of UAV U_i, E_{min} is the minimum energy required for safe return, γ is the per-unit-distance energy consumption. This constraint ensures that UAVs can safely return after completing their missions.

For tasks belonging to critical regions (i.e., $\omega_{d,b} = \omega_{marjor}$), the bandwidth constraint must also be satisfied. A UAV U_i can only be assigned to such a task if its allocated bandwidth satisfies $R_i \ge R_{\text{req}}$. If no UAV meets this requirement, the assignment is temporarily deferred until sufficient bandwidth becomes available or the task is rescheduled, expressed as Equation (14) where $T_{critical}$ denotes the set of all critical task regions. This mechanism ensures that critical regions are not covered with low-quality imaging, thereby preserving the integrity of key data acquisition.

4.2. Utility function design and centralized solution procedure

The comprehensive utility function $\Delta_{i,j}$ reflects the trade-off between task value, execution cost, and system resources:

$$\Delta_{i,j} = I_j \cdot \left(\frac{1}{d_{i,j}}\right) \cdot 1_{band_i \ge band_{req}} \tag{15}$$

where: I_j is the importance weight of task T_j (from the spherical model $\omega_{d,b}$, $d_{i,j}$ is the shortest distance from UAV U_i to task T_j in 3D space, $1_{\{\cdot\}}$ is the indicator function ensuring that UAV U_i only contributes positive utility to high-quality inspection tasks when its allocated bandwidth $band_i$ meets the required minimum $band_{req}$, which takes the value 1 only when UAV U_i 's available bandwidth $band_{req}$ is greater than or equal to the minimum bandwidth required for high-quality imaging $band_{req}$; otherwise, it is 0. This ensures that a UAV contributes positive utility to high-quality inspection tasks only when sufficient bandwidth is available.

The central controller collects all UAV status information (positions, energy, available bandwidth) and task attributes (positions, importance, bandwidth requirements). It calculates the utility $\Delta_{i,j}$ for every UAV-task pair and removes infeasible pairs that violate energy or bandwidth constraints.

The feasible utility matrix is then modeled as a maximum-weight bipartite graph matching problem, solved efficiently using the Hungarian algorithm. Through matrix transformations, independent zero-element searches, and augmenting path adjustments, the algorithm computes the optimal allocation $\{x\}_{i,j}^{*}$ in polynomial time. This ensures: Each UAV is assigned at most one task, Each task is assigned no more UAVs than its allowed limit, System-wide utility is maximized.

Finally, the allocation results are dispatched to the UAVs to serve as constraints for subsequent reinforcement learning-based decision-making. This mechanism provides near-optimal solutions while keeping computational complexity manageable.

5. Inspection path planning based on distributed reinforcement learning

As shown in Figure 2, within the proposed cooperative inspection framework, the distributed action control layer is trained using an Actor-Critic (AC) network, aiming to enable each UAV to learn optimal flight trajectories and imaging strategies in continuous space. To ensure trajectory continuity, a collision-prevention mechanism is introduced as a pre-filter before reinforcement learning decisions, rejecting proposed actions that may lead to path overlap. Specifically, the action network in the AC framework generates preliminary action proposals based on local observations, including movement direction, imaging decisions, and bandwidth operations. The collision-prevention mechanism then filters these proposals according to the real-time environment and collaborative state, eliminating inefficient behaviors that could result in area enclosure, self-enclosure, or mutual enclosure. Finally, the critic network evaluates the filtered feasible action set, estimating its long-term value under the current global state. This guides UAVs to optimize cooperative inspection efficiency and imaging quality while avoiding path overlaps

5.1. Path overlap prevention mechanism in multi-uav cooperative inspection

In dynamic 3D environments, local path optimization and temporary failures of group coordination can lead UAVs into three typical inefficient states: area enclosure, self-enclosure, and mutual enclosure. These states cause path overlap, increasing energy and time costs, and reducing overall coverage efficiency and image quality. To address this, a real-time state monitoring and dynamic replanning mechanism is introduced. It continuously tracks UAVs' local motion features and relative positions, identifies enclosure risks, and triggers cooperative replanning strategies. Overlap penalties are incorporated into trajectory reward design, encouraging UAVs to maintain efficient, balanced spatial distributions.

Let the position of UAV U_i at time t be $p_i(t)$, and its neighborhood be denoted as $\mathcal{N}_{cube}$. The historical trajectory projected onto the horizontal plane forms a polygon P .

5.1.1. Area enclosure

As shown in Figure 3a, area enclosure occurs when a single UAV's inspection trajectory forms a closed loop in space, fully enclosing one or more uncovered surface elements (facelets), resulting in blank regions. If other UAVs need to inspect the enclosed area, they must pass through already-inspected regions, causing trajectory overlap:

$$c_{d,b} \in Int\,(P) \wedge c_{d,b} = 0 \wedge \omega_{d,b} \neq 0 \tag{16}$$

As shown in Equation (16), $Int(P)$ denotes the interior of a polygon P. If there exists a surface element $c_{d,b} \in Int(P)$ such that $\omega_{d,b} \neq 0$, a trajectory enclosure is considered to have occurred. Therefore, if a candidate path and historical paths form a closed polygon in projection, and there exist critical surface elements within this polygon that are yet to be inspected, the candidate path is prohibited.

To address this situation, UAV U_i evaluates whether its trajectory may form a closed region during inspection, and avoids entering paths that could create such enclosures.

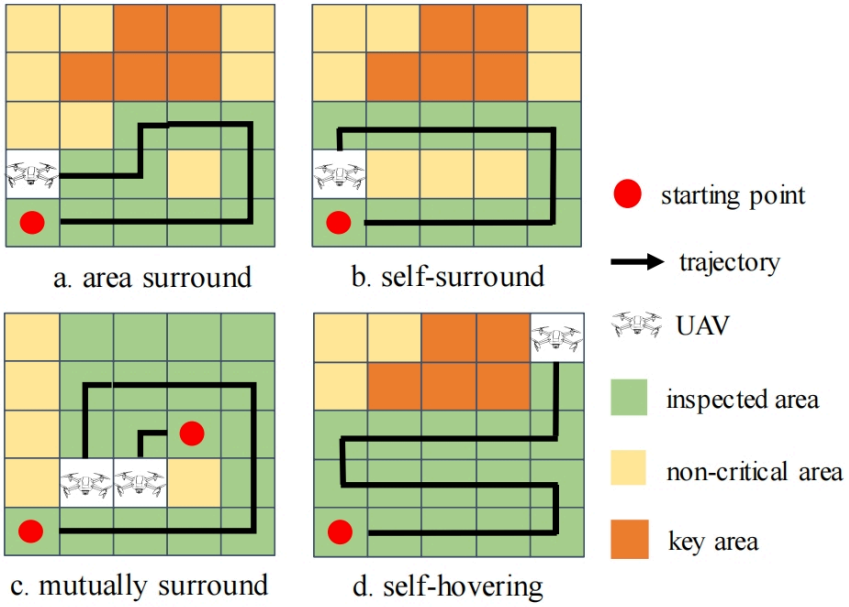

Figure 3. Classification of trajectory overlap phenomena

5.1.2. Trajectory enclosure

As illustrated in Figures 3b and 3c, self-enclosure refers to a single UAV forming a small-radius loop or spiral in a local area, causing it to be surrounded by its previously inspected dense region. Mutual enclosure occurs when multiple UAVs collaborate and one UAV's path partially or fully surrounds another, creating a collective deadlock.

Both scenarios can cause UAVs to be trapped by previously inspected areas; any UAV inside the enclosure must overlap with already covered regions when moving to new tasks, leading to redundant coverage and reduced efficiency.

To prevent self-enclosure, divergence constraints are imposed on UAV trajectories. By evaluating recent trajectory turning consistency, if a UAV is determined to be performing sustained rotations in the same direction, a corrective reverse-flight command is issued to steer the UAV toward areas with higher densities of uninspected elements, thereby breaking imminent loops.

For mutual enclosure, UAV U_i determines whether it can move to the next step via $AllowMove$. Before selecting its next target p_{next}, it checks the positions of nearby UAVs U_n. If the distance to any UAV U_n is less than a safety distance d_{safe}, the move is canceled. This predictive coordination mechanism resolves potential conflicts at the planning stage, preventing UAVs from entering mutual enclosure during execution, as formalized in Equation (17):

$$AllowMove\left(p_i^{next}\right) = \begin{cases} 0, & if\ \exists U_n \in U : ||p_n(t) - p_i^{next}|| < d_{safe} \\ 1, & otherwise \end{cases} \tag{17}$$

5.1.3. Self-hovering

If UAV U_i detects insufficient bandwidth to support high-quality imaging, it temporarily forgoes capturing high-quality areas and enters a hover state, as illustrated in Figure 3d. During hovering, the UAV's trajectory is temporarily interrupted, forming a static obstacle in space and time. Other UAVs nearby may take detours to avoid potential collisions, which may unintentionally cause local trajectory overlaps. Due to limited local sensing and other practical constraints, complete avoidance of all trajectory overlaps is infeasible; the core objective of this mechanism is to suppress overlaps to an acceptably low level.

5.2. Reinforcement learning model definition

The multi-UAV collaborative inspection problem is formulated as a Markov Decision Process (MDP). The UAV action policy model is defined as follows:

5.2.1. UAV state

The state of UAV U_i is denoted by S, as shown in Equation (18). $tr_i = \{pos_1, pos_2 \dots pos_i\}$, $pos_i = (x_i, y_i, z_i)$, records the UAV's historical trajectory in a time series. $band_i$ represents the current available bandwidth, T contains all task information, and B_i is the UAV's remaining energy.

$$S = \{tr_i, band_i, T, B_i\} \tag{18}$$

5.2.2. UAV actions

UAV U_i 's action in continuous space is represented as a triplet, as in Equation (19): M denotes 3D flight actions, Cap denotes capture actions (shoot or not shoot), and $G_i = \{G_{get}, G_{release}\}$ represents UAV-initiated bandwidth request or release actions.

$$A = \{M, Cap, G\} \tag{19}$$

5.3. Collaborative UAV reward design

5.3.1. Inspection quality reward

After performing capture actions, UAV U_i receives a quality reward R_{qual} :

$R_{qual} = Q(c_{d,b}) + \Phi\#(20)$ Where $Q(c_{d,b})$ accounts for imaging clarity and transmission fidelity:

$$Q(c_{d,b}) = \frac{d_{i,c} \cdot s_{pix}}{f} \cdot 1_{band_i \geq band_{req}} \tag{21}$$

Here, $d_{i,c}$ is the distance between UAV U_i and the surface element $c_{d,b}$, s_{pix} is the camera sensor pixel size, f is the lens focal length, and $1_{(\cdot)}$ indicates whether the image transmission quality meets bandwidth requirements $band_i \geq band_{req}$.

Φ penalizes unnecessary repeated inspections; the lower the value, the better:

$$\Phi = \frac{1}{|\mathscr{C}_j|} \sum_{c \in \mathscr{C}} max(0, |\mathscr{V}(c_{d,b}) - 1) \tag{22}$$

Where $\mathscr{C}_j$ is the number of surface elements under task T_j, and $|\mathscr{V}(c_{d,b})|$ is the total number of times element $c_{d,b}$ has been covered. This encourages UAVs to explore uninspected regions and reduce redundant information collection.

5.3.2. Anti-overlap reward

The movement reward R_{move} prevents trajectory overlaps due to enclosure phenomena, as in Equation (23), with $\gamma_1 < \gamma_2 < 0$:

$$R_{move} = \begin{cases} \gamma_1, & if\ regional\ enclosure\ occurs \\ \gamma_2, & if\ trajectory\ enclosure\ occurs \end{cases} \tag{23}$$

5.3.3. Flight safety reward

Safety reward R_{safe} guides UAVs to maintain safe distances from each other and obstacles:

$$R_{safe} = -\frac{1}{d_{min}} \tag{24}$$

Where d_{min} is the minimum distance to all other UAVs and known obstacles.

5.3.4. Resource optimization reward

To prevent waste of bandwidth, d_{min} evaluates UAV U_i's bandwidth allocation decisions:

$$E_{bw} = \begin{cases} \psi_1, & band_i \geq R_{req}\ and\ G_i = G_{release} \\ \psi_2, & band_i < R_{req}\ and\ G_i = G_{get} \\ \psi_3, & band_i > R_{req}\ and\ G_i = G_{get} \end{cases} \tag{25}$$

Where ψ_1, ψ_2, ψ_3 are reward/penalty parameters, and $\psi_1 > \psi_2$. This guides UAVs to allocate bandwidth efficiently under constrained conditions.

The total reward $R_{total}(i,t)$ for UAV U_i at time ttt is a weighted sum of all components:

$$R_{total}(i,t) = \theta_1 R_{qual}(i,t) + \theta_2 R_{move}(i,t) + \theta_3 R_{safe}(i,t) + \theta_4 R_{bw}(i,t) \tag{26}$$

6. Experimental design and results analysis

To evaluate the performance of the proposed method, simulation experiments were conducted in a Python environment. The proposed method was compared with existing approaches in terms of average flight distance, number of trajectory overlaps, bandwidth utilization, and final inspection quality.

Table 1. Parameter configuration

Parameter	Setting	Parameter	Setting
Number of UAVs	7	Camera focal length	12mm
Maximum flight speed	5dist/t	Safety distance	5m
Initial energy	100ene	Movement reward γ1, γ2	-2,-3
Unit energy consumption	0.5ene/dist	Bandwidth reward ψ1, ψ2, ψ3	3,1,-3
Minimum bandwidth for high-quality capture	25bps	Reward weights θ1, θ2, θ3, θ4	0.4,0.2,0.2,0.2
Number of task points	40	Actor network learning rate	0.0001

Total system bandwidth	100bps	Critic network learning rate	0.0001
Camera pixel size	2.0um		

The deep reinforcement learning model designed in this study was implemented based on the PyTorch framework and trained on an NVIDIA RTX 4060 GPU. In the task allocation and path planning simulations, appropriately sized virtual inspection regions were established. The initial positions of UAVs and task points were randomly generated within these regions, while ensuring a reasonable distance between UAV starting points and task locations. The units of energy, distance, and time used in the experiments are abstract simulation units, denoted as ene (energy unit), dist (distance unit), and time unit, respectively.

To objectively evaluate the performance of the proposed ADEPT method, three representative algorithms were selected for comparison in collaborative inspection tasks, with identical UAV numbers and task points across all comparative experiments. The selected algorithms are as follows:

1) Greedy-Target Deep Q-Network (GT-DQN) [18]: This method combines centralized task allocation with short-distance-first priority and deep reinforcement learning path planning. Each UAV learns a local path planning policy using a classical Deep Q-Network (DQN) framework. In the task allocation layer, a simple greedy strategy of "shortest distance first" is adopted, where each UAV selects the nearest uninspected task point as its target.

Bandwidth is treated as a fixed resource evenly allocated to all inspection UAVs.

2) Centralized-Task Deep Q-Network (CT-DQN) [19]: This method integrates the centralized task allocation model designed in this study with DQN-based path planning. It uses a hybrid "planning-execution" architecture: the high-level layer performs global task allocation via a centralized model, while the execution layer allows each UAV to learn local flight and obstacle avoidance strategies using DQN. In this algorithm, communication bandwidth is neither treated as a decision variable nor as a constraint during task allocation or path planning.

3) ADEPT Variant-1: To quantify the contribution of the anti-overlap mechanism in the proposed ADEPT method, a variant was designed. This variant retains all modules of ADEPT, including the joint imaging-transmission optimization model and the hierarchical importance strategy, but removes the geometric state detection used to prevent and resolve trajectory overlaps.

Figure 4 illustrates the differences in path optimization efficiency among the algorithms, measured by the average UAV travel distance. With increasing training episodes, the average travel distance of all algorithms gradually decreases, indicating that agents have learned shorter inspection paths. ADEPT achieves the fastest convergence and the shortest final average travel distance, significantly outperforming the other algorithms. Variant-1 shows a rapid initial decrease but has limited optimization in later stages, resulting in a final distance higher than ADEPT. This demonstrates that the anti-overlap mechanism not only prevents path conflicts but also indirectly shortens flight paths through more efficient coordination. GT-DQN and CT-DQN, lacking global coordination, exhibit limited path efficiency improvement and ultimately maintain significantly longer average distances, particularly stagnating in the late training phase.

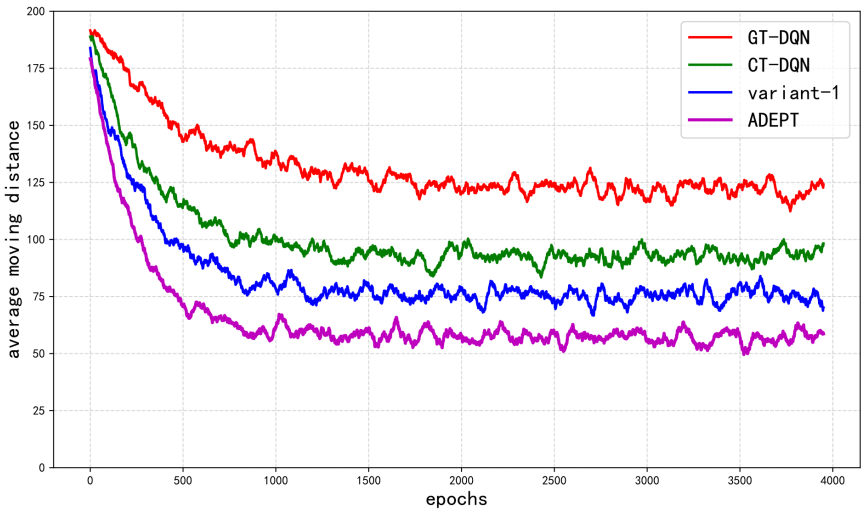

Figure 4. Comparison of average travel distance across different algorithms

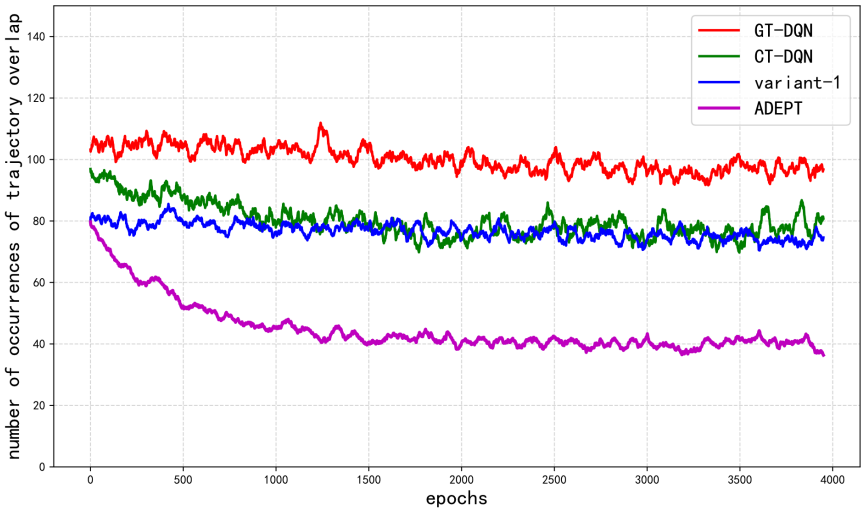

Figure 5. Comparison of the number of instances of trajectory overlap across differentalgorithms

Figure 5 compares the number of trajectory overlap occurrences during collaborative inspection. ADEPT consistently maintains the lowest overlap frequency, stabilizing in the later training stage. Variant-1, lacking the anti-overlap mechanism, exhibits significantly higher overlap counts with slower reduction. This validates that the proposed anti-overlap mechanism plays a key role in preventing and resolving multi-UAV path conflicts, effectively avoiding system deadlocks. Both CT-DQN and variant-1 adopt centralized utility-based task allocation, which allows them to progressively achieve optimal task allocation and reduce overlap frequency as training progresses. In contrast, GT-DQN maintains high overlap levels, highlighting that conventional reinforcement learning approaches struggle to resolve complex multi-UAV path conflicts without targeted mechanisms.

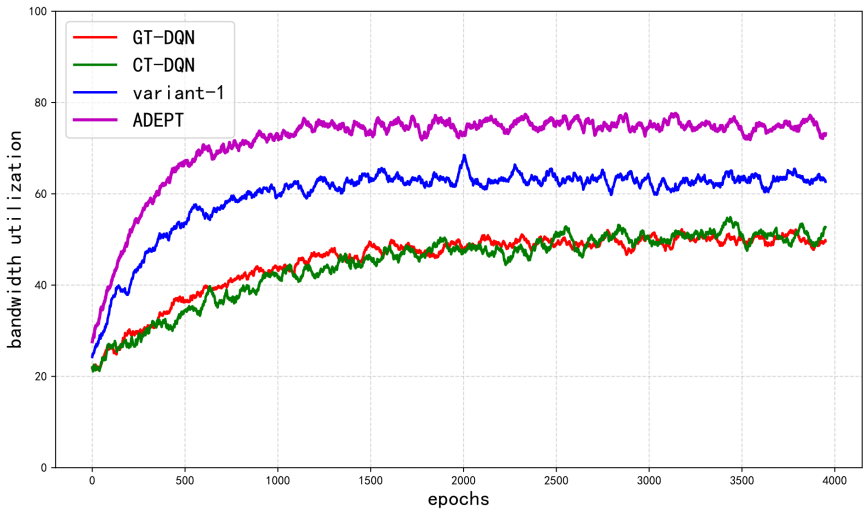

Figure 6. Comparison of bandwidth utilization among different algorithms

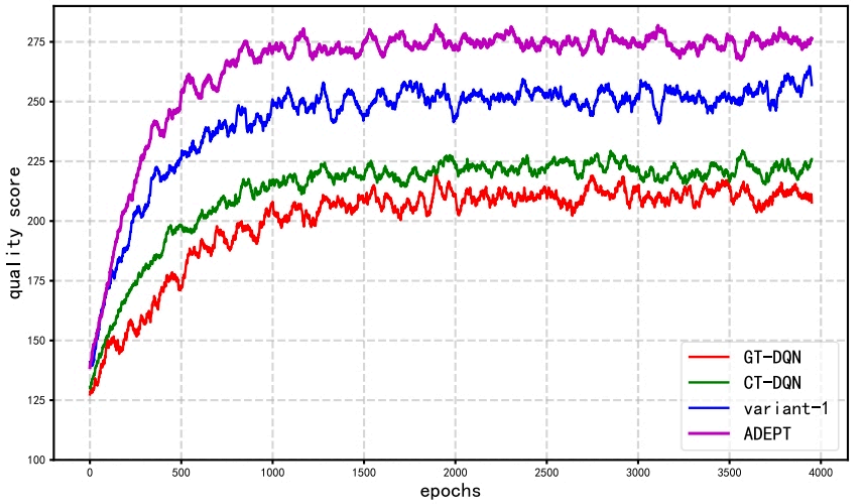

Figure 7. Comparison of overall quality across different algorithms

Figure 6 compares bandwidth utilization efficiency among the algorithms, defined as the ratio of effectively used bandwidth to total bandwidth consumed. The ADEPT algorithm quickly learns and consistently maintains high bandwidth utilization, indicating that its bandwidth-aware decision module effectively schedules communication resources, ensuring image transmission quality while avoiding idle resources. Variant-1 exhibits larger fluctuations and lower overall utilization, showing that without the anti-overlap mechanism, path conflicts and repeated captures lead to bandwidth waste. GT-DQN and CT-DQN, which completely ignore bandwidth constraints, show higher utilization curves than ADEPT and variant-1 but fail to achieve intelligent resource allocation.

Figure 7 compares inspection quality across the algorithms. ADEPT demonstrates steady improvement from the early training phase and stabilizes after approximately 2,000 episodes, significantly surpassing other algorithms. Variant-1 achieves high performance during mid-training but experiences a slight decline in later stages, reflecting the negative impact of lacking the anti-overlap mechanism on path conflicts and redundant captures, which reduces the stability and ceiling of inspection quality. GT-DQN and CT-DQN, lacking

differentiated quality control, show limited improvement, clearly underperforming ADEPT and variant-1, indicating poor adaptability of traditional RL methods in this type of multi-UAV inspection task.

Overall, the results from Figures 4–7 indicate that the proposed ADEPT algorithm, by integrating the anti-overlap mechanism with bandwidth-aware optimization, achieves significant improvements in collaborative robustness, path efficiency, resource utilization, and final inspection quality, providing a reliable technical solution for practical multi-UAV cooperative inspection applications.

7. Conclusion

This study addresses two critical challenges in multi-UAV cooperative inspection systems: trajectory overlaps and bandwidth constraints. To tackle these issues, we proposed an innovative collaborative optimization framework, ADEPT. Experimental results demonstrate that ADEPT significantly outperforms conventional methods across multiple performance dimensions. Overlap prevention: The system reduces inspection stalling time caused by path conflicts by over 30%. Path efficiency: The average flight distance of UAVs is shortened by more than 25% compared to baseline methods. Communication resource utilization: Under dynamic bandwidth constraints, the system maintains a stable utilization rate exceeding 75%. These results validate the comprehensive advantages of ADEPT in improving collaborative robustness, operational efficiency, final inspection quality, and resource effectiveness. The main contribution of this work lies in integrating proactive overlap prevention with quality assurance under bandwidth constraints into a unified optimization framework, providing a solution that combines theoretical rigor with practical engineering applicability for real-world multi-UAV deployments. Future work could further enhance the adaptability, robustness, and generalization of the algorithm by: Incorporating attention mechanisms for improved decision-making. Extending experiments to real-world UAV platforms. Integrating continuous dynamic task scenarios for more complex operational conditions. Such advancements are expected to promote the broader application of intelligent UAV swarms in industrial inspection, disaster response, urban governance, and other complex environments.

References

[1] RAHMAN M, SARKAR N I, LUTUI R. A Survey on Multi-UAV Path Planning: Classification, Algorithms, Open Research Problems, and Future Directions [J]. *Drones, 2025, 9*(4): 263.

[2] HUI ZHANG, JIAN ZHAO, JIANGSHUN YU. Application of UAV in Intelligent Patrol Inspection of Transmission Line [EB/OL]. arXiv preprint, 2023.

[3] LIU Y, WANG C, ZHAO X, et al. Reinforcement-Learning-Based Multi-UAV Cooperative Search for Moving Targets in 3D Scenarios [J]. *Drones, 2024, 8*(8): 378.

[4] HUO L Y, WANG Z L, XU M, et al. A Task-Agnostic Regularizer for Diverse Subpolicy Discovery in Hierarchical Reinforcement Learning [J]. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2023, 53*(3): 1932-1944.

[5] HOU Z M, FEI J J, DENG Y L, et al. Data-Efficient Hierarchical Reinforcement Learning for Robotic Assembly Control Applications [J]. *IEEE Transactions on Industrial Electronics, 2021, 68*(11): 11565-11575.

[6] EKECHI C C, ELFOULY T, ALOUANI A, et al. A Survey on UAV Control with Multi-Agent Reinforcement Learning [J]. *Drones, 2025, 9*(7): 484.

[7] BAYERLEIN H, GAFVERT M, KARAMAN S. Multi- UAV Path Planning for Wireless Data Harvesting with Deep Reinforcement Learning [J]. *IEEE Transactions on Robotics, 2020, 37*(4): 1346-1362.

[8] LI X, YAO L, LI M, et al. Reinforcement Learning Based Collaborative Path Planning for UAVs and Unmanned Vehicles [C]//Proceedings of Machine Learning Research. 2025, 278: 595-603.

[9] LIU J, ZHANG Q, WANG W, et al. Dual-Timescale Hierarchical MADDPG for Multi-UAV Cooperative Path Planning [J]. *Applied Intelligence, 2025, 55*(3): 123-145.
[10] WANG Z, YI M, LIU J, et al. Cooperative Data Collection with Multiple UAVs for Information Freshness in IoT [EB/OL]. arXiv preprint, 2023.
[11] KIM M, LEE H, HWANG S, et al. Cooperative Multi-Agent Deep Reinforcement Learning Methods for UAV-Aided Mobile Edge Computing Networks [EB/OL]. arXiv preprint, 2024.
[12] XU L, WANG H, MING J. Cooperative Path Planning Optimization for Multiple UAVs Considering Communication Constraints with PSO [J]. Knowledge-Based Systems, 2023, 249: 110164.
[13] WANG J, TIAN G, WU Y. Multi-UAV Cooperative Path Planning via Graph Neural Networks and Proximal Policy Optimization [J]. *IEEE Transactions on Neural Networks and Learning Systems, 2024, 35*(2): 345-356.
[14] SHEN G, LEI L, ZHANG X, et al. Multi-UAV Cooperative Search Based on Reinforcement Learning With a Digital Twin Driven Training Framework [J]. *IEEE Transactions on Vehicular Technology, 2023, 72*(5): 5678-5692.
[15] ZHOU Y, ZHANG X, LI Z, et al. Multi-UAV Cooperative 3D Path Planning with Communication Constraints: A Deep Reinforcement Learning Approach [J]. *IEEE Transactions on Vehicular Technology, 2024, 73*(5): 6821-6836.
[16] WESTHEIDER J, RÜCKIN J, POPOVIĆ M. Multi-UAV Adaptive Path Planning Using Deep Reinforcement Learning [J]. IEEE Robotics and Automation Letters, 2023, 8(4): 2120-2127.
[17] SHEN G, LEI L, ZHANG X, et al. Multi-UAV Cooperative Search Based on Reinforcement Learning with a Digital Twin Driven Training Framework [J]. *IEEE Transactions on Vehicular Technology, 2023, 72*(5): 5678-5692.
[18] XU Y, WEI Y, JIANG K, et al. Multi-UAV Dynamic Path Planning with Dueling Deep Q-Network under Communication Denial [J]. *Mathematics, 2023, 11*(2): 405.
[19] CUI J, LIU Y, NALLANATHAN A. Multi-Agent Reinforcement Learning Based Resource Allocation for UAV Networks [J]. *IEEE Transactions on Wireless Communications, 2019, 18*(4): 2045-2059.