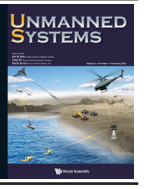
Unmanned Systems, Vol. 14, No. 2 (2026) 461–479

DOI: 10.1142/S2301385026500111

UNMANNED SYSTEMS
https://www.worldscientific.com/worldscinet/us



Action Correction-Enhanced Multi-Agent Reinforcement Learning for Path Planning in Urban Environments

Haixia Pan [*,‡], Linfeng Han [*,§], Jiaming Yan [†,¶], Ruijun Liu [*,‖]

[*]*School of Software, Beihang University, Beijing 100191, P. R. China*

[†]*Nanjing Research Institute of Electronic Engineering, Nanjing 210000, P. R. China*

In urban environments, the path planning (PP) of unmanned aerial vehicles (UAVs) presents significant challenges, particularly since they are tasked with executing various operations in crowded areas. This scenario can be framed as a Multiple Traveling Salesman Problem (MTSP), where multiple drones must efficiently visit a set of target locations while ensuring safety and collision avoidance. The high density of obstacles, such as buildings, trees, and other aerial vehicles, increases the risk of collisions, making effective PP essential for operational safety. This paper proposes a two-stage PP approach to address these challenges. In the first stage, we introduce an improved Particle Swarm Optimization (PSO) algorithm for task allocation (TA), assigning each UAV a unique task queue to minimize the overall flight distance while ensuring efficient coverage of the target area. In the second stage, we employ a multi-agent reinforcement learning algorithm for PP and incorporate a safe action correction module that operates independently to adjust actions, thereby enhancing collision avoidance capabilities. Experimental results demonstrate that our approach reduces the probability of collisions with obstacles by 9% compared to the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm while also increasing the success rate of drone mission execution by 10%. This validates the effectiveness of our strategies for safe and efficient multi-drone operations in urban environments.

Keywords: Path planning; multi-agent reinforcement learning; collision avoidance.

US

1. Introduction

In recent years, the advancement and deployment of UAVs [1] have gained substantial attention due to their versatility across a wide range of applications [2, 3]. Efficient close-loop [4–8] path planning (PP) for multi-UAV systems is essential, as it enables these UAVs to perform complex tasks [9–11] in a coordinated manner, improving mission success and resource utilization [12]. Multi-UAV PP applications in urban environments are broad, covering inspection tasks [13–18] delivery [19–21], exploration missions [22–24], field survey [25–28], disaster response [29–32], and target searching [33–35]. For instance, in multi-UAV goods or food delivery, coordinated UAVs enable efficient and timely distribution across multiple locations. In disaster scenarios, UAVs can rapidly assess damage [32] and deliver aid to disaster zone [36]. Additionally, in clinical emergencies, optimized PP [37] allows UAVs to quickly transport medical supplies [19–21] in 3D space, significantly reducing delivery time from supply points to patients [38] while ensuring safe low-altitude flight operations.

The diverse applications [39, 40] of drones in urban environments [41–44] highlight the critical need for robust multi-UAV PP systems in 3D. Although 2D system [41, 45–51] has been well explored, expanding 2D solution [52] into multi-agent 3D navigation [31, 53, 54] is still relatively challenging. Consequently, this study focuses on solving the PP problem for multiple UAVs operating in low-altitude

Received 23 November 2024; Accepted 21 January 2025; Published 18 March 2025. This paper was recommended for publication in its revised form by editorial board member, Shenghai Yuan.
Email Addresses: [‡]*haixiapan@buaa.edu.cn*, [§]*zy2221110@buaa.edu.cn*, [¶]*benjaminyan2494@163.com*, [‖]*liuruijun@buaa.edu.cn*
[‡]Corresponding author.

urban areas [3, 55], where effective coordination is essential to fulfill complex mission requirements [44].

In the delivery task scenario illustrated in Fig. 1, several key operational constraints are considered to ensure successful and safe flight missions. First, multiple UAVs are deployed from different starting locations, each assigned to reach a unique set of target points, with only one drone assigned per target to avoid redundancy. Second, all designated target points within the environment must be visited efficiently to optimize mission coverage. Third, UAVs are required to navigate through the urban landscape while actively avoiding collisions with obstacles [56, 57], such as buildings [58], and other structures [59] as well as maintaining safe distances from fellow drones in shared airspace.

Various methodologies have been developed for this problem, primarily categorized into Node-Based Optimal Algorithms [60], Sampling-Based Algorithms, Bio-Inspired Algorithms, and Reinforcement Learning Methods [61]. Node-Based Optimal Algorithms, such as Dijkstra's Algorithm [62] and A-star Search Algorithm [63], focus on finding the shortest paths by evaluating discrete nodes, making them suitable for static environments. In contrast, Sampling-Based Algorithms like Rapidly-exploring Random Trees (RRT) [64] and Probabilistic Roadmaps (PRM) [65] utilize probabilistic sampling to navigate complex, high-dimensional spaces, enabling efficient pathfinding in dynamic urban landscapes. Bio-inspired algorithms, including Genetic Algorithms (GAs) [66] and Ant Colony Optimization (ACO) [67], draw inspiration from natural processes, effectively solving optimization problems through mechanisms like natural selection and collective behavior. Reinforcement Learning Methods, such as Deep Q-Networks (DQN) [68], and Proximal Policy Optimization (PPO) [69] enable drones to learn optimal PP strategies through interactions with their environment, adapting to dynamic conditions and improving decision-making over time.

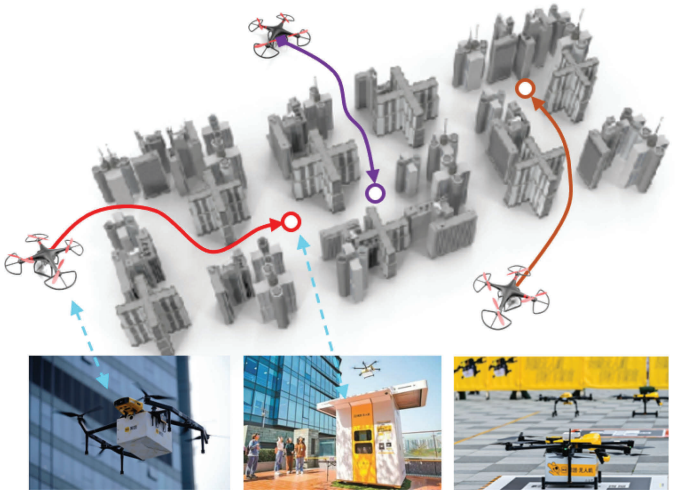


Fig. 1. This image illustrates the PP application of the UAV swarm in the logistics.

Sampling-Based and Node-Based Optimal Algorithms [24, 70, 71] often suffer from high computational complexity in complex scenarios, making them less suitable for real-time applications. Biologically inspired methods, however, face challenges in obstacle avoidance and performance instability in dynamic urban environments due to their reliance on stochastic processes. Additionally, in environments with numerous obstacles, these methods may converge to suboptimal solutions, adversely affecting the efficiency and reliability of drone operations. The effectiveness of reinforcement learning algorithms in obstacle avoidance and task allocation (TA) [72] highly depends on the design of the reward function. There is no mechanism to ensure safety by itself, and the path length optimization also depends on the reward function. A reasonable reward function is crucial, and improper design may lead to poor learning results. Existing methods often exhibit high computational complexity in complex scenes [44], which restricts their use in real-time tasks. Additionally, changes in obstacles within dynamic environments pose a safety risk for UAVs.

Therefore, to effectively implement multi-drone PP, this paper divides the entire workflow into two main phases: TA and PP. In the TA phase, we begin by identifying and defining the specific tasks that need to be accomplished by the drones. This process involves analyzing environmental conditions to ensure that each drone is allocated to the most suitable task. By optimizing TA, we can enhance the overall efficiency and task completion rate of the system. In the PP phase, we design optimal flight paths for each drone based on the assigned tasks. This process requires consideration of the interrelationships among the drones, avoidance of potential collision risks, and minimization of flight time.

To avoid collisions, we design a safety action correction module for multi-agent systems that operates independently of the multi-agent reinforcement learning algorithm. This module can be trained using customized safety constraint functions based on different task scenarios and can be applied after the policy network of any multi-agent reinforcement learning algorithm to correct the outputs of the policy layer. Specifically, our safety action correction module enables coordination between agents. This functionality ensures that different UAVs can effectively avoid potential collision risks while executing tasks and achieve safe and efficient collaboration in dynamic environments.

In particular, our work entailed as follows:

- We propose a two-phase multi-agent intelligent framework to achieve both safety and efficiency in PP.
- We design a safety action correction module independent of the multi-agent reinforcement learning algorithm.
- Our safety action correction module enables coordination among UAVs to ensure efficient collaboration.
- We will open-source our proposed methods and simulation environment to contribute to the community and foster advancements in UAV PP and urban navigation research at https://github.com/ruifeng24/GACM-MARL.

2. Related Work

Traditional UAV PP methods usually rely on basic algorithms and heuristic rules. These include evolutionary algorithms, simulated annealing algorithms, shortest path algorithms, like Dijkstra's algorithm and A-star algorithm. Bai *et al.* [73] combined the A-star algorithm and artificial potential field method to realize the PP of UAV. However, the A-star algorithm [74] presents challenges for multi-UAV PP due to its single-agent characteristics, reliance on global environmental information, and focus on finding a single optimal path, rather than considering system-level global optimization objectives [74].

As deep reinforcement learning has advanced [75–82], an increasing number of researchers have started to investigate its potential for use in UAV PP. The multi-layer DQN algorithm proposed by Cui and Wang [83] uniquely combines global and local information, significantly enhancing the overall performance. In this approach, the bottom layer manages the long-term strategies based on the global information, while the upper layer focuses on the short-term tactics using the local data [84–98] to avoid collisions [99–106]. The end-to-end deep neural network [107, 108] enables obstacle avoidance in mobile robots by mapping depth images to control commands, using convolutional layers for feature extraction and fully connected layers to optimize action selection for safe navigation. The Bnd$*$-DDQN algorithm [109] combines Bnd$*$ for long-term PP with DDQN for short-term control, enabling efficient and adaptive autonomous steering in complex environments. In order to design three-dimensional paths, Xie *et al.* [110] provided a deep reinforcement learning method that uses relative distance and local information instead of global information. The TDPP-Net [111] enables efficient 3D PP for autonomous navigation, reducing computational time while maintaining path quality through action decomposition and composition. Wu *et al.* [112] proposed OTDPP-Net, a DNN-based architecture for real-time 3D PP in unknown environments, achieving near-optimal navigation with enhanced path quality via recurrent value iterations and line-of-sight checks. The recent DRL model [113] integrates a novel feature extractor, a double-source experience scheme, and a dual-soft-actor-critic setup to enhance learning efficiency and performance, surpassing conventional and state-of-the-art methods in success rate, trajectory quality, and generalization.

However, for complex tasks, multiple UAVs can perform flight tasks in parallel to improve efficiency. The use of multiple UAVs can more flexibly adapt to the changing environment. When one agent encounters obstacles or new situations, other agents can adjust their policies to support the overall goal, improving the robustness and adaptability of the system. Luo *et al.* [114] studied the problem of edge computing enabling multiple UAVs to cooperate in target search, and developed a deep reinforcement learning framework based on DQN to effectively optimize mission unloading decision and flight direction selection, minimizing the uncertainty of the search area. Liu *et al.* [115] propose a new multi-UAV trajectory design framework based on multi-agent Q-learning. Yan *et al.* [116] use the dueling double deep Q-Networks (D3QN) method to take a set of observations as inputs to approximate the Q values corresponding to all candidate operations, and the proposed algorithm can achieve higher cumulative rewards and success rates compared to the DQN algorithm. Hong *et al.* [117] extended the TD3 model to the continuous action space of the UAV. They trained the modified TD3 model through Offline reinforcement learning to reduce the training overhead of the reinforcement learning model [118–121]. Especially, the improved TD3 model can select the energy-saving path in real-time. The multi-agent deep deterministic policy gradient (MADDPG) [122] algorithm is a state-of-the-art MARL algorithm that performs well in multi-agent collaborative, competitive or mixed environments. Many researchers have implemented multi-agent PP [123] based on MADDPG. Qie *et al.* [12] propose an AI approach based on the MADDPG algorithm for simultaneous target assignment and path planning (STAPP). The MADDPG framework is used to train the system according to the corresponding reward structure to solve both goal assignment and PP problems. The proposed system can effectively handle dynamic environments because its execution requires only the location of the UAV, the target, and the threat area. The system can solve the problem of target assignment and PP at the same time, and deal with the dynamic environment due to the change of environment in training. Wang *et al.* [124] proposed a trajectory control algorithm based on MADDPG to independently manage the trajectory of each drone.

Nevertheless, numerous challenges arise in real-world scenarios. Many new solutions have been proposed to keep

drone clusters from colliding with obstacles. Model predictive control (MPC) can handle several restrictions and cost functions simultaneously through optimization [125]. Zhang *et al.* [126] realized the PP in a 3D environment and used the hierarchical PP algorithm as the global path planner to calculate the near-optimal path composed of a group of path points. And design a local path planner that uses nonlinear MPC to track the path calculated by the global path planner while satisfying multiple constraints of dynamics and environment. Alrifaee *et al.* [127] proposed to apply MPC to vehicle collision avoidance. The vehicle model adopts the Taylor linearization method, linearizes around the steady-state operating point, takes vehicle collision avoidance as a constraint condition, and expresses it as a quadratic function. Finally, the vehicle is controlled by optimizing MPC. Another way is to incorporate safety constraints into the reinforcement learning algorithm. Cai *et al.* [128] proposed a combination of distributed control barrier functions (CBF) and the MADDPG algorithm to achieve collision avoidance in multi-agent cooperation scenarios to a certain extent. CBF is integrated into every agent and can ensure the security of the system by forcing the invariance of the security set. Gu *et al.* [129] constructed a theoretical framework based on the Constrained Markov Decision Process (CMDP) model, incorporating a trust region approach to ensure that each iteration incrementally enhances the reward function while adhering to the specified safety constraints. Sheebaelhamd *et al.* [130] enhanced the well-established MADDPG framework by incorporating a safety layer, which employs soft constraints, into the deep policy network. Recent approaches made use of attention mechanism [131–133] for long horizon PP [134, 135], but often for simple low-dimensional space environments.

3. Preliminaries

3.1. *Reinforcement learning*

In the field of artificial intelligence, a reinforcement-learning agent is engineered to make a series of decisions while actively engaging with its environment. This environment is often conceptualized as an ongoing, long-term discounted Markov decision process (MDP), providing the agent with a framework for learning and adapting its decision-making strategies over time. The MDP is defined by $\langle S, A, P, R, \gamma \rangle$, where S and A represent the state and action spaces. P denotes the transition probability from any state s_t to s_t' for any given a_t. R refers to the reward obtained by taking action a_t at state s_t to reach state s_t'. γ is the discount factor that trades off the instantaneous and future rewards.

Multi-agent reinforcement learning is a branch of reinforcement learning that involves multiple agents interacting, collaborating, or competing in a shared environment to reach individual or collective goals. MARL aims to enable multiple agents to autonomously learn and make decisions to achieve task goals in complex and dynamic environments. The multi-agent environment is modeled by a Markov game [136], which is defined as

$$\langle N, S, {A^i}_{i\in 1,\ldots,N}, P, {R^i}_{i\in 1,\ldots,N}, \gamma \rangle.$$

- N represents the number of agents.
- S represents the environment shared by all agents.
- A^i represents the set of actions that the ith agent can take.
- P represents the probability of transitioning from state s to s' when taking action A.
- R^i represents the reward received by the ith agent for taking action a in state s.
- γ represents the discount factor.

3.2. *Safe Multi-agent reinforcement learning*

Safe Reinforcement Learning can be characterized as the procedure of acquiring policies that maximize the expected return in scenarios where it is crucial to guarantee adequate system performance and adhere to safety restrictions throughout the learning and implementation processes [137]. This approach is particularly useful when the learning process poses a potential risk to the system or environment or when the system is required to operate within certain safety limitations.

The Safe MARL problem is typically defined as a multi-agent Constrained Markov Game (MA-CMG) [129]. MA-CMG can be considered as the tuple $\langle N, S, {A^i}_{i\in 1,\ldots,N}, P, {R^i}_{i\in 1,\ldots,N}, \gamma, C \rangle$:

$$C = \left\{ C_j^i \right\}_{1\le j\le m_i}^{i\in N}. \tag{1}$$

- C represents the constraint functions.
- m_i represents the number of constraint functions for agent i.

The agents' goal is to maximize the joint return, given by

$$J(\pi) = E_{s_0\sim\rho, a_0\sim\pi, s_1\sim P}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right]. \tag{2}$$

The equation delineates the expected return, denoted by $J(\pi)$, that is anticipated when following a policy π in a MDP. Here, E signifies the expectation operator, which averages over all possible occurrences of random variables. The sequence $s_0, a_0, s_1, \ldots$ represents the trajectory generated by interacting with the environment under policy π, where s_t denotes the state at time step t, and a_t represents the action taken at that same time step. The initial state s_0 is sampled from the distribution ρ, typically representing the start state distribution of the MDP. Actions a_t are chosen according to the policy π, while the next state s_{t+1} is determined by the

transition probability P of the environment given the current state-action pair. The summation inside the expectation runs over an infinite horizon, with each reward $R(s_t, a_t)$ discounted by a factor of γ^t, where γ is the discount factor ensuring convergence of the series for $\gamma \in [0, 1)$. This formulation encapsulates the long-term value of a policy, balancing immediate rewards against potential future gains.

Meanwhile, it obeys the safety constraints

$$J_j^i(\pi) = E_{s_0 \sim \rho, a_0 \sim \pi, s_1 \sim P}\left[\sum_{t=0}^{\infty} \gamma^t C_j^i(s_t, a_t)\right]. \tag{3}$$

The function $C_j^i(s_t, a_t)$ represents a safety constraint evaluating whether taking action a_t in state s_t complies with the ith safety criterion concerning the jth aspect of the system, ensuring responsible and bounded behavior in decision-making processes.

4. Proposed Solution

Our main objective is to achieve multi-UAV PP in this 3D urban environment while ensuring the safety of the path to a certain extent.

4.1. *Environment*

The city comprises diverse buildings that differ in size, height, and geometric configuration. Each of these buildings is uniquely identified by B_i, where i is the index of the ith building, $i \in 1, 2, \ldots, N_b$, with N_b representing the total number of buildings in the area. Diverse buildings not only make the scene more realistic but also bring challenges to the agent's PP and obstacle avoidance.

We used the Unity engine to build a 3D model of the city environment and developed custom scripts to programmatically control the agent's behavior and reward system. Furthermore, by implementing the Remote Procedure Call (RPC) mechanism, the seamless communication between the reinforcement learning algorithm and the Unity environment was realized, so as to effectively convert the decision output of the learning algorithm into the dynamic movement and interaction actions of the agent in the virtual city environment.

4.1.1. *Representations*

State estimation plays a vital role in the autonomous navigation of drones, as it provides essential observational feedback to the system, ensuring accurate and reliable operation. The observation space, representing the information available to each UAV i at time t, is denoted by the state vector s_i^t, which encapsulates the UAVs properties, environmental perceptions, and inter-agent spatial relationships. Formally, this state is defined as $s_i^t = [p_i^t, v_i^t, R_i^t, d_{io_1}^t, \ldots, d_{io_m}^t, d_{ig}^t, d_{it_1}^t, \ldots, d_{it_k}^t]$, where $p_i^t \in \mathbb{R}^3$ is the position, $v_i^t \in \mathbb{R}^3$ is the velocity, $R_i^t \in \mathbb{R}^3$ is the angle, $d_{io_*}^t \in \mathbb{R}^3$ are the relative positions to obstacles, $d_{ig}^t \in \mathbb{R}^3$ is the relative position to the goal, and $d_{it_*}^t \in \mathbb{R}^3$ are the relative positions to other teammates. The relative positions of UAV i and other observation j can be calculated from Eq. (4). This comprehensive state representation provides each UAV with a nuanced understanding of its dynamic environment, which is crucial for informed decision-making:

$$d_{ij} = \left[\frac{x_i - x_j}{||P_i - P_j||_2}, \frac{y_i - y_j}{||P_i - P_j||_2}, \frac{z_i - z_j}{||P_i - P_j||_2}\right]. \tag{4}$$

Building upon this comprehensive observation space, the action space is defined to delineate the range of control inputs available to each UAV, enabling them to interact with and navigate through their environment. Specifically, for UAV i at time t, the actions comprise two distinct components: the angular velocity, denoted by $\omega_i^t \in \mathbb{R}$, which controls the rotational movement of the UAV, and the acceleration vector, denoted by $a_i^t \in \mathbb{R}^3$, which governs the translational motion of the UAV in three-dimensional space.

The interaction between the observation and action spaces is governed by the propagation model, which dictates the temporal evolution of each UAVs state. By representing the UAVs pose using the Special Euclidean group $\mathrm{SE}(3)$, we denote the pose at time t as follows:

$$T_i^t = \begin{bmatrix} R_i^t & p_i^t \\ 0^T & 1 \end{bmatrix}, \tag{5}$$

where $R_i^t \in SO(3)$ is the rotation matrix and $p_i^t \in \mathbb{R}^3$ is the position vector. The orientation update is determined by the angular velocity ω_i^t through the exponential map on $SO(3)$, yielding as follows:

$$R_i^{t+1} = R_i^t \exp(\hat{\omega}_i^t \Delta t), \tag{6}$$

where $\hat{\omega}_i^t$ is the skew-symmetric matrix of ω_i^t and Δt is the time step. Velocity v_i^t is updated using the first-order Euler integration as follows:

$$v_i^{t+1} = v_i^t + R_i^t a_i^t \Delta t \tag{7}$$

reflecting the change due to acceleration. Finally, Position p_i^t is updated based on the new velocity, resulting in the following equation:

$$p_i^{t+1} = p_i^t + v_i^{t+1} \Delta t. \tag{8}$$

This unified formulation provides a coherent and mathematically rigorous framework for multi-UAV PP, enabling the agents to learn effective policies for navigating complex environments through the iterative process of reinforcement learning.

4.1.2. *Rewards*

In the PP task of a multi-UAV system in a city environment, designing a reasonable reward mechanism is crucial to ensure the UAV can learn a policy to navigate to the goal point safely and efficiently. In order for the UAV to perform better in this complex environment, multiple rewards need to be designed. We construct an aggregate reward function R.

$$R = w_a \cdot f_a(t) + r_{\text{arr}} + r_c + r_b, \tag{9}$$

$$f_a(t) = ||P_u(t), P_g(t)||_2 - ||P_u(t-1) - P_g(t-1)||_2, \tag{10}$$

- $f_a(t)$ represents the reward for approaching the goal at time t.
- r_{arr} represents the reward for arriving at the target at time t.
- r_c represents the reward for colliding the obstacles and other UAVs at time t.
- r_b represents the reward beyond the scene boundary at time t.
- w_a represents the weight coefficient.

Based on several experiments, the optimal configurations of the hyperparameters in the reward function are shown in Table 1.

4.1.3. *Safety constraints*

To ensure that UAVs do not collide with obstacles or each other during flight, we construct a comprehensive mathematical model that clearly defines the trajectories and their associated constraints. First, let N denote the number of UAVs. The motion trajectory of each drone can be expressed as a time-parameterized function as follows:

$$\begin{aligned}\mathbf{r}_i(t) = (x_i(t), y_i(t), z_i(t)), \quad \forall i \in \{1, 2, \ldots, N\}, \\ \times\, t \in [t_0, t_f].\end{aligned} \tag{11}$$

In this equation, $\mathbf{r}_i(t)$ denotes the three-dimensional position of the UAV $\mathbf{u}_i$ at time t, where $x_i(t)$, $y_i(t)$, and $z_i(t)$ represent the drone's position along the x-, y-, and z-axes, respectively. Next, to ensure the safe flight of the drones, we need to define the trajectory set $\mathcal{T}_i$, which contains all possible states of the drones within a specified time range, expressed as

$$\mathcal{T}_i = \{\mathbf{r}_i(t) : t \in [t_0, t_f]\} \quad \forall i \in \{1, 2, \ldots, N\}. \tag{12}$$

Table 1. Reward setting.

Symbol	Value	Remark
ω_a	1	The weight of $f_a(t)$
r_{arr}	20	Reward for reaching the goal
r_c	−5	Reward for collision
r_b	−1	Reward for beyond boundaries

In the PP process, we need to consider two main types of constraints to ensure the safe operation of the drones. The first is the obstacle constraint. We define the obstacle region O mathematically as

$$O = \{(x, y, z) \in \mathbb{R}^3 : g(x, y, z) \leq 0\}, \tag{13}$$

where $g(x, y, z)$ is a function that describes the boundaries of the obstacle. To ensure that the trajectories of the drones do not intersect with the obstacle region O, we require the following conditions to be satisfied:

$$\mathcal{T}_i \cap O = \emptyset, \quad \forall i \in \{1, 2, \ldots, N\}. \tag{14}$$

This condition signifies that the trajectory set of the drones does not overlap with the obstacle region, thereby ensuring that the drones will not collide with obstacles during flight. The second important constraint is related to inter-UAV collision avoidance. In a multi-UAV system, it is crucial to ensure that different drones do not collide with each other at the same time. This can be expressed mathematically as follows:

$$\begin{aligned}&\forall t \in [t_0, t_f], \\ &\mathcal{T}_i(t) \cap \mathcal{T}_j(t) = \emptyset, \\ &\forall i, j \in \{1, 2, \ldots, N\}, \quad i \neq j.\end{aligned} \tag{15}$$

Here, $\mathcal{T}_i(t)$ and $\mathcal{T}_j(t)$ represent the position of drones $\mathbf{u}_i$ and $\mathbf{u}_j$ at time t, respectively. This notation states that the positions of any two drones cannot be the same at any given time t, preventing collisions at that specific moment.

Based on these constraints, we can construct a safe trajectory set $\mathcal{T}_{\text{safe}}$, ensuring that all drones execute their tasks without intersecting with obstacles or the trajectories of other drones. Specifically, the safe trajectory set can be defined as

$$\begin{aligned}\mathcal{T}_{\text{safe}} = \Big\{ &\mathbf{r}_i(t) : \mathcal{T}_i \cap O = \emptyset \\ &\text{and } \forall t \in [t_0, t_f],\ \mathcal{T}_i(t) \cap \mathcal{T}_j(t) = \emptyset \\ &\text{for all } i, j \in \{1, 2, \ldots, N\},\ i \neq j \Big\}.\end{aligned} \tag{16}$$

4.2. *Multi-UAV path planning*

To effectively complete the multi-target point-reaching task of multiple UAVs in urban environments, a two-stage PP algorithm is proposed in this study (see Fig. 2). The first stage of the algorithm is devoted to solving the TA problem,

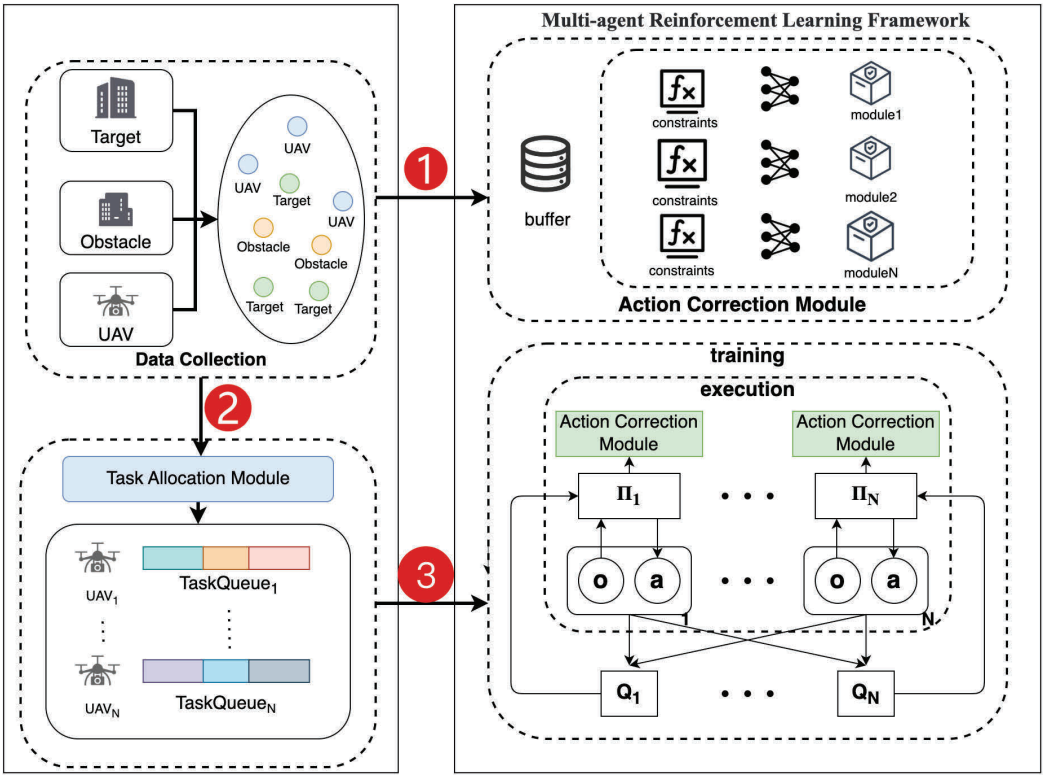


Fig. 2. This is the overall framework of UAV PP.

and the mapping relationship between each UAV and the target point is determined through our improved PSO algorithm. The second stage focuses on the safety optimization of flight paths and improves the safety and efficiency of the overall operation by integrating an advanced reinforcement learning module. In the following section, these two stages will be elaborated on separately.

4.2.1. *Task allocation*

In the first stage, we proposed an improved Particle Swarm Optimization (PSO) algorithm. In this paper, particles are defined as the task queues of all UAVs (see Fig. 3). By modeling the distances between each drone and various target points, the algorithm can dynamically adjust the target allocation strategy for the drones to ensure the minimization of the total path length for all drones.

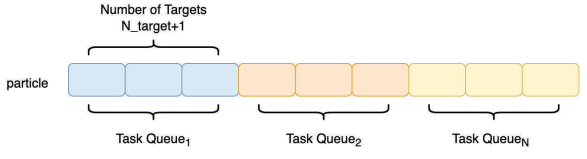


Fig. 3. This image shows the definition of particle.

To achieve this, an objective function is first constructed, which optimizes for path length while considering the starting positions of the drones and the coordinates of the target points. Through iterative updates of the particles' positions and velocities, the algorithm is able to gradually converge to the optimal solution over multiple iterations, effectively achieving an optimized matching between the drones and the target points.

Our objective is to minimize the total path length from all UAVs to their respective goals as Function 17.

$$\mathbb{L} = \sum_{i=1}^{N} \sum_{j=1}^{T} ||P_u^i - P_g^j||_2. \tag{17}$$

First, randomly generate the positions and velocities of the particles. Due to the limitations imposed by fixed initialization strategies in swarm intelligence algorithms when solving optimization problems, this paper introduces chaos mapping as a new initialization method to enhance algorithm performance. Chaos mapping can uniformly distribute the initial population in the solution space, thereby strengthening global search capabilities and increasing the likelihood of finding the optimal solution. Additionally, the generated population exhibits good diversity, effectively preventing the algorithm from falling into local optima and accelerating convergence speed. Among these, Chebyshev

chaos mapping, with its favorable mathematical properties and rapid convergence, can produce significant changes in each iteration as Function (18). Subsequent chapters will validate the effectiveness of this chaos mapping through experimental results.

$$x_{n+1} = \cos(\lambda \cdot \cos^{-1}(x_n)). \tag{18}$$

For each particle, update its personal optimal position p_i^o as Function (19) and global optimal position p_g^o as Function (20).

$$\mathbf{p}_i^o = \begin{cases} \mathbf{p}_i & \text{if } f(\mathbf{p}_i) < f(\mathbf{p}_i^o), \\ \mathbf{p}_i^o & \text{otherwise.} \end{cases} \tag{19}$$

$$\mathbf{p}_g^o = \arg\min_i f(\mathbf{p}_i^o). \tag{20}$$

Adjust the velocity and position of the particles based on the updated individual best and global best solutions as follows:

$$\mathbf{v}_i' = w \cdot \mathbf{v}_i + c_1 \cdot r_1 \cdot (\mathbf{p}_i^o - \mathbf{p}_i) + c_2 \cdot r_2 \cdot (\mathbf{p}_g^o - \mathbf{p}_i), \tag{21}$$

$$\mathbf{p}_i' = \mathbf{p}_i + \mathbf{v}_i'. \tag{22}$$

In this context, w represents the inertia weight, c_1 and c_2 are acceleration constants, while r_1 and r_2 are random numbers uniformly distributed in the interval [0, 1].

In complex search spaces, local optima often hinder the discovery of global optima. This paper introduces a Lévy flight mutation operator, which can overcome the limitations of the current solution and explore new regions of the solution space, thereby effectively avoiding convergence to local optima. In the context of multi-drone multi-objective TA, Lévy flight is an exceptionally suitable model. The characteristics of Lévy flight enable drones to exhibit superior efficiency in search and exploration. By combining long-distance jumps with short-range fine searches, it allows for rapid coverage of extensive search areas. This flexible movement pattern enables drones to effectively respond to dynamic environments and targets, thereby facilitating efficient TA and execution.

This paper introduces an adaptive step length adjustment mechanism. Specifically, the step length L can be dynamically adjusted through a feedback mechanism based on the drone's performance in the search process (the number of successful target detections), establishing an adaptive step length L_a, as indicated in the following equation:

$$L_a = L_0 \cdot (1 + \alpha \cdot P_s). \tag{23}$$

In this context, L_0 represents the initial step length, α is a positive adjustment factor, and P_s refers to the proportion of tasks successfully completed by the drone over a given period. A higher success rate indicates that the algorithm performs well in the current search area. When the success rate is high, the adaptive step length L_a increases, which aligns with the heavy-tailed characteristic of Lévy distribution; the randomness of long step lengths facilitates rapid exploration of new regions in the solution space.

The generated step length will be used to update the position of the drone as follows:

$$p_i'' = p_i' + L_a^*(p_i^o - p_i'). \tag{24}$$

4.2.2. *Generalized action correction module for MARL*

We introduce a generic Generalized Action Correction Module (GACM) that aims to enhance the safety of multi-agent reinforcement learning (MARL) algorithms for PP, shown as Fig. 4 and pseudocode 1. GACM is designed as a pluggable component that can be compatible with any multi-agent reinforcement learning algorithm, thereby improving the safety of the action policies generated during learning.

Importantly, the GACM module is not dependent on a specific MARL algorithm, and its generality allows it to be integrated into various MARL frameworks, whether value-based, policy gradient or actor-critic approaches. In this way, the GACM module significantly improves the overall safety of multi-agent systems in the face of dynamic and uncertain environments without requiring large-scale modifications to the underlying MARL algorithm. This feature not only improves the practicality of the module, but also greatly expands its application scope for multi-UAV PP and wider multi-agent systems.

Following Dalal *et al.* [138], we design the GACM for each UAV. Different from the existing safety layer designed for single agents, GACM is designed to fully consider the interaction and coordination between multi-agents. GACM introduced the security considerations of multi-agent interaction to achieve collective security control. Moreover, GACM has the ability to resolve dynamic conflicts, which can adjust the action strategy between multi-agents in real-time.

The training process of the GACM can be completely independent of the MARL algorithm. We can train a linear safety-signal model for each drone at the beginning which can approximate the immediate-constraint functions $c_i^j(s, a)$.

$$\overline{c}_i^j(s_i') = c_i^j(s_i, a_i) \approx \overline{c}_i^j(s_i) + g(s_i; w_i^j)^T a_i, \tag{25}$$

$\overline{c}_i^j(s_i)$ represents the constraint value at state s, where i represents the ith UAV and j represents the jth constraints. w_i^j represents the weights of a Neural Network. $g(s_i; w_i^j)$ takes s as input and outputs a vector of the same dimension as a_i. This linear safety-signal model can approximate the immediate constraint value at state s after taking action a.

To train this model, we generate $\mathcal{D} = \{s_i, a_i, s_i', c_i^j\}$ by initializing agents with a random state and choosing actions randomly. By using the generated dataset, we can train with

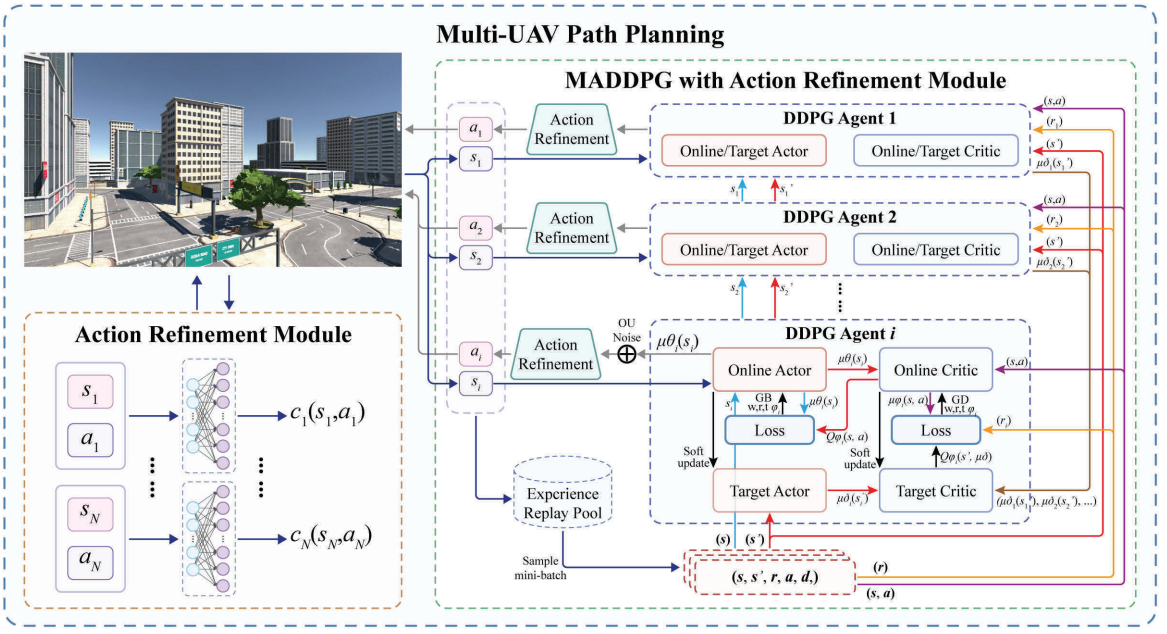


Fig. 4. This is the overall framework of Multi-agent Reinforcement Learning. We start by training an action correction module that outputs a constraint value based on state s and action a. Second, this action correction layer is used to correct the action output by the actor-network. The critic network is trained to judge the quality of the decision based on the input s and a.

Algorithm 1. Generalized Action Correction Module for Multi-Agent Reinforcement Learning (GACM-MARL)

Initialize critic networks $Q(s, a_1, \ldots, a_N|\theta^Q)$, actor networks $\mu(s|\theta^\mu)$ for each agent
Initialize target networks Q', μ' with weights equal to the original networks
Initialize cost network $g(s; w)$ for each agent with random weights
Initialize replay buffers R for MADDPG and R_c for GACM
Phase 1: Train GACM Independently
for episode $= 1$ to E **do**
    Receive initial observation state s
    for $t = 1$ to T **do**
        Select action a randomly
        Execute action a and observe new state s' and cost c
        Store (s, a, s', c) in safety replay buffer R_c
        $s \leftarrow s'$
    end for
    Randomly sample a batch of transitions from R_c
    Update C by minimizing the cost prediction error
end for
Phase 2: Integrate GACM with MADDPG
for episode $= 1$ to M **do**
    Initialize a random process N for action exploration
    Receive initial observation state s
    for $t = 1$ to T **do**
        for each agent i **do**
            Select action $a_i = \mu(s|\theta_i^\mu) + N_t$
            project a_i to the safety set by solving
            $L_i(a_i, \lambda) = \frac{1}{2}\|a_i - \mu_i(s_i)\|^2 + \sum_{i=1}^{K} \lambda_i(\bar{c}_i^j(s_i) + g(s_i; w_i^j)^T a_i - C_i^j)$
        end for
        Execute actions $a = (a_1, \ldots, a_N)$, observe reward r and new state s'
        Store (s, a, r, s') in replay buffer R
        $s \leftarrow s'$
        Proceed with MADDPG training steps (updating Q, μ, and target networks)
    end for
end for

the following loss function:

$$L(w_i^j) = \sum_{(s_i,a_i,s_i')\in\mathcal{D}} (c_i^j(s') - (c_i^j(s) + g(s_i;w_i^j)^T a_i))^2. \quad (26)$$

With the linear safety-signal model, we can augment the policy networks of MARL by adding an additional action Correction module. When choosing a reinforcement learning algorithm for multi-agent systems, MADDPG is an ideal choice because of its characteristics of centralized training and distributed execution. The algorithm can effectively deal with continuous action space and realize complex policy learning in a multi-agent interaction environment through its Actor-Critic architecture. In addition, MADDPG is designed to consider the interaction between agents, which shows excellent learning effects and adaptability in multi-agent cooperative and competitive tasks. Therefore, MADDPG is an attractive algorithm choice for application scenarios that require a high degree of collaboration and strategic interaction between agents.

This framework is composed of an actor-network that approximates the policy and a critic network that estimates the value function. The actor-network outputs' actions are based on the current state, while the Critic network evaluates the quality of these actions. The overall objective of MADDPG is to enable agents to learn cooperative behavior in complex environments where the optimal action for one agent depends on the actions of other agents. By utilizing this approach, agents can learn to coordinate their actions and achieve higher rewards, leading to improved performance in multi-agent systems.

The actor network's objective is to maximize the expected cumulative reward. The optimization objective can be formulated as follows:

$$\nabla_{\theta_i} J(\mu_i) = \mathbb{E}_{s,a\sim D}\Big[\nabla_{\theta_i}\mu_i(a_i|o_i) \times \nabla_{a_i} Q_i^{\mu}(s,a_1,\ldots,a_N)|_{a_i=\mu_i(o_i)}\Big], \quad (27)$$

where θ_i represents the actor-network parameters, μ_i represents the continuous policies, s is the observation of the UAV, the experience replay buffer D contains the tuples $(s, s', a_1, \ldots, a_N, r_1, \ldots, r_N)$, Q_i^{μ} represents the critic network.

The Critic network's goal is to evaluate the actions selected by the actor-network. The optimization objective can be expressed as follows:

$$\mathcal{L}_{(\theta_i)} = \mathbb{E}_{s,a,r,s'}[Q_i^{\mu}(s,a_1,\ldots,a_N) - y]^2,$$
$$y = r_i + \gamma Q_i^{\mu'}(s',a_1',\ldots,a_N')\Big|_{a_j'=\mu_j'(o_j)}, \quad (28)$$

where μ represents the critic-network with delayed parameters θ_i', $Q_i^{\mu}(s,a_1,\ldots,a_N)$ denotes the Critic value function for state s and action a, and y is the target value function calculated through the Bellman equation.

We can then add the GACM after μ_i to refine action, which enhances safety by solving

$$L_i(a_i,\lambda) = \frac{1}{2}||a_i - \mu_i(s_i)||^2 + \sum_{i=1}^{K}\lambda_i(\overset{j}{c}_i(s_i) + g(s_i;w_i^j)^T a_i - C_i^j), \quad (29)$$

where K represents the number of constraints and $\mu_i(s_i)$ represents the actor network of the ith UAV. Following Dalal [138], we can solve the above formula to obtain the correction function of the action. Incorporating an action correction layer into the policy network of each UAV can significantly mitigate the occurrence of perilous actions. The effectiveness of the correction layer can be verified in the subsequent experiments.

5. Experiment

5.1. *Setting*

In order to simulate different situations that UAVs may face in real life, we have built an urban environment that consists of multiple scenarios. We construct a highly simulated urban scene on the simulation platform, in which the layout, location and height of urban buildings accurately simulate the characteristics of the real world urban skyline, and effectively reproduce the complexity of urban airspace. To simulate the diversity of real urban environments, we have randomly distributed the buildings in terms of their height and location.

In this experiment, a total of three UAVs are involved in the PP task. Each UAV has the same size and needs to pass through the gaps and voids in the urban environment. We set a maximum flight speed for the UAV to ensure that the UAV can maintain a balance between fast response and collision avoidance. We set three target points in the city. The randomness of the target points ensures true mission variability in the flight environment.

The experiments were conducted on a high-performance computing setup featuring an NVIDIA GeForce RTX 2080 Ti GPU, complemented by 32 GB of RAM. We use Python 3.7, leveraging the PyTorch 1.7.1 framework for its efficient and flexible neural network capabilities.

Specifically, we designed and developed a city environment using Unity version 2022.3.8f1c1. This platform allowed us to create detailed urban scenarios with various interactive elements, thereby providing a dynamic setting for multi-agent interactions [139].

Each agent within the MADDPG framework was equipped with an actor network and a critic network. Both networks comprised three fully connected layers with hidden layer sizes of [64], utilizing ReLU activation functions for nonlinearity. The actor network output was subjected to a Tanh activation function, while the Critic network maintained linear activations for direct reward prediction.

The Adam optimization algorithm was utilized for updating the network weights, with learning rates set at 1×10^{-4} for the actor and 1×10^{-3} for the critic, respectively. A weight decay of 0.0001 was applied to promote generalization and prevent overfitting.

Each training episode consisted of 500 time steps, and a total of 10,000 episodes were executed to ensure thorough learning. At intervals of every 50 episodes, we conduct 20 validations to assess the model's efficacy.

5.2. *Results*

5.2.1. *Multi-UAV task allocation*

In the research of multi-UAV TA, the main objective of this paper is to find a TA result that minimizes the total flight path length of all UAVs. Therefore, the total flight path length is used as the evaluation metric, as shown in Eq. (17). Another evaluation metric is N_c, which is the number of iterations that reach the desired accuracy.

N represents the number of UAVs, and T represents the number of tasks. The total flight path length is calculated by summing the path lengths of each UAV as it reaches all target points in its task queue.

For the research on multi-UAV TA algorithms, a chaotic PSO algorithm based on Lévy flight mutation is ultimately proposed. Before each actual flight mission execution, a unified system performs the TA to ensure that the targets each UAV needs to reach are nonredundant and that the total flight path length is minimized. In terms of baseline algorithm selection, a set of comparative experiments was conducted, as shown in Table 2. This study compares the performance of five different baseline algorithms in multi-UAV multi-target PP tasks, specifically including GA, ACO, PSO, Pigeon-Inspired Optimization (PIO), and the standard PSO algorithm. The experimental results of the improved PSO algorithm are also listed.

From the experimental results, it can be seen that the improved PSO algorithm performs the best among all algorithms, with a total flight path length of 287.67 m. This indicates that the algorithm can effectively reduce the flight distance of UAVs in PP, enhancing the optimization degree of the path. In contrast, the Pigeon-Inspired Optimization (PIO) algorithm has the highest total flight path length, reaching 319.54 m, showing its lower efficiency in this specific task. This result may be related to the limitations of the PIO algorithm in handling multi-target environments, as it mainly relies on local search strategies and may not effectively explore the global optimal solution.

Table 2. Total flight path length for different algorithms. $\mathbb{L}$ is the total path length. N_c is the number of iterations that reach the desired accuracy. The best results are in bold. The rest of the paper follows the same notation.

Algorithm	$\mathbb{L}\downarrow$	$N_c\downarrow$
GA [140]	305.34	1800
ACO [141]	301.89	1500
PIO [142]	319.54	1200
PSO [143]	298.77	1100
Proposed	**287.67**	**750**

ACO and GA have similar performances, with total flight path lengths of 301.89 m and 305.34 m, respectively. Both algorithms perform comparably in PP, but ACO is slightly better. This could be attributed to ACO's adaptability and search efficiency in dynamic environments, allowing it to better handle the complex relationships between target points. The PSO algorithm has a total flight path length of 314.81 m. Although its performance is not as good as the aforementioned algorithms, it still shows some optimization potential, especially in certain specific scenarios.

It is evident that among the baseline algorithms, the PSO algorithm is the most effective, and the optimization effect of the improved PSO algorithm is the best. Subsequent experiments will further verify the effectiveness of the improvements to the PSO algorithm, which mainly include chaotic mapping in the initialization phase and fusion mutation during the iterative process.

To ensure a more uniform distribution of the initial population, this study conducted comparative experiments using different chaotic mapping algorithms. As shown in Table 3, the experimental results indicate that the Chebyshev mapping algorithm performs the best in terms of total flight path length, achieving 289.74 m. This result is significantly better than the other algorithms, which may be attributed to the Chebyshev mapping's ability to explore the local search space more effectively, enabling the metaheuristic algorithm to find better solutions. The results for the Logistic mapping and Bernoulli mapping are relatively close, at 296.30 m and 295.68 m, respectively.

Table 3. Comparative experiments of chaotic mapping algorithms.

Algorithm	$\mathbb{L}\downarrow$	$N_c\downarrow$
Tent mapping	299.67	1500
Logistic mapping	296.30	1200
Bernoulli mapping	295.68	1100
Sinusoidal mapping	300.33	1600
Chebyshev mapping	**289.74**	**900**

This indicates that these two mapping methods can provide reasonable path lengths during the initialization phase, although their efficiency is still inferior to that of the Chebyshev mapping. This outcome is related to the limitations of the Logistic and Bernoulli mappings in solving complex optimization problems, particularly in handling nonlinear relationships and multimodal functions, where their lack of adaptability may lead to higher path lengths. Therefore, this study ultimately chose the Chebyshev mapping to optimize the initialization phase of the population solutions.

Similarly, during the iterative process, the population solutions are optimized towards the optimal direction. However, to avoid the population solutions from falling into local optima, different fusion mutation algorithms were experimented with on the updated solutions. The experimental results are shown in Table 4. From the results, it can be observed that the Lévy flight algorithm performs the best in terms of total flight path length, achieving 288.91 m. This is due to its ability to achieve broader exploration in the search space and effectively jump out of local optima, thereby significantly reducing the flight path length.

The Cauchy mutation perturbation also performs well, with a total flight path length of 289.85 m. This is related to the heavy-tailed characteristic of the Cauchy distribution, which provides strong exploration capabilities near local optima, helping to find better path configurations. In contrast, the Gaussian mutation and *t*-distribution perturbation perform relatively poorly, with total flight path lengths of 290.18 m and 294.56 m, respectively. Therefore, this study ultimately chose the Lévy flight algorithm for fusion mutation.

5.2.2. *Multi-UAV path planning*

To more accurately assess the performance of multi-UAV PP, this study introduces several widely used metrics for different tasks.

Success Rate $P_{\mathrm{sr}}, P_{\mathrm{sr}} \in [0,1]$ is the proportion of rounds in which UAVs complete all tasks without collisions, as shown in the following equation:

$$P_{\mathrm{sr}} = \frac{\sum_{i=1}^{N} I(S_i)}{N}. \tag{30}$$

Here, N is the total number of experiments conducted. $I(S_i)$ is an indicator function. If all UAVs reach all target points without collisions in the ith round, then $I(S_i) = 1$. Equation (30) is used to calculate the success rate of UAVs in task execution.

Obstacle collision probability $P_{\mathrm{oc}}, P_{\mathrm{oc}} \in [0,1]$ is the proportion of episodes in which UAVs collide with obstacles, as shown in the following equation:

$$P_{\mathrm{oc}} = \frac{\sum_{i=1}^{E} I(C_{o,i} > 0)}{E}, \tag{31}$$

where $I(C_{(o,i)} > 0)$ represents the number of collisions with obstacles in the ith episode. If the number of collisions is greater than 0, then $I(C_{(o,i)} > 0) = 1$. This formula is used to calculate the proportion of rounds in which UAVs collide with obstacles.

Teammate collision probability $P_{\mathrm{tc}}, P_{\mathrm{tc}} \in [0,1]$ is the proportion of episodes in which UAVs collide with their teammates, as shown in the following equation:

$$P_{\mathrm{tc}} = \frac{\sum_{i=1}^{E} I(C_{\mathrm{t},i} > 0)}{E}, \tag{32}$$

where $C_{(t,i)}$ presents the number of collisions with UAV teammates in the ith episode. If $C_{(t,i)}$ is greater than 0, then $I(C_{(t,i)} > 0) = 1$. This formula is used to calculate the proportion of rounds in which UAVs collide with their teammates.

For the multi-UAV PP task, the central system first performs unified TA for the UAVs, where each UAV has its own target to reach. Then, multi-agent reinforcement learning algorithms are used for trajectory planning. The GACM-MARL algorithm proposed in this paper is based on the MADDPG algorithm, with the addition of an attention mechanism-based UAV communication module and a safety reinforcement learning-based action correction module.

For the selection of baseline models, a series of comparative experiments were conducted, as shown in Table 5. The focus was on evaluating the performance of four baseline models: IPPO, COMA, MAPPO and MADDPG. The improved model's performance was also listed. The experimental results were quantitatively evaluated using four key metrics, there are the average reward, mission success rate, the

Table 4. Comparative experiments of fusion mutation algorithms.

Algorithm	$\mathbb{L}\downarrow$	$N_c\downarrow$
Gaussian mutation	290.18	1500
t-Distribution perturbation	294.56	1200
Cauchy mutation	289.85	1000
Differential mutation perturbation	296.33	1300
Lévy flight	**288.91**	**800**

Table 5. Performance metrics of algorithms.

Algorithm	Mean reward↑	$P_{\mathrm{sr}}\uparrow$	$P_{\mathrm{oc}}\downarrow$	$P_{\mathrm{tc}}\downarrow$
IPPO [144]	75.57	0.43	0.42	0.12
COMA [145]	138.04	0.56	0.28	0.12
MAPPO [146]	147.07	0.58	0.21	0.08
MADDPG [122]	185.80	0.66	0.22	0.10
Proposed	**207.89**	**0.76**	**0.13**	**0.04**

Table 6. Results of the ablation study.

Task allocation	Obstacle collision	UAV collision	Average reward↑	P_{sr}↑	P_{sr}↓	P_{tc}↓
—	—	—	185.80	0.66	0.22	0.10
✓	—	—	192.20 (+6.40)	0.71(+0.05)	0.24(+0.02)	0.13(+0.03)
✓	✓	—	205.74 (+19.94)	0.74(+0.08)	**0.13**(−0.09)	0.08(−0.02)
✓	✓	✓	**207.89**(+22.09)	**0.76**(+0.10)	**0.13**(−0.09)	**0.04**(−0.06)

proportion of rounds with collisions with obstacles and proportion of rounds with collisions with UAV teammates.

The experimental results show that GACM-MARL performed excellently across all evaluation metrics. The average reward reached 207.89, the mission success rate was 0.76, the proportion of rounds with collisions with obstacles dropped to 0.13, and the proportion of rounds with collisions with teammates further decreased to 0.04. This indicates that the introduction of the UAV communication module and the action correction module significantly improved the algorithm's ability to avoid obstacles in multi-UAV cooperation and PP.

Among the baseline models, the MADDPG algorithm performed exceptionally well. This is attributed to MADDPGs strategy of centralized training and decentralized execution, which allows agents to share global information

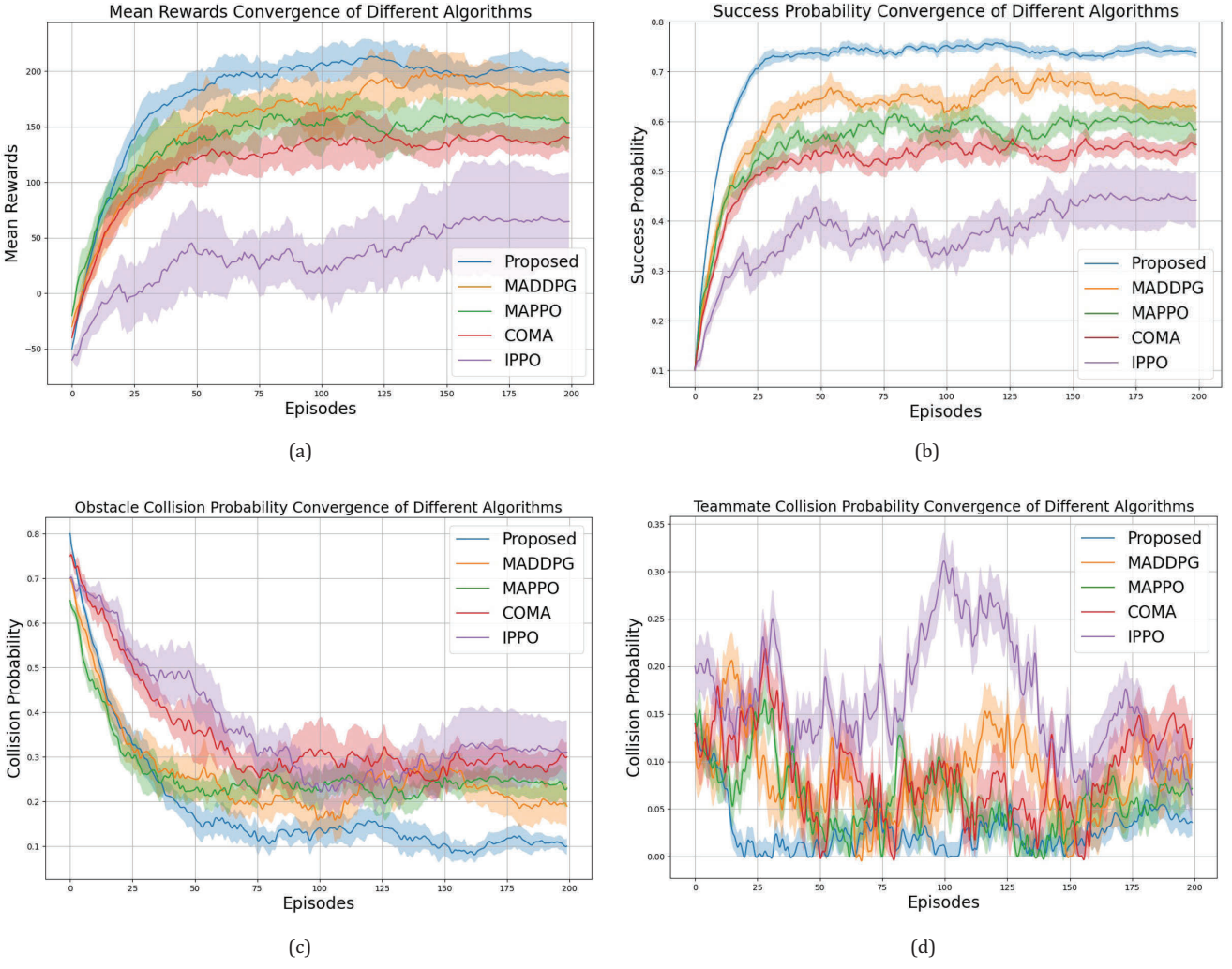
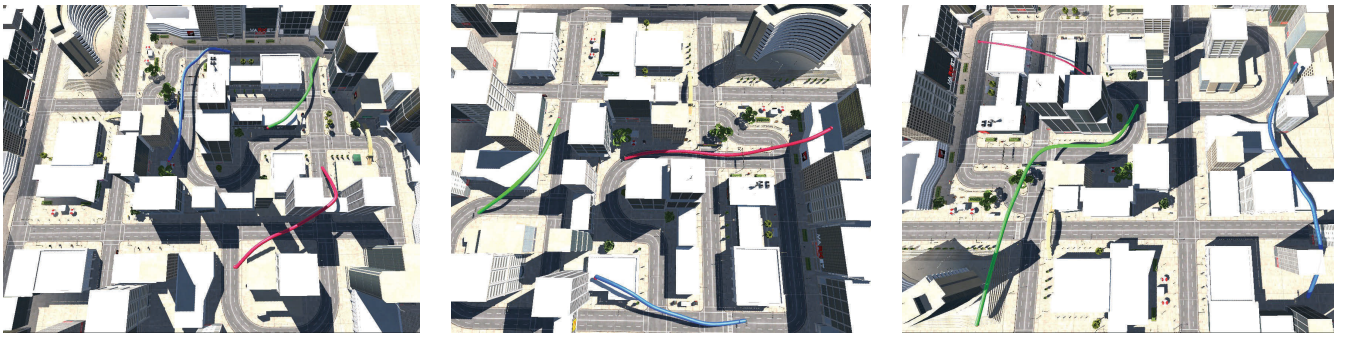


Fig. 5. Comparative Experiments on Multi-UAV PP. (a) Mean rewards. (b) Success rate (c) Collision rate of obstacles. (d) Collision rate of UAVs.



(a) (b) (c)

Fig. 6. (Color online) Trajectory diagram of multi-UAV path planning (Red Line for UAV_1, Green Line for UAV_2, Blue Line for UAV_3).

during the training phase, thereby enhancing cooperation among agents. This information-sharing mechanism enables MADDPG to effectively handle PP and obstacle avoidance in complex environments.

Although the MAPPO algorithm showed slightly better obstacle avoidance capabilities than MADDPG, it still lagged behind MADDPG in terms of overall mission success rate.

The IPPO algorithm's performance was noticeably insufficient in this experiment. The policy gradient method used by IPPO lacks global awareness of other agents' states in a multi-agent environment, making it difficult to effectively coordinate actions with other UAVs. This limitation negatively impacted its mission success rate and overall performance.

Based on the above analysis, it is evident that using the MADDPG algorithm as a baseline model is the optimal choice. Moreover, the improvements proposed in this study, building upon this algorithm, have proven to be highly effective. The convergence curve throughout the entire process is illustrated in Fig. 5.

For the multi-UAV PP task, this paper proposes a series of improvements based on the MADDPG model. An improved PSO algorithm is integrated at the central end to allocate tasks to all UAVs, ensuring reasonable task distribution for each UAV. To further enhance obstacle avoidance capabilities, an action correction module is added to the policy layer network of each UAV. This action correction module corrects the outputs of the policy layer network, ensuring that the UAVs actions comply with defined safety constraints. Specifically, two types of safety constraint functions are introduced to control UAVs, preventing collisions with obstacles and dynamic UAV teammates.

To assess the contributions of each component, an ablation study was conducted, where modules were incrementally added to the standard MADDPG model, allowing a clear evaluation of the impact on performance and informing potential future improvements.

As shown in Table 6, this study assessed the impact of different module combinations on the multi-UAV PP task. The experimental results indicate that with the gradual introduction of the TA module, communication module, and collision correction module, both the average reward and success rate exhibit an upward trend.

When the TA module is introduced alone, the average reward increases to 192.20, and the success rate rises to 0.71, although the collision probability slightly increases (0.24). This indicates that the module plays a positive role in optimizing UAV scheduling strategies. Since the maximum compensation per round is limited to 300 in this study, without a good scheduling strategy, UAVs cannot reach all target points within the limited steps. However, it still shows inadequacies in handling obstacles.

With the introduction of the obstacle and UAV collision correction modules, the average reward and success rate are observed to reach 205.74 and 0.74, respectively, eventually achieving 207.89 and 0.76 in Experiment 4. Notably, the addition of the obstacle collision correction module significantly reduces the probability of colliding with obstacles to 0.13, while the proportion of collisions with UAV teammates also decreases to 0.04. This demonstrates that the collision correction module plays a crucial role in enhancing the safety of PP, reflecting the continuous improvement in system safety. The action correction module in this study optimizes constrained problems to correct the output of the action by the policy layer network, thereby achieving obstacle avoidance. See Fig. 6 for the trajectory diagram of multi-UAV PP obtained using the proposed method.

6. Conclusion and Future Works

This paper presents a two-stage multi-UAV PP framework. In the first stage, we employ an improved PSO algorithm for TA, assigning a completely unique task queue to each UAV, enabling the UAV swarm to efficiently complete its missions. In the second stage, we utilize an improved MADDPG

algorithm for PP, incorporating an action correction module independent of the reinforcement learning algorithm for each UAV, which enhances obstacle avoidance capabilities. Experimental results demonstrate that this framework excels in multi-UAV collaborative tasks, significantly improving task completion efficiency and path safety.

Future work will focus on optimizing the real-time performance and adaptability of the algorithms. We plan to explore online learning mechanisms based on deep learning, allowing UAVs to dynamically adjust their TA and path planning strategies in response to environmental changes. Additionally, we will investigate collaborative working methods for different types of UAVs to further enhance the system's flexibility and efficiency.

Acknowledgments

Haixia Pan and Linfeng Han contributed equally to this work.

ORCID

Haixia Pan https://orcid.org/0000-0001-5243-348X
Linfeng Han https://orcid.org/0009-0007-8858-4512
Jiaming Yan https://orcid.org/0009-0004-3196-1725
Ruijun Liu https://orcid.org/0009-0008-1938-8574

References

[1] T.-M. Nguyen, S. Yuan, M. Cao, Y. Lyu, T. H. Nguyen and L. Xie, Ntu viral: A visual-inertial-ranging-lidar dataset, from an aerial vehicle viewpoint, *Int. J. Robot. Res.* **41**(3) (2022) 270–280.

[2] M. J. Er, S. Yuan and N. Wang, Development control and navigation of octocopter, in *2013 10th IEEE Int. Conf. Control and Automation (ICCA)* (IEEE, Hangzhou, China, 2013), pp. 1639–1643.

[3] Y. Lyu, T.-M. Nguyen, L. Liu, M. Cao, S. Yuan, T. H. Nguyen and L. Xie, Spins: A structure priors aided inertial navigation system, *J. Field Robot.* **40**(4) (2023) 879–900.

[4] M. A. Esfahani, K. Wu, S. Yuan and H. Wang, From local understanding to global regression in monocular visual odometry, *Int. J. Pattern Recognit. Artif. Intell.* **34**(01) (2020) 2055002.

[5] T.-M. Nguyen, S. Yuan, M. Cao, T. H. Nguyen and L. Xie, Viral slam: Tightly coupled Camera-IMU-UWB-Lidar slam (2021), arXiv: 2105.03296.

[6] T.-M. Nguyen, S. Yuan, M. Cao, L. Yang, T. H. Nguyen and L. Xie, Miliom: Tightly coupled multi-input lidar-inertia odometry and mapping, *IEEE Robot. Autom. Lett.* **6**(3) (2021) 5573–5580.

[7] T.-M. Nguyen, M. Cao, S. Yuan, Y. Lyu, T. H. Nguyen and L. Xie, Liro: Tightly coupled lidar-inertia-ranging odometry, in *2021 IEEE Int. Conf. Robotics and Automation (ICRA)* (IEEE, Xi'an, China, 2021), pp. 14484–14490.

[8] M. A. Esfahani, H. Wang, B. Bashari, K. Wu and S. Yuan, Learning to extract robust handcrafted features with a single observation via evolutionary neurogenesis, *Appl. Soft Comput.* **106** (2021) 107424.

[9] M. A. Esfahani, K. Wu, S. Yuan and H. Wang, A new approach to train convolutional neural networks for real-time 6-DOF camera relocalization, *2018 IEEE 14th Int. Conf. Control and Automation (ICCA)* (IEEE, Anchorage, Alaska, USA, 2018), pp. 81–85.

[10] M. A. Esfahani, H. Wang, K. Wu and S. Yuan, Aboldeepio: A novel deep inertial odometry network for autonomous vehicles, *IEEE Trans. Intell. Transp. Syst.* **21**(5) (2019) 1941–1950.

[11] M. A. Esfahani, K. Wu, S. Yuan and H. Wang, Deepdsair: Deep 6-dof camera relocalization using deblurred semantic-aware image representation for large-scale outdoor environments, *Image Vis. Comput.* **89** (2019) 120–130.

[12] H. Qie, D. Shi, T. Shen, X. Xu, Y. Li and L. Wang, Joint optimization of multi-UAV target assignment and path planning based on multi-agent reinforcement learning, *IEEE Access* **7** (2019) 146264–146272.

[13] Y. Lyu, M. Cao, S. Yuan and L. Xie, Vision-based plane estimation and following for building inspection with autonomous UAV, *IEEE Trans. Syst. Man Cybern.: Syst.* **53**(12) (2023) 7475–7488.

[14] M. Cao, Y. Lyu, S. Yuan and L. Xie, Online trajectory correction and tracking for facade inspection using autonomous UAV, in *2020 IEEE 16th Int. Conf. Control & Automation (ICCA)* (IEEE, Singapore, 2020), pp. 1149–1154.

[15] M. Cao, K. Cao, X. Li, S. Yuan, Y. Lyu, T.-M. Nguyen and L. Xie, Distributed multi-robot sweep coverage for a region with unknown workload distribution, *Auton. Intell. Syst.* **1**(1) (2021) 13.

[16] Y. Lyu, S. Yuan and L. Xie, Structure priors aided visual-inertial navigation in building inspection tasks with auxiliary line features, *IEEE Trans. Aerosp. Electron. Syst.* **58**(4) (2022) 3037–3048.

[17] S. Yuan and H. Wang, Autonomous object level segmentation, in *Proc. Int. Conf. Control, Automation, Robotics and Vision (ICARCV 2014)* (Singapore, 2014), pp. 33–37.

[18] M. Cao, T.-M. Nguyen, S. Yuan, A. Anastasiou, A. Zacharia, S. Papaioannou, P. Kolios, C. G. Panayiotou, M. M. Polycarpou, X. Xu, M. Zhang, F. Gao, B. Zhou, B. M. Chen and L. Xie, Cooperative aerial robot inspection challenge: A benchmark for heterogeneous multi-UAV planning and lessons learned (2025), arXiv:2501.06566.

[19] L. Zou, R. Zhou, S. Zhao and Q. Ding, Decentralized cooperative guidance for multiple missile groups in salvo attack, *Acta Aeronaut. Astronaut. Sin.* **32**(2) (2011) 281–290.

[20] Z. Shiyu and Z. Rui, Cooperative guidance for multimissile salvo attack, *Chin. J. Aeronaut.* **21**(6) (2008) 533–539.

[21] K. Li, J. Wang, C.-H. Lee, R. Zhou and S. Zhao, Distributed cooperative guidance for multivehicle simultaneous arrival without numerical singularities, *J. Guid. Control Dyn.* **43**(7) (2020) 1365–1373.

[22] M. Cao, X. Xu, S. Yuan, K. Cao, K. Liu and L. Xie, Doublebee: A hybrid aerial-ground robot with two active wheels, in *2023 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)* (IEEE, Detroit, Michigan, USA, 2023), pp. 6962–6969.

[23] X. Xu, M. Cao, S. Yuan, T. H. Nguyen, T.-M. Nguyen and L. Xie, A cost-effective cooperative exploration and inspection strategy for heterogeneous aerial system, in *Proc. 2024 IEEE Int. Conf. Control and Automation (ICCA)* (IEEE, Reykjavík, Iceland, 2024), pp. 673–678.

[24] R. Bai, S. Yuan, H. Guo, P. Yin, W.-Y. Yau and L. Xie, Collaborative graph exploration with reduced pose-slam uncertainty via submodular optimization, in *Proc. 2024 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)* (IEEE, Abu Dhabi, United Arab Emirates, 2024), pp. 10229–10236.

[25] Y. Lyu, M. Cao, S. Yuan and L. Xie, Vision based autonomous UAV plane estimation and following for building inspection (2021), arXiv:2102.01423.

[26] X. Ji, S. Yuan, J. Li, P. Yin, H. Cao and L. Xie, Sgba: Semantic gaussian mixture model-based lidar bundle adjustment, *IEEE Robot. Autom. Lett.* **9**(12) (2024) 10922–10929.

[27] S. Chen, K. Liu, C. Wang, S. Yuan, J. Yang and L. Xie, Salient sparse visual odometry with pose-only supervision, *IEEE Robot. Autom. Lett.* **9**(5) (2024) 4774–4781.

[28] M. A. Esfahani, H. Wang, K. Wu and S. Yuan, Unsupervised scene categorization, path segmentation and landmark extraction while traveling path, in *2020 16th Int. Conf. Control, Automation, Robotics and Vision (ICARCV)* (IEEE, Shenzhen, China, 2020), pp. 190–195.

[29] K. Xu, Y. Hao, S. Yuan, C. Wang and L. Xie, Airvo: An illumination-robust point-line visual odometry, in *2023 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)* (IEEE, Detroit, MI, USA, 2023), pp. 3429–3436.

[30] Y. Yang, S. Yuan, M. Cao, J. Yang and L. Xie, Av-pedaware: Self-supervised audio-visual fusion for dynamic pedestrian awareness, in *2023 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)* (IEEE, Detroit, Michigan, USA, 2023), pp. 1871–1877.

[31] M. Cao, K. Cao, S. Yuan, T.-M. Nguyen and L. Xie, Neptune: Non-entangling trajectory planning for multiple tethered unmanned vehicles, *IEEE Trans. Robot.* **39**(4) (2023) 2786–2804.

[32] M. Cao, K. Cao, S. Yuan, K. Liu, Y. L. Wong and L. Xie, Path planning for multiple tethered robots using topological braids, *Robot.: Sci. Syst.* (2023).

[33] S. Yuan, Y. Yang, T. H. Nguyen, T.-M. Nguyen, J. Yang, F. Liu, J. Li, H. Wang and L. Xie, Mmaud: A comprehensive multi-modal anti-UAV dataset for modern miniature drone threats, in *2024 IEEE Int. Conf. Robotics and Automation (ICRA)* (Yokohama, Japan, 2024), pp. 2745–2751.

[34] Q. Wang, G. Wan, X. Cao and L. Xie, A modeling method of multiple targets assignment under multiple UAVs' cooperation, *IOP Conf. Ser.: Mater. Sci. Eng.* **187** (2017) 012006.

[35] Y. Yang, S. Yuan, J. Yang, T. H. Nguyen, M. Cao, T.-M. Nguyen, H. Wang and L. Xie, Av-fdti: Audio-visual fusion for drone threat identification, *J. Autom. Intell.* **3**(3) (2024) 144–151.

[36] F. Liu, S. Yuan, W. Meng, R. Su and L. Xie, Non-cooperative stochastic target encirclement by anti-synchronization control via range-only measurement, in *2023 IEEE Int. Conf. Robotics and Automation (ICRA)* (IEEE, London, United Kingdom, 2023), pp. 5480–5485.

[37] K. Cao, M. Cao, S. Yuan and L. Xie, Direct: A differential dynamic programming based framework for trajectory generation, *IEEE Robot. Autom. Lett.* **7**(2) (2022) 2439–2446.

[38] M. A. Esfahani, H. Wang, K. Wu and S. Yuan, Orinet: Robust 3-d orientation estimation with a single particular IMU, *IEEE Robot. Autom. Lett.* **5**(2) (2019) 399–406.

[39] T.-M. Nguyen, M. Cao, S. Yuan, Y. Lyu, T. H. Nguyen and L. Xie, Viral-fusion: A visual-inertial-ranging-lidar sensor fusion approach, *IEEE Trans. Robot.* **38**(2) (2021) 958–977.

[40] S. Yuan, H. Wang and L. Xie, Survey on localization systems and algorithms for unmanned systems, *Unmanned Syst.* **9**(02) (2021) 129–163.

[41] Y. Liao, J. Li, S. Kang, Q. Li, G. Zhu, S. Yuan, Z. Dong and B. Yang, Se-calib: Semantic edge-based lidar–camera boresight online calibration in urban scenes, *IEEE Trans. Geosci. Remote Sens.* **61** (2023) 1–13.

[42] J. Li, Q. Leng, J. Liu, X. Xu, T. Jin, M. Cao, T.-M. Nguyen, S. Yuan, K. Cao and L. Xie, Helmetposer: A helmet-mounted IMU dataset for data-driven estimation of human head motion in diverse conditions (2024), arXiv:2409.05006.

[43] T.-M. Nguyen *et al.*, MCD: Diverse large-scale multi-campus dataset for robot perception, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* (Seattle, Washington, USA, 2024), pp. 22304–22313.

[44] F. Liu, S. Yuan, K. Cao, W. Meng and L. Xie, Distance-based multiple noncooperative ground target encirclement for complex environments, *IEEE Trans. Control Syst. Technol.* **33**(1) (2025) 261–273.

[45] P. Yin, S. Yuan, H. Cao, X. Ji, S. Zhang and L. Xie, Segregator: Global point cloud registration with semantic and geometric cues, in *2023 IEEE Int. Conf. Robotics and Automation (ICRA)* (IEEE, London, United Kingdom, 2023), pp. 2848–2854.

[46] Y. Zhou, H. Huang, S. Yuan, H. Zou, L. Xie and J. Yang, Metafi++: Wifi-enabled transformer-based human pose estimation for metaverse avatar simulation, *IEEE Internet Things J.* **10**(16) (2023) 14128–14136.

[47] J. Yang, H. Huang, Y. Zhou, X. Chen, Y. Xu, S. Yuan, H. Zou, C. X. Lu and L. Xie, Mm-fi: Multi-modal non-intrusive 4d human dataset for versatile wireless sensing, *Adv. Neural Inf. Process. Syst.* **36** (2024) 18756–18768.

[48] L. Deng, J. Yang, S. Yuan, H. Zou, C. X. Lu and L. Xie, Gaitfi: Robust device-free human identification via wifi and vision multimodal learning, *IEEE Internet Things J.* **10**(1) (2022) 625–636.

[49] Y. Yang, S. Yuan and L. Xie, Overcoming catastrophic forgetting for semantic segmentation via incremental learning, in *2022 17th Int. Conf. Control, Automation, Robotics and Vision (ICARCV)* (IEEE, Singapore, 2022), pp. 299–304.

[50] T. Ji, S. Yuan and L. Xie, Robust RGB-D slam in dynamic environments for autonomous vehicles, in *2022 17th Int. Conf. Control, Automation, Robotics and Vision (ICARCV)* (IEEE, Singapore, 2022), pp. 665–671.

[51] T. H. Nguyen, S. Yuan and L. Xie, Vr-slam: A visual-range simultaneous localization and mapping system using monocular camera and ultra-wideband sensors (2023), arXiv:2303.10903.

[52] H. Cao, Y. Xu, J. Yang, P. Yin, S. Yuan and L. Xie, Multi-modal continual test-time adaptation for 3d semantic segmentation, in *Proc. IEEE/CVF Int. Conf. Computer Vision* (Paris, France, 2023), pp. 18809–18819.

[53] T.-M. Nguyen, D. Duberg, P. Jensfelt, S. Yuan and L. Xie, Slict: Multi-input multi-scale surfel-based lidar-inertial continuous-time odometry and mapping, *IEEE Robot. Autom. Lett.* **8**(4) (2023) 2102–2109.

[54] F. Liu, S. Yuan, W. Meng, R. Su and L. Xie, Multiple noncooperative targets e ncirclement by relative distance-based positioning and neural antisynchronization control, *IEEE Trans. Ind. Electron.* **71**(2) (2023) 1675–1685.

[55] H. Wang, W. Mou, X. Mou, S. Yuan, S. Ulun, S. Yang and B.-S. Shin, An automatic self-calibration approach for wide baseline stereo cameras using sea surface images, *Unmanned Syst.* **3**(04) (2015) 277–290.

[56] Z. Wu, W. Li, P. Wang, K. Cao and Y. Pan, Vision-based potential field path planning for robot obstacle avoidance under field-of-view constraints, in *2023 IEEE 19th Int. Conf. Automation Science and Engineering (CASE)* (IEEE, Auckland, New Zealand, 2023), pp. 1–7.

[57] C. Guo, F. Liu, Y. Yang, X. Ling and W. Meng, Multiagent formation control for obstacle traversing using distance-only measurements in GPS-denied environments, *IEEE Trans. Ind. Inf.* **20**(7) (2024) 9120–9129.

[58] H. Wang, S. Yuan and K. Wu, Heterogeneous stereo: A human vision inspired method for general robotics sensing, in *TENCON 2017–2017 IEEE Region 10 Conf.* (IEEE, Penang, Malaysia, 2017), pp. 793–798.

[59] H. Wang, X. Mou, W. Mou, S. Yuan, S. Ulun, S. Yang and B.-S. Shin, Vision based long range object detection and tracking for unmanned surface vehicle, in *2015 IEEE 7th Int. Conf. Cybernetics and Intelligent Systems (CIS) and IEEE Conf. Robotics, Automation and Mechatronics (RAM)* (IEEE, Siem Reap, Cambodia., 2015), pp. 101–105.

[60] R. Bai, R. Zheng, Y. Xu, M. Liu and S. Zhang, Hierarchical multi-robot strategies synthesis and optimization under individual and collaborative temporal logic specifications, *Robot. Auton. Syst.* **153** (2022) 104085.

[61] L. Yang, J. Qi, J. Xiao and X. Yong, A literature review of UAV 3d path planning, in *Proc. 11th World Congress on Intelligent Control and Automation* (IEEE, Shenyang, China, 2014), pp. 2376–2381.

[62] J. Wang, Y. Li, R. Li, H. Chen and K. Chu, Trajectory planning for UAV navigation in dynamic environments with matrix alignment dijkstra, *Soft Comput.* **26**(22) (2022) 12599–12610.

[63] Z. Zhang, J. Wu, J. Dai and C. He, Optimal path planning with modified a-star algorithm for stealth unmanned aerial vehicles in 3d network radar environment, *Proc. Inst. Mech. Eng. Pt. G: J. Aerosp. Eng.* **236**(1) (2022) 72–81.

[64] Y. Guo, X. Liu, X. Liu, Y. Yang and W. Zhang, FC-RRT*: An improved path planning algorithm for UAV in 3d complex environment, *ISPRS Int. J. Geo-Inf.* **11**(2) (2022) 112.

[65] W. Li, L. Wang, A. Zou, J. Cai, H. He and T. Tan, Path planning for UAV based on improved PRM, *Energies* **15**(19) (2022) 7267.

[66] Y. V. Pehlivanoglu and P. Pehlivanoglu, An enhanced genetic algorithm for path planning of autonomous UAV in target coverage problems, *Appl. Soft Comput.* **112** (2021) 107796.

[67] L. Chen, W.-L. Liu and J. Zhong, An efficient multi-objective ant colony optimization for task allocation of heterogeneous unmanned aerial vehicles, *J. Comput. Sci.* **58** (2022) 101545.

[68] W. Su, M. Gao, X. Gao, X. Zhu and D. Fang, Adaptive decision-making with deep q-network for heterogeneous unmanned aerial vehicle swarms in dynamic environments, *Comput. Electr. Eng.* **119** (2024) 109621.

[69] G. Zhan, X. Zhang, Z. Li, L. Xu, D. Zhou and Z. Yang, Multiple-UAV reinforcement learning algorithm based on improved PPO in ray framework, *Drones* **6**(7) (2022) 166.

[70] R. Bai, H. Guo, W.-Y. Yau and L. Xie, Graph-based slam-aware exploration with prior topo-metric information, *IEEE Robot. Autom. Lett.* **9**(9) (2024) 7597–7604.

[71] R. Bai, S. Yuan, H. Guo, P. Yin, W.-Y. Yau and L. Xie, Multi-robot active graph exploration with reduced pose-slam uncertainty via submodular optimization, in *2024 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)* (2024), pp. 10229–10236.

[72] R. Bai, R. Zheng, M. Liu and S. Zhang, Multi-robot task planning under individual and collaborative temporal logic specifications, in *2021 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)* (IEEE, Prague, Czech Republic, 2021), pp. 6382–6389.

[73] X. Bai, H. Jiang, J. Cui, K. Lu, P. Chen and M. Zhang, UAV path planning based on improved A-star and DWA algorithms, *Int. J. Aerosp. Eng.* **2021** (2021) 1–12.

[74] T. Hu, S. Yuan, R. Bai and X. Xu, Swept volume-aware trajectory planning and MPC tracking for multi-axle swerve-drive AMRs (2024), arXiv:2412.16875.

[75] Q. Li and S. Yuan, Jacquard V2: Refining datasets using the human In the loop data correction method, in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, Yokohama, Japan, 2024, pp. 7932–7938, doi: 10.1109/ICRA57147.2024.10611652.

[76] T. Jin, X. Xu, Y. Yang, S. Yuan, T.-M. Nguyen, J. Li and L. Xie, Robust loop closure by textual cues in challenging environments, *IEEE Robot. Autom. Lett.* **10**(1) (2025) 812–819.

[77] Z. Qi, S. Yuan, F. Liu, H. Cao, T. Deng, J. Yang and L. Xie, Air-embodied: An efficient active 3dgs-based interaction and reconstruction framework with embodied large language model (2024), arXiv:2409.16019.

[78] T.-M. Nguyen, Y. Yang, T.-D. Nguyen, S. Yuan and L. Xie, Uloc: Learning to localize in complex large-scale environments with ultra-wideband ranges (2024), arXiv:2409.11122.

[79] H. Cao, Y. Xu, J. Yang, P. Yin, S. Yuan and L. Xie, Mopa: Multi-modal prior aided domain adaptation for 3d semantic segmentation, in *2024 IEEE Int. Conf. Robotics and Automation (ICRA)* (IEEE, Yokohama, Japan, 2024), pp. 9463–9470.

[80] P. Yin, H. Cao, T.-M. Nguyen, S. Yuan, S. Zhang, K. Liu and L. Xie, Outram: One-shot global localization via triangulated scene graph and global outlier pruning, in *2024 IEEE Int. Conf. Robotics and Automation (ICRA)* (IEEE, Yokohama, Japan, 2024), pp. 13717–13723.

[81] H. Cao, Y. Xu, J. Yang, P. Yin, X. Ji, S. Yuan and L. Xie, Reliable spatial-temporal voxels for multi-modal test-time adaptation, in *Computer Vision — ECCV 2024*, eds. A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler and G. Varol (Springer Nature, Cham, Switzerland, 2025), pp. 232–249.

[82] L. Geng, C. Haozhi, L. Mingyang, Y. Shenghai and Y. Jianfei, Gera: Geometric embedding for efficient point registration analysis (2024), arXiv:2410.00589.

[83] Z. Cui and Y. Wang, UAV path planning based on multi-layer reinforcement learning technique, *IEEE Access* **9** (2021) 59486–59497.

[84] M. A. Esfahani, K. Wu, S. Yuan and H. Wang, Towards utilizing deep uncertainty in traditional slam, in *2019 IEEE 15th Int. Conf. Control and Automation (ICCA)* (Edinburgh, Scotland, 2019), pp. 344–349.

[85] Z. Chen, Y. Xu, S. Yuan and L. Xie, iG-LIO: An incremental GICP-based tightly-coupled lidar-inertial odometry, *IEEE Robot. Autom. Lett.* **9**(2) (2024) 1883–1890.

[86] X. Ji, S. Yuan, P. Yin and L. Xie, LIO-GVM: An accurate, tightly-coupled lidar-inertial odometry with gaussian voxel map, *IEEE Robot. Autom. Lett.* **9**(3) (2024) 2200–2207.

[87] T. Deng, N. Wang, C. Wang, S. Yuan, J. Wang, D. Wang and W. Chen, Incremental joint learning of depth, pose and implicit scene representation on monocular camera in large-scale scenes (2024), arXiv:2404.06050.

[88] T.-M. Nguyen, X. Xu, T. Jin, Y. Yang, J. Li, S. Yuan and L. Xie, Eigen is all you need: Efficient lidar-inertial continuous-time odometry with internal association, *IEEE Robot. Autom. Lett.* **9**(6) (2024) 5330–5337.

[89] J. Li, S. Yuan, M. Cao, T.-M. Nguyen, K. Cao and L. Xie, Hcto: Optimality-aware lidar inertial odometry with hybrid continuous time optimization for compact wearable mapping system, *ISPRS J. Photogramm. Remote Sens.* **211** (2024) 228–243.

[90] Z. Yang, K. Xu, S. Yuan and L. Xie, A fast and light-weight non-iterative visual odometry with RGB-D cameras, *Unmanned Syst.* (2024) 1–13, https://www.worldscientific.com/doi/10.1142/S2301385025500591?srsltid=AfmBOorXdeA2FkP5ZvZTYt-F7wErKhyk5wbQSJnkRsf_Wbj9UhUE5cJx.

[91] J. Xu, W. Yu, S. Huang, S. Yuan, L. Zhao, R. Li and L. Xie, M-DIVO: Multiple ToF RGB-D cameras-enhanced depth–inertial–visual odometry, *IEEE Internet Things J.* **11**(23) (2024) 37562–37570.

[92] K. Xu, Y. Hao, S. Yuan, C. Wang and L. Xie, AirSLAM: An efficient and illumination-robust point-line visual SLAM system, in *IEEE Transactions on Robotics*, doi: 10.1109/TRO.2025.3539171.

[93] C. Zhao, K. Hu, J. Xu, L. Zhao, B. Han, K. Wu, M. Tian and S. Yuan, Adaptive-lio: Enhancing robustness and precision through environmental adaptation in lidar inertial odometry, *IEEE Internet Things J.* **PP**(99) (2024) 1–1.

[94] T.-M. Nguyen, Z. Cao, K. Li, S. Yuan and L. Xie, Gptr: Gaussian process trajectory representation for continuous-time motion estimation (2024), arXiv:2410.22931.

[95] K. Xu, Z. Jiang, H. Cao, S. Yuan, C. Wang and L. Xie, An efficient scene coordinate encoding and relocalization method (2024), arXiv:2412.06488.

[96] S. Yuan, B. Lou, T.-M. Nguyen, P. Yin, M. Cao, X. Xu, J. Li, J. Xu, S. Chen and L. Xie, Large-scale UWB anchor calibration and one-shot localization using gaussian process (2024), arXiv:2412.16880.

[97] J. Xu, G. Huang, W. Yu, X. Zhang, L. Zhao, R. Li, S. Yuan and L. Xie, Selective Kalman filter: When and how to fuse multi-sensor information to overcome degeneracy in slam (2024), arXiv:2412.17235.

[98] Y. Lai, S. Yuan, Y. Nassar, M. Fan, T. Weber and M. Rätsch, NVP-HRI: Zero shot natural voice and posture-based human-robot interaction via large language model, *Expert Syst. Appl.* **268** (2025) 126360–126378.

[99] T. Deng, Y. Chen, L. Zhang, J. Yang, S. Yuan, J. Liu, D. Wang, H. Wang and W. Chen, Compact 3d gaussian splatting for dense visual slam (2024), arXiv:2403.11247.

[100] Y. Ma, J. Xu, S. Yuan, T. Zhi, W. Yu, J. Zhou and L. Xie, MM-LINS: A multi-map LiDAR-inertial system for over-degenerate environments, *IEEE Trans. Intell. Veh.* (Abu Dhabi, United Arab Emirates, 2024) 1–11, https://ieeexplore.ieee.org/document/10557776.

[101] J. Li, T.-M. Nguyen, S. Yuan and L. Xie, PSS-BA: LiDAR bundle adjustment with progressive spatial smoothing, in *2024 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, (2024), pp. 1124–1129.

[102] W. Yu, J. Xu, C. Zhao, L. Zhao, T.-M. Nguyen, S. Yuan, M. Bai and L. Xie, I2EKF-LO: A dual-iteration extended Kalman filter based lidar odometry, in *2024 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)* (Abu Dhabi, United Arab Emirates, 2024), pp. 10453–10460.

[103] J. Li, T.-M. Nguyen, M. Cao, S. Yuan, T.-Y. Hung and L. Xie, Graph optimality-aware stochastic lidar bundle adjustment with progressive spatial smoothing (2024), arXiv:2410.14565.

[104] Z. Xiao, Y. Yang, G. Xu, X. Zeng and S. Yuan, Av-dtec: Self-supervised audio-visual fusion for drone trajectory estimation and classification (2024), arXiv:2412.16928.

[105] H. Liang, Y. Yang, J. Hu, J. Yang, F. Liu and S. Yuan, Unsupervised UAV 3d trajectories estimation with sparse point clouds, in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)* (IEEE, Hyderabad, India, 2025).

[106] A. Lei, T. Deng, H. Wang, J. Yang and S. Yuan, Audio array-based 3d UAV trajectory estimation with lidar pseudo-labeling, in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)* (IEEE, Hyderabad, India, 2025).

[107] K. Wu, M. Abolfazli Esfahani, S. Yuan and H. Wang, Learn to steer through deep reinforcement learning, *Sensors* **18**(11) (2018) 3650.

[108] K. Wu, M. A. Esfahani, S. Yuan and H. Wang, Depth-based obstacle avoidance through deep reinforcement learning, in *Proc. 5th Int. Conf. Mechatronics and Robotics Engineering* (Rome, Italy, 2019), pp. 102–106.

[109] K. Wu, H. Wang, M. A. Esfahani and S. Yuan, BND*-DDQN: Learn to steer autonomously through deep reinforcement learning, *IEEE Trans. Cogn. Devel. Syst.* **13**(2) (2019) 249–261.

[110] R. Xie, Z. Meng, L. Wang, H. Li, K. Wang and Z. Wu, Unmanned aerial vehicle path planning algorithm based on deep reinforcement learning in large-scale and dynamic environments, *IEEE Access* **9** (2021) 24884–24900.

[111] K. Wu, M. A. Esfahani, S. Yuan and H. Wang, Tdpp-net: Achieving three-dimensional path planning via a deep neural network architecture, *Neurocomputing* **357** (2019) 151–162.

[112] K. Wu, H. Wang, M. A. Esfahani and S. Yuan, Achieving real-time path planning in unknown environments through deep neural networks, *IEEE Trans. Intell. Transp. Syst.* **23**(3) (2020) 2093–2102.

[113] K. Wu, H. Wang, M. A. Esfahani and S. Yuan, Learn to navigate autonomously through deep reinforcement learning, *IEEE Trans. Ind. Electron.* **69**(5) (2021) 5342–5352.

[114] Q. Luo, T. H. Luan, W. Shi and P. Fan, Deep reinforcement learning based computation offloading and trajectory planning for multi-UAV cooperative target search, *IEEE J. Sel. Areas Commun.* **41**(2) (2022) 504–520.

[115] X. Liu, Y. Liu, Y. Chen and L. Hanzo, Trajectory design and power control for multi-UAV assisted wireless networks: A machine learning approach, *IEEE Trans. Veh. Technol.* **68**(8) (2019) 7957–7969.

[116] C. Yan, X. Xiang and C. Wang, Towards real-time path planning through deep reinforcement learning for a UAV in dynamic environments, *J. Intell. Robot. Syst.* **98** (2020) 297–309.

[117] D. Hong, S. Lee, Y. H. Cho, D. Baek, J. Kim and N. Chang, Energy-efficient online path planning of multiple drones using reinforcement learning, *IEEE Trans. Veh. Technol.* **70**(10) (2021) 9725–9740.

[118] P. Xu, Y. Cui, Y. Shen, W. Zhu, Y. Zhang, B. Wang and Q. Tang, Reinforcement learning compensated coordination control of multiple mobile manipulators for tight cooperation, *Eng. Appl. Artif. Intell.* **123** (2023) 106281.

[119] P. Xu, Z. Li, X. Liu, T. Zhao, L. Zhang and Y. Zhao, Reinforcement learning-based distributed impedance control of robots for compliant operation in tight interaction tasks, *Eng. Appl. Artif. Intell.* **136** (2024) 108913.

[120] L. Mei and P. Xu, Path planning for robots combined with zero-shot and hierarchical reinforcement learning in novel environments, *Actuators* **13**(11) (2024) 458.

[121] J. Liang, Z. Wang, Y. Cao, J. Chiun, M. Zhang and G. A. Sartoretti, Context-aware deep reinforcement learning for autonomous robotic navigation in unknown area, in *Conf. Robot Learning, Proc. Machine Learning Research* (Atlanta, Georgia, USA, 2023), pp. 1425–1436.

[122] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel and I. Mordatch, Multi-agent actor-critic for mixed cooperative-competitive environments, *Adv. Neural Inf. Process. Syst.* **30** (2017) 6382–6393.

[123] J. Liang, Y. Cao, Y. Ma, H. Zhao and G. Sartoretti, Hdplanner: Advancing autonomous deployments in unknown environments through hierarchical decision networks, *IEEE Robot. Autom. Lett.* **10**(1) (2025) 256–263.

[124] L. Wang, K. Wang, C. Pan, W. Xu, N. Aslam and L. Hanzo, Multi-agent deep reinforcement learning-based trajectory planning for multi-UAV assisted mobile edge computing, *IEEE Trans. Cogn. Commun. Netw.* **7**(1) (2020) 73–84.

[125] S. Huang, R. S. H. Teo and K. K. Tan, Collision avoidance of multi unmanned aerial vehicles: A review, *Annu. Rev. Control* **48** (2019) 147–164.

[126] X. Zhang, J. Ma, S. Huang, Z. Cheng and T. H. Lee, Integrated planning and control for collision-free trajectory generation in 3d environment with obstacles, in *2019 19th Int. Conf. Control, Automation and Systems (ICCAS)* (IEEE, Jeju, South Korea, 2019), pp. 974–979.

[127] B. Alrifaee, K. Kostyszyn and D. Abel, Model predictive control for collision avoidance of networked vehicles using lagrangian relaxation, *IFAC-PapersOnLine* **49**(3) (2016) 430–435.

[128] Z. Cai, H. Cao, W. Lu, L. Zhang and H. Xiong, Safe multi-agent reinforcement learning through decentralized multiple control barrier functions, in *Proc. Int. Conf. Learning Representations (ICLR 2021)* (2021), https://openreview.net/forum?id=P6_q1BRxY8Q.

[129] S. Gu, L. Yang, Y. Du, G. Chen, F. Walter, J. Wang and A. Knoll, A review of safe reinforcement learning: Methods, theories, and applications, *IEEE Trans. Pattern Anal. Mach. Intell.* **46**(12) (2024) 11216–11235.

[130] Z. Sheebaelhamd, K. Zisis, A. Nisioti, D. Gkouletsos, D. Pavllo and J. Kohler, Safe deep reinforcement learning for multi-agent systems with continuous action spaces (2021), arXiv:2108.03952.

[131] C. He, T. Yang, T. Duhan, Y. Wang and G. Sartoretti, Alpha: Attention-based long-horizon pathfinding in highly-structured areas, in *2024 IEEE Int. Conf. Robotics and Automation (ICRA)* (IEEE, Yokohama, Japan, 2024), pp. 14576–14582.

[132] Y. Cao, R. Zhao, Y. Wang, B. Xiang and G. Sartoretti, Deep reinforcement learning-based large-scale robot exploration, *IEEE Robot. Autom. Lett.* **9**(5) (2024) 4631–4638.

[133] T. Yang, Y. Cao and G. Sartoretti, Intent-based deep reinforcement learning for multi-agent informative path planning, in *2023 Int. Symp. Multi-Robot and Multi-Agent Systems (MRS)* (IEEE, 2023), pp. 71–77.

[134] D. M. S. Tan, Y. Ma, J. Liang, Y. C. Chng, Y. Cao and G. Sartoretti, Ir 2: Implicit rendezvous for robotic exploration teams under sparse intermittent connectivity, in *2024 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)* (IEEE, Abu Dhabi, United Arab Emirates, 2024), pp. 13245–13252.

[135] Y. Ma, J. Liang, Y. Cao, D. M. S. Tan and G. Sartoretti, Privileged reinforcement and communication learning for distributed, bandwidth-limited multi-robot exploration (2024), arXiv:2407.20203.

[136] K. Zhang, Z. Yang and T. Başar, Multi-agent reinforcement learning: A selective overview of theories and algorithms, in *Handbook of Reinforcement Learning and Control*, Vol. 325 (Springer, 2021), pp. 321–384.

[137] J. Garcıa and F. Fernández, A comprehensive survey on safe reinforcement learning, *J. Mach. Learn. Res.* **16**(1) (2015) 1437–1480.

[138] G. Dalal, K. Dvijotham, M. Vecerik, T. Hester, C. Paduraru and Y. Tassa, Safe exploration in continuous action spaces (2018), arXiv:1801.08757.

[139] M. Vukić, B. Grgić, D. Dinčir, L. Kostelac and I. Marković, Unity based urban environment simulation for autonomous vehicle stereo vision evaluation, in *2019 42nd Int. Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (IEEE, 2019), pp. 949–954.

[140] J. H. Holland, Genetic algorithms, *Sci. Am.* **267**(1) (1992) 66–73.

[141] M. Dorigo, M. Birattari and T. Stutzle, Ant colony optimization, *IEEE Comput. Intell. Mag.* **1**(4) (2006) 28–39.

[142] H. Duan and P. Qiao, Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning, *Int. J. Intell. Comput. Cybern.* **7**(1) (2014) 24–37.

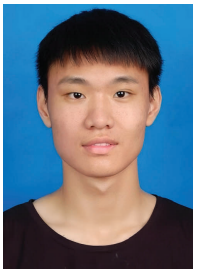
[143] J. Kennedy and R. Eberhart, Particle swarm optimization, in *Proc. ICNN'95 — Int. Conf. Neural Networks*, Vol. 4 (IEEE, Perth, Western Australia, 1995), pp. 1942–1948.

[144] C. S. De Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. Torr, M. Sun and S. Whiteson, Is independent learning all you need in the starcraft multi-agent challenge? (2020), arXiv:2011.09533.

[145] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli and S. Whiteson, Counterfactual multi-agent policy gradients, in *Proc. AAAI Conf. Artificial Intelligence*, Vol. 32 (New Orleans, Louisiana, USA, 2018).

[146] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen and Y. Wu, The surprising effectiveness of PPO in cooperative multi-agent games, *Adv. Neural Inf. Process. Syst.* **35** (2022) 24611–24624.

Haixia Pan received Ph.D. degree in Computer Science from the Beihang University, Beijing, China. She is currently Professor and Master's Supervisor at the School of Software, the Beihang University. Her research interests include software engineering, artificial intelligence, pattern recognition, computer vision, and image processing. Dr. Pan is Senior Member of the China Computer Federation (CCF). She serves as Expert for Graduate Education Evaluation and monitoring at the Academic Degrees & Graduate Education Development Center of the Ministry of Education.



Linfeng Han is currently pursuing his degree in Software Engineering, the Beihang University. His research interests include reinforcement learning and multi-agent system.

Jiaming Yan received his Ph.D. degree in Information and Communication Engineering from the Nanjing University of Science and Technology, Nanjing, China, in 2019. From 2016 to 2017, he was Research Assistant at the Texas Tech University, Lubbock, TX, USA. He is currently Senior Engineer at Nanjing Research Institute of Electronic Engineering. His research interests are UAV Path Planning and Reinforcement Learning.

Ruijun Liu received his Ph.D. degree in Engineering from Centrale Nantes, France. He is currently Professor at the School of Software, the Beihang University and a Senior Member of the China Computer Federation (CCF). Dr. Liu serves as Committee Member of the CCF Virtual Reality and Visualization Technology Committee, the Virtual Technology and Applications Committee of the China Society for Simulation, and the Visualization and Visual Analytics Committee of the China Society of Image and Graphics (CSIG). His research interests include industrial software, virtual reality.