frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 22 January 2024
DOI 10.3389/fnbot.2023.1302898

OPEN ACCESS

EDITED BY
Di Wu,
Southwest University, China

REVIEWED BY
Qinbing Fu,
Guangzhou University, China
Ziheng Chen,
Walmart Labs, United States
Jose Balsa Barreiro,
Massachusetts Institute of Technology, United States

*CORRESPONDENCE
Yatong Zhou
✉ zyt@hebut.edu.cn

RECEIVED 27 September 2023
ACCEPTED 26 December 2023
PUBLISHED 22 January 2024

CITATION
Kong X, Zhou Y, Li Z and Wang S (2024) Multi-UAV simultaneous target assignment and path planning based on deep reinforcement learning in dynamic multiple obstacles environments. *Front. Neurorobot.* 17:1302898. doi: 10.3389/fnbot.2023.1302898

Multi-UAV simultaneous target assignment and path planning based on deep reinforcement learning in dynamic multiple obstacles environments

Xiaoran Kong[1], Yatong Zhou[1]*, Zhe Li[2] and Shaohai Wang[3]

[1]School of Electronic and Information Engineering, HeBei University of Technology, Tianjin, China, [2]Institute of Digital Economy Industry Research, Hebei University of Technology, Shijiazhuang, China, [3]School of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China

Target assignment and path planning are crucial for the cooperativity of multiple unmanned aerial vehicles (UAV) systems. However, it is a challenge considering the dynamics of environments and the partial observability of UAVs. In this article, the problem of multi-UAV target assignment and path planning is formulated as a partially observable Markov decision process (POMDP), and a novel deep reinforcement learning (DRL)-based algorithm is proposed to address it. Specifically, a target assignment network is introduced into the twin-delayed deep deterministic policy gradient (TD3) algorithm to solve the target assignment problem and path planning problem simultaneously. The target assignment network executes target assignment for each step of UAVs, while the TD3 guides UAVs to plan paths for this step based on the assignment result and provides training labels for the optimization of the target assignment network. Experimental results demonstrate that the proposed approach can ensure an optimal complete target allocation and achieve a collision-free path for each UAV in three-dimensional (3D) dynamic multiple-obstacle environments, and present a superior performance in target completion and a better adaptability to complex environments compared with existing methods.

KEYWORDS

multiple unmanned aerial vehicles, target assignment, path planning, deep reinforcement learning, partially observable Markov decision process

1 Introduction

Recently, unmanned aerial vehicles (UAV) have been widely applied to a variety of fields due to their advantages of high flexibility, low operating cost, and ease of deployment. In the military field, UAVs have become an important part of modern warfare and can be used for missions such as reconnaissance (Qin et al., 2021), strikes (Chamola et al., 2021), and surveillance (Liu et al., 2021), reducing casualties and enhancing combat efficiency. In the field of agriculture, UAVs have good applications in plant protection (Xu et al., 2019; Chen et al., 2021), agricultural monitoring (Zhang et al., 2021), and so on, improving the efficiency and precision of agricultural operations. In the field of environmental protection, UAVs are extensively employed in environmental monitoring (Yang et al., 2022), pollution source tracking (Liu et al., 2023), nature reserve inspection (Su et al., 2018), and other tasks, effectively supporting environmental protection work. In addition, for search and

rescue tasks (Fei et al., 2022; Lyu et al., 2023), UAVs can quickly obtain disaster information through airborne sensors to provide efficient and timely assistance for subsequent rescue. However, it is difficult to apply lone single UAVs to complex and diverse missions due to their limited functionality and payload. Cooperation between multiple UAVs (Song et al., 2023) has greatly expanded the ability and scope of task execution, and has gradually replaced the single UAV as the nontrivial technology for various complex tasks. The key to solving the multi-UAV cooperative problems (Wang T. et al., 2020; Xing et al., 2022; Wang et al., 2023) is target assignment and path planning for UAVs, which is the guarantee of task completion.

The above problem consists of two fundamental sub-problems. Target assignment (Gerkey and Matarić, 2004) means assigning one UAV for each target to maximize the overall efficiency or minimize the total costs. It has many effective solutions such as the Genetic algorithm (GA) (Tian et al., 2018) and the Hungarian algorithm (Kuhn, 1955). Lee et al. (2003) introduced greedy eugenics to GA to improve the performance of GA in weapon-target assignment problems. Aiming at the multi-task allocation problem, Samiei et al. (2019) proposed a novel cluster-based Hungarian algorithm. Path planning (Aggarwal and Kumar, 2020) refers to each drone planning an optimal path from its initial location to its designated target with the collision-free constraint. It has been studied extensively, and A* (Grenouilleau et al., 2019), rapidly-exploring random tree algorithm (RRT) (Li et al., 2022) and particle swarm optimization (PSO) (Fernandes et al., 2022) are classical methods. Fan et al. (2023) incorporated the artificial potential field method into RRT to reduce the cost of path planning. He W. et al. (2021) proposed a novel hybrid algorithm for UAV path planning by combining PSO with the symbiotic organism search. While most previous works tackle the problem in static environments, and a common feature of these solutions is that they rely on global information of the task environment for explicit planning, which may lead to unexpected failure in the face of uncertain circumstances or unpredictable obstacles.

Therefore, some studies resort to learning-based approaches such as deep learning (DL) (Kouris and Bouganis, 2018; Mansouri et al., 2020; Pan et al., 2021). Pan et al. (2021) combined DL and GA to plan the path for UAV data collection. The proposed method collected various paths and states in different task environments by GA, and used them to train the neural network of DL, which can give an optimal path in familiar scenarios with real-time requirements. Kouris and Bouganis (2018) proposed a self-supervised CNN-based approach for indoor UAV navigation. This method used an indoor-flight dataset to train the CNN and utilized the CNN to predict collision distance based on an on-board camera. However, Deep learning-based approaches require labels for learning and they are infeasible when the environment is highly variable.

Unlike DL methods, reinforcement learning (RL) (Thrun and Littman, 2000; Busoniu et al., 2008; Zhang et al., 2016) can optimize strategies directly through trial-and-error iteration interacting with the environment without prior knowledge, which is adaptable to dynamic environments. Moreover, deep reinforcement learning (DRL) (Mnih et al., 2015) combines DL and RL to implement end-to-end learning. It makes RL no longer limited to low-dimensional space and greatly expands the scope of application of RL (Wang C. et al., 2020; Chane-Sane et al., 2021; He L. et al., 2021; Kiran et al., 2021; Luo et al., 2021; Wu et al., 2021; Yan et al., 2022; Yue et al., 2023; Zhao et al., 2023). Wu et al. (2021) introduced a curiosity-driven method into DRL to improve training efficiency and performance in autonomous driving tasks. Yan et al. (2022) proposed a simplified, unified, and applicable DRL method for vehicular systems. Chane-Sane et al. (2021) designed a new RL method with imagined possible subgoals to facilitate learning of complex tasks such as challenging navigation and vision-based robotic manipulation. Luo et al. (2021) designed a DRL-based method to generate solutions for the missile-target assignment problem autonomously. He L. et al. (2021) presented an autonomous path planning method based on DRL for quadrotors in unknown environments. Wang C. et al. (2020) proposed DRL algorithm with nonexpert helpers to address the autonomous navigation problem for UAVs in large-scale complex environments.

DRL is suitable to solve the target assignment problem and path planning problem of UAVs, but there are still some challenges when multiple UAVs perform tasks in dynamic environments. The first challenge is inefficient target assignment. Typically, UAVs execute target assignment first and then perform path planning based on the result of the target assignment. However, the dynamism and uncertainty of the environment always lead to an inaccurate assignment result, which directly affects the subsequent path planning. In this respect, UAVs need to perform autonomous target assignment and path planning simultaneously. There are only a few scholars who have studied this field. Qie et al. (2019) constructed the multiple UAVs target assignment and path planning problem as a multi-agent system and used the multi-agent deep deterministic policy gradient (MADDPG) (Lowe et al., 2017) framework to train the system to solve two problems simultaneously. They traverse all targets and select the agent closest to each target after each step of the agent, which often results in an incomplete assignment of targets when two agents are at the same and shortest distance from one target. Han et al. (2020) proposed a navigation policy for multiple robots in a dynamic environment based on the Proximal Policy Optimization (PPO) (Schulman et al., 2017) algorithm. The target assignment scheme was proposed depending on the distance between robots and targets. However, this assignment method does not take into account the obstacles in the task environment, which is vulnerable to leads to inaccurate allocation in a multi-obstacle environment similar to the real world. The second challenge is that UAVs' onboard sensors have limited detection range. The real-time decision-making of UAVs depends on observation returned by sensors, especially in dynamic and uncertain environments. If the detection range of sensors is limited, the current state cannot fully represent the global environmental information, which greatly increases the difficulty of autonomous flight.

To overcome these challenges, this article models the multi-UAV target assignment and path planning problem as a partially observable Markov decision process (POMDP) (Spaan, 2012) and designs a simultaneous target assignment and path planning method based on DRL to settle it. Among the DRL-based methods, the twin-delayed deep deterministic policy gradient (TD3) (Fujimoto et al., 2018) is a state-of-the-art (SOTA) DRL

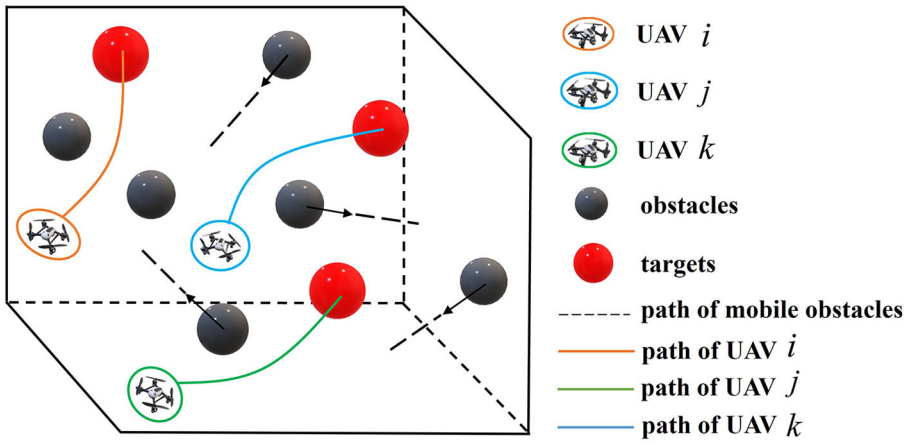

FIGURE 1
Schematic diagram of multi-UAV target assignment and path planning.

algorithm and has been widely used in training the policy of UAVs. It significantly improves the learning speed and performance of deep deterministic policy gradient (DDPG) (Lillicrap et al., 2015) algorithm by reducing the overestimation of DDPG. Zhang et al. (2022) introduced the spatial change information of environment to the TD3, and used it to guide a UAV to complete navigation tasks in complex environments with multiple obstacles. Hong et al. (2021) proposed an advanced TD3 model to perform energy-efficient path planning at the edge-level drone. In this regard, a more effective DRL algorithm based on TD3 is proposed to solve the POMDP in this article.

The main contributions of this article can be summarized as follows:

- A DRL framework for multi-UAV target assignment and path planning is developed in 3D dynamic multiple obstacles environments, where the target assignment and path planning problem is modeled as a POMDP.
- A simultaneous target assignment and path planning method taking into account UAVs, targets, and moving obstacles is proposed, which can achieve an optimal target assignment and complete collision-free path planning for each UAV simultaneously.
- A 3D stochastic complex simulation environment is built to train an algorithm, and the experimental results validate the effectiveness of the proposed method.

The remainder of this article is organized as follows: The background is presented in Section 2, Section 3 introduces the formulation of the multi-UAV problem. In Section 4, a detailed introduction to our method is provided. Section 5 presents the simulation experiments and results. Finally, the conclusion of this paper and future work are summarized in Section 6.

2 Background

This section gives a brief introduction to the multi-UAV target assignment and path planning problem in this article first, followed by the multi-UAV problem formulated as a POMDP in 3D dynamic environments.

2.1 Multi-UAV target assignment and path planning problem

The multiple UAVs target assignment and path planning scenario of this paper is shown in Figure 1:
(1) A series of UAVs are commanded to fly across a 3D mission area until they reach the targets distributed in different locations.
(2) The mission area is scattered with some static or irregularly moving obstacles.
(3) UAVs are required to avoid collision with each other and obstacles.
(4) UAVs are isomorphic and targets are identical.

The object of multi-UAV target assignment and path planning is to minimize the total flight path length of all UAVs [Equation (1)] under the constraints of target completely assignment and collision-free:

$$\min\left(\sum_{i}^{N_U} d_i\right) \tag{1}$$

$$\begin{cases} \bigcup_{i=1}^{N_U} T_i = \mathbf{T},\ i \in \{1,\ldots,N_T\}, \\ T_i \neq T_j,\ i \neq j. \end{cases} \tag{2}$$

$$\begin{cases} \left\| U_i^t, U_j^t \right\| > 2r_u,\ i,j = 1,2,\ldots,N_U, \\ \left\| U_i^t, O_k^t \right\| > r_u + r_o,\ i = 1,2,\ldots,N_U,\ k = 1,2,\ldots,M. \end{cases} \tag{3}$$

where $T_i \in \mathbf{T}$, $i \in \{1,\ldots,N_T\}$ denotes the targets, U_i, $i = 1,2,\ldots,N_U$ denotes the UAVs and O_k, $k = 1,2,\ldots,M$ denotes the obstacles. U_i^t, O_k^t represents the positions of UAV i and obstacle k at time t, respectively. d_i is the flight length of UAV i, r_u and r_o are the radius of UAVs and obstacles. Equation (2) denotes the target complete assignment constraint, which means each target is only assigned to one UAV. Equation (3) defines the collision-free constraint, where the first one means any two UAVs cannot

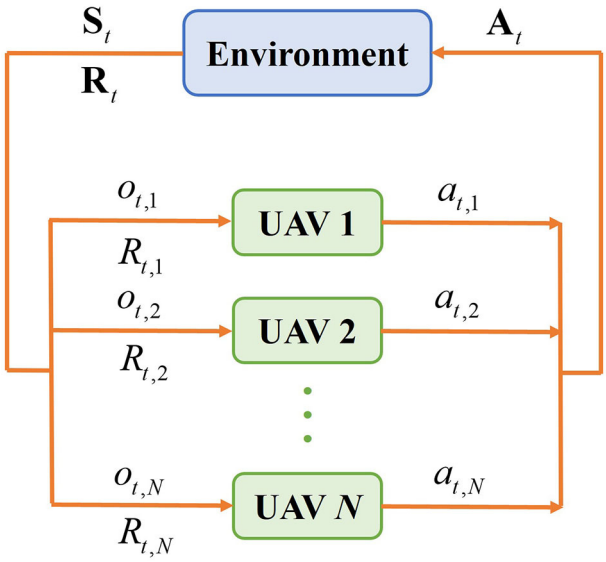

FIGURE 2
Multi-UAV reinforcement learning process in partially observable environment.

collide at all times, while the second defines each UAV's path is collision-free with obstacles.

2.2 Modeling multiple UAVs problem as an POMDP

The multi-UAV problem can be modeled as POMDP, which is composed of a tuple $\langle \mathbf{N}, \mathbf{S}, \mathbf{O}, \mathbf{A}, \mathbf{P}, \mathbf{R} \rangle$. In this tuple, $\mathbf{N} = \{1, 2, \cdots, N\}$ represents the collection of N UAVs, $\mathbf{S}$ is the state space of UAVs, $\mathbf{O} = \{o_1, o_2, \cdots, o_N\}$ is the observation of all UAVs, where o_i represents the observation of UAV i. When the environment is partially observable, at time t, each UAV only obtains its own local observation $o_{t,i} \in o_i$. $\mathbf{A} = \{a_1, a_2, \cdots, a_N\}$ is the action collection of UAVs, where a_i is the action taken by UAV i; $\mathbf{P} : \mathbf{S} \times \mathbf{A} \times \mathbf{S}' \in [0, 1]$ denotes the probability that state transfers from $\mathbf{S}$ to $\mathbf{S}'$ after performing action $\mathbf{A}$; $\mathbf{R} = \{R_1, R_2, \cdots, R_N\}$ is the reward collection of UAVs, where R_i denotes the reward of UAV i received from the environment.

The multi-UAV reinforcement learning process in a partially observable environment is shown in Figure 2. At each epoch t, UAV i selects its optimal action $a_{t,i}$ based on the policy π to maximize the joint cumulative reward of all UAVs, and $\pi(a|s) = P[A_t = a|S_t = s]$ represents the probability of action a under state s. Then the joint action $\mathbf{A}_t = \{a_{t,1}, a_{t,2}, \ldots, a_{t,N}\}$ of UAVs is executed to control the movement of UAVs, the joint state is changed to $\mathbf{S}_{t+1}$ and the reward received by the UAV i is $R_{t,i}$. The cumulative reward of UAV i is defined as Equation (4),

$$J_i^{\pi} = \sum_{t=0}^{T} \gamma^t R_{t,i} \tag{4}$$

where $\gamma \in [0, 1]$ is the discount factor that balances the current rewards and the future rewards.

3 Problem formulation

UAVs use onboard sensors to acquire their internal state information and environmental state information, execute actions according to the DRL model, and obtain the corresponding reward. Figure 3 describes the problem formulation.

3.1 State space

The state space consists of the internal state of the UAV and the environmental information within the max detection distance d_{det} of onboard sensors. The state space of UAV i can be defined as $s_i = (s_{ui}, o_i)$. $s_{ui} = (p_i, v_i, r_i)$ is the internal state of UAV i, which is composed of the position $p_i = (x_i, y_i, z_i)$, the velocity v_i and the radius r_i of UAV i. $o_i = (s_T, s_U, s_O)$ is the environmental information observed by UAV i, where $s_T = (p_t, r_t)$ is the relative position $p_t = (x_t - x_i, y_t - y_i, z_t - z_i)$ to the target with the radius r_t, $s_U = (p_u, v_u, r_u)$ is the relative position $p_u = (x_u - x_i, y_u - y_i, z_u - z_i)$ to other UAVs with the velocity v_u and the radius r_u. s_O represents the state of obstacles. If obstacles are within the max detection range, $s_O = (p_o, v_o, r_o)$ is the relative position $p_o = (x_o - x_i, y_o - y_i, z_o - z_i)$, the velocity v_o and the radius r_o of the obstacles, otherwise, $s_O = (\pm d_{\text{det}}, \pm d_{\text{det}}, \pm d_{\text{det}}, 0, 0)$.

3.2 Action space

In this paper, the action space of UAV i is defined as $a_i = (F_{iX}, F_{iY}, F_{iZ})$ as shown in Figure 3B, where the F_{iX}, F_{iY}, F_{iZ} represent the component forces applied to UAV i in X, Y, and Z three directions, respectively. The force produces an acceleration to change the velocity of the UAV.

3.3 Reward function

In this paper, the goal of the reward function is to guide UAVs to fly to the assigned target without any collision. In order to address the problem of underperforming training efficiency caused by sparse rewards, the reward function in this article uses a combination of guided rewards and sparse rewards. In the process of interacting with the environment, if a UAV reaches the target, collides with other UAVs, or hits an obstacle, a sparse reward is applied; when none of these three situations occurs, a guided reward is applied.

(1) Approaching the target

This reward function is to guide the UAV to head for the target and reach the target. When a UAV moves away from the target, it will receive a larger penalty related to the distance between the UAV and the target, and a reward of value 0 will be given to the UAV when it arrives at the target. Consequently, the reward for UAV i

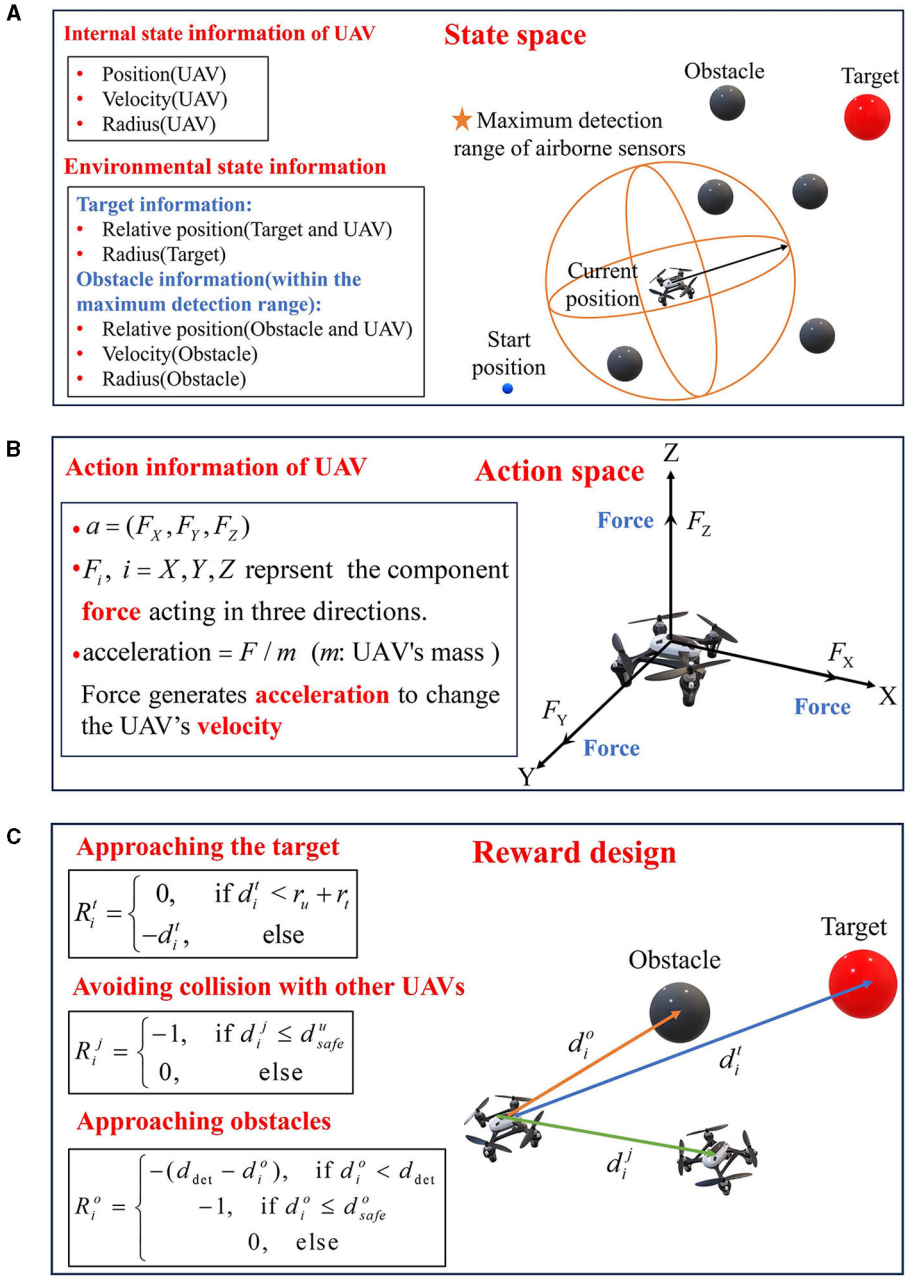

FIGURE 3
Problem formulation. **(A)** State space. **(B)** Action space. **(C)** Reward design.

approaching the target can be defined as Equation (5),

$$R_i^t = \begin{cases} 0, & \text{if } d_i^t < r_u + r_t \\ -d_i^t, & \text{else} \end{cases} \tag{5}$$

where d_i^t denotes the distance between UAV i and the target, r_u is the radius of UAVs, r_t is the radius of targets.

(2) Avoiding collision with other UAVs

This reward is to avoid collision with other UAVs in the process of approaching the target. When the distance between UAV i and UAV j is shorter than their minimum safe distance, a collision will occur and the penalty value is set as Equation (6),

$$R_i^j = \begin{cases} -1, & \text{if } d_i^j \le d_{safe}^u \\ 0, & \text{else} \end{cases} \tag{6}$$

where d_i^j represents the distance between UAV i and UAV j, $d_{safe}^u = 2r_u$ is the minimum safe distance between UAVs.

(3) Avoiding obstacles

The aim of this reward function is to keep UAVs away from obstacles. If the obstacle appears within the detection range d_{det}, the UAV will obtain a punishment, and the closer the UAV gets to the obstacle, the greater the penalty. When the distance between

the UAV and the obstacle is less than their minimum safe distance, a penalty of −1 will be given to the UAV.

$$R_i^o = \begin{cases} -(d_{\text{det}} - d_i^o), & \text{if } d_i^o < d_{\text{det}} \\ -1, & \text{if } d_i^o \le d_{safe}^o \\ 0, & \text{else} \end{cases} \tag{7}$$

In Equation (7), d_i^o is the distance between UAV i and the nearest obstacle within the detection range, $d_{safe}^o = r_u + r_o$ is the minimum safe distance between UAV and obstacles, r_o is the radius of obstacles.

In conclusion, the reward function received by UAV i can be summarized as Equation (8),

$$R_i = R_i^t + R_i^j + R_i^o \tag{8}$$

As can be seen from the reward function designed in this article, the guided reward functions are all negative. It means that each additional step taken by the UAV will have a negative value as a step penalty before reaching the target. Therefore, the reward value in this article can reflect the length of the flight path. A longer flight path corresponds to a smaller reward value.

4 Algorithm

In this section, the proposed algorithm, TANet-TD3, is illustrated in detail.

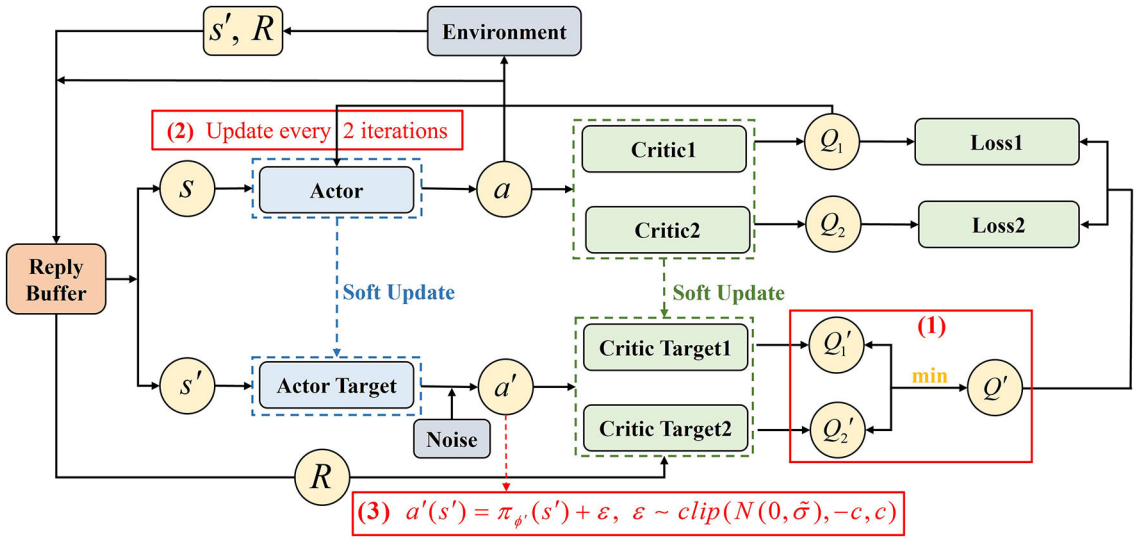

FIGURE 4
TD3 algorithm architecture.

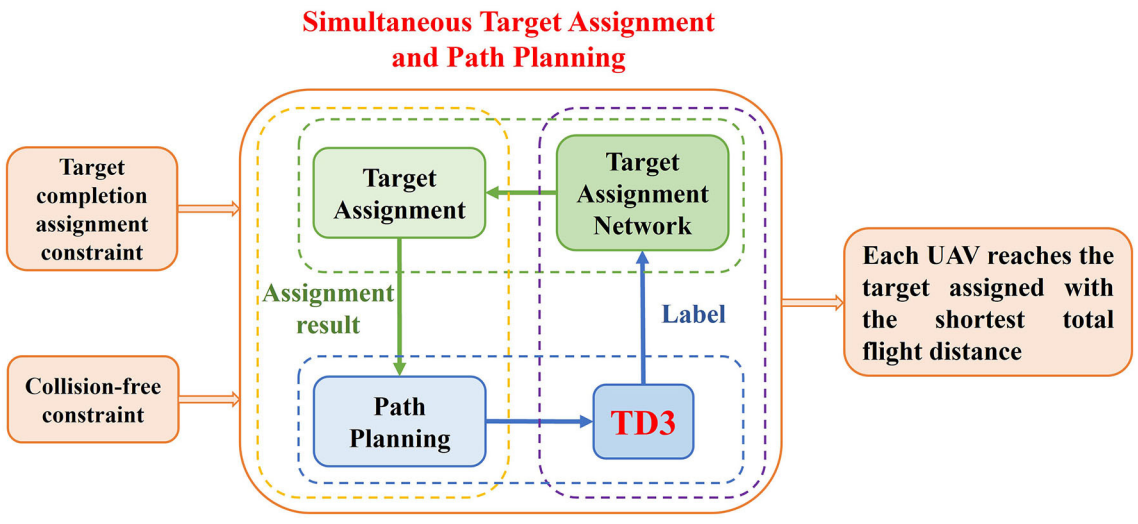

FIGURE 5
The framework of TANet-TD3.

4.1 TD3 algorithm

This paper uses the TD3 as a basic algorithm to address the multi-UAV target assignment and path planning problem. As an improvement of the DDPG algorithm, TD3 also uses an Actor-Critic structure, but it introduces three technologies to prevent the overestimation problem of DDPG:

(1) Clipped double-Q learning.

TD3 has two Critic networks Q_{θ_n} parameterized by $\theta_n, n = 1, 2$ and two Critic target networks $Q_{\theta'_n}$ parameterized by $\theta'_n, n = 1, 2$. The smaller one of two target Q-values is used to calculate the target value function y [Equation (9)] to alleviate the overestimation problem of the value function, as shown in (1) of Figure 4.

$$y = R(s, a) + \gamma \min_{n=1,2} Q_{\theta'_n}(s', a') \tag{9}$$

Therefore, the two Critic networks are updated by minimizing the loss function as Equation (10),

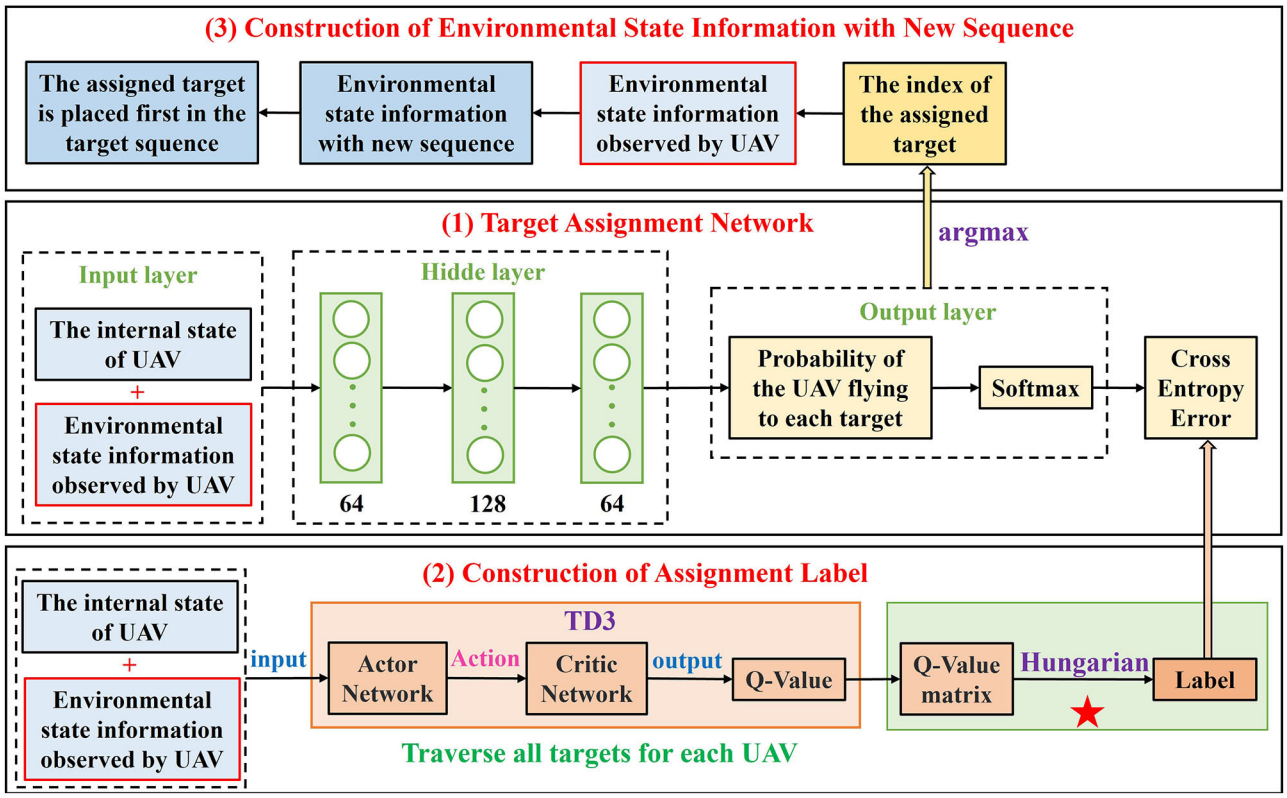

FIGURE 6
The framework of target assignment.

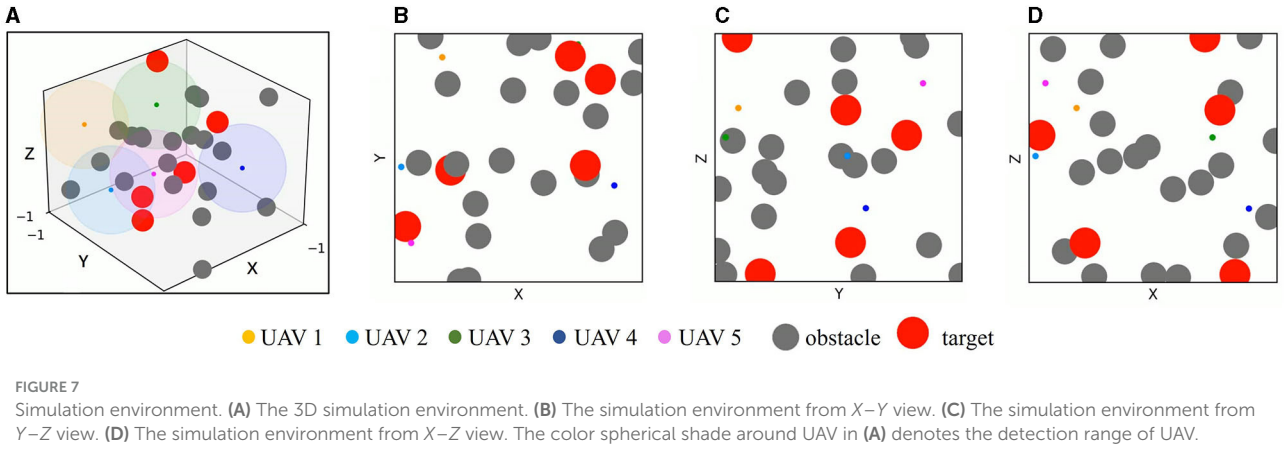

FIGURE 7
Simulation environment. **(A)** The 3D simulation environment. **(B)** The simulation environment from $X-Y$ view. **(C)** The simulation environment from $Y-Z$ view. **(D)** The simulation environment from $X-Z$ view. The color spherical shade around UAV in **(A)** denotes the detection range of UAV.

$$\theta_n \leftarrow \arg\min_{\theta_n} N^{-1} \sum (Q_{\theta_n}(s,a) - y)^2,\ \mathrm{n} = 1,2 \qquad (10)$$

(2) Delayed policy update.

TD3 updates the policy after getting an accurate estimation of the value function to ensure more stable training of the Actor-network. Usually updating the Actor once when the Critic is updated twice, as shown in (2) of Figure 4.

(3) Target policy smoothing.

The policy in DDPG is susceptible to influence by the function approximation error. TD3 adds the clipped noise into the target policy to make the value estimate more accurate, as shown in (3) of Figure 4.

$$a'(s') = \pi_{\phi'}(s') + \varepsilon,\ \varepsilon \sim clip(N(0,\tilde{\sigma}), -c, c) \qquad (11)$$

In Equation (11), s' and a' represent the state and action at the next time, respectively. $\pi_{\phi'}$ represents the Actor target network with the parameter ϕ' and ε denotes the clipped noise.

Each UAV executes action a to transform the state s to next state s', and obtains a reward R from the environment. The data (s, a, R, s') is stored in the replay buffer D as a tuple. Sample a minibatch transition randomly from D, and input the s' into the Actor target network $\pi_{\phi'}$ to get the next action a'. Then, input the (s', a') into the two Critic target networks $Q_{\theta'_1}$, $Q_{\theta'_2}$ to calculate the Q-values and select the smaller one to calculate the target value y. In the meantime, input (s, a) into the two Critic network Q_{θ_1}, Q_{θ_2} and calculate the MSE with y to update the parameters θ_1, θ_2 of two Critic networks. After that, input the Q-value acquired from Critic network Q_{θ_1} into the Actor-network π_ϕ, and update its parameter ϕ in the direction of increasing the Q-value as Equation (12),

$$\nabla_\phi J(\phi) = N^{-1} \sum \nabla_a Q_{\theta_1}(s,a)|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s) \qquad (12)$$

Finally, the target Actor network' parameter ϕ' and the two target Critic networks' parameters θ'_1, θ'_2 are updated by soft update as follows Equation (13) and Equation (14),

$$\theta_n{}' = \tau\theta_n + (1-\tau)\theta_n{}',\ n = 1,\ 2, \qquad (13)$$

$$\phi' = \tau\phi + (1-\tau)\phi'. \qquad (14)$$

4.2 TANet-TD3

4.2.1 Framework of the TANet-TD3

This paper proposed the twin-delayed deep deterministic policy gradient algorithm with target assignment network (TANet-TD3), different from the existing methods that assign targets for the whole task first and then planning the path according to the assignment results, TANet-TD3 can solve the multiple UAVs target assignment and path planning simultaneously in dynamic multi-obstacle environments. The framework of the TANet-TD3 is shown in Figure 5, it can be seen that the object of the task is to minimize the total flight path length of all UAVs with the complete target assignment constraint and collision-free constraint. TANet-TD3 introduces a target assignment network into the framework of TD3 to solve the two problems simultaneously. Among the overall process, the target assignment network provides the optimal complete assignment of targets for each step of UAVs (the green dashed box), and then the TD3 algorithm guides each UAV plan a feasible path for this step (the blue dashed box) according to the assigned result (the yellow dashed box). In the meantime, the training labels of assignment network are obtained from the process of path planning driven by TD3 algorithm (the purple dashed box). This method not only takes into account the distance between UAVs and targets but also considers the dynamic obstacles in task environments, so it can generate an optimal assignment and path.

4.2.2 Framework of target assignment

Figure 6 illustrates the overall framework of target assignment. It is composed of three parts, including the target assignment network, construction of the assignment label, and construction of the environmental state information with new sequence.

(1) Target assignment network

The network structure of target assignment network is designed as the middle section of Figure 6, it consists of a $(7 + 4(N_U - 1) + 4N_T + 7N_O) \times 64 \times 128 \times 64 \times N_T$ fully-connected neural network

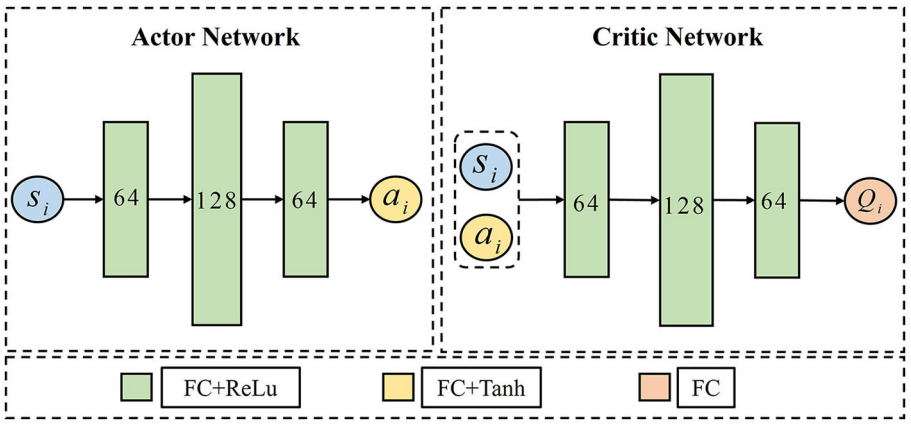

FIGURE 8
The network architecture of Actor and Critic in TD3.

layers, where $(7 + 4(N_U - 1) + 4N_T + 7N_O)$ represents the state information $s_i = (s_{ui}, o_i)$ of each UAV under the scenario of N_U UAVs, N_T targets and N_O obstacles within the detection range. For UAV i, after four FCs, the target assignment network maps the state information $s_i = (s_{ui}, o_i)$ to the probability $(P_{iT_1}, P_{iT_2}, \cdots, P_{iT_{N_T}})$ of UAV i flying to targets $(T_1, T_2, \cdots, T_{N_T})$. The probability is first normalized by the Softmax function [Equation (15)],

$$p_{ij} = \frac{p_{iT_j}}{\sum_j^{N_T} p_{iT_j}} \tag{15}$$

and then the Cross-Entropy calculation is performed with the assigned labels to update the assignment network [Equation (16)],

$$H(L, P) = -\sum_j^{N_T} l_j \log p_{ij} \tag{16}$$

(2) Construction of the assignment label

From the bottom section of the Figure 6, it can be seen that the training labels of the assignment network are provided by TD3 framework. The task objective is to achieve a complete assignment and minimize the total flight path, but it is not accurate to only consider the distance between UAVs and targets to make decisions in random and dynamic environments. As mentioned in Section 2.2, a multi-UAV problem means to maximize the joint cumulative reward of all UAVs in DRL, that is, each UAV will choose the action that maximized the Q-value based on its current state. Compared with selecting the target only according to distance, this method determines the assigned target according to the Q-value comprehensively taking into account UAVs, targets, and obstacles, even if obstacles are moving, so the targets can get an optimal assignment.

For UAV i, a $1 \times N_T$ Q-value list $(Q_{i1}, Q_{i2}, \ldots, Q_{iN_T})$ can be obtained for each step from the initial position by considering each target T_j, $j = 1, 2, \ldots, N_T$ as the destination the UAV i will eventually reach, and for N_U UAVs, a $N_U \times N_T$ value matrix is formed as Equation (17) by traversing all targets,

$$\begin{pmatrix} Q_{11}, & Q_{12}, & \ldots, & Q_{1N_T} \\ Q_{21}, & Q_{22}, & \ldots, & Q_{2N_T} \\ \ldots, & \ldots, & \ldots, & \ldots, \\ Q_{N_U1}, & Q_{N_U2}, & \ldots, & Q_{N_UN_T} \end{pmatrix} \tag{17}$$

In order to ensure the constraints of complete target assignment, among many methods, the Hungarian algorithm has fast solution speed and stable solution quality, and with the aid of the independent 0 element theorem, it can obtain the exact solution of the problem by making elementary changes for the matrix with finite steps. Therefore, the Hungarian algorithm is introduced to achieve a complete allocation for targets in this article. After the Hungarian transformation, the Q-value matrix can be transformed into a permutation matrix with only 0 and 1 elements such as in Equation (18)

$$\begin{pmatrix} Q_{11}, & Q_{12}, & \ldots, & Q_{1N_T} \\ Q_{21}, & Q_{22}, & \ldots, & Q_{2N_T} \\ \ldots, & \ldots, & \ldots, & \ldots, \\ Q_{N_U1}, & Q_{N_U2}, & \ldots, & Q_{N_UN_T} \end{pmatrix} \xrightarrow{\text{Hungarian}} \begin{array}{c} \\ i \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{matrix} \quad & \quad & j & \quad \end{matrix} \\ \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 1 & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 \end{pmatrix}_{N_U \times N_T} \end{array} \tag{18}$$

if element 1 of row i is located in column j, it means that the j-th target is assigned to the i-th UAV. Thus, the target assignment can be achieved according to the Q-value $\tilde{Q}_{i,j}$ of each step, and the result of Hungarian transformation is used as the training label of the target assignment network.

(3) Construction of environmental state information with a new sequence

After the target assignment network of UAV i has been fully trained, a list of probabilities of UAV i moving to each target in the current state can be obtained, among which the assigned target has the largest probability in the list. As shown in the top section of the Figure 6, if the target T_j is assigned to UAV i, then the index of target T_j can be calculated by Equation (19),

$$index(T_j) = \text{argmax}(P_{i1}, P_{i2}, \ldots, P_{iN_T}) \tag{19}$$

The original environmental state information $o_i = (s_{T_1}, s_{T_2}, \cdots, s_{T_{N_T}}, s_U, s_O)$ can be transformed into the environmental state information with new targets sequence $\tilde{o}_i = (s_{T_j}, s_{T_1}, \cdots, s_{T_{j-1}}, s_{T_{j+1}}, \cdots, s_{T_{N_T}}, s_U, s_O)$, that is the assigned target T_j is placed in the first place of the target sequence to guide UAV i to recognize its own target.

The target assignment network realizes the optimal target assignment every step in the dynamic environment, and then UAV i uses the TD3 algorithm to plan the path for the assigned target according to the new state information $\tilde{s}_i = (s_{ui}, \tilde{o}_i)$. The Actor network updates according to the $\tilde{Q}_{i,n}$ and the state information $\tilde{s}_i$ with new sequence using Equation (20),

$$\nabla_{\phi_i} J(\phi_i) = N^{-1} \sum \nabla_a \tilde{Q}_{\theta_{i,1}}(\tilde{s}_i, a_i)|_{a=\pi_{\phi_i}(\tilde{s}_i)} \nabla_{\phi_i} \pi_{\phi_i}(\tilde{s}_i) \tag{20}$$

the TANet-TD3 is described in Algorithm 1.

5 Experiments and results

In this section, the simulation environment is introduced first. Then, the training experiments, testing experiments, and statistical experiments are presented to verify the effectiveness of the proposed method in different scenarios.

TABLE 1 The hyperparameters of TANet-TD3.

No	Hyperparameters	Values
1	Max episodes number of TD3	5,000
2	Max episodes number of TANet-TD3	10,000
3	Max episodes length	100
4	Discount factor	0.9
5	Critic learning rate	1E-3
6	Actor learning rate	1E-4
7	Reply buffer size	5E5
8	Batch size	256
9	Soft update factor	0.01

Initialize Critic networks $Q_{\theta_{i,1}}$, $Q_{\theta_{i,2}}$ and Actor-network π_{ϕ_i} with random parameters $\theta_{i,1}$, $\theta_{i,2}$, ϕ_i for each UAV i;
Initialize target networks for each UAV i, $\theta_{i,1}{}' \leftarrow \theta_{i,1}$, $\theta_{i,2}{}' \leftarrow \theta_{i,2}$, $\phi_i{}' \leftarrow \phi_i$;
Initialize rely buffer D;
For episode=1 to Max-episodes do
  For step=1 to Max-step do
    Select action with exploration noise for each UAV i
    Observe reward R and new state s'_i.
    Store transition tuple (s_i, a, R, s'_i) in D
    Randomly sample a mini-batch samples from D
    Calculate target actions using Equation (11)
    Calculate Q-targets using Equation (9)
    Update $\theta_{i,1}$ and $\theta_{i,2}$ using Equation (10)
    Calculate the $(Q_{i1}, Q_{i2}, \ldots, Q_{iN_T})$ for each UAV i
    Obtain the assignment label using Equation (18)
    Update target assignment network using Equation (16)
    Construct the observation $\tilde{s}_i$ with the new sequence
    If t mod d **then**
      Update Actor using Equation (20)
      Update target networks:
      Update $\theta'_{i,1}$ and $\theta'_{i,2}$ using Equation (13)
      Update ϕ'_i using Equation (14)
    End if
  End for
End for

Algorithm 1. TANet-TD3

5.1 Experimental settings

A 3D simulation environment with two-dimensional three views is constructed based on the OpenAI platform to implement multi-UAV simultaneous target assignment and path planning in dynamic multiple obstacle environments. As shown in Figure 7, the simulation environment covers a 2 × 2 × 2 cubic area, UAVs, targets, and obstacles are simplified to a sphere and randomly initialized in this area. The radius of UAVs $r_u = 0.02$, and the maximum detection range d_{det} of UAVs is set as 0.5, which is denoted by the color spherical shades around UAVs. The radius of targets is set to $r_t = 0.12$. The obstacles have static mode and mobile mode with a radius $r_o = 0.1$, In motion mode, they move in a linear motion with a randomly initialized direction and velocity $v_i \in [-0.05, 0.05]$, $i \in [X, Y, Z]$. v_i represents the sub-velocity of obstacles in the X, Y, Z three directions. When it hits the boundary of the simulation environment, it moves in the opposite direction with the same velocity.

In this paper, the network of TD3 is shown in Figure 8, N UAVs include N Actor-Critic structures. For UAV i, the Actor network is constructed by $s_i \times 64 \times 128 \times 64 \times a_i$, where the input s_i represents the state of UAV i, and the output a_i represents the action performed by UAV i. The first three layers use a rectified linear unit (Relu) as the activation function, and the last layer uses a hyperbolic tangent (tanh) activation function to limit the output of action within the range of [−1,1]. The Critic owns a network structure of $(s_i + a_i) \times 64 \times 128 \times 64 \times Q_i$, after three fully connected neural network layers (FCs) activated by Relu, the Critic maps the combination of state and action of UAV i to the Q-value evaluated by UAV i. The hyperparameters of TANet-TD3 and TD3 are given in Table 1.

5.2 Training experiments

Training experiments include two sections, the first section is to verify the advantages of TD3 in path planning, and the second section is to validate the effectiveness of TANet-TD3 in multi-UAV simultaneous target assignment and path planning. These algorithms have been trained in dynamic and mixed task environments as depicted in Figure 7, and in each episode, UAVs, targets, and obstacles are randomly initialized in the task area. There are three indicators used to measure the performance of training shown in Equation (21), including the average reward, the average arrival rate and the average target completion rate, where N_{ver} is the number of verification episodes, $R_{i\text{-th}}$ is the reward of the i-th verification episode, N_i^U is the number of UAVs that reach the target in the i-th verification episode and N_i^T is the number of targets that have UAV reached in the i-th verification episode.

$$\begin{cases} \text{Average reward} = \sum_{i=1}^{N_{ver}} R_{i-\text{th}}/N_{ver} \\ \text{Average arrival rate} = \sum_{i=1}^{N_{ver}} N_i^U/(N_{ver} \times N_U) \\ \text{Average target completion rate} = \sum_{i=1}^{N_{ver}} N_i^T/(N_{ver} \times N_T) \end{cases} \tag{21}$$

5.2.1 Training experiments for path planning tasks

Firstly, the TD3 algorithm is trained in single-UAV and multi-UAV dynamic scenarios respectively. Scenario I: one UAV, one target, and 20 moving obstacles; Scenario II: three UAVs, one target, and 20 moving obstacles. Each experiment is trained for 5,000 episodes, and 50 episodes of verification are conducted after every 10 episodes of training. The average reward and average arrival rate of these 50 verification episodes are counted to evaluate the algorithm. As a comparison, the DDPG algorithm is trained in the same task scenarios as TD3 with the same hyperparameters in Table 1.

As can be seen from the training results depicted in Figures 9A, B, after adequate training, the TD3 algorithm has a good average arrival rate for path planning tasks, whether the scenario I of a single UAV (95%) or the scenario II of multiple UAVs (90%). It is evident that compared to the DDPG algorithm, TD3 has a better convergence effect and a faster convergence speed.

Therefore, in this paper, the TD3 algorithm is used as the basic algorithm for path planning, which can provide accurate assignment labels for the training of the target assignment network.

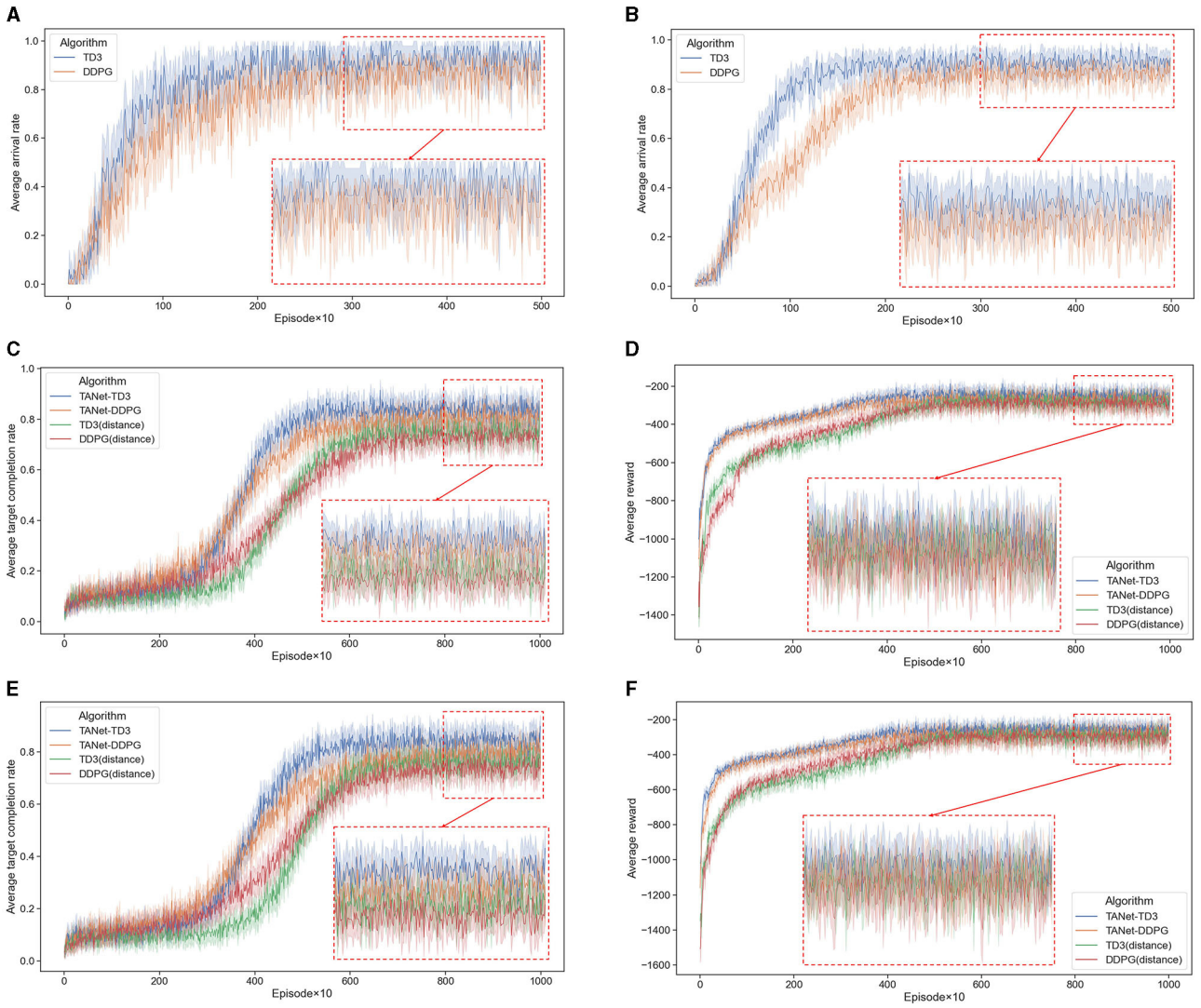

FIGURE 9
Convergence curves. **(A)** The average arrival rate of DDPG and TD3 in scenario I. **(B)** The average arrival rate of DDPG and TD3 in scenario II. **(C)** The average target completion rate of TANet-DDPG and TANet-TD3 in dynamic environment. **(D)** The average reward of TANet-DDPG and TANet-TD3 in dynamic environment. **(E)** The average target completion rate of TANet-DDPG and TANet-TD3 in mixed environment. **(F)** The average reward of TANet-DDPG and TANet-TD3 in mixed environment. The solid line denotes the statistical means and the 95% confidence interval of the means is shown shaded.

5.2.2 Training experiments for simultaneous target assignment and path planning tasks

Next, the proposed algorithm TANet-TD3 is trained in the dynamic environment (five UAVs, five targets, and 20 moving obstacles) and the mixed environment (five UAVs, five targets, 10 static obstacles, and 10 moving obstacles). Each experiment has 10,000 episodes, and 50 episodes of verification are conducted after every 10 episodes of training. To verify the feasibility of the assignment network of TANet-TD3, DDPG with the target assignment network (TANet-DDPG) is introduced for comparison. In addition, the scheme of target assignment based on the distance between the target and UAV is introduced to the DDPG (DDPG(distance)) and TD3 (TD3(distance)) respectively to verify the advantages of TANet-TD3. Four algorithms are trained with the same hyperparameters in Table 1, and the target completion rate and the average reward are used as indicators to evaluate the performance of algorithms.

As shown in Figures 9C, E, in the initial stage, all algorithms generated training samples by the interaction process between UAVs and the environment, and the training started when the number of samples reached the capacity of batch size. The reward is very low and UAVs do not know what the goal is before the first 3,000 episodes. With the gradual rise of the samples in the reply buffer, each UAV gradually began to learn more intelligent strategies and finally reached the convergence result. The training results are listed in Table 2.

Figures 9C, D present that TANet-TD3 has the fastest convergence rate in the dynamic environment, reaching

TABLE 2 The training results of TANet-DDPG and TANet-TD3.

Environment	Algorithm	In episode 5,000		Last 1,000 episodes	
		Average target completion rate	Average reward	Mean average target completion rate	Mean average reward
Dynamic	TANet-DDPG	71.20%	−303.5	80.70%	−271.5
	TANet-TD3	**81.54%**	**−241.4**	**83.77%**	**−253.0**
	DDPG(distance)	51.07%	−301.0	73.06%	−290.4
	TD3(distance)	55.27%	−296.5	75.10%	−275.0
Mixed	TANet-DDPG	63.20%	−305.9	80.38%	−281.1
	TANet-TD3	**78.21%**	**−220.5**	**84.27%**	**−255.2**
	DDPG(distance)	52.38%	−300.3	73.78%	−299.2
	TD3(distance)	48.00%	−295.9	76.28%	−289.2

The bold values represents the training result of our algorithm, and it is optimal among the four algorithms.

TABLE 3 The test statical results of TANet-DDPG and TANet-TD3.

Environments	Algorithms	The number of targets reached by UAVs	Rewards
Dynamic	TANet-DDPG	4	−256.87
	TANet-TD3	5	−141.97
Mixed	TANet-DDPG	2	−452.70
	TANet-TD3	5	−335.05

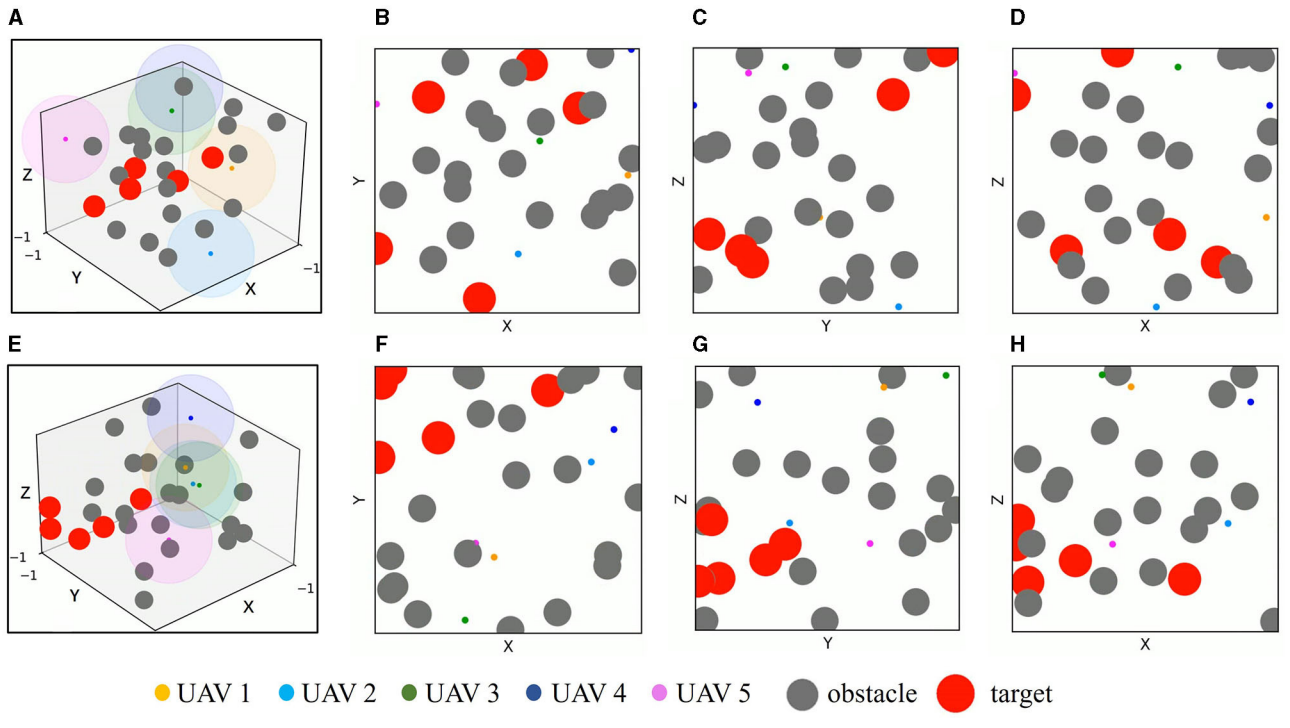

FIGURE 10
The test scenarios. **(A)** The 3D scenario of dynamic environment. **(B)** The dynamic environment from X-Y view. **(C)** The dynamic environment from Y-Z view. **(D)** The dynamic environment from X-Z view. **(E)** The 3D scenario of mixed environment. **(F)** The mixed environment from X-Y view. **(G)** The mixed environment from Y-Z view. **(H)** The mixed environment from X-Z view.

convergence about the 5,000th episode; followed by TANet-DDPG, while the TD3(distance) and DDPG(distance) algorithms reach convergence at about the 7,000th episode. Similarly, Figures 9E, F depict that TANet-TD3 has about 500 accelerated

convergence compared to TANet-DDPG and about 2,000 accelerated convergence compared to TD3 and DDPG. Additionally, the relevant statistics in Table 2 illustrate that TANet-TD3 has the highest average target completion rate and average reward in both the dynamic and mixed environments. Compared to TANet-DDPG, TANet-TD3 leads an increase of (3.07%, 18.5) in a dynamic environment and (3.89%, 25.9) in a mixed environment. It has the largest average target completion rate difference of 10.71% (dynamic environment) and 10.49% (mixed environment) among TANet-TD3 and DDPG (distance).

Overall, the improvement of TANet-TD3 and TANet-DDPG is remarkable compared to DDPG(distance) and TD3(distance), this demonstrated that it is effective for the assignment method proposed in the paper, which can achieve simultaneous target assignment and path planning. Moreover, the training results also illustrated that TANet-TD3 outperforms TANet-DDPG in terms of convergence effect and convergence speed, which is mainly due to the superiority of TD3 in completing path planning tasks, which provides better Q-value labels for the optimization of target assignment network.

5.3 Testing experiments and results

In order to evaluate the application efficiency of the algorithm after convergence and further verify the advantages of the TANet-TD3 algorithm in the simultaneous target assignment and path planning of multiple UAVs, a series of test experiments are conducted, in which the network parameters after convergence of TANet-TD3 and TANet-DDPG are used to control UAVs move in two environments. One is a dynamic environment, where all obstacles are mobile; the other environment is a mixed environment, where obstacles are static or mobile. As shown in Figure 10, UAVs, targets, and obstacles are randomly deployed in task areas, Figure 10A presents the 3D scenario of a dynamic environment with five UAVs, five targets, and 20 mobile obstacles, Figure 10B depicts the 3D scenario of the mixed environment with five UAVs, five targets, 10 static obstacles, and 10 mobile obstacles. Note that the colored spherical shades around UAVs represent the detection range of UAVs.

Figures 11, 12 present the 3D trajectories and corresponding 2D three views of five UAVs derived by TANet-DDPG and TANet-TD3 respectively in the dynamic scenario of Figure 10A. As can be

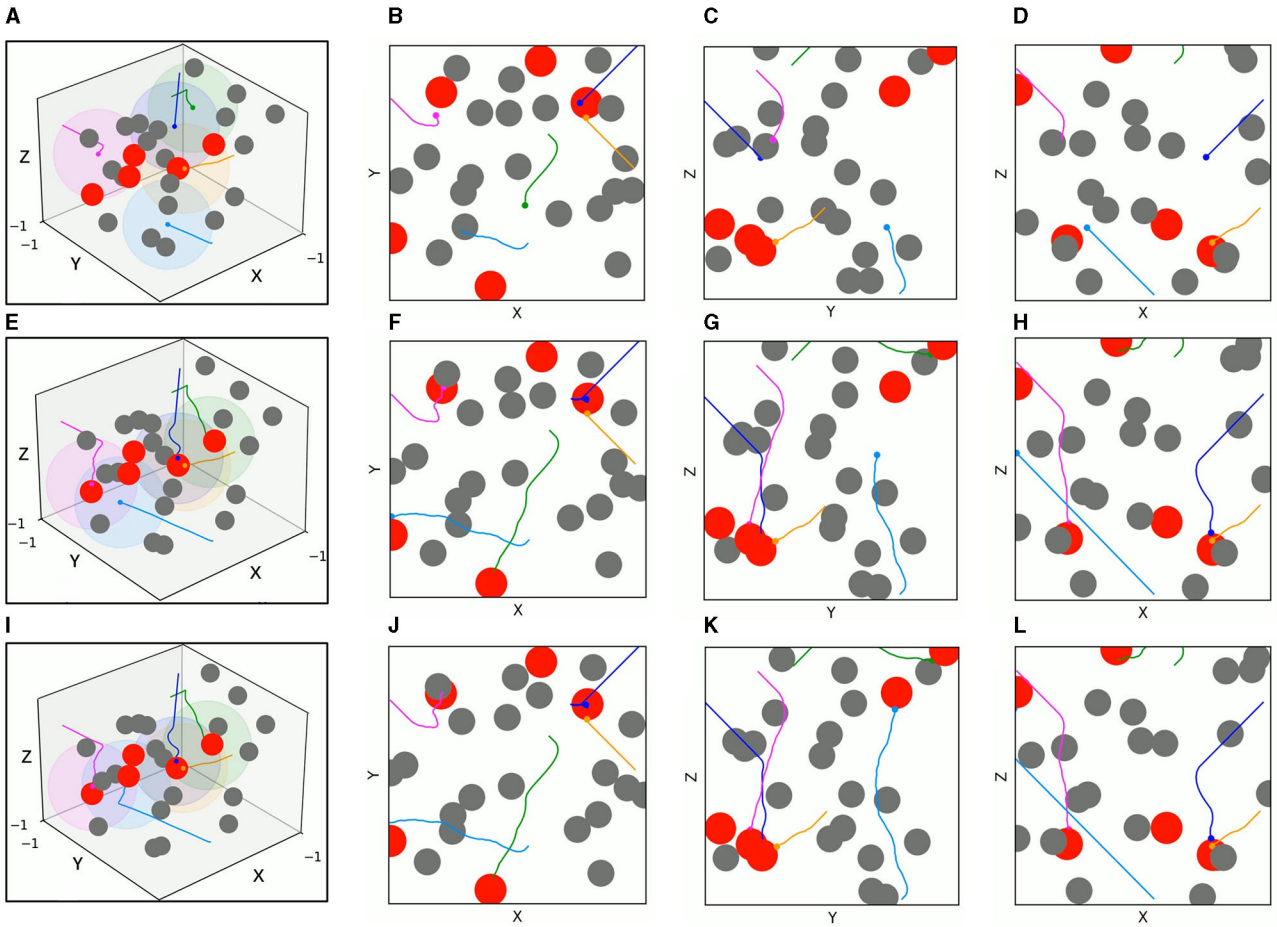

FIGURE 11
The 3D trajectories and corresponding 2D three views of UAVs driven by TANet-DDPG at different times in a dynamic environment. **(A)** The 3D trajectories at $t = 4s$. **(B–D)** The corresponding 2D three views of **(A)**. **(E)** The 3D trajectories at $t = 8s$. **(F–H)** The corresponding 2D three views of **(E)**. **(I)** The 3D trajectories at $t = 12s$. **(J–L)** The corresponding 2D three views of **(I)**.

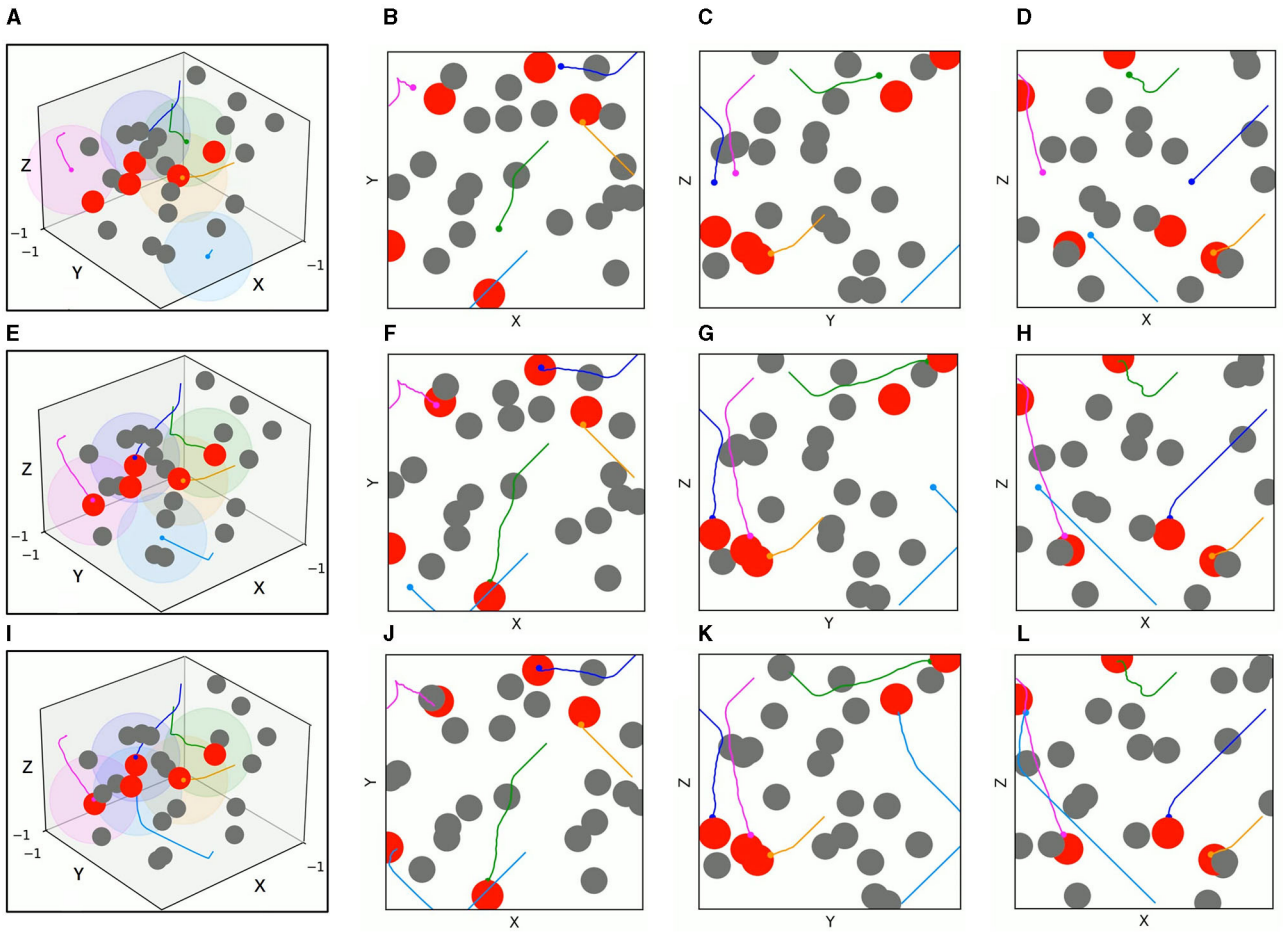

FIGURE 12
The 3D trajectories and corresponding 2D three views of UAVs driven by TANet-TD3 at different times in dynamic environment. **(A)** The 3D trajectories at $t = 4s$. **(B–D)** The corresponding 2D three views of **(A)**. **(E)** The 3D trajectories at $t = 8s$. **(F–H)** The corresponding 2D three views of **(E)**. **(I)** The 3D trajectories at $t = 12s$. **(J–L)** The corresponding 2D three views of **(I)**.

seen in Figure 11, all five UAVs driven by TANet-DDPG reached targets, but UAV 1 and UAV 4 reached the same target resulting in one of five targets not having a UAV arriving for the next mission. Unlike TANet-DDPG, UAVs driven by TANet-TD3 achieve a full assignment of targets, that is a one-to-one correspondence between targets and UAVs. Meanwhile, it is evident that multiple UAVs execute simultaneous target assignment and path planning under TANet-TD3 and have a superior performance in the capability of obstacle avoidance. The trajectories of UAV 2 and UAV 5 in Figure 12 avoided the obstacle intentionally choosing a safer path as they approached the obstacle.

The test results of the two algorithms in the mixed environment of Figure 10B are depicted in Figures 13, 14, respectively. First, UAV 2 and UAV 3 driven by TANet-DDPG failed to reach their assigned target due to hitting moving obstacles during flight, but they adapted well to the uncertain environment driven by TANet-TD3, and both succeeded in reaching their respective targets. Then, UAV 1 flew to the target reached by UAV 5 under the TANet-DDPG planning. In contrast, the test result derived by TANet-TD3 provides a complete assignment and a path without collision.

As a result, TANet-TD3 presents a better adaptability to dynamic environments compared to TANet-DDPG. Besides, the test statical results shown in Table 3 illustrate that TANet-TD3 exceeds TANet-DDPG in both the number of targets reached by UAVs and the reward value. According to the design of the reward function in Section 3.3, the value of reward also reflects the length of the flight path of UAVs, which also indicates that the path derived by TANet-TD3 is shorter than that derived by TANet-DDPG.

5.4 Statistical experiments

In this section, the statistical experiments about different numbers of UAVs and different numbers of obstacles are presented to further verify the advantage of TANet-TD3.

5.4.1 Adaptability to different numbers of UAVs

In this experiment, the average target completion rate of the TANet-DDPG and TANet-TD3 are sequentially

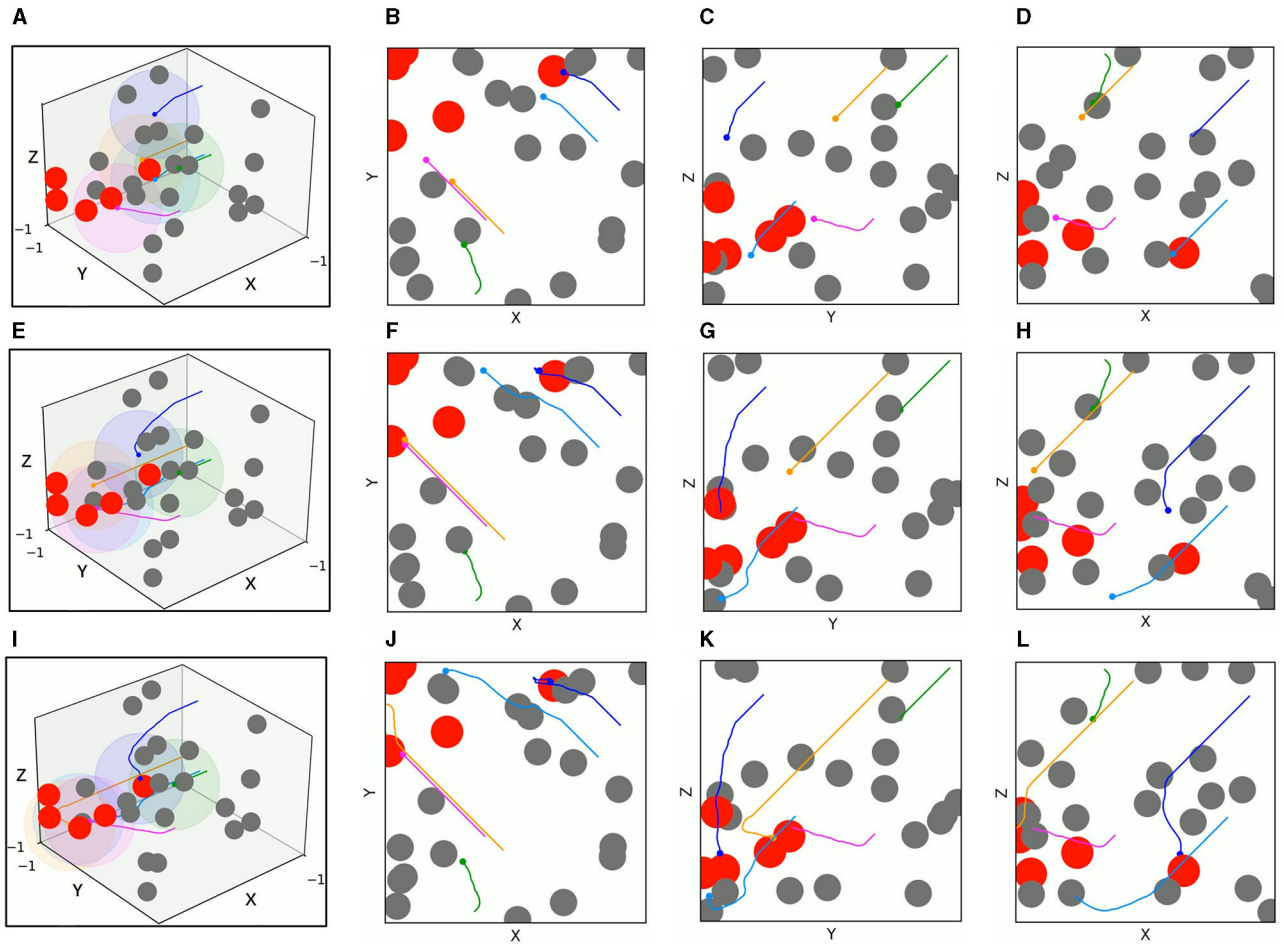

FIGURE 13
The 3D trajectories and corresponding 2D three views of UAVs driven by TANet-DDPG at different times in mixed environments. **(A)** The 3D trajectories at $t = 4s$. **(B–D)** The corresponding 2D three views of **(A)**. **(E)** The 3D trajectories at $t = 8s$. **(F–H)** The corresponding 2D three views of **(E)**. **(I)** The 3D trajectories at $t = 12s$. **(J–L)** The corresponding 2D three views of **(I)**.

compared in terms of the number of UAVs from 3 to 7. The obstacles are set to 20 moving obstacles in a dynamic environment, and 10 static obstacles and 10 moving obstacles in a mixed environment. Each experiment with a specific number of UAVs is repeated 1,000 episodes, and in each episode, UAVs, targets, and obstacles are initialization with random position and velocity. Figures 15A, B depict the statistical results in dynamic environment and mixed environment, respectively.

As the number of UAVs increases, the difficulty of the simultaneous target assignment and path planning tasks increases dramatically, and the average target completion rate of TANet-DDPG and TANet-TD3 gradually decreases in both dynamic and mixed environments. Faced with the complex mission scenario of seven UAVs and 20 obstacles, TANet-TD3 can maintain an average target completion rate of more than 71% (71.54%, 71.06%). In contrast, TANet-DDPG has dropped to just over 70% (70.35%, 70.45%) at six UAVs and falls sharply below 65% (64.93%, 63.03%) at the seven UAVs. In addition, as the number of UAVs increases, the gap between TANet-DDPG and TANet-TD3 grows wider.

5.4.2 Adaptability to different number of obstacles

This experiment verifies the effect of different numbers of obstacles on TANet-TD3 and TANet-DDPG. Specifically, the two algorithms are compared in dynamic and mixed environments with five UAVs and different numbers of obstacles including 10, 15, 20, 25, and 30, respectively. Each experiment is repeated 1,000 episodes, and the state of UAVs, targets, and obstacles are randomly initialized for each episode. The comparison results of the average target completion rate are presented in Figures 15C, D.

As shown in Figure 15, the increase in the number of obstacles has affected the performance of two algorithms both in dynamic and mixed environments, but TANet-TD3 consistently outperforms TANet-DDPG in all scenarios.

10.3389/fnbot.2023.1302898

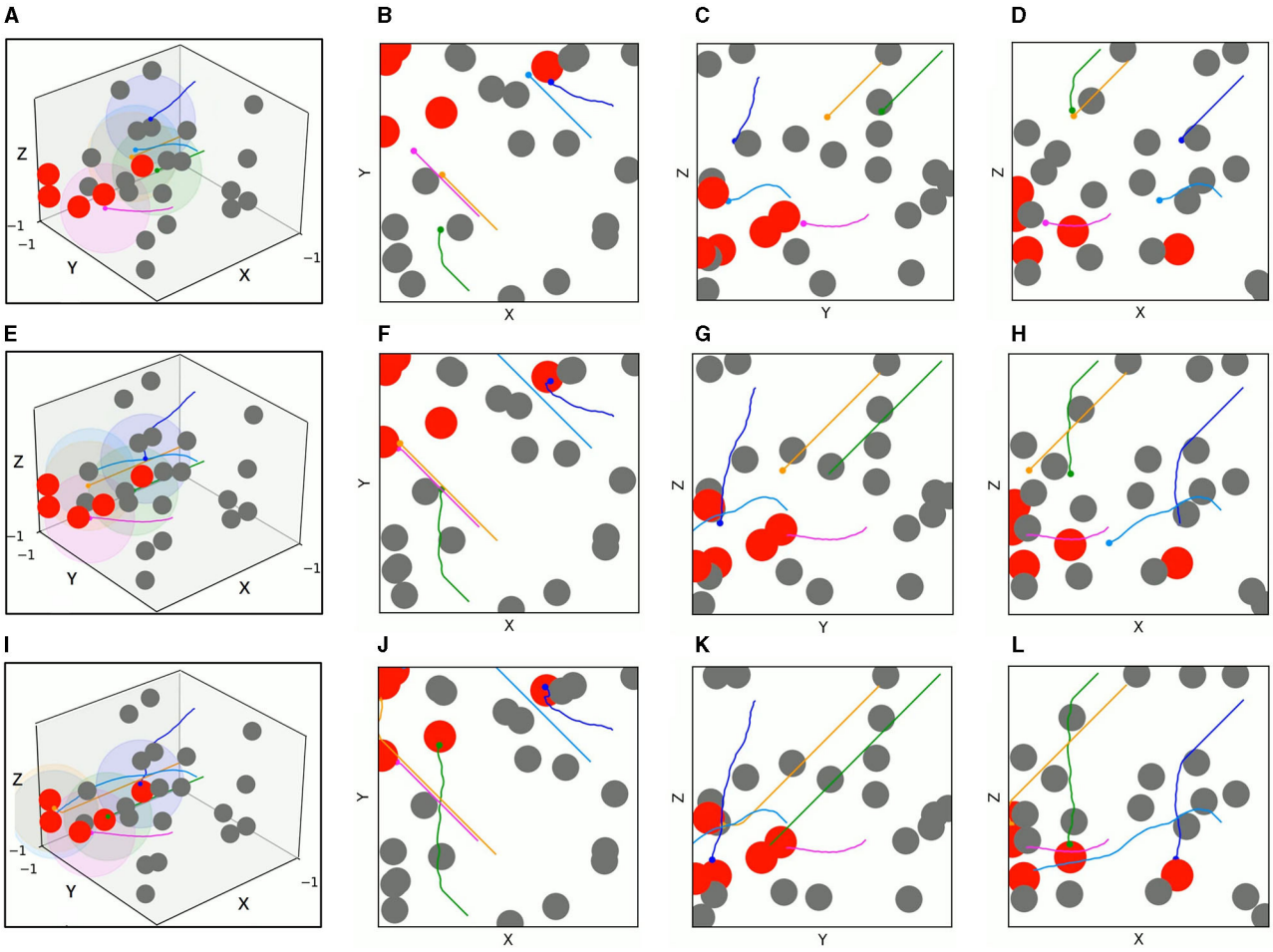

FIGURE 14
The 3D trajectories and corresponding 2D three views of UAVs driven by TANet-TD3 at different times in mixed environment. **(A)** The 3D trajectories at $t = 4s$. **(B–D)** The corresponding 2D three views of **(A)**. **(E)** The 3D trajectories at $t = 8s$. **(F–H)** The corresponding 2D three views of **(E)**. **(I)** The 3D trajectories at $t = 12s$. **(J–L)** The corresponding 2D three views of **(I)**.

Additionally, when the number of obstacles is 25, the average target completion rate of TANet-DDPG is below 80% in both dynamic (79.36%) and mixed environments (79.35%), while the average target completion rate of TANet-TD3 remains above 81% (81.36%, 82.40%) under the complex environment with 30 obstacles.

In summary, TANet-TD3 can effectively complete simultaneous target assignment and path planning. Besides, it has demonstrated that TANet-TD3 has a better adaptability to dynamic and random environments compared with TANet-DDPG.

6 Conclusion and discussion

This paper proposes a novel DRL-based method TANet-TD3 for multiple UAVs target assignment and path planning in dynamic multi-obstacle environments. The problem is formulated as a POMDP and a target assignment network is introduced to the TD3 algorithm to complete the target assignment and path planning simultaneously. Specifically, each UAV considers each target as its final target to be reached in turn and executes its action derived by TD3 for the next step. A Q-value matrix can be obtained by reward function and the Hungarian algorithm is used to act on the Q-value matrix to achieve an exact match between UAVs and targets. The matching result is used as labels to train the target assignment network, so as to obtain the optimal allocation for targets. Then each UAV moves to its assigned target under the planning of the TD3 algorithm. The experiment results demonstrate that TANet-TD3 can achieve simultaneous target assignment and path planning in dynamic multiple obstacle environments, and the performance of TANet-TD3 outperforms the existing methods in both convergence speed and target completion rate.

For future research, we will further improve the proposed method by combining it with specific applications, such as multi-UAV target search tasks and multi-UAV target-tracking tasks. Additionally, we will study the method of calculating the Q-value matrix in high-dimensional scenarios to deal with complex tasks with a large number of targets. Furthermore, we will build a

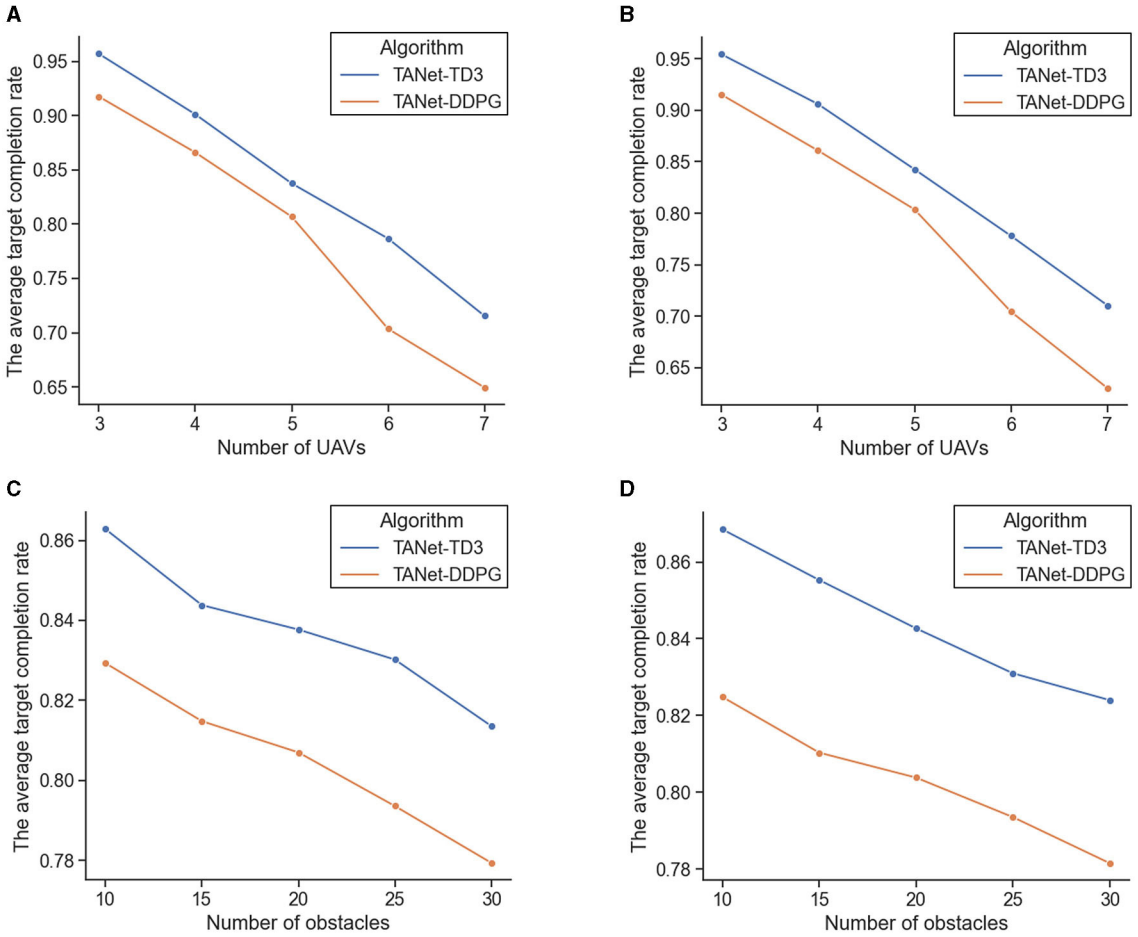

FIGURE 15
The comparison result of the average target completion rate of TANet-DDPG and TANet-TD3. **(A)** The comparison result under different numbers of UAVs in a dynamic environment. **(B)** The comparison result under different numbers of UAVs in mixed environments. **(C)** The comparison result under different numbers of obstacles in a dynamic environment. **(D)** The comparison result under different numbers of obstacles in a mixed environment.

more realistic simulation environment, in which the shape and movement of obstacles are more complex, to verify the effectiveness of the proposed algorithm.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

XK: Conceptualization, Methodology, Software, Validation, Visualization, Writing—original draft, Writing—review & editing. YZ: Supervision, Writing—review & editing. ZL: Supervision, Writing—review & editing. SW: Supervision, Writing—review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work was supported by the Special Foundation for Beijing Tianjin Hebei Basic Research Cooperation (J210008, H2021202008), and the Inner Mongolia Discipline Inspection and Supervision Big Data Laboratory (IMDBD202105).

Acknowledgments

The authors would like to thank all reviewers and editors for their comments on this study.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Aggarwal, S., and Kumar, N. (2020). Path planning techniques for unmanned aerial vehicles: a review, solutions, and challenges. *Comput. Commun*. 149, 270–299. doi: 10.1016/j.comcom.2019.10.014

Busoniu, L., Babuska, R., and De Schutter, B. (2008). A comprehensive survey of multiagent reinforcement learning. *IEEE Trans. Syst. Man. Cybern. C. Appl. Rev*. 38, 156–172. doi: 10.1109/TSMCC.2007.913919

Chamola, V., Kotesh, P., Agarwal, A., Gupta, N., Guizani, M., et al. (2021). A comprehensive review of unmanned aerial vehicle attacks and neutralization techniques. *Ad hoc Netw*. 111:102324. doi: 10.1016/j.adhoc.2020.102324

Chane-Sane, E., Schmid, C., and Laptev, I. (2021). "Goal-conditioned reinforcement learning with imagined subgoals," in *International Conference on Machine Learning* (Stockholm: PMLR), 1430–1440.

Chen, H., Lan, Y., Fritz, B. K., Hoffmann, W. C., and Liu, S. (2021). Review of agricultural spraying technologies for plant protection using unmanned aerial vehicle (UAV). *Int. J. Agric. Biol. Eng.* 14, 38–49. doi: 10.25165/j.ijabe.20211401.5714

Fan, J., Chen, X., and Liang, X. (2023). UAV trajectory planning based on bi-directional APF-RRT* algorithm with goal-biased. *Expert Syst. Appl*. 213:119137. doi: 10.1016/j.eswa.2022.119137

Fei, B., Bao, W., Zhu, X., Liu, D., Men, T., Xiao, Z., et al. (2022). Autonomous cooperative search model for multi-UAV with limited communication network. *IEEE Internet Things J*. 9, 19346–19361. doi: 10.1109/JIOT.2022.3165278

Fernandes, P. B., Oliveira, R., and Neto, J. F. (2022). Trajectory planning of autonomous mobile robots applying a particle swarm optimization algorithm with peaks of diversity. *Appl. Soft Comput*. 116:108108. doi: 10.1016/j.asoc.2021.108108

Fujimoto, S., Hoof, H., and Meger, D. (2018). "Addressing function approximation error in actor-critic methods," in *International Conference on Machine Learning* (PMLR), 1587–1596.

Gerkey, B. P. Matarić, M. J. (2004). A formal analysis and taxonomy of task allocation in multi-robot systems. *Int. J. Robot. Res*. 23, 939–954. doi: 10.1177/0278364904045564

Grenouilleau, F., Van Hoeve, W-.J., and Hooker, J. N. (2019). "A multi-label a* algorithm for multi-agent pathfinding," *in Proceedings of the International Conference on Automated Planning and Scheduling* (Berkeley), Volume 29, 181–185. doi: 10.1609/icaps.v29i1.3474

Han, R., Chen, S., and Hao, Q. (2020). "Cooperative multi-robot navigation in dynamic environment with deep reinforcement learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)* (Paris: IEEE), 448–454. doi: 10.1109/ICRA40945.2020.9197209

He, L., Aouf, N., and Song, B. (2021). Explainable deep reinforcement learning for UAV autonomous path planning. *Aerosp. Sci. Technol*. 118:107052. doi: 10.1016/j.ast.2021.107052

He, W., Qi, X., and Liu, L. (2021). A novel hybrid particle swarm optimization for multi-UAV cooperate path planning. *Appl. Intell*. 51, 7350–7364. doi: 10.1007/s10489-020-02082-8

Hong, D., Lee, S., Cho, Y. H., Baek, D., Kim, J., Chang, N., et al. (2021). Energy-efficient online path planning of multiple drones using reinforcement learning. *IEEE Trans. Veh. Technol*. 70, 9725–9740. doi: 10.1109/TVT.2021.3102589

Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A. A., Yogamani, S., et al. (2021). Deep reinforcement learning for autonomous driving: a survey. *IEEE Trans. Intell. Transp. Syst*. 23, 4909–4926. doi: 10.1109/TITS.2021.3054625

Kouris, A., and Bouganis, C.-S. (2018). "Learning to fly by myself: a self-supervised cnn-based approach for autonomous navigation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Madrid: IEEE), 1–9. doi: 10.1109/IROS.2018.8594204

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Nav. Res. Logist. Q*. 2, 83–97. doi: 10.1002/nav.3800020109

Lee, Z.-J., Su, S.-F., and Lee, C.-Y. (2003). Efficiently solving general weapon-target assignment problem by genetic algorithms with greedy eugenics. *IEEE Trans. Syst. Man Cybernet. B* 33, 113–121. doi: 10.1109/TSMCB.2003.808174

Li, J., Li, C., Chen, T., and Zhang, Y. (2022). Improved rrt algorithm for auv target search in unknown 3d environment. *J. Mar. Sci. Eng*. 10:826. doi: 10.3390/jmse10060826

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., et al. (2015). Continuous control with deep reinforcement learning. *arXiv* [Preprint]. doi: 10.48550/arXiv.1509.02971

Liu, D., Bao, W., Zhu, X., Fei, B., Men, T., Xiao, Z., et al. (2021). Cooperative path optimization for multiple uavs surveillance in uncertain environment. *IEEE Internet Things J*. 9, 10676–10692. doi: 10.1109/JIOT.2021.3125784

Liu, X., Li, H., Xue, J., Zeng, T., and Zhao, X. (2023). Location and tracking of environmental pollution sources under multi-UAV vision based on target motion model. *Soft Comput*. 27, 1–15. doi: 10.1007/s00500-023-07981-9

Lowe, R., Wu, Y., and Tamar, A. (2017). "Multi-agent actor-critic for mixed cooperative? competitive environments," in *31st International Conference on Neural Information Processing Systems* (Curran Associates Inc.), 6379–6390.

Luo, W. Lü, J., Liu, K., Chen, L. (2021). Learning-based policy optimization for adversarial missile-target assignment. *IEEE Trans. Syst. Man Cybernet. Syst*. 52, 4426–4437. doi: 10.1109/TSMC.2021.3096997

Lyu, M., Zhao, Y., Huang, C., and Huang, H. (2023). Unmanned aerial vehicles for search and rescue: a survey. *Remote Sens*. 15:3266. doi: 10.3390/rs15133266

Mansouri, S. S., Kanellakis, C., Kominiak, D., and Nikolakopoulos, G. (2020). Deploying mavs for autonomous navigation in dark underground mine environments. *Robot. Auton. Syst*. 126, 103472. doi: 10.1016/j.robot.2020.103472

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level control through deep reinforcement learning. *Nature* 518, 529–533. doi: 10.1038/nature14236

Pan, Y., Yang, Y., and Li, W. (2021). A deep learning trained by genetic algorithm to improve the efficiency of path planning for data collection with multi-UAV. *IEEE Access* 9, 7994–8005. doi: 10.1109/ACCESS.2021.3049892

Qie, H., Shi, D., Shen, T., Xu, X., Li, Y., Wang, L., et al. (2019). Joint optimization of multi-UAV target assignment and path planning based on multi-agent reinforcement learning. *IEEE Access* 7, 146264–146272. doi: 10.1109/ACCESS.2019.2943253

Qin, Z., Wang, H., Wei, Z., Qu, Y., Xiong, F., Dai, H., et al. (2021). Task selection and scheduling in UAV-enabled mec for reconnaissance with time-varying priorities. *IEEE Internet of Things Journal*, 8, 17290–17307. doi: 10.1109/JIOT.2021.3078746

Samiei, A., Ismail, S., and Sun, L. (2019). "Cluster-based hungarian approach to task allocation for unmanned aerial vehicles" in *2019 IEEE National Aerospace and Electronics Conference (NAECON)* (Dayton, OH: IEEE), 148–154. doi: 10.1109/NAECON46414.2019.9057847

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv* [Preprint]. doi: 10.48550/arXiv.1707.06347

Song, J., Zhao, K., and Liu, Y. (2023). Survey on mission planning of multiple unmanned aerial vehicles. *Aerospace* 10:208. doi: 10.3390/aerospace10030208

Spaan, M. T. (2012). Partially observable markov decision processes. *Reinforcement learning: State-of-the-art*, eds M. Wiering, and M. van Otterlo (Cham: Springer), 387–414. doi: 10.1007/978-3-642-27645-3_12

Su, X., Dong, S., Liu, S., Cracknell, A. P., Zhang, Y., Wang, X., et al. (2018). Using an unmanned aerial vehicle (UAV) to study wild yak in the highest desert in the world. *Int. J. Remote Sens*. 39, 5490–5503. doi: 10.1080/01431161.2018.1441570

Thrun, S., and Littman, M. L. (2000). Reinforcement learning: an introduction. *AI. Mag*. 21, 103–103. doi: 10.1609/aimag.v21i1.1501

Tian, J., Wang, Y., and Fan, C. (2018). "Research on target assignment of multiple-uavs based on improved hybrid genetic algorithm," in *2018 IEEE 4th International Conference on Control Science and Systems Engineering (ICCSSE)* (Wuhan: IEEE), 304–307. doi: 10.1109/CCSSE.2018.8724841

Wang, C., Wang, J., Wang, J., and Zhang, X. (2020). Deep-reinforcement-learning-based autonomous UAV navigation with sparse rewards. *IEEE Internet. Things. J*. 7, 6180–6190. doi: 10.1109/JIOT.2020.2973193

Wang, T., Li, M., and Zhang, M.-Y. (2020). "Cooperative coverage reconnaissance of multi- UAV," in *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)* (Chongqing: IEEE), 1647–1651. doi: 10.1109/ITOEC49072.2020.9141873

Wang, X., Wang, H., Zhang, H., Wang, M., Wang, L., Cui, K., et al. (2023). A mini review on UAV mission planning. *J. Ind. Manag. Optim*. 19, 3362–3382. doi: 10.3934/jimo.2022089

Wu, Y., Liao, S., Liu, X., Li, Z., and Lu, R. (2021). Deep reinforcement learning on autonomous driving policy with auxiliary critic network. *IEEE Trans. Neural. Netw. Learn. Syst*. 34, 3680–3690. doi: 10.1109/TNNLS.2021.3116063

Xing, L., Fan, X., Dong, Y., Xiong, Z., Xing, L., Yang, Y., et al. (2022). Multi-UAV cooperative system for search and rescue based on YOLOv5. *Int. J. Disaster Risk Sci*. 76:102972. doi: 10.1016/j.ijdrr.2022.102972

Xu, Y., Xue, X., Sun, Z., Chang, C., Gu, W., Chen, C., et al. (2019). Online spraying quality assessment system of plant protection unmanned aerial vehicle based on android client. *Comput. Electron. Agric*. 166:104938. doi: 10.1016/j.compag.2019.104938

Yan, Z., Kreidieh, A. R., Vinitsky, E., Bayen, A. M., and Wu, C. (2022). Unified automatic control of vehicular systems with reinforcement learning. *IEEE Trans. Autom. Sci. Eng*. 20, 789–804. doi: 10.1109/TASE.2022.3168621

Yang, Z., Yu, X., Dedman, S., Rosso, M., Zhu, J., Yang, J., et al. (2022). UAV remote sensing applications in marine monitoring: knowledge visualization and review. *Sci. Total Environ*. 838:155939. doi: 10.1016/j.scitotenv.2022.155939

Yue, L., Yang, R., Zhang, Y., and Zuo, J. (2023). Research on reinforcement learning-based safe decision-making methodology for multiple unmanned aerial vehicles. *Front. Neurorobot*. 16:1105480. doi: 10.3389/fnbot.2022.1105480

Zhang, H., Jiang, H., Luo, Y., and Xiao, G. (2016). Data-driven optimal consensus control for discrete-time multi-agent systems with unknown dynamics using reinforcement learning method. *IEEE Trans. Ind. Electron*. 64, 4091–4100. doi: 10.1109/TIE.2016.2542134

Zhang, H., Wang, L., Tian, T., and Yin, J. (2021). A review of unmanned aerial vehicle low-altitude remote sensing (UAV-LARS) use in agricultural monitoring in china. *Remote Sens*. 13:1221. doi: 10.3390/rs13061221

Zhang, S., Li, Y., and Dong, Q. (2022). Autonomous navigation of UAV in multi-obstacle environments based on a deep reinforcement learning approach. *Appl. Soft. Comput*. 115:108194. doi: 10.1016/j.asoc.2021.108194

Zhao, M., Wang, G., Fu, Q., Guo, X., Chen, Y., Tengda, L., et al. (2023). MW-MADDPG: a meta-learning based decision-making method for collaborative UAV swarm. *Front. Neurorobot*. 17:1243174. doi: 10.3389/fnbot.2023.1243174