

Article

Multi-UAVs Searching and Tracking for USV Swarm: A Center-Sub-Critics Reinforcement Learning Approach

Ye Hou , Bo Li * and Xueru Miao

Institute of Logistics Science and Engineering, Shanghai Maritime University, Shanghai 201306, China

* Correspondence: libo@shmtu.edu.cn

Highlights

What are the main findings?

- In the searching and tracking USV swarm task, an observation fusion mechanism with spatio-temporal decay is designed to improve the perception abilities of multi-UAVs.
- The proposed center-sub-critics deep deterministic policy gradient algorithm mitigates credibility imbalance and policy variance from both cause and effect sides through employing multiple center-critic networks coupled with sub-critic networks.

What are the implications of the main findings?

- The confidence probability map considering the distance timeliness of neighbor data can deliver more precise target locations to multi-UAVs in dynamic complex environments.
- The proposed algorithm provides a multiple unmanned systems cooperative trajectory planning scheme, improving the searching and tracking efficiency and fairness.

Abstract

This work proposes a multiple unmanned aerial vehicles (UAVs) cooperative trajectory planning scheme constructed by multi-agent reinforcement learning with hybrid critics, improving the searching and tracking efficiency and fairness when the dynamic unmanned surface vehicle (USV) swarm exceeds the number of UAVs. A confidence map of targets' existence probability with spatio-temporal decay is first established through a local information fusion mechanism based on Bayesian update theory. It leads to a reformulation of the problem model into a communication-enhanced partially observable Markov decision process. To suppress policy variance and credibility imbalance of the multi-UAVs, a center-sub-critics deep deterministic policy gradient algorithm is then proposed, combining multiple centralized critics with decentralized critics. Meanwhile, a segmented reward function is designed to incentivize the UAV to revisit detected targets. Finally, the simulation results compared with diverse baseline algorithms demonstrate the efficacy and scalability of the proposed scheme in this paper.



Academic Editor: Xiwang Dong

Received: 29 December 2025

Revised: 7 February 2026

Accepted: 9 February 2026

Published: 11 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: multi-target tracking; unmanned aerial vehicle swarm; multi-agent deep reinforcement learning; confidence probability map

1. Introduction

Recently, multi-target searching and tracking (MTST) has been a hot research issue, which can be widely used in various fields [1], such as security surveillance, military reconnaissance, and emergency communication. The increasing number of vessels navigating near public waters, including tourist motorboats, yachts, and a growing number

of unmanned surface vehicles (USVs) deployed for scientific research and other purposes, has made the maritime environment increasingly complex. Consequently, the task of searching and tracking multiple moving targets on the sea surface is still gaining significant attention due to its high demand and high complexity. Whether cooperating with USVs for maritime rescue, environmental monitoring, and oceanographic surveys, or addressing non-cooperative targets such as illegal fishing and vessel traffic management, multi-target searching and tracking solutions hold broad practical motivation.

Benefiting from the inherent characteristics of lightweight, flexible, and low-cost UAVs, are being widely used to solve MTST problems. Limited by the energy and execution efficiency issues of a single UAV, multi-UAV tasks are preferred for MTST. Multi-UAVs can achieve efficient searching and tracking of multiple dynamic targets by implementing cooperative control strategies [2] through information interaction. The cooperative trajectory planning method of MTST is proved to be NP-hard [3]. Since the number and the movement of targets in the task area are unknown, MTST requires multi-UAVs to continuously explore the unknown area while tracking back the discovered targets to update their positions. In addition, the multi-UAVs need to plan the optimal detection trajectories cooperatively to minimize the duplicate path and realize the continuous and efficient searching and tracking of dynamic targets. This further raises the fairness definition in dynamic target searching and tracking tasks, namely ensuring all dynamic targets receive as equal attention as possible during the task duration, thereby minimizing disparities in the number of times each target is observed. A large number of linear programming and heuristic methods [4–6] have been applied to MTST. To reduce the energy consumption, Al-Turjman et al. [7] proposed an optimized placement approach to model the multi-agent coverage search problem as a numerical optimization model and solve it accurately. Zou et al. [8] predicted the target motion and guided the multi-agent movement through quad-tree and integer linear programming. Huang et al. [9] modeled the MTST problem as mixed-integer linear programming and achieved efficient cooperative searching by task decomposition and reorganization. Scherer et al. [10] designed heuristic operators for local and global paths, respectively, to reduce the delay of the searching task at a lower cost.

Inspired by natural biological clusters, a large number of swarm intelligence-based optimization algorithms have also been used in the MTST problem. Zhen et al. [11] improved the global planning ability of multi-agent to search and strike multi-target through distributed ant colony optimization (ACO). Pérez-Carabaza et al. [12] improved the ACO algorithm to deal with complex multi-target searching tasks by fusing heuristic operators and pheromones with a utility function. Phung et al. [13] encoded multi-agent trajectories as particle swarm optimization (PSO) to retain the community cognitive ability of agents and ultimately accomplish the MTST task. Xu et al. [14] introduced dynamic multi-swarm and comprehensive learning to enhance the performance of PSO in dynamic tracking problems. Duan et al. [15] proposed a dynamic discrete pigeon-inspired optimization method to solve the task planning problem of multi-agent cooperative searching. The above methods have excellent performance for MTST. However, their application in dynamic environments is limited [16] by the dependence on prior environmental knowledge and the constrained cooperation among multi-agents.

As an alternative way, deep reinforcement learning (DRL) has achieved notable success [17] in agent sequential decision and control by modeling problems with the Markov decision process (MDP). Multi-agent deep reinforcement learning (MADRL) focusing on more complex multi-agent systems is also emerging [18], which provides new ideas for solving MTST problems. MADRL allows multi-agent to explore collaborative relationships and executes distributed decisions through the paradigm termed centralized training with decentralized execution (CTDE) [19], significantly improving the scalability and fault tol-

erance of the algorithm. For fully cooperative multi-agent systems like MTST, MADRL facilitates cooperation mainly from two perspectives: centralized critic and value function decomposition. The centralized critic uses a global network that can collect multi-agent joint actions and observations to estimate the task's cumulative reward expectation. Through this, the multi-agent proximal policy optimization (MAPPO) [20] and multi-agent deep deterministic policy gradient (MADDPG) [21] extend the PPO and DDPG to multi-agent systems. Decomposition of the value function distributes the joint action value into individual value for each agent, subsequently estimating the joint value by aggregating the local results through a specific strategy. The value-decomposition network (VDN) [22] utilizes a simple summation form, while QMIX [23] employs a monotonicity mixing network strategy.

Benefiting from the ability to efficiently execute complex decisions, researchers have applied MADRL to the multi-target searching and tracking field. Hou et al. [24] utilized MADDPG to split the task scene and improve the distributed cooperative searching capability of multi-agent. In order to ensure accurate position of multi-target, Zhang et al. [25] proposed double critic DDPG with Dempster–Shafer theory to describe uncertainty. Shen et al. [23] proposed the DNQMIX algorithm by adding a noise layer for QMIX to solve the trade-off between training speed and environmental realism. The aforementioned methods experience a loss in searching performance when dealing with dynamic targets. Zhou et al. [26] proposed the reciprocal reward multi-agent actor-critic framework to guide agents in handling dynamic targets. Aiming to increase the generalization of MADRL in the MTST task, Cao et al. [27] proposed a multi-criteria negative feedback hierarchical deep Q-network with a two-stage reward function. To enhance dynamic target tracking performance, Yue et al. [28] incorporated the VDN concept into the multi-agent soft actor-critic framework.

The issues of value overestimation and policy fluctuation in existing MADRL methods are particularly exacerbated in dynamic target search and tracking problems like those involving USV swarm. Moreover, the mathematical expressiveness of the target confidence maps employed in current methods is compromised in dynamic environments. Inspired by the related works, a multi-UAVs cooperative trajectories planning scheme based on MADRL with hybrid critics is proposed to address the limitations of MTST efficiency and fairness when the USV swarm is regarded as a target in multi-constraint environments. This work achieves the following primary contributions:

- Unlike prior studies [25,29] that ignore the distance timeliness of neighbor data, a local observation fusion rule with spatio-temporal decay is designed to improve the environmental perception of the multi-UAV system. With the update utilizing the rule, the target confidence map guides multi-UAVs to balance the searching and tracking of the USV swarm.
- In existing decentralized partially observable Markov decision processes (Dec-POMDP) [30,31], agents maintain independent observation spaces. This paper proposes a communication-enhanced POMDP with additional masking mechanisms to support variable-dimensional fusion communication observations, and the multi-UAV MTST problem is reformulated based on this model.
- Previous MADRL approaches [24,32] addressed value overestimation and policy collapse from either a causal or an outcome perspective. In contrast, the proposed center-sub-critics deep deterministic policy gradient (CSC-DDPG) algorithm mitigates credibility imbalance and policy variance from dual perspectives through employing multiple center-critic networks coupled with sub-critic networks.

2. System Model and Problem Formulation

In this section, the multi-UAV MTRT for the USV swarm problem is systematically modeled. We clarify the environment model and the UAV's mathematical model studied in this paper, followed by a mathematical formulation of the MTRT objectives for further optimization analysis in subsequent sections.

2.1. Environment Model

This study investigates MTST within a rectangular sea surface region $C \subset \mathbb{R}^2$ of dimensions $C_x \times C_y$, whose spatial configuration is depicted in Figure 1. The region C is divided into a grid map using squares with side length C_l . A Cartesian coordinate system defines spatial positions, where the bottom-left cell serves as the coordinate origin. $c_{x,y}$, where $x \in \{1, 2, \dots, c_x^{max}\}$ and $y \in \{1, 2, \dots, c_y^{max}\}$ correspond to the cell in column x and row y . Meanwhile, a binary state $c_{x,y} \in \{0, 1\}$ is used to encode the target existence status in $c_{x,y}$, where $c_{x,y} = 1$ signifies target existence and $c_{x,y} = 0$ denotes target absence.

The task timeline is discretized into T intervals of duration Δt , with t indexing the current time step. It is assumed that there are N_k USVs $K = \{K_1, K_2, \dots, K_{N_k}\}$ as targets with distributed positions $k_i \in C$, and each cell obeys the single-target occupancy principle. The USVs can move in a random direction with a fixed velocity v_k within each time step. Concurrently, N_b no-fly zones $B = \{B_1, B_2, \dots, B_{N_b}\}$ are arbitrarily positioned within C . Each no-fly zone occupies a whole cell, with coordinates defined as $b_i = (x, y)$. In addition, the multi-UAVs do not have any prior knowledge of the locations or numbers of USVs and no-fly zones.

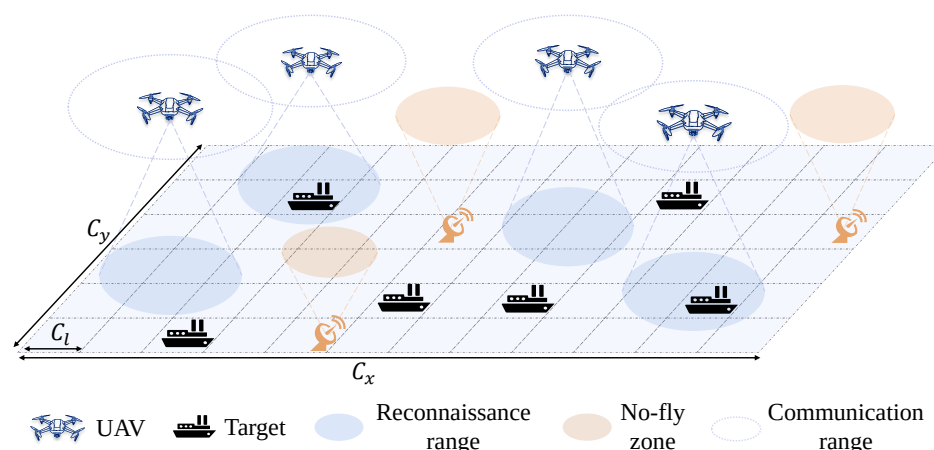


Figure 1. The environment model of the multi-target searching and tracking by multi-UAVs.

2.2. UAV's Mathematical Model

In MTST, both multi-UAVs and USV swarms are homogeneous. The UAVs do not need to identify targets, which makes more extensive and flexible cooperation among multi-UAVs possible. The multi-UAVs $U = \{U_1, U_2, \dots, U_{N_u}\}$ with N_u UAVs operate at a common altitude. Their movement is constrained within a planar space congruent with C . For UAV $U_i \in U$, let $u_i \in C$ denote its planar coordinates, with the corresponding cell coordinate $g_i = \lfloor u_i \rfloor$, where $\lfloor \cdot \rfloor$ represents the discretization operator rounding down u_i . The UAV kinematic model is simplified as a particle undergoing two-dimensional translation. Meanwhile, the multi-UAVs are subjected to a maximum velocity amplitude

constraint v_{max} and an axial maximum acceleration bound a_{max} . Then the kinematic model of UAV U_i is described in the form of

$$\begin{cases} \mathbf{a}_{i,t}^{clip} = \text{clip}(\mathbf{a}_{i,t}, -a_{max}, a_{max}) \\ \mathbf{v}_{i,t+1} = \frac{\min(\|\mathbf{v}_{i,t} + \mathbf{a}_{i,t}^{clip} \Delta t\|, v_{max})}{\|\mathbf{v}_{i,t} + \mathbf{a}_{i,t}^{clip} \Delta t\|} (\mathbf{v}_{i,t} + \mathbf{a}_{i,t}^{clip} \Delta t), \\ \mathbf{u}_{i,t+1} = \mathbf{u}_{i,t} + \mathbf{v}_{i,t+1} \Delta t \end{cases} \quad (1)$$

where $\mathbf{a}_{i,t}$, $\mathbf{v}_{i,t}$, and $\mathbf{u}_{i,t}$ represent the acceleration, velocity, and position vector of U_i at time step t , respectively. The *clip* function represents applying minimum and maximum value truncation to each element of the vector, while $\min(\cdot, \cdot)$ corresponds to the minimization operator between two scalars. Each UAV features a sensor with limited performance. It is assumed that, at a fixed altitude, the sensing range of the sensor of UAV U_i forms a square region I_i centered at $\mathbf{g}_i$ with side length $2r_s$, where r_s denotes the effective radius of the sensor. At time step t , U_i can obtain the no-fly zones and target positions information in $I_{i,t}$. To support the confidence map analysis, let $\theta_{i,t} \in \{0, 1\}^{2r_s \times 2r_s}$ encode detection results at time t of U_i ; then $\theta_{i,t}^{x,y} = 1$ signifies that the target is reconnoitered in $c_{x,y}$, while $\theta_{i,t}^{x,y} = 0$ indicates nothing found. Sensor performance is characterized by the true positive probability $p_t = \mathbb{P}(\theta_{i,t}^{x,y} = 1 | c_{x,y} = 1)$ and the false alarm probability $p_f = \mathbb{P}(\theta_{i,t}^{x,y} = 1 | c_{x,y} = 0)$, accounting for sensor imperfections and environmental disturbances. In practical deployments, p_t and p_f are influenced by numerous environmental factors such as viewing occlusion and sea-clutter intensity. For analytical simplicity, p_t and p_f are simplified to fixed values primarily affected by the rated performance of the UAV sensor.

2.3. Confidence Map and Information Fusion

The perception of environments is represented by independently maintaining a confidence map of target existence probability decaying over time and distance for each UAV, called the spatio-temporal decay target confidence map (SDTCM). The SDTCM M_i for UAV U_i constitutes a $c_x^{max} \times c_y^{max}$ probabilistic grid map, then $M_i^{x,y} \in [0, 1]$ represents the posterior probability $\mathbb{P}(c_{x,y} = 1)$ of target existence in $c_{x,y}$. We initialize the SDTCM of U_i with a uniform prior distribution as $M_{i,t_0}^{x,y} = 0.5, \forall x \in \{1, \dots, c_x^{max}\}, y \in \{1, \dots, c_y^{max}\}$, which reflects complete uncertainty about target distribution across C . The UAV U_i continuously updates M_i by incorporating $\theta_{i,t}$ from successive time steps $t \in \{1, 2, \dots, T\}$ through Bayesian update theory, which is formulated as [33]

$$M_{i,t}^{x,y} = \begin{cases} \frac{p_t M_{i,t-1}^{x,y}}{p_t M_{i,t-1}^{x,y} + p_f (1 - M_{i,t-1}^{x,y})}, & \text{if } \theta_{i,t}^{x,y} = 1 \\ \frac{(1-p_t) M_{i,t-1}^{x,y}}{(1-p_t) M_{i,t-1}^{x,y} + (1-p_f)(1 - M_{i,t-1}^{x,y})}, & \text{if } \theta_{i,t}^{x,y} = 0 \\ M_{i,t-1}^{x,y}, & \text{otherwise} \end{cases} \quad (2)$$

The belief update mechanism for $M_{i,t}^{x,y}$ in (2) exhibits nonlinear characteristics. Following the nonlinear mapping proposed in [34], one can reformulate (2) through $m_{i,t}^{x,y} = -\ln(M_{i,t}^{x,y})$, obtaining the equivalent representation of (2), yields to

$$m_{i,t}^{x,y} = m_{i,t-1}^{x,y} + \Delta \theta_{i,t}^{x,y}, \text{ where } \Delta \theta_{i,t}^{x,y} = \begin{cases} \frac{p_f}{p_t}, & \text{if } \theta_{i,t}^{x,y} = 1 \\ \frac{(1-p_f)}{(1-p_t)}, & \text{if } \theta_{i,t}^{x,y} = 0 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Given the known sensor detection likelihood probability, the update for SDTCM in (3) demonstrates linear characteristics. This linear property substantially mitigates computational overhead in maintaining the SDTCM.

Operating with latency-free assumptions, the multi-UAVs possess communication capabilities with a bounded range to enhance cooperation. At time t , the communication neighbors $U_{i,t}^n$ for U_i are defined as $U_{i,t}^n = \{U_j \mid \|u_{i,t} - u_{j,t}\| < r_c, j \in \{1, \dots, N_u\}, i \neq j\}$, where r_c denotes the communication radius. Each UAV U_i can exchange data bidirectionally with all $U_j \in U_{i,t}^n$, facilitating distributed information sharing across the multi-UAVs. During each time step t , U_i performs the above information exchange within $U_{i,t}^n$:

- Dissemination of self-position $u_{i,t}$, detected no-fly zones, and detection result $\theta_{i,t}$.
- Acquirement of neighbor UAV positions $u_{j,t}^n = \{u_{j,t} \mid U_j \in U_{i,t}^n\}$, neighbors detected no-fly zones, and detection results set $\Theta_{i,t}^n = \{\theta_{j,t} \mid U_j \in U_{i,t}^n\}$.

In order to represent the spatio-temporal uncertainty of the environments and make efficient utilization of self and neighbor information, this paper designs a fusion mechanism based on spatio-temporal decay to update SDTCM M_i . It can be expressed as

$$m_{i,t}^{x,y} = \alpha m_{i,t-1}^{x,y} + \Delta \theta_{i,t}^{x,y} + \sum_{j=1}^{N_{i,t}^n} \left(1 - \frac{\|u_{i,t} - u_{j,t}\|}{r_c}\right) \Delta \theta_{j,t}^{x,y}, \quad (4)$$

where α denotes the time decay factor, and $N_{i,t}^n$ is the UAV number in $U_{i,t}^n$. This distributed fusion mechanism enables autonomous maintenance of SDTCM while facilitating real-time awareness of neighbors. The synergistic integration of localized sensing and neighbor information achieves enhanced cooperative searching and tracking capability under joint constraints. Simultaneously, utilizing neighbor observations for probability updates within the independently maintained SDTCM reflects the real-world effect of corroborating evidence, while enabling more efficient responses to global state. The inherent conservatism of dual spatio-temporal decay ensures that posterior probabilities do not become unrealistically overconfident.

2.4. Problem Formulation

The optimization aim of multi-UAVs MTST for USV swarms is to simultaneously discover new targets while maintaining persistent tracking of detected USVs. Due to the indeterminate scale of the USV swarm within C , it is necessary for continuous exploration to minimize the environment's uncertainty. The spatial uncertainty within C is quantifiable through Shannon's information entropy of SDTCM, expressed as

$$e_t^{x,y} = -M_t^{x,y} \ln M_t^{x,y} - (1 - M_t^{x,y}) \ln(1 - M_t^{x,y}), \quad (5)$$

where $M_t^{x,y}$ denotes the value of cell $c_{x,y}$ in global SDTCM M , and it is updated by all UAVs without spatial decay. The reconnoitered state of target K_k at time step t can be represented as

$$I_{k,t} = \begin{cases} 1, & \text{if } \exists \theta_{i,t}^{x,y} = 1, U_i \in U \\ 0, & \text{otherwise} \end{cases}, \quad (6)$$

with $(x, y) = \lfloor k_k \rfloor$. Then the optimization objective of MTST can be formulated as achieving maximization of the following functions $f_i, i \in \{1, 2, 3\}$

$$\begin{cases} f_1 = \sum_{x=1}^{c_x^{max}} \sum_{y=1}^{c_y^{max}} 1_{M_{t=T}^{x,y} > \zeta} \\ f_2 = \frac{1}{N_k} \frac{1}{T} \sum_{k=1}^{N_k} \sum_{t=1}^T I_{k,t}, \\ f_3 = - \sum_{x=1}^{c_x^{max}} \sum_{y=1}^{c_y^{max}} e_{t=T}^{x,y} \end{cases} \quad (7)$$

where ζ represents the confidence threshold for target existence. We admit the existence of a target in cell $c_{x,y}$ when global SDTCM satisfies $M_t^{x,y} > \zeta$. This work addresses the multi-UAVs MTST problem under dynamic environments. The resultant MTST formulation under composite constraints is mathematically encapsulated in

$$\begin{cases} \max_{i \in \{1,2,3\}} f_i \\ s.t. \quad \begin{aligned} & \|v_{i,t}\| < v_{max} \\ & -a_{max} < a_{i,t}^x, a_{i,t}^y < a_{max} \\ & \|u_{i,t} - u_{j,t}\| < r_{sn}, \forall u_j \in U_{i,t}^n \\ & \|u_{i,t} - b_j\| < r_{sb}, \forall b_j \in B \\ & u_{i,t} \in C \end{aligned} \end{cases}, \quad (8)$$

where the Cartesian components $a_{i,t}^x$ and $a_{i,t}^y$ represent the acceleration vector $a_{i,t}$ of the UAV U_i . The minimum separation thresholds r_{sn} and r_{sb} govern collision prevention between neighbor UAVs and no-fly zones B spatial relationships, respectively.

3. Multi-Agent Deep Reinforcement Learning-Based MTST Method

3.1. MADRL Based on Communication-Enhanced POMDP

The Dec-POMDP establishes a joint sequential decision-making task model in which agents utilize their own local observation in the multi-agent environment. However, Dec-POMDP is not adapted to the communication interaction of multi-agent. This paper extends it to the communication-enhanced POMDP model to express that agents can make decisions through local observation and fusion information in a distributed manner. Taking the MTST problem as an example, the communication-enhanced POMDP can be described as a tuple $(U, S, A, O^c, R, P_s, J)$ omitting the time subscripts, where

- $U = \{U_1, \dots, U_i, \dots, U_{N_u}\}$ is the set of N_u UAVs in the multi-UAVs;
- S is the global state space of the environment, and $s \in S$ denotes a specific global state;
- $A = \{A_1, \dots, A_i, \dots, A_{N_u}\}$ defines the joint action space, with $a_i \in A_i$ corresponding to the action of U_i ;
- $O^c = \{O_1^c, \dots, O_i^c, \dots, O_{N_u}^c\}$ is the joint communication observation space, where $o_i^c \in O_i^c$ is the fusion communication observation of U_i ;
- $R(s, a)$ represents the global reward function for joint action $a \in A$ of multi-UAVs;
- $P_s(s'|s, a)$ is the state transition probability function, which describes the probability of transferring to the next state s' after the joint action a is executed;
- $J(s, U_i, U_i^n)$ is the fusion observation function, denoting the fusion communication observation generated by U_i after fusing local observation and neighbor information.

In each time step t , UAV U_i obtains fusion observation $o_{i,t}^c$ via $J(s_t, U_i, U_{i,t}^n)$ and performs action $a_{i,t}$ according to the policy $\pi_i(o_{i,t}^c) \in \pi$ to form the joint action a_t , where $\pi = \{\pi_1, \dots, \pi_i, \dots, \pi_{N_u}\}$ is the joint policy. The global state s_{t+1} for the next time step is

then obtained by $P_s(s_{t+1}|s_t, \mathbf{a}_t)$, while the global reward r_t is acquired by $R(s_t, \mathbf{a}_t)$. An optimal joint policy π^* is the fundamental objective of MADRL to maximize the task cumulative reward

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}, \quad (9)$$

where γ is the reward discount factor over time. The action value function is given by

$$Q_{\pi}(s, \mathbf{a}) = E_{\pi} \left(\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, \mathbf{a}_t = \mathbf{a} \right), \quad (10)$$

which expresses the cumulative reward expectation of executing the joint action $\mathbf{a}$ with policy π at state s . According to Bellman's optimality principle [35], the optimal action value function can be described as

$$Q_{\pi^*}(s, \mathbf{a}) = E_{\pi^*} \left[r_t + \gamma \max_{\mathbf{a}_{t+1}} Q_{\pi^*}(s_{t+1}, \mathbf{a}_{t+1}) \right], \quad (11)$$

then the optimal joint policy can be formulated as

$$\pi^*(s) = \arg \max_{\mathbf{a}} Q_{\pi^*}(s, \mathbf{a}). \quad (12)$$

In communication-enhanced POMDP, U_i obtains fusion observations via $J(s, U_i, U_i^n)$, so $Q_{\pi}(s, \mathbf{a})$ can be modified as $Q_{\pi}(o_1^c, \dots, o_{N_u}^c, \mathbf{a})$ while the joint policy is described as policy set $\pi(s) \rightarrow \{\pi_1(o_1^c), \dots, \pi_{N_u}(o_{N_u}^c)\}$ utilizing joint communication observations.

3.2. Action and Observation Space Design

In order to improve the flexibility and scalability of the algorithm, a continuous action space is used to express the acceleration of the UAV. The UAV U_i outputs the interval mapping $\mathbf{a}_{i,t}^e \in [-1, 1]^{1 \times 2}$ of the acceleration $\mathbf{a}_{i,t}$ by the policy π_i at each time step t , and through $a_{max} \cdot \mathbf{a}_{i,t}^e$ obtains actual acceleration. Subsequently, $\mathbf{v}_{i,t+1}$ and $\mathbf{u}_{i,t+1}$ are acquired iteratively according to (1) and employed to update the environment state. Throughout the task duration T , U_i generates a motion trajectory based on sequential decisions. Using continuous acceleration as the policy output can facilitate the training network to adapt to diversified flight control systems during real-world deployment.

The UAV U_i outputs the action in each time step t based on the fusion observation $o_{i,t}^c$. With the aim of increasing the effective information density, the observation space is composed of two-dimensional spatial information and one-dimensional structural information. By preprocessing the direct observation and communication data, the fusion communication observation is expressed as

$$o_{i,t}^c = \{\mathbf{M}_{i,t}^o, \mathbf{u}_{i,t}^e, \mathbf{v}_{i,t}, \mathbf{u}_{i,t}^{mask}, \mathbf{b}_{i,t}^{mask}\}, \quad (13)$$

where $\mathbf{M}_{i,t}^o$ represents the data in a square region of $\mathbf{M}_{i,t}$ centered on $\mathbf{g}_{i,t}$ with a side length of $2r_o$, and r_o is the observation radius of UAV. The remaining items are structural information, where $\mathbf{u}_{i,t}^e = (u_{i,t}^x/c_x^{max}, u_{i,t}^y/c_y^{max})$ denotes the normalized formulation of the position of U_i . $\mathbf{u}_{i,t}^{mask}$ and $\mathbf{b}_{i,t}^{mask}$ denote the normalized relative positions of the neighbor UAVs and no-fly zones based on a mask mechanism, respectively. Owing to the limited communication range, the dimension of the fusion observation is variable over time. This contradicts the fixed dimension input requirement of the policy network, which further leads to the low scalability of the algorithm. A feature representation of observation space based on a mask mechanism is proposed to help. Suppose that the observation space is fixed to accept N_u^{mask}

positions of nearby UAVs; if $N_u^{mask} > N_{i,t}^n$, then the normalized relative position of the neighbor UAVs based on the mask mechanism can be represented as

$$\mathbf{u}_{i,t}^{mask} = \left\{ \frac{\mathbf{u}_{j,t} - \mathbf{u}_{i,t}}{r_c} \middle| \mathbf{u}_{j,t} \in \mathbf{u}_{i,t}^n \right\} \cup \{(1, 1), \dots\}, \quad (14)$$

where $|\{(1, 1), \dots\}| = N_u^{mask} - N_{i,t}^n$, implying that the UAV is not concerned with the neighbor UAVs whose distance exceeds the communication radius r_c . If $N_u^{mask} \leq N_{i,t}^n$, one can obtain

$$\mathbf{u}_{i,t}^{mask} = \left\{ \frac{\mathbf{u}_{j,t} - \mathbf{u}_{i,t}}{r_c} \middle| \mathbf{u}_{j,t} \in \mathbf{u}_{i,t}^{rank} \right\}, \quad (15)$$

where $\mathbf{u}_{i,t}^{rank}$ is the set of communication neighbors $U_{i,t}^n$ sorted by the relative distance to U_i . Then the nearest N_u^{mask} position is taken, i.e., $|\mathbf{u}_{i,t}^{rank}| = N_u^{mask}$. Similarly, $\mathbf{b}_{i,t}^{mask}$ can also be represented by the masking mechanism as shown in (14) and (15). The mask mechanism has solved the time-varying dimension problem through fixing the dimensions of $\mathbf{u}_{i,t}^{mask}$ and $\mathbf{b}_{i,t}^{mask}$ to $2N_u^{mask}$ and $2N_b^{mask}$, respectively. Furthermore, the input quality of the observation is improved by relative position transformation and communication radius normalization, resulting in enhanced scalability of the algorithm. To prevent introducing artificial sequence information into the observations that could cause irreversible permutation sensitivity and bias in model training, the individual information within $\mathbf{u}_{i,t}^{mask}$ and $\mathbf{b}_{i,t}^{mask}$ processed through the masking mechanism is randomly shuffled before being combined into the final observations. Moreover, a one-hot encoding of each UAV's unique ID is incorporated into its observation vector. This encoding enables individual identification within asymmetric spatial configurations and promotes the emergence of specialized, role-based cooperation among agents.

3.3. Reward Function Design

To enhance the distributed execution capability of UAVs, the global reward function $R(s, \mathbf{a})$ is modified to the sum of independent reward functions $R_i(s, \mathbf{a})$ of homogeneous UAVs, i.e., the global reward $r_t = \sum_{i=1}^{N_u} r_{i,t}$ is the sum of independent reward $r_{i,t}$ of all UAVs at time step t . Following the optimization objectives and constraints defined in Section 2.4, each $r_{i,t}$ at time step t incorporates the following incentive components and penalty metrics.

(1) *Centralized reward*: The centralized reward assesses the overall contribution of the joint action to the cooperative tasks. The designed centralized reward includes the global spatial information entropy reduction reward and the USV swarm average detected rate increase reward, which can be described as

$$r_{i,t}^1 = \omega_1 \sum_{x=1}^{c_x^{max}} \sum_{y=1}^{c_y^{max}} e_{t-1}^{x,y} - e_t^{x,y} + \omega_2 \frac{1}{N_k} \frac{1}{t} \sum_{k=1}^{N_k} \sum_{t_i=1}^t I_{k,t_i}, \quad (16)$$

where ω_1 and ω_2 denote the weights of global spatial entropy and average target detected rate, respectively.

(2) *Detected reward*: To reward the detective operations to the USV swarm while promoting tracking behaviors, we formulate a two-phase detection reward structure designed as

$$r_{i,t}^2 = \omega_3 \sum_{x=1}^{c_x^{max}} \sum_{y=1}^{c_y^{max}} 1 \left| \begin{array}{c} M_{i,t-1}^{x,y} < \zeta, M_{i,t}^{x,y} \geq \zeta \end{array} \right| + \omega_4 \sum_{x=1}^{c_x^{max}} \sum_{y=1}^{c_y^{max}} 1 \left| \begin{array}{c} M_{i,t-1}^{x,y} < \bar{\zeta}, M_{i,t}^{x,y} \geq \bar{\zeta} \end{array} \right|, \quad (17)$$

where ω_3 and ω_4 denote respective phase reward weights, and the activation threshold ζ for primary detection satisfies $0.5 < \zeta < \bar{\zeta}$, establishing the hierarchical incentive requirement.

(3) *Exploration Reward*: The exploration reward mechanism drives UAVs towards sustained investigation of unknown regions. Affected by SDTCM time decay, the entropy of C increases over time, resulting in a detection value again. We define the detection value threshold ς to denote a probability interval $\sigma = [0.5 - \varsigma, 0.5 + \varsigma]$ for possessing a detection value, then the exploration reward can be formulated as

$$r_{i,t}^3 = \omega_5 \sum_{x=1}^{c_x^{max}} \sum_{y=1}^{c_y^{max}} 1 \Bigg|_{M_{i,t-1}^{x,y} \in \sigma, M_{i,t}^{x,y} \notin \sigma}, \quad (18)$$

where ω_5 is the exploration reward weight.

(4) *Boundary penalty*: UAV mobility is strictly confined to the MTST task region C . A boundary-approaching penalty mechanism governs UAV spatial deployment through a dual-modal reward structure, which is formulated as

$$r_{i,t}^4 = \begin{cases} -\omega_6 \exp \frac{r_{so} - d_{i,t}^0 - 1}{r_{so}}, & \text{if } d_{i,t}^0 < 1 \\ -\omega_7, & \text{if } \mathbf{u}_{i,t} \notin C \\ 0, & \text{otherwise} \end{cases} \quad (19)$$

The ω_6 and ω_7 are penalty coefficients of boundary proximity violations and egregious exits, respectively, while r_{so} denotes the safety buffer distance from the boundary. The variable $d_{i,t}^0$ quantifies the minimum boundary separation of UAV U_i at time step t .

(5) *Collision Penalty*: The collision penalty mechanism mitigates trajectory overlap risks and prohibits incursions into no-fly zones. This proximity-based penalty operates through exponential decay terms described as

$$r_{i,t}^5 = -\omega_8 \sum_{\mathbf{u}_{j,t} \in \mathbf{u}_{i,t}^{mask}} \exp \frac{r_{sn} - d_{i,t}^n(\mathbf{u}_{j,t})}{r_{sn}} \Bigg|_{d_{i,t}^n < r_{sn}} - \omega_9 \sum_{\mathbf{b}_j \in \mathbf{b}_{i,t}^{mask}} \exp \frac{r_{sb} - d_{i,t}^b(\mathbf{b}_j)}{r_{sb}} \Bigg|_{d_{i,t}^b < r_{sb}}, \quad (20)$$

where ω_8 and ω_9 govern multi-UAV and no-fly zone collision penalty weight, respectively. The formulation employs two distance metrics for UAV U_i at time step t , where $d_{i,t}^n(\mathbf{u}_{j,t}) = \|\mathbf{u}_{i,t} - \mathbf{u}_{j,t}\|$ quantifies the spatial separation between UAV U_i and U_j , while $d_{i,t}^b(\mathbf{b}_j) = \|\mathbf{u}_{i,t} - \mathbf{b}_j\|$ measures the distance relative to no-fly zone $\mathbf{b}_j$.

The comprehensive reward function design integrates the above incentive and restraint components, thereby enabling the independent reward $r_{i,t}$ of UAV U_i at time step t to be formulated as

$$r_{i,t} = r_{i,t}^1 + r_{i,t}^2 + r_{i,t}^3 + r_{i,t}^4 + r_{i,t}^5. \quad (21)$$

Following the $r_{i,t}$ computation through the independent reward function $R_i(s, \mathbf{a})$, the multi-UAVs global reward r_t emerges through summation of all independent incentives. This distributed reward aggregation mechanism preserves individual autonomy while ensuring cooperative objective alignment across the multi-UAVs.

3.4. Network Architectures of Actors and Center-Sub-Critics

In pure cooperative scenarios such as multi-UAVs MTST, mainstream MADRL methodologies face dual challenges, i.e., the policy instability and collapse caused by training variance [36] and the value overestimation aggravated by imbalanced credit assignment among agents. To address these challenges, this work develops the CSC-DDPG algorithm featuring hybrid critic networks that combine multiple centralized critics and decentral-

ized sub-critics. Building upon the AC foundation, the CSC-DDPG framework is derived from the CTDE paradigm. In CTDE, the centralized critic network integrates global state information during the training phase to assess joint action value, while decentralized actor networks generate deterministic policies using local observations during the execution phase. A neural network is employed within the actor model to approximate the optimal control policy $\pi_i(o_i^c, \psi_i)$ for U_i , where ψ_i represents the policy network parameters. Using the fusion communication observation o_i^c as input, the policy network generates the action a_i taken under the strategy. The architecture of the policy network is depicted in Figure 2a.

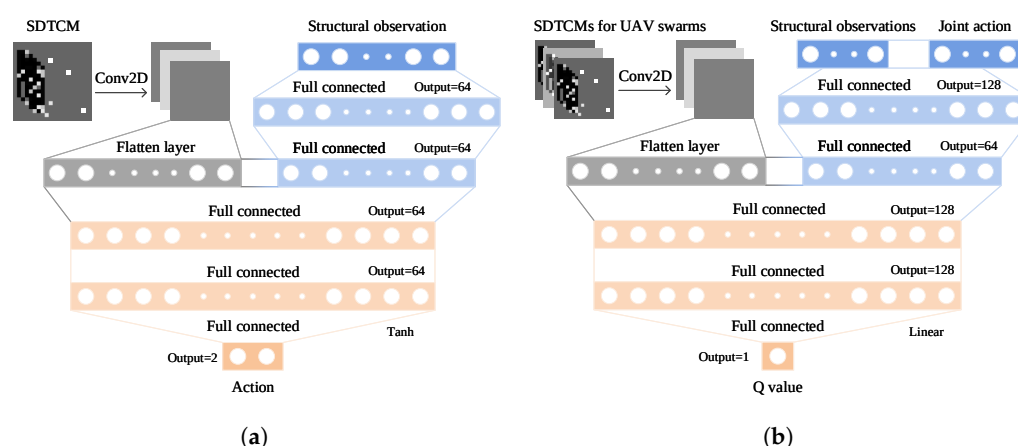


Figure 2. The architecture of CSC-DDPG networks. (a) Policy network. (b) Center-critic network.

To adapt to the multi-dimensional mixed observation space described in Section 3.2, the branch input network architecture is designed. The SDTCM two-dimensional local slices and structural vector groups, such as position and velocity from the fusion communication observations, are input into a branch convolutional neural network (CNN) and a branch multi-layer perceptron (MLP), respectively. The branch CNN extracts spatial features from the SDTCM slices and unfolds them into one-dimensional feature vectors, while the branch MLP preliminarily analyzes the mathematical relationships between the structural vector groups. The outputs of the branch network are then concatenated and input into a master MLP, which ultimately maps the concatenated feature vectors to the action space.

Guided by deterministic policy gradient principles [37], the policy network optimization objective maximizes the expectation

$$J(\pi_i) = E_{o_i^c \sim \rho} [Q_i(o_i^c, a_i) |_{a_i = \pi_i(o_i^c, \psi_i)}], \quad (22)$$

where ρ signifies the probability distribution governing fusion communication observations o_i^c , derived from the state transitions and fusion observation function. The action value function for UAV U_i is represented by Q_i . Update for the parameter of policy network ψ_i employs gradient ascent optimization. Under the assumption of a differentiable policy $\pi_i(o_i^c, \psi_i)$ and action value function $Q_i(o_i^c, a_i)$, the policy gradient follows from chain rule differentiation as

$$\nabla_{\psi_i} J(\pi_i) = E_{o_i^c \sim \rho} [\nabla_{\psi_i} \pi_i(o_i^c, \psi_i) \nabla_{a_i} Q_i(o_i^c, a_i)]. \quad (23)$$

However, direct acquisition of the action-value function Q_i in (23) proves to be intractable, while MADDPG utilizes a centralized critic network implemented with deep neural networks to approximate Q_i ; CSC-DDPG improves through a hybrid framework. The proposed critic model synergistically couples independent sub-critic networks with a group of center-critic networks.

Inspired by the idea of decomposing the centralized value function into independent and distributed value functions in VDN, the sub-critic network $Q_{si}(o_i^c, a_i, \phi_{si})$ serves a single UAV U_i , where ϕ_{si} denotes its network parameters. This dedicated network processes the fusion communication observations and actions of U_i to produce localized action value estimates. Mirroring the policy network implementation, each UAV autonomously maintains its own sub-critic network, focusing exclusively on evaluating self-action values. This distributed model mitigates non-uniform agent contribution issues inherent in the centralized-only critic model. Meanwhile, by avoiding more uncertain global information dependencies, the sub-critic networks can achieve reduced estimation variance. The sub-critic network follows the same architecture as the policy network shown in Figure 2a, except that its output layer produces a scalar Q-value and uses a linear activation function.

The utilization of sub-critic networks results in a fully decentralized scheme bearing resemblance to the independent DDPG (IDDPG) algorithm. IDDPG struggles to converge stably due to the dynamic strategies of other agents, hindering the learning of cooperative relationships. The center-critic network is introduced to directly address this challenge. The network architecture is illustrated in Figure 2b. Compared to the sub-critic network, the local SDTCM observations of the multi-UAVs are integrated at the input layer of the branch CNN, while the joint action vector is concatenated at the branch MLP input layer. The neuron count is correspondingly increased to accommodate the expanded input dimension. The CSC-DDPG framework incorporates N_c center-critic networks $Q_{ck}(\mathbf{o}, \mathbf{a}, \phi_{ck})$, which possess the same architecture yet independent parameters, where $k \in \{1, 2, \dots, N_c\}$ and ϕ_{ck} represent the network parameters. Each Q_{ck} accepts the joint observation $\mathbf{o}$ and joint action $\mathbf{a}$ as inputs to produce an estimated joint action value. To derive the final centralized value assessment for the joint action, the multi-center-critic function $Q_c(\mathbf{o}, \mathbf{a}, \phi_c)$ utilizing the N_c outputs can be expressed as

$$Q_c(\mathbf{o}, \mathbf{a}, \phi_c) = \frac{1}{N_c - 1} \left(\sum_{k=1}^{N_c} Q_{ck} - \max_{1 \leq k \leq N_c} Q_{ck} \right) \quad (24)$$

where ϕ_c denotes the set of center-critic networks parameters and $Q_{ck} \triangleq Q_{ck}(\mathbf{o}, \mathbf{a}, \phi_{ck})$. By averaging the value estimates that remain once the maximum is excluded, Q_c determines the final joint action value evaluation, directly alleviating the value function overestimation issue from an outcome perspective. The purpose of center-critic networks is to promote the cooperation abilities acquisition of policy networks through accurate joint action value estimation.

3.5. Center-Sub-Critics Deep Deterministic Policy Gradient

CSC-DDPG utilizes sub-critic networks and center-critic networks to jointly approximate the action value function Q_i of UAV U_i , which can be expressed as

$$Q_i \approx \varepsilon Q_c(\mathbf{o}, \mathbf{a}, \phi_c) + (1 - \varepsilon) Q_{si}(o_i^c, a_i, \phi_{si}), \quad (25)$$

where ε denotes the multi-center-critic function weight. The gradient descent method is employed to optimize ϕ_{ck} and ϕ_{si} . The target function Q_i is fitted by minimizing the temporal difference error (TD-Error), defined as the discrepancy between the current action value estimate and the next time step estimate combined with the action reward. For the center-critic parameters ϕ_{ck} , the loss function can be described as

$$L(\phi_{ck}) = E_{(\mathbf{o}, \mathbf{a}') \sim \rho} \left[\left(Q_{ck}(\mathbf{o}, \mathbf{a}) - r - \gamma Q_c(\mathbf{o}', \mathbf{a}' | \mathbf{a}' = \pi'(\mathbf{o}')) \right)^2 \right], \quad (26)$$

where $(\cdot)'$ indicates the subsequent state of the variables, while r represents the instantaneous joint reward acquired by the multi-UAVs. Correspondingly, the loss of Q_{si} is defined as

$$L(\phi_{si}) = E_{(o_i^c, o_i^{c'}) \sim \rho} \left[\left(Q_{si}(o_i^c, a_i) - r_i - \gamma Q_{si}(o_i^{c'}, a_i' | a_i' = \pi'(o_i^{c'})) \right)^2 \right], \quad (27)$$

where r_i indicates the immediate collective reward obtained by the UAV U_i .

Figure 3 depicts the framework of the CSC-DDPG algorithm. To improve the stability of the target for action value estimation during the training phase, similar to DDPG, a series of target networks is configured for the policy and critic model. The target networks output the action value estimates for the next time step, which serves as the update target in TD-Error. The target networks achieve smooth and stable soft updates by summing the parameters of the current networks and the target networks with a tiny weight τ at each update time step.

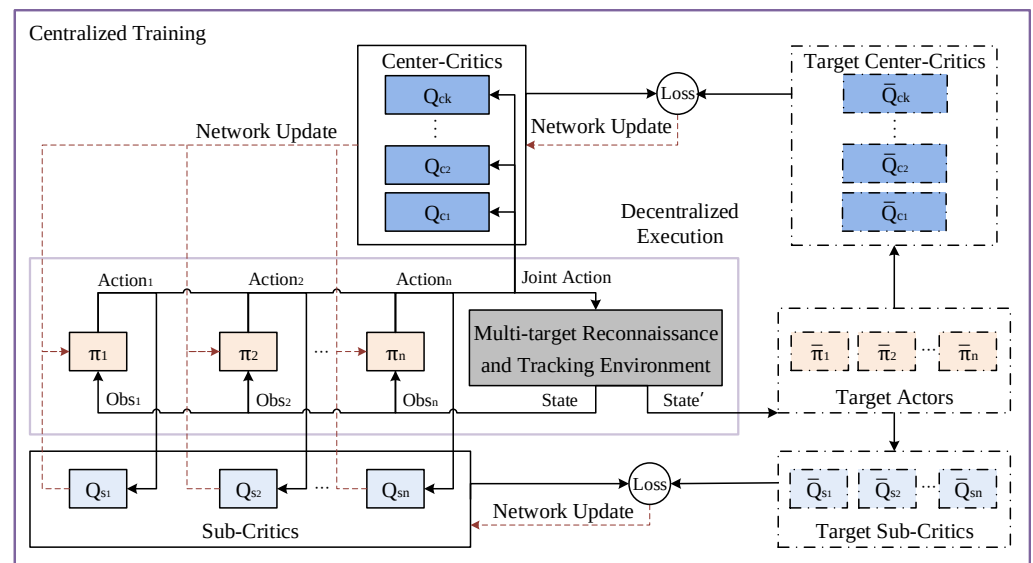


Figure 3. The framework of the proposed CSC-DDPG.

Given the homogeneity of the UAVs, both the policy network $\pi(o_i^c, \psi)$ and the sub-critic network $Q_s(o_i^c, a_i, \phi_s)$ are shared across all agents. This sharing mechanism significantly reduces the computational load of the CSC-DDPG framework while also mitigating the non-stationarity issue inherent in MADDPG. During training, UAV experiences remain independent of agent identities, enabling cross-swarm experience sharing that critically augments training data diversity and elevates training efficiency. At each time step, experience data (o_t, a_t, r_t, o_{t+1}) are stored in the replay buffer E . Network parameters undergo gradient update using N_d samples randomly drawn from E per update cycle. The loss functions for the shared network $\pi(\psi)$, $Q_s(\phi_s)$, and the center-critic networks $Q_{ck}(\phi_{ck})$ are consequently rewritten as

$$\begin{cases} L(\psi) = \frac{1}{N_d N_u} \sum_{d=1}^{N_d} \sum_{i=1}^{N_u} -Q_i^{(d)} \\ L(\phi_s) = \frac{1}{N_d N_u} \sum_{d=1}^{N_d} \sum_{i=1}^{N_u} (Q_{si}^{(d)} - r_i^{(d)} - \gamma \bar{Q}_{si}^{(d)})^2, \\ L(\phi_{ck}) = \frac{1}{N_d} \sum_{d=1}^{N_d} (Q_{ck}^{(d)} - r^{(d)} - \gamma \bar{Q}_c^{(d)})^2 \end{cases} \quad (28)$$

where d refers to a set of experiences and $\bar{\cdot}$ represents the corresponding target network. Furthermore, during the training phase, Gaussian noise featuring linearly damped variance over time is introduced to the policy network outputs, enhancing the exploration capability of the deterministic policy approach while preventing convergence to the local optima. The pseudocode for the CSC-DDPG is detailed in Algorithm 1.

Algorithm 1 CSC-DDPG for MTST

```

Initialize a shared policy network  $\pi$  with random parameters  $\psi$  for multi-UAVs.
Initialize sub-critic network  $Q_s$  and  $N_c$  center-critic networks  $Q_{ck}$  with random parameters  $\phi_s$  and  $\phi_{ck}$ .
Initialize target networks  $\bar{\pi}$ ,  $\bar{Q}_s$  and  $\bar{Q}_{ck}$  with  $\bar{\psi} \leftarrow \psi$ ,  $\bar{\phi}_s \leftarrow \phi_s$  and  $\bar{\phi}_{ck} \leftarrow \phi_{ck}$ .
Initialize experience replay buffer  $E$ .
for  $episode = 1$  to  $end\_episode$  do
  for  $env = 0$  to  $parallel\_envs$  do
    Initialize MTST environment and global state  $s_{t=0}$ .
    for  $t = 1$  to  $T$  do
      for  $i = 1$  to  $N_u$  do
         $\mathbf{o}_t \leftarrow \mathbf{o}_{i,t}^c = J(s_t, U_i, U_{i,t}^n)$ 
         $\mathbf{a}_t \leftarrow \mathbf{a}_{i,t} = \pi(\mathbf{o}_{i,t}^c) + \text{Gaussian noise}$ 
         $\mathbf{r}_t \leftarrow \mathbf{r}_{i,t} = R_i(s_t, \mathbf{a}_t)$ 
      end for
       $\mathbf{o}_{t+1} \leftarrow s_{t+1} = P_s(s_{t+1}|s_t, \mathbf{a}_t)$ 
       $E$  stores  $(\mathbf{o}_t, \mathbf{a}_t, \mathbf{r}_t, \mathbf{o}_{t+1})$ .
    end for
  end for
  if  $episode \% update\_frequency == 0$  do
    Randomly sample  $N_d$  experiences from  $E$ .
     $\mathbf{a}_{t+1} \leftarrow \mathbf{a}_{i,t+1} = \bar{\pi}(\mathbf{o}_{i,t+1}^c)$ 
    For  $\psi$ ,  $\phi_s$  and  $\phi_{ck}$  as  $\Lambda$  do:
       $\Lambda \leftarrow \Lambda + learning\_rate \nabla L(\Lambda)$ 
    For  $\bar{\psi}$ ,  $\bar{\phi}_s$  and  $\bar{\phi}_{ck}$  as  $\bar{\Lambda}$  do:
       $\bar{\Lambda} \leftarrow \tau \Lambda + (1 - \tau) \bar{\Lambda}$ 
    Update  $learning\_rate$  with cosine annealing
  end if
end for

```

4. Simulation Results

4.1. Simulation Parameters Setting and Hyperparameter Analysis

The CSC-DDPG algorithm workflow is implemented on a personal computer equipped with an RTX 4060 GPU. The task region C is configured with $c_x^{max} = c_y^{max} = 20$ and a grid size $C_l = 1$ to facilitate practical implementation. A total of 4 UAVs, 4 no-fly zones, and 8 USV targets are randomly initialized within C . The entire task duration was discretized into $T = 30$ time steps. Other essential parameters are provided in Table 1.

Regarding the selection of center-critic networks number N_c , since the training phase requires removing the maximum Q-value while taking the average of the remaining outputs, N_c must be greater than 2. However, excessive networks will increase the number of training parameters and even introduce greater variance in policy updates. This paper employs three center-critic networks to minimize computational overhead while implementing the CSC-DDPG framework. This paper utilizes the hyperparameter ε to balance the utility of the centralized critics and decentralized critics within CSC-DDPG, which serves as a key parameter for suppressing value overestimation and achieving a variance-bias trade-off. To determine an appropriate range for ε , $[0, 1]$ is divided into ten intervals with a step size of 0.1. The training is carried out five times with random values in each interval. To

analyze convergence and evaluate policy performance, the average episode reward for the final 20% of training steps was calculated, and a box plot of the statistics is shown in Figure 4.

Table 1. Key simulation parameters.

Environment		Training	
Parameter	Value	Parameter	Value
v_{max}, a_{max}	1.5, 1	ω_1, ω_2	1, 10
r_s, r_c	1.5, 6	$\omega_3, \omega_4, \omega_5$	10, 10, 5
r_{sn}, r_{sb}	3, 1.5	ω_6, ω_7	2, 5
p_t	0.95	ω_8, ω_9	2, 2
p_f	0.05	l_ψ, l_ϕ	0.0003, 0.0001
v_k	0.5	Total training step	5,000,000
r_o	7	Replay buffer size	100,000
α	0.7	Update frequency	1

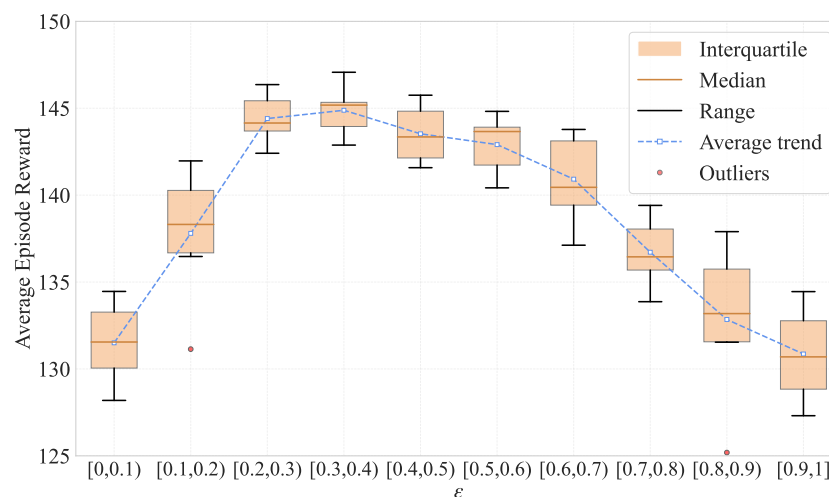


Figure 4. Variation in episode reward across multi-center-critic function weight ϵ .

As ϵ approaches 1, the center-critic networks play a dominant role, making the training framework more similar to MADDPG. Conversely, as ϵ approaches 0, the sub-critic networks take the lead, aligning the framework more closely with IDDPG. Analysis of the results in Figure 4 reveals that as ϵ increases, the average episode reward shows a declining trend in the later stages, with only a rapid rise observed in the early stages. This is related to the ability of the center-critic network to capture the joint actions of the multi-UAVs, enabling better learning of cooperative strategies. However, these joint actions also encompass a larger range of data variance, which further leads to strategy collapse and getting stuck in a local optimum. Conversely, on the one hand, when sub-critic networks predominantly assess the independent actions of multi-UAVs, each data variance of the sub-critic network originates solely from its own UAV action sampling. This mitigates evaluation instability, reducing the likelihood of policy collapse into suboptimal traps. On the other hand, excessive reliance on sub-critic networks amplifies state capture errors, as they focus solely on their own actions while neglecting how neighboring UAVs alter the global state. This is reflected in the interquartile range of Figure 4, where the interquartile range is larger when ϵ is closer to 0 than closer to 1. Additionally, note that the probability of outliers increases as ϵ approaches 1, which correlates with the further increased probability of policy collapse caused by the center-critic network.

Building upon the preceding analysis, $\varepsilon = 0.3$ is chosen for the subsequent simulations in this work. Figure 5 illustrates how the Q-value estimates of the CSC network and its dependent center-critic and sub-critic networks evolve with training steps. It is evident that the Q-value of the CSC network is between those of the centralized critics and the decentralized critic, leaning closer to the decentralized part. Additionally, due to random parameter initialization, estimation bias exists among the center-critic networks during the early training phase. However, as the training progresses, the network parameters converge to be consistent, and this bias progressively diminishes.

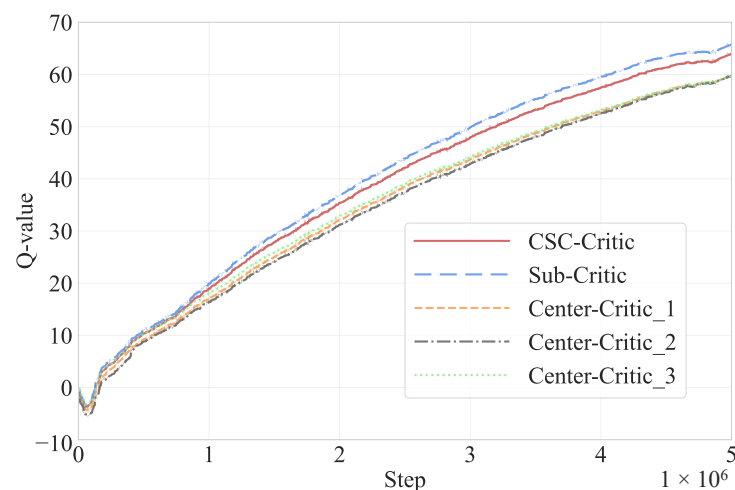


Figure 5. Variation of Q-value with the number of training steps.

4.2. Convergence Analysis

MADDPG, IDDPG, and the Multi-Agent Twin Delayed Deep Deterministic Policy Gradient (MATD3) were selected as baseline algorithms for comparative analysis of CSC-DDPG convergence. MATD3, a modification of TD3 [38] for multi-agent environments, employs two critic networks and a delayed policy update mechanism to improve MADDPG. Figure 6 shows the variation of the episode rewards with the training steps for the four algorithms. The dark lines represent the average of five training results, while the light shaded areas denote the corresponding standard deviation.

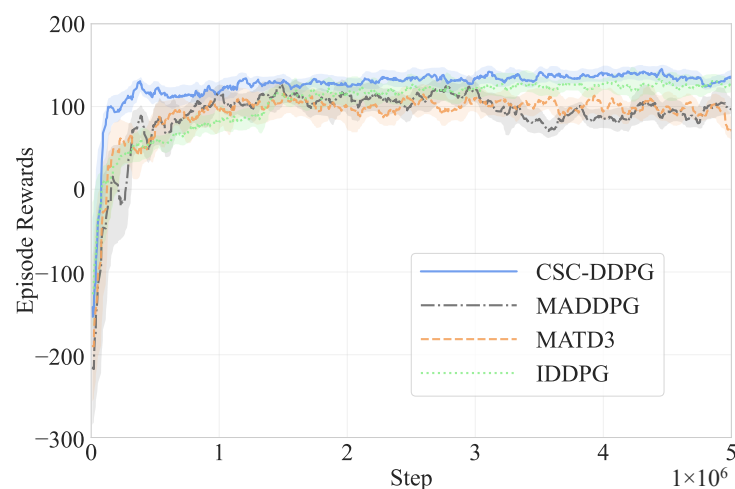


Figure 6. Variation of episode reward with the training steps for CSC-DDPG and baselines.

It can be seen from Figure 6 that CSC-DDPG converges to a higher average reward and a lower policy variance, indicating that the proposed algorithm learned a more optimal and stable policy. Meanwhile, both MADDPG and MATD3 exhibited certain degrees of policy

collapse. This stems from relying solely on the center-critic network while ignoring the negative contributions of suboptimal actions by individual agents. The suboptimal actions will increase the variance in policy updates, potentially leading to negative optimizations or even abrupt falls. Benefiting from the hybrid advantages, CSC-DDPG also outperforms baselines in convergence speed. Compared to IDDPG in particular, the introduction of center-critic networks enables the individual UAV to consider changes of the global state and execute more efficient actions, thereby acquiring higher-quality experience at the early stage of training. Simultaneously, the proposed method also mitigates the tendency of IDDPG to become trapped in local optima, demonstrated by CSC-DDPG converging to a higher episode reward.

To validate the performance of the CSC-DDPG after convergence, the policy network is deployed, and the state of the task area at different time steps during a single episode is displayed. As shown in Figure 7, the first and second rows represent the state of the environment at steps 15 and 30, respectively. Figure 7a,d illustrate the evolution of the SDTCM, where darker areas indicate regions with higher USV existence probability, while lighter areas denote regions with lower probability. Due to the spatio-temporal decay of SDTCM, probability information gradually dilutes over time, guiding the UAV to revisit the detected targets. This also explains why the edge regions of the SDTCM predominantly appear blue. Figure 7b,e visualize the SDTCM information entropy. At step 15, numerous lighter grids represent unexplored areas. By step 30, the darker low-information areas have expanded further, and their edge information continues to guide the multi-UAVs to explore unknown regions. Figure 7c,f display the trajectory maps of the multi-UAVs. It can be seen that the majority of targets have been detected at step 15. Subsequently, while exploring unknown areas, the multi-UAVs repeatedly revisit detected USVs to update their position information.

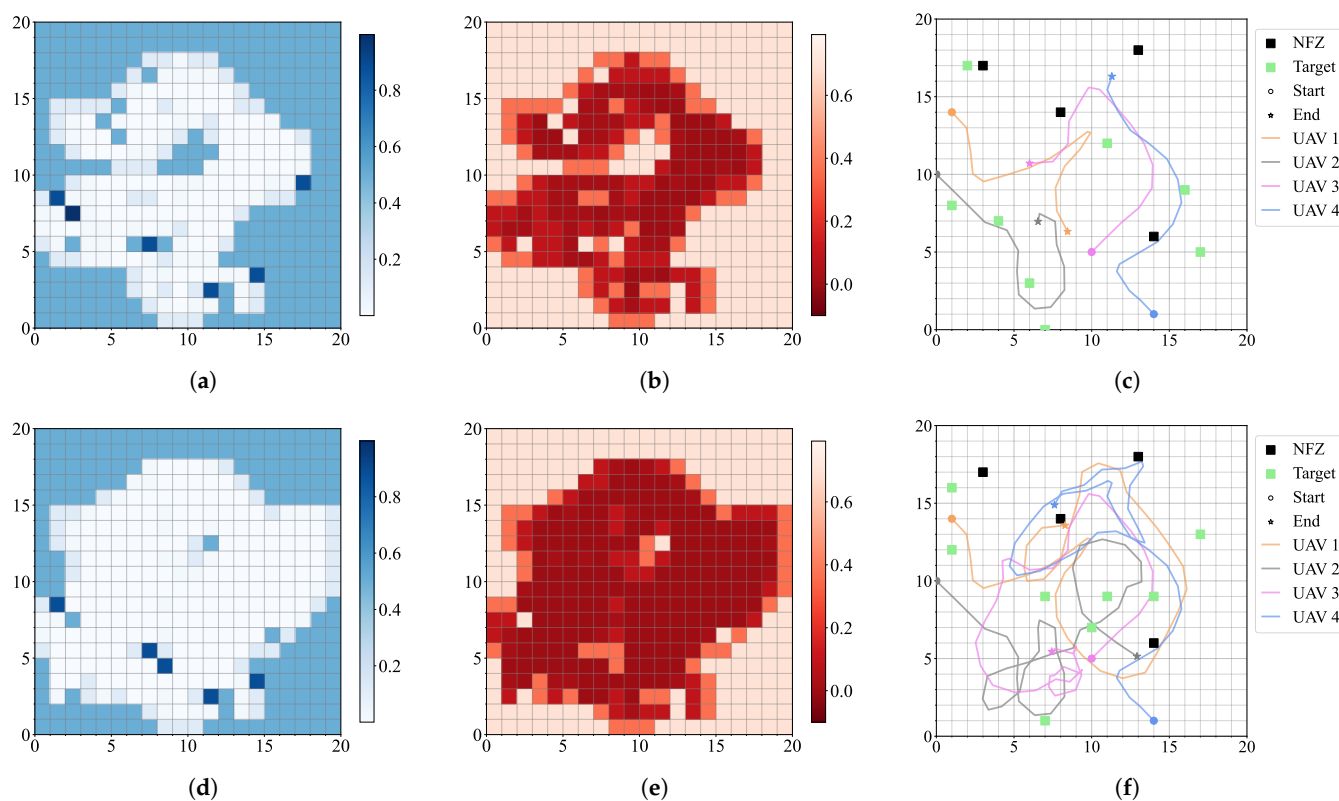


Figure 7. The changes of environment state with steps. (a) SDTCM at step 15. (b) Entropy at step 15. (c) Trajectories at step 15. (d) SDTCM at step 30. (e) Entropy at step 30. (f) Trajectories at step 30.

4.3. Performance Comparison Analysis

The performances of CSC-DDPG with non-reinforcement learning algorithms in multi-UAVs MTST for USV swarm are compared to demonstrate the efficacy of the proposed algorithm, while the generalization and scalability of CSC-DDPG in different scale environments are discussed. The selected comparison algorithms are cooperative multi-robot observation of multi-moving targets (CMOMMT), ACO, and random algorithms. CMOMMT [3,39–41] guides agent motion through the resultant force derived from target attraction and partner repulsion. CMOMMT is improved by generating reliable target attraction based on SDTCM while incorporating target existence probability and information entropy as force weights. The modified approach is termed CMOMMT with SDTCM (CMOMMT-M). Based on the idea of ACO, the action space of UAV is modified to discrete 8-directional actions. The pheromone table is designed as an $8 \times c_x^{max} \times c_y^{max}$ grid action guidance map. At the same time, SDTCM and the locations of neighboring UAVs and no-fly zones serve as real-time dynamic heuristic information. Together with pheromones, these factors determine the action probability distribution of UAV, which is denoted as ACO with SDTCM (ACO-M). In the random algorithm, each drone obtains its action by randomly selecting a resultant force with constrained magnitude in any direction.

The performance evaluation metrics for the MTST task are defined as the average of the target detected rate, the standard deviation of the detected rate, and the sum or mean of grid information entropy. The target detected rate is calculated as the number of USVs detected times divided by the current time step. All metric data were obtained by averaging results from testing under 50 randomly initialized configurations of no-fly zones, targets, and multi-UAV positions. Scalability and generalization testing encompass two aspects: variations in UAV quantity and environmental scale. Once trained in the benchmark environment, the CSC-DDPG policy model is deployed in test environments of various configurations. No retraining is required for different configurations, and it is also a challenge for the generalization of CSC-DDPG when facing unknown environments.

Firstly, based on the benchmark environment, different numbers of UAVs are deployed for algorithm comparison tests, observing the impact of multi-UAVs scale changes on performance metrics. As shown in Figure 8a,b, throughout the transition from under-saturation to over-saturation of multi-UAVs, CSC-DDPG consistently maintained the average detected rate within the top tier, performing on par with comparison algorithms. This validates the effectiveness and scalability of CSC-DDPG. The proposed algorithm demonstrates its advantage in terms of detected rate standard deviation, showing that the model can perform more uniform revisit detection on detected USVs across different numbers of UAVs. Focusing on ACO-M, non-reinforcement learning (non-RL) algorithms are observed to tend to be satisfied with continuously tracking the same USV to maintain decent detected rates while ignoring the fairness of importance among multi-UAVs.

Figure 8c shows that all the algorithms except the random algorithm have carried out a considerable degree of autonomous exploration in unknown environments to reduce the environmental uncertainty.

Subsequently, based on the benchmark environment, the number of UAVs remained constant while varying the side length of the square task area to explore the performance of multi-UAVs in smaller and larger environments. While changing the task area scale, the number of no-fly zones and USVs, and the maximum time step of one episode are scaled proportionally to match the benchmark environment. As shown in Figure 9a,b, the trained model maintains comparable target detection capabilities to the baseline algorithms when deployed in environments smaller or larger than the training environment. In extremely large-scale environments, it even slightly outperforms the baseline algorithms. Furthermore, CSC-DDPG consistently completes the MTST task with a lower detected standard

deviation than the baseline algorithms across most environments, further validating the generalization and scalability of the proposed algorithm. It is noted that the random algorithm achieves lower detected standard deviation in the large-scale environment. This occurs because the fairness inherent to a random strategy, which lacks targeted intent, is amplified as the environment scale increases. Figure 9c illustrates that, compared to random algorithms, other algorithms exhibit purposeful environmental exploration behavior, rather than merely wandering around their initial positions. Additionally, CSC-DDPG achieves lower average grid information entropy in large-scale environments, demonstrating more intelligent multi-UAV coordination capability than the comparison algorithms by avoiding redundant inspection of areas recently explored by neighbor UAVs.

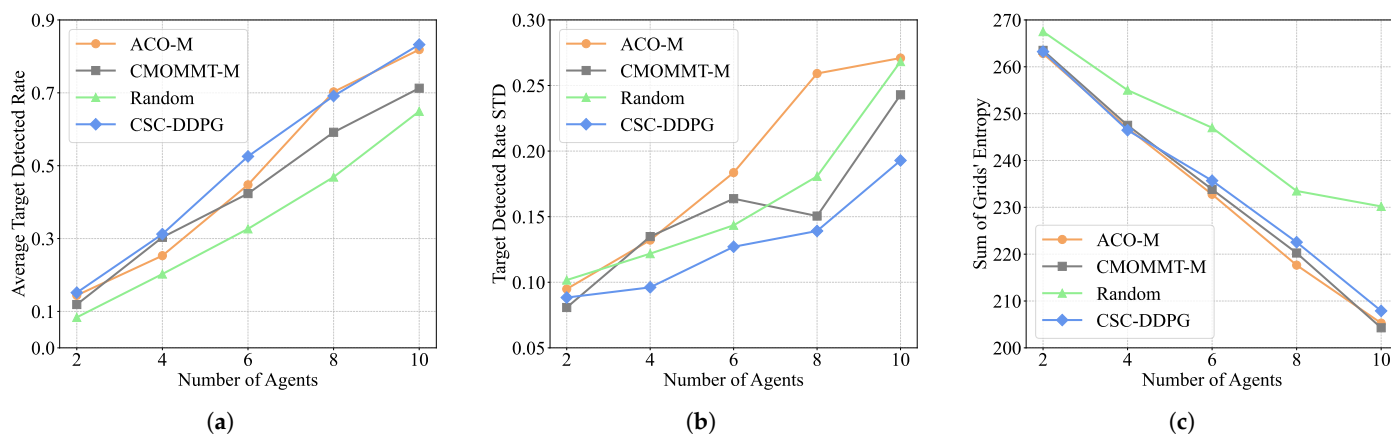


Figure 8. Variation of the performance metrics with the number of UAVs. (a) Average target detected rate. (b) Target detected rate STD. (c) Sum of grids' entropy.

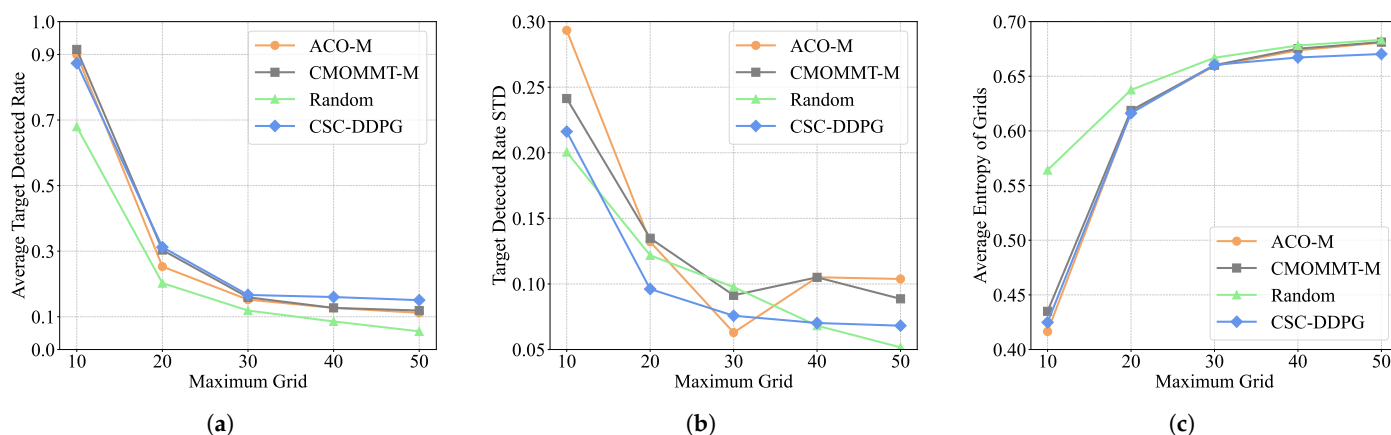


Figure 9. Variation of the performance metrics with the scale of environment. (a) Average target detected rate. (b) Target detected rate STD. (c) Average entropy of grids.

5. Conclusions

A hybrid critics MARL framework is proposed in this paper to address the multi-UAVs MTST for USV swarm problem considering detection errors. The multi-UAVs information fusion mechanism is first established by Bayesian update, and based on this, the SDTCM is designed to support the fusion communication observation. Building upon communication-enhanced POMDP, the CSC-DDPG algorithm is introduced, which comprehensively utilizes multiple center-critic networks and local sub-critic networks to address variance-bias balance and overestimation in model training. Finally, through simulation comparisons with various RL and non-RL algorithms across environments of different scales, the efficacy and scalability of the proposed algorithm are verified, and the

cooperative searching and tracking capabilities of the multi-UAVs system are improved. Future research will be carried out in more complex high-dimensional environments and real-world deployment experiments of multi-UAVs, in order to achieve contributions with greater practical engineering significance. The transition from simulation to real-world deployment, taking into account communication topology and latency, is the primary challenge that must be addressed. Developing dynamic models that track how UAV observation errors vary with environmental factors represents a key research direction for accelerating practical deployment in the future. In terms of theoretical research, the core hyperparameters and weights of adaptive learning represent future research directions.

Author Contributions: Conceptualization, Y.H.; methodology, Y.H.; software, Y.H.; validation, Y.H.; formal analysis, B.L.; investigation, Y.H.; resources, B.L.; data curation, Y.H.; writing—original draft preparation, Y.H. and B.L.; writing—review and editing, Y.H., X.M. and B.L.; visualization, Y.H. and X.M.; supervision, X.M.; project administration, X.M. and B.L.; funding acquisition, B.L. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was supported in part by National Natural Science Foundation of China (62473249), Natural Science Foundation of Shanghai (23ZR1426600).

Data Availability Statement: Data are contained within the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wei, D.; Zhang, L.; Liu, Q.; Chen, H.; Huang, J. UAV Swarm Cooperative Dynamic Target Search: A MAPPO-Based Discrete Optimal Control Method. *Drones* **2024**, *8*, 214. [\[CrossRef\]](#)
2. Liu, H.; Li, B.; Ahn, C.K. Asymptotically Stable-Learning-Based Formation Control for Multi-quadrotor UAVs. *IEEE Internet Things J.* **2025**, *12*, 24985–24995. [\[CrossRef\]](#)
3. Parker, L.; Emmons, B. Cooperative multi-robot observation of multiple moving targets. In *Proceedings of the International Conference on Robotics and Automation, Albuquerque, NM, USA, 20–25 April 1997*; IEEE: New York, NY, USA, 1997; Volume 3, pp. 2082–2089.
4. Gammell, J.D.; Strub, M.P. Asymptotically Optimal Sampling-Based Motion Planning Methods. *Annu. Rev. Control. Robot. Auton. Syst.* **2021**, *4*, 295–318. [\[CrossRef\]](#)
5. Li, B.; Gong, W.; Yang, Y.; Xiao, B.; Ran, D. Appointed Fixed Time Observer-Based Sliding Mode Control for a Quadrotor UAV Under External Disturbances. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 290–303. [\[CrossRef\]](#)
6. Zhong, X.; Huo, Y.; Dong, X.; Liang, Z. QoS-Compliant 3-D Deployment Optimization Strategy for UAV Base Stations. *IEEE Syst. J.* **2021**, *15*, 1795–1803.
7. Al-Turjman, F.; Zahmatkesh, H.; Al-Oqily, I.; Daboul, R. Optimized Unmanned Aerial Vehicles Deployment for Static and Mobile Targets' Monitoring. *Comput. Commun.* **2020**, *149*, 27–35. [\[CrossRef\]](#)
8. Zou, J.; Shi, D. Fair Observation of Multiple Moving Targets in Cooperative Multi-UAV Systems. In *Proceedings of the 2021 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 28–30 June 2021*; IEEE: New York, NY, USA, 2021; pp. 562–568.
9. Huang, H.; Savkin, A.V. Aerial Surveillance in Cities: When UAVs Take Public Transportation Vehicles. *IEEE Trans. Autom. Sci. Eng.* **2023**, *20*, 1069–1080. [\[CrossRef\]](#)
10. Scherer, J.; Rinner, B. Multi-UAV Surveillance With Minimum Information Idleness and Latency Constraints. *IEEE Robot. Autom. Lett.* **2020**, *5*, 4812–4819. [\[CrossRef\]](#)
11. Zhen, Z.; Chen, Y.; Wen, L.; Han, B. An intelligent cooperative mission planning scheme of UAV swarm in uncertain dynamic environment. *Aerosp. Sci. Technol.* **2020**, *100*, 105826. [\[CrossRef\]](#)
12. Pérez-Carabaza, S.; Besada-Portas, E.; López-Orozco, J.A. Minimizing the searching time of multiple targets in uncertain environments with multiple UAVs. *Appl. Soft Comput.* **2024**, *155*, 111471. [\[CrossRef\]](#)
13. Phung, M.D.; Ha, Q.P. Motion-encoded particle swarm optimization for moving target search using UAVs. *Appl. Soft Comput.* **2020**, *97*, 106705. [\[CrossRef\]](#)
14. Xu, L.; Cao, X.; Du, W.; Li, Y. Cooperative path planning optimization for multiple UAVs with communication constraints. *Knowl.-Based Syst.* **2023**, *260*, 110164. [\[CrossRef\]](#)

15. Duan, H.; Zhao, J.; Deng, Y.; Shi, Y.; Ding, X. Dynamic Discrete Pigeon-Inspired Optimization for Multi-UAV Cooperative Search-Attack Mission Planning. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 706–720. [[CrossRef](#)]
16. Li, B.; Liu, H.; Ahn, C.K.; Gong, W. Optimized intelligent tracking control for a quadrotor unmanned aerial vehicle with actuator failures. *Aerosp. Sci. Technol.* **2024**, *144*, 108803. [[CrossRef](#)]
17. Garaffa, L.C.; Basso, M.; Konzen, A.A.; de Freitas, E.P. Reinforcement Learning for Mobile Robotics Exploration: A Survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 3796–3810. [[CrossRef](#)] [[PubMed](#)]
18. Ning, Z.; Xie, L. A survey on multi-agent reinforcement learning and its application. *J. Autom. Intell.* **2024**, *3*, 73–91. [[CrossRef](#)]
19. Oliehoek, F.A.; Spaan, M.T.J.; Vlassis, N. Optimal and approximate Q-value functions for decentralized POMDPs. *J. Artif. Int. Res.* **2008**, *32*, 289–353. [[CrossRef](#)]
20. Yu, C.; Velu, A.; Vinitzky, E.; Gao, J.; Wang, Y.; Bayen, A.; Wu, Y. The surprising effectiveness of ppo in cooperative multi-agent games. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24611–24624.
21. Liu, J.; Wei, S.; Li, B.; Wang, T.; Qi, W.; Han, X.; Hou, G.; Li, K.; Lin, Y.; Xue, D.; et al. Dual-timescale hierarchical MADDPG for Multi-UAV cooperative search. *J. King Saud Univ. Comput. Inf. Sci.* **2025**, *37*, 142. [[CrossRef](#)]
22. Gong, Z.; Xu, Y.; Luo, D. UAV Cooperative Air Combat Maneuvering Confrontation Based on Multi-agent Reinforcement Learning. *Unmanned Syst.* **2023**, *11*, 273–286.
23. Shen, G.; Lei, L.; Zhang, X.; Li, Z.; Cai, S.; Zhang, L. Multi-UAV Cooperative Search Based on Reinforcement Learning with a Digital Twin Driven Training Framework. *IEEE Trans. Veh. Technol.* **2023**, *72*, 8354–8368. [[CrossRef](#)]
24. Hou, Y.; Zhao, J.; Zhang, R.; Cheng, X.; Yang, L. UAV Swarm Cooperative Target Search: A Multi-Agent Reinforcement Learning Approach. *IEEE Trans. Intell. Veh.* **2024**, *9*, 568–578.
25. Zhang, B.; Lin, X.; Zhu, Y.; Tian, J.; Zhu, Z. Enhancing Multi-UAV Reconnaissance and Search Through Double Critic DDPG With Belief Probability Maps. *IEEE Trans. Intell. Veh.* **2024**, *9*, 3827–3842. [[CrossRef](#)]
26. Zhou, W.; Li, J.; Liu, Z.; Shen, L. Improving multi-target cooperative tracking guidance for UAV swarms using multi-agent reinforcement learning. *Chin. J. Aeronaut.* **2022**, *35*, 100–112. [[CrossRef](#)]
27. Cao, X.; Luo, H.; Tai, J.; Jiang, R.; Wang, G. Multi-agent target search strategy optimization: Hierarchical reinforcement learning with multi-criteria negative feedback. *Appl. Soft Comput.* **2023**, *149*, 110999.
28. Yue, L.; Yang, R.; Zuo, J.; Yan, M.; Zhao, X.; Lv, M. Factored Multi-Agent Soft Actor-Critic for Cooperative Multi-Target Tracking of UAV Swarms. *Drones* **2023**, *7*, 150.
29. Liu, Y.; Li, X.; Wang, J.; Wei, F.; Yang, J. Reinforcement-Learning-Based Multi-UAV Cooperative Search for Moving Targets in 3D Scenarios. *Drones* **2024**, *8*, 378. [[CrossRef](#)]
30. Tilak, O.; Mukhopadhyay, S. Partially decentralized reinforcement learning in finite, multi-agent Markov decision processes. *AI Commun.* **2011**, *24*, 293–309. [[CrossRef](#)]
31. Wang, C.; Wang, J.; Zhang, X.; Zhang, X. Autonomous navigation of UAV in large-scale unknown complex environment with deep reinforcement learning. In *Proceedings of the 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Montreal, QC, Canada, 14–16 November 2017*; IEEE: New York, NY, USA, 2017; pp. 858–862.
32. Shen, G.; Lei, L.; Li, Z.; Cai, S.; Zhang, L.; Cao, P.; Liu, X. Deep Reinforcement Learning for Flocking Motion of Multi-UAV Systems: Learn from a Digital Twin. *IEEE Internet Things J.* **2022**, *9*, 11141–11153. [[CrossRef](#)]
33. Hou, Y.; Li, B.; Miao, X. Cooperative Target Search for UAV Swarm Based on Multi-Hybrid Critic DDPG. In *Proceedings of the 2025 IEEE 23rd International Conference on Industrial Informatics (INDIN), Kunming, China, 12–15 July 2025*; IEEE: New York, NY, USA, 2025; pp. 1–6.
34. Dibangoye, J.S.; Amato, C.; Buffet, O.; Charpillet, F. Optimally solving Dec-POMDPs as continuous-state MDPs. *J. Artif. Int. Res.* **2016**, *55*, 443–497. [[CrossRef](#)]
35. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; A Bradford Book: Cambridge, MA, USA, 2018.
36. Oroojlooy, A.; Hajinezhad, D. A review of cooperative multi-agent deep reinforcement learning. *Appl. Intell.* **2023**, *53*, 13677–13722. [[CrossRef](#)]
37. Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; Riedmiller, M. Deterministic policy gradient algorithms. In *Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 22–24 June 2014*; ML Research Press: Norfolk, MA, USA, 2014; Volume 32, pp. I-387–I-395.
38. Fujimoto, S.; van Hoof, H.; Meger, D. Addressing Function Approximation Error in Actor-Critic Methods. In *Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018*; ML Research Press: Norfolk, MA, USA, 2018; Volume 80, pp. 1587–1596.
39. Parker, L.E. Distributed algorithms for multi-robot observation of multiple moving targets. *Auton. Robot.* **2002**, *12*, 231–255. [[CrossRef](#)]

40. Ding, Y.; Zhu, M.; He, Y.; Jiang, J. P-CMOMMT Algorithm For The Cooperative Multi-Robot Observation of Multiple Moving Targets. In *Proceedings of the 6th World Congress on Intelligent Control and Automation, Dalian, China, 21–23 June 2006*; IEEE: New York, NY, USA, 2006; Volume 2, pp. 9267–9271.
41. Kolling, A.; Carpin, S. Cooperative observation of multiple moving targets: An algorithm and its formalization. *Int. J. Robot. Res.* **2007**, *26*, 935–953. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.