drones

MDPI

Article

Factored Multi-Agent Soft Actor-Critic for Cooperative Multi-Target Tracking of UAV Swarms

Longfei Yue, Rennong Yang, Jialiang Zuo, Mengda Yan, Xiaoru Zhao and Maolong Lv *

Air Traffic Control and Navigation College, Air Force Engineering University, Xi'an 710051, China
* Correspondence: maolonglv@163.com

Abstract: In recent years, significant progress has been made in the multi-target tracking (MTT) of unmanned aerial vehicle (UAV) swarms. Most existing MTT approaches rely on the ideal assumption of a pre-set target trajectory. However, in practice, the trajectory of a moving target cannot be known by the UAV in advance, which poses a great challenge for realizing real-time tracking. Meanwhile, state-of-the-art multi-agent value-based methods have achieved significant progress for cooperative tasks. In contrast, multi-agent actor-critic (MAAC) methods face high variance and credit assignment issues. To address the aforementioned issues, this paper proposes a learning-based factored multi-agent soft actor-critic (FMASAC) scheme under the maximum entropy framework, where the UAV swarm is able to learn cooperative MTT in an unknown environment. This method introduces the idea of value decomposition into the MAAC setting to reduce the variance in policy updates and learn efficient credit assignment. Moreover, to further increase the detection tracking coverage of a UAV swarm, a spatial entropy reward (SER), inspired by the spatial entropy concept, is proposed in this scheme. Experiments demonstrated that the FMASAC can significantly improve the cooperative MTT performance of a UAV swarm, and it outperforms existing baselines in terms of the mean reward and tracking success rates. Additionally, the proposed scheme scales more successfully as the number of UAVs and targets increases.

Keywords: UAV swarm; multi-target tracking; multi-agent reinforcement learning; soft actor-critic; value decomposition; credit assignment

check for updates

Citation: 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. https://doi.org/10.3390/drones7030150

Academic Editor: Diego González-Aguilera

Received: 26 January 2023
Revised: 20 February 2023
Accepted: 20 February 2023
Published: 22 February 2023

Copyright: © 2023 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) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Multi-target tracking (MTT) with an unmanned aerial vehicle (UAV) swarm has attracted widespread research interest due to its important role in highly mobile networks [1–4]. In the MTT task [5,6], numerous small UAVs are deployed in a distributed manner across the task area to search for unknown vehicle user targets and track the perceived targets cooperatively. This technique addresses the requirement of tracking moving targets for periods of time by taking advantage of the flexibility, low cost, and high mobility of UAVs. A key challenge in MTT is how to coordinate multiple UAVs to track diverse unknown targets accurately and consecutively [1].

MTT with UAV swarm is essentially a multi-agent cooperative decision-making optimization problem. A large number of MTT studies based on intelligent optimization algorithms have been published in recent decades; studied methods include particle swarm optimization [7], multi-objective optimization [8], and cyclic stochastic optimization algorithms [9,10]. Traditional intelligent optimization algorithms can find the optimal or sub-optimal solution of the objective function through a parallel heuristic search. Such algorithms are gradient-free and have a strong optimization performance in complex non-convex objective functions. However, these algorithms also require online searches to solve the problem, which do not meet the real-time calculation requirements. Moreover, the optimization speed is greatly reduced as the problem's scale grows. Therefore, these methods still present great difficulties in practical applications.

With the great progress of reinforcement learning (RL) and multi-agent reinforcement learning (MARL), some scholars have also attempted to use RL to solve the MTT problem [11–14]. Examples include the mixed noises deep deterministic policy gradient (MN-DDPG) [11] and the reciprocal reward multi-agent actor–critic (MAAC-R) [14]. RL and MARL utilize the agent to interact with the environment by trial and error to allow the UAV to learn and track the target. However, it is not easy to cooperatively realize the multi-target tracking of a UAV swarm using MARL. Generally, cooperative MARL adopts a centralized training with decentralized execution (CTDE) paradigms, which suffer from the global action–value function, whose complexity grows exponentially with the number of agents [15]. Value decomposition enables CTDE to be used in value-based MARL [16–21], but policy-based MARL or multi-agent actor-critic (MAAC) [22–27] frameworks that learn through continuous actions create high variance. This problem occurs because the exploration or sub-optimality of other agents' policies might propagate through the centralized critic and aggravate the training of other agents. Therefore, determining how to realize efficient cooperation for multiple agents remains a significant challenge.

In addition, reward design plays a key role in RL settings. A reasonable and effective reward can significantly improve MTT performance. Information entropy is a measure of the uncertainty of the entropy. Spatial entropy is a measure of the spatial coverage [28], which is important for target searching and tracking tasks. Under this backdrop, using spatial entropy to guide the UAV swarm to search and track targets might improve the cooperation behavior between UAVs.

In this paper, we propose a cooperative MARL scheme called factored multi-agent soft actor-critic (FMASAC) to achieve MTT for a UAV swarm. First, we describe the MTT problem and formulize this multi-agent cooperative decision-making problem as a decentralized partially observable Markov decision process (Dec-POMDP) model. Then, inspired by the idea of value decomposition, we improve the vanilla multi-agent soft actor-critic (MASAC) by factorizing the centralized critic as the composition of the individual critic and using factored soft policy iterations to optimize the policy. On this basis, the FMASAC scheme using spatial entropy reward (SER) is proposed to learn and solve the MTT tasks of the UAV swarm. Finally, simulation experiments are conducted to test the effectiveness, generalization, and scalability of the proposed method. The contributions of this paper are summarized as follows:

1. We formulate a UAV swarm-based MTT problem within a Dec-POMDP setting. The UAVs are partially observable, can only perceive the targets within the observation range, and can communicate with the neighboring UAVs. A recurrent neural network (RNN) is added to the actor-critic network to gather the historical information from the hidden state of the RNN, which solves the problem of incomplete information caused by partial observations. To increase the detection coverage and boost the tracking efficiency, inspired by the concept of spatial entropy, we design a shaping-reward SER. In addition, safe distance constraints are considered in the reward function to avoid collisions.
2. MARL is used to solve the MTT optimization problem, which does not need a pre-set target trajectory and can learn to track targets in an unknown environment. Here, the CTDE paradigm is adopted, where global observation–action history can be accessed during centralized training, and trained policies are executed conditioned only on local information in a decentralized way. Moreover, the trained model can generalize to an unknown and dynamic changing environment.
3. We propose using the FMASAC algorithm, which adopts an entropy maximization MARL for greater exploration and introduces the idea of value decomposition. This algorithm effectively combines the advantages of the value decomposition and MASAC methods, which reduces the variance in policy updates, achieves efficient credit assignment [29], and enables the scalable learning of a centralized critic in Dec-POMDP.

The rest of this paper is organized as follows. Section 2 summarizes related work about the target tracking problem. Section 3 provides a preliminary outline of Dec-POMDP and MASAC. Section 4 formulates the UAV kinematic model and the cooperative MTT

optimization model. Section 5 details the proposed FMASAC-based MTT methods. Experimental results and analyses are presented in Section 6. Finally, Section 7 concludes this paper and envisages future work. For the sake of clarity, all the symbols and notations used in this paper are summarized in Table 1, and the definition of the default symbols in MARL is omitted.

Table 1. Summary of notations.

Symbol	Definition
n, m	The numbers of the UAVs and targets.
i, j, k	The indexes of each UAV, each neighbor, and each target.
$(x_i, y_i), v_i, \varphi_i$	The position, velocity, and heading of UAV i.
$a_{\max}, a_i^t$	The maximum heading angular rate and the action of UAV i.
N_a, n_a	The UAV's cardinality of the discrete action space, and the corresponding index of its discrete action.
d_o, d_c	The maximum observation distance and maximum communication distance.
$d_{i,j}, d_{i,k}, d_{i,O}$	The distance between UAV i and UAV j, target k, and scenario center O.
$o_{i,k} = (x_k^T, y_k^T, v_k^{Tx}, v_k^{Ty})$	The UAV i's observation information about target k.
$c_{i,j} = (x_j, y_j, v_j^x, v_j^y, a_j^{t-1})$	The UAV i's communication information from neighbor j.
$o_i = (x_i, y_i, \varphi_i, o_{i,k}, c_{i,j})$	The UAV i's local observation information.
s	The global state of the environment.

2. Related Work

In Section 2, we review some recent publications on target tracking from two perspectives: traditional optimization methods and reinforcement learning methods. A brief introduction of these two optimization methods is provided in Table 2.

Table 2. The common optimization methods for target tracking.

Methods	Advantage	Limitation
Traditional optimization	Powerful non-convex optimization and convergence performance; robust.	Computationally expensive; poor real-time performance of online searching; limitations in large-scale variable problems.
Reinforcement learning	Implicit modeling unknown environment; data-driven; offline training and online decision framework; near real-time solving speed.	Learning inefficiency; credit assignment; poor interpretation; scalability.

2.1. Traditional Optimization Methods

Target tracking is a classic optimization problem. Therefore, most published works are based on traditional optimization algorithms. Pitre et al. [7] described a joint search and tracking mission for UAVs and presented an objective function that integrates target detection, target tracking, and UAV survivability. A modified particle swarm optimization algorithm was proposed to optimize the detection and tracking trajectory. Jilkov et al. [8] developed a multiple objective optimization algorithm to optimize the search and tracking path under multiple objectives for one or multiple UAVs in uncertain environments. A cyclic stochastic optimization algorithm was adopted in [9] to solve the target tracking problem, which demonstrated that multiple agents can cooperatively search a region and track all sensed targets. More related works can be found in [10].

As we can conclude from the above studies, traditional optimization methods formulate the target tracking task as an optimization problem to solve. Despite powerful optimization and convergence performance, these methods require an online search for an unknown and dynamic changing environment, which is computationally expensive for onboard computers and can reduce the real-time performance of target tracking.

2.2. Reinforcement Learning Methods

With the tremendous advances in RL in recent years, some studies have tried to adopt RL to solve the target tracking problem. Li et al. [11] proposed an MN-DDPG-based online path-planning approach, which can achieve the perception of the target and good self-adaptive flight control for UAVs in the tasks related to maneuvering target tracking. Wang et al. [12] developed a centralized-RL-based cooperative target searching and tracking method for UAV fleets that handles the non-stationarity in multi-agent learning. Rosello et al. [13] formulated the MTT problem as a typical motion planning problem and used a MARL method to learn the tracking task. Zhou et al. [14] designed a MAAC-R algorithm using the maximum reciprocal reward to enable multiple UAVs to autonomously track multiple moving targets according to only the past and current position of the UAVs and the target. Notably, the learned policy can directly scale to other complex scenarios without re-training.

For the RL method, agents learn to track through repeated interactions with the environment and other agents and also learn potential cooperative relationships between the agents. Compared with traditional optimization methods, RL can implicitly encode the complex cooperative tracking rules into the neural networks to realize flexible and real-time tracking in an unknown environment through an offline training and online decision framework. However, existing MARL methods have encountered the issues of learning inefficiency, credit assignment, and scalability.

2.3. UAV Swarm Communication

Moreover, communication is the cornerstone of the UAV swarm's ability to transmit information and achieve cooperation [30,31]. To better realize cooperative target tracking with UAV swarms, many methods has been proposed, including traditional optimization methods [32], learning-based methods [33], etc. More detailed communication examples, such as communication protocol, communication structure, communication object, and communication timing, can be found in Ref. [31].

In summary, different from the above works, our study realizes cooperative and scalable MTT for UAV swarms using a learning-based factored MASAC method. This method requires only the positional and velocity information and the local observation and communication information of the UAVs and targets, and it does not need to acquire the target motion pattern in advance. Moreover, the improved CTDE paradigm enables the efficient cooperative learning and the trained UAV model to be deployed in a distributed manner, with the real-time tracking of moving targets in unknown and dynamic changing environments.

3. Preliminary Analysis

In this section, we give the preliminary evaluation of Dec-POMDP and MASAC, which is the basis of subsequent modeling.

3.1. Dec-POMDP

Here, we consider the UAV swarm cooperative MTT task in which each UAV interacts with the environment only according to its local information (including observation and communication information) to achieve the same team goal. This task can be modeled as a Dec-POMDP $G = \langle I, S, A, P, R, \Omega, O, n, \gamma \rangle$ [34,35], where I is the set of agents, S is the global state of the environment, n is the total number of agents, and $\gamma \in [0, 1]$ is the discount factor. Due to partial observability, at every timestep, each agent i perceives a local observation $o_i \in \Omega$ from the observation function $O(s, i)$ and selects an action $a_i \in A$, forming a joint action $\boldsymbol{a} \in A^n$ leading to the next state s', according to the state transition function $P(s'|s, \boldsymbol{a})$ and a shared team reward $r = R(s, \boldsymbol{a})$ [26]. Each agent learns a deterministic policy $\mu_{\theta_i}(\tau_i)$ or a stochastic policy $\pi_{\theta_i}(a_i|\tau_i)$, conditioned only on the local observation-action history τ_i. The joint stochastic policy $\pi_{\boldsymbol{\theta}} = \langle \pi_{\theta_1}, \cdots, \pi_{\theta_n} \rangle$ induces a joint action-value function $Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a}) = \mathbb{E}_{s_{0:\infty}, \boldsymbol{a}_{0:\infty}}[\sum_{t=0}^{\infty} \gamma^t R(s_t, \boldsymbol{a}_t) | s_0 = s, \boldsymbol{a}_0 = \boldsymbol{a}, \pi_{\boldsymbol{\theta}}]$. Similarly, the joint deterministic policy $\mu_{\boldsymbol{\theta}} = \langle \mu_{\theta_1}, \cdots, \mu_{\theta_n} \rangle$ induces a joint action value function $Q_{tot}(\boldsymbol{\tau}, \boldsymbol{a})$.

3.2. MASAC

Soft Actor-Critic (SAC) is an off-policy, actor-critic algorithm in maximum entropy reinforcement learning [36], which simultaneously maximizes the expected return and the entropy of the policy to encourage exploration. SAC optimizes the critic through temporal-difference (TD) learning by minimizing the loss function:

$$\mathcal{L}_Q(\phi) = \mathbb{E}_{(s,a,r,s')\sim D}[(Q_\phi(s,a) - y)^2] \tag{1}$$

where $y = r(s,a) + \gamma(Q_{\overline{\phi}}(s', \pi_{\overline{\theta}}(a'|s')) - \alpha \log(\pi_{\overline{\theta}}(a'|s')))$. Here, D is the replay buffer of past experiences, and ϕ, $\overline{\phi}$, and $\overline{\theta}$ are the parameters of the critic, target critic, and target actor, respectively. Notably, SAC adopts a soft action-value function by incorporating an entropy term. The policy is learned by the gradient ascent optimizer:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{s\sim D, a\sim\pi}[\nabla_\theta \log(\pi_\theta(a|s))(\alpha \log(\pi_\theta(a|s)) - Q_\phi(s,a))]. \tag{2}$$

MASAC is an extension of SAC to multi-agent learning that adopts the CTDE paradigm to learn stochastic policies in both discrete and continuous action spaces. All critics are optimized together to minimize the joint loss function through parameter sharing:

$$\mathcal{L}_Q(\phi) = \sum_{i=1}^{n} \mathbb{E}_{(\tau,a,r,\tau')\sim D}[(Q_i^\phi(\boldsymbol{\tau}, \boldsymbol{a}) - y_i)^2] \tag{3}$$

where $y_i = r_i + \gamma(Q_i^{\overline{\phi}}(\boldsymbol{\tau}', a'_1, \ldots, a'_n|_{\pi_{\overline{\theta}_i}(a'_i|\tau'_i)}) - \alpha \log(\pi_{\overline{\theta}_i}(a'_i|\tau'_i)))$. Here, Q_i^ϕ is the centralized action-value function for agent i. The individual policies are updated by gradient ascent with the following loss:

$$\nabla_{\theta_i} J(\pi_\theta) = \mathbb{E}_{\tau\sim D, a\sim\pi}[\nabla_{\theta_i} \log(\pi_{\theta_i}(a_i|\tau_i))(\alpha \log(\pi_{\theta_i}(a_i|\tau_i)) - Q_i^\phi(\boldsymbol{\tau}, \boldsymbol{a}))] \tag{4}$$

where all actions are sampled from all agents' current policies to prevent overgeneralization [24]. However, MASAC cannot scale easily to the large joint action spaces inherent in multi-agent settings, as the joint action space grows exponentially with the number of agents.

4. Problem Formulation

In this section, we first establish a UAV kinematic model and then formulate the UAV-swarm-based MTT task as a Dec-POMDP, as illustrated in Figure 1. UAV swarms are deployed in a task area to search and track the target cooperatively. Each UAV needs to keep a minimum safe distance from others to avoid collision. Additionally, each UAV can be modeled as an agent whose observation and communication ranges are limited. Finally, SER is introduced to help increase the search and tracking coverage.

4.1. UAV Kinematic Model

In an MTT task, the UAV always flies at a fixed altitude [31]. Without a loss of generality, we adopt a two-dimensional kinematic model as follows:

$$\begin{cases} x_i^{t+1} = x_i^t + v_i \cos\varphi_i^t \Delta t, & 0 \le x_i^t \le x_{\max} \\ y_i^{t+1} = y_i^t + v_i \sin\varphi_i^t \Delta t, & 0 \le y_i^t \le y_{\max} \\ \varphi_i^{t+1} = \varphi_i^t + a_i^t \Delta t, & -a_{\max} \le a_i^t \le a_{\max} \end{cases} \tag{5}$$

where (x_i^t, y_i^t), v_i, and φ_i^t are the position, velocity, and heading angle of UAV i at time step t, respectively. Here, v_i is a constant for simplicity, and a_i^t is the heading angular rate whose maximum value is $a_{\max}$. Therefore, each UAV can be defined by its position and heading (x_i, y_i, φ_i).

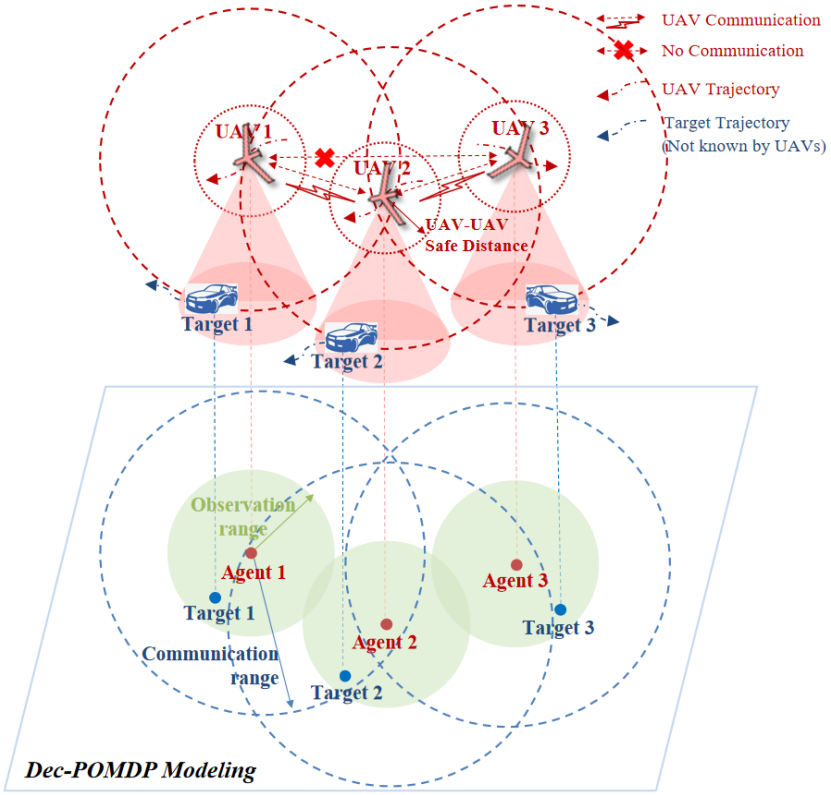


Figure 1. Example of cooperative MTT for UAV swarm under the settings of Dec-POMDP. Red dots are UAV agents and blue dots target agents.

4.2. Dec-POMDP Modeling

The Dec-POMDP model includes the design of the observation space, action space, and reward function, which are vital for agents to learn to cooperate.

4.2.1. Observation Space

Considering the limits of the observation and communication range of UAV in practice, we suppose that the UAV's maximum observation range is d_o, and the maximum communication range is d_c, as depicted in Figure 2. If the distance between UAV i and j is less than d_c, the UAVs can communicate and share information with each other to promote cooperation. The communication information received by UAV i can be denoted as $c_{i,j} = (x_j, y_j, v_j^x, v_j^y, a_j^{t-1})$, where (v_j^x, v_j^y) is the velocity of UAV j in direction (x, y), and a_j^{t-1} is the UAV's last action. Similarly, if the distance between UAV i and target k is less than d_o, the UAV can perceive the target and acquire the target's position and velocity information $o_{i,k} = (x_k^T, y_k^T, v_k^{Tx}, v_k^{Ty})$. Therefore, the observation space of the UAV is composed of its own position, heading, local observation, and communication information, which is denoted as $o_i = (x_i, y_i, \varphi_i, o_{i,k}, c_{i,j})$.

4.2.2. Action Space

The action space of the UAV consists of the heading angle, according to Section 4.1. To promote learning efficiency, the heading is discretized into N_a discrete values [14], which can be denoted as

$$a = \frac{2n_a - N_a - 1}{N_a - 1}\dot{\varphi}_{\max}, n_a \in [1, N_a] \tag{6}$$

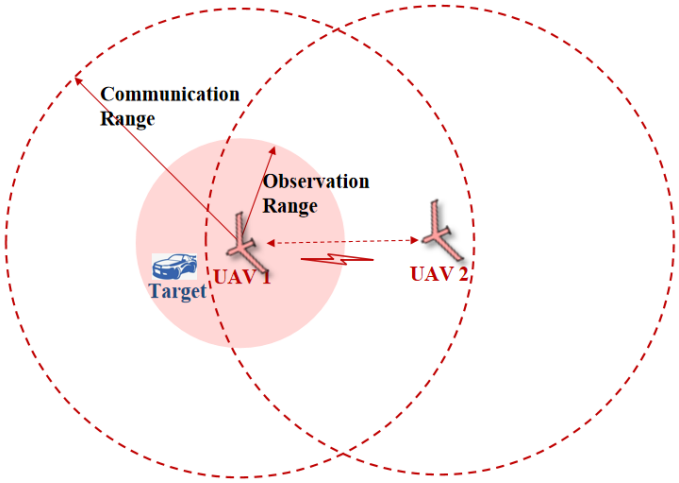


Figure 2. The observation and communication range of each UAV.

4.2.3. Reward Function

In RL settings, the reward function is used to guide agent learning. Therefore, the design of the reward function is particularly critical and can greatly affect the learning speed and final performance of the agent.

In an MTT task, each UAV needs learn to search and track the target, while keeping the target in its observation range as much as possible. In addition, the UAVs should avoid collisions or flying out of the scenario boarder. Therefore, the reward consists of three parts: the tracking reward r_i^{tracking}, the collision penalty $r_i^{\text{collision}}$, and the boundary penalty r_i^{bound}.

The tracking reward can be calculated by

$$r_i^{\text{tracking}} = \max_k (r_{i,k}^{\text{tracking}}) \tag{7}$$

$$r_{i,k}^{\text{tracking}} = \begin{cases} (d_o - d_{i,k})/d_o + 1, & d_{i,k} \le d_o \\ 0, & d_{i,k} > d_o \end{cases} \tag{8}$$

where $d_{i,k}$ is the distance between UAV i and target k. When UAV i detects multiple targets, it only concentrates on tracking the nearest target and obtains the reward $\max_k (r_{i,k}^{\text{tracking}})$.

The collision penalty can be calculated by

$$r_i^{\text{collision}} = \min_j (r_{i,j}^{\text{collision}}) \tag{9}$$

$$r_{i,j}^{\text{collision}} = \begin{cases} (d_{i,j} - d_s)/d_s - 0.5, & d_{i,j} \le d_s \\ 0, & d_{i,j} > d_s \end{cases} \tag{10}$$

where $d_{i,j}$ is the distance between UAV i and j, and d_s is the safe distance needed between UAVs to avoid collision. If $d_{i,j} \le d_s$, the UAV will receive a negative collision penalty to improve safety.

The boundary penalty can be calculated by

$$r_i^{\text{bound}} = \begin{cases} 1 - \mathrm{e}^{(d_{i,O} - d^{\text{bound}}/2)}, & d_{i,O} > d^{\text{bound}}/2 \\ 0, & d_{i,O} \le d^{\text{bound}}/2 \end{cases} \tag{11}$$

where $d_{i,O}$ is the distance between UAV i and scenario center O, and d^{bound} is the scenario boundary length. If UAV i flies out of the boundaries, that UAV will be punished for ineffective searching and tracking.

Because the UAV swarm is cooperative, each UAV shares a global team reward, which can be represented by $r = \sum_{i=1}^{n} (r_i^{\text{tracking}} + r_i^{\text{collision}} + r_i^{\text{bound}})$.

4.3. Spatial Entropy Reward

Due to insufficient exploration, there are always some targets left without UAVs to track for the MTT task, which means becoming trapped in the local optima. To address this issue, one straightforward idea is to encourage the UAVs to keep a certain distance between each other to increase the detection coverage of the UAV swarm. Inspired by the spatial entropy concept [28], SER is proposed to enlarge the detection coverage of the UAV swarm as much as possible to increase the target detection probability.

First, we provide the definition of spatial entropy. Considering a two-dimensional polar coordinate space (ρ, φ), the origin is the position of a UAV. Thus, each position is defined by polar coordinates, where ρ is the distance from the pole, and φ is the relative angle. Then, the spatial density function is

$$p(\rho, \varphi) = Ke^{-\lambda\rho} \tag{12}$$

where λ is a parameter related to the mean distance to the pole, and K is a normalizing constant that ensures the following:

$$\int_0^{2\pi} \int_0^{\overline{R}} p(\rho, \varphi)\rho d\rho d\varphi = 1 \tag{13}$$

where $\overline{R}$ is the radius. Thus, according to Equations (12) and (13), we obtain

$$K = \frac{\lambda}{2\pi(\frac{1}{\lambda}(1 - e^{-\lambda\overline{R}}) - e^{-\lambda\overline{R}}\overline{R})} \tag{14}$$

Then, similar to information entropy, spatial entropy $\mathcal{H}_{SE}$ is defined as

$$\begin{aligned} \mathcal{H}_{SE} &= -\int_0^{2\pi} \int_0^{\overline{R}} p(\rho, \varphi) \ln p(\rho, \varphi)\rho d\rho d\varphi \\ &= -2\pi K \int_0^{\overline{R}} (\ln K - \lambda\rho)\rho e^{-\lambda\rho} d\rho \\ &= -2\pi K \ln K \int_0^{\overline{R}} \rho e^{-\lambda\rho} d\rho + 2\pi K\lambda \int_0^{\overline{R}} \rho^2 e^{-\lambda\rho} d\rho \\ &= -\ln\lambda + \ln 2\pi + \ln(\tfrac{1}{\lambda}(1 - e^{-\lambda\overline{R}}) - e^{-\lambda\overline{R}}\overline{R}) + \frac{\frac{2}{\lambda} - \lambda e^{-\lambda\overline{R}}(\overline{R}^2 + \frac{2\overline{R}}{\lambda^2} + \frac{2}{\lambda})}{\frac{1}{\lambda}(1 - e^{-\lambda\overline{R}}) - e^{-\lambda\overline{R}}\overline{R}} \end{aligned} \tag{15}$$

The evaluation clearly shows that $\mathcal{H}_{SE}$ increases monotonously with $\overline{R}$. Therefore, we can control the size of $\overline{R}$ by adjusting the entropy:

$$d_{SE} = \mathcal{H}_{SE}^{-1}(\mathcal{H}_1) \tag{16}$$

where $\mathcal{H}_1$ is the expected minimum spatial entropy, and d_{SE} is the equivalent spatial entropy distance.

Based on the definitions of $\mathcal{H}_{SE}$ and d_{SE}, we propose a concept called SER r^{SE}, which can be calculated as follows:

$$r^{SE} = \frac{\overline{d}_n - d_s}{d_{SE} - d_s} \tag{17}$$

where $\overline{d}_n = \frac{2}{n(n-1)} \sum_{i \neq j}^{n} d_{i,j}$ is the average distance of the UAV swarm within the communication range. If $\overline{d}_n > d_{SE}$, then $r^{SE} > 1$. Therefore, r^{SE} encourages UAVs to keep as far away as possible to increase detection coverage during the search process.

Thus, the final reward function is shaped as

$$r = \sum_{i=1}^{n} (r_i^{\text{tracking}} + r_i^{\text{collision}} + r_i^{\text{boundary}}) + \kappa r^{SE} \tag{18}$$

where $\kappa = \max(1 - n_e/1000, 0)$, and n_e is the current episode number. κ linearly anneals from one to zero after 1000 episodes to reduce the influence of r^{SE} for tracking performance after a target is searched.

Finally, the objective function of this MTT optimization problem is

$$\max_{\pi_\theta} \mathbb{E}[\sum_{t=0}^{T} r_t + \alpha \mathcal{H}(\pi_\theta)] \tag{19}$$

where $\mathcal{H}(\pi_\theta)$ is the entropy of the policy. Next, we will solve this problem in the next section by using a new MARL algorithm.

5. Method

In Section 5, we propose a new algorithm called FMASAC, which incorporates the idea of value decomposition into MASAC. To address the issues of high variance and credit assignment in the cooperative MAAC framework, the centralized critic is factored to assign credit for each agent to improve its policy and thus increase the rewards. Then, factored soft policy iteration is presented to realize off-policy learning for the multi-agent stochastic policy gradient. Finally, the procedure of the FMASAC algorithm is given.

5.1. Learning a Centralized but Factored Critic

Learning a centralized critic conditioning the joint observation and action can be impractical and difficult due to the growth in the number of agents. Therefore, here, we employ the value decomposition in the MASAC framework to enable the scalable learning of a centralized critic in Dec-POMDP. We factor the centralized critic as a linear weighted combination of individual critics across agents:

$$Q_{tot}^{\phi,\omega}(\boldsymbol{\tau}, \boldsymbol{a}, s) = g_\omega(s, \{Q_i^{\phi_i}(\tau_i, a_i)\}_{i=1}^{n}) \tag{20}$$

where ϕ and ϕ_i are the parameters of the centralized critic $Q_{tot}^{\phi,\omega}$ and the agent-wise utilities $Q_i^{\phi_i}$, respectively. Here, g_ω is a linear monotonic function parametrized as a mixing network without the activation function, and s is the global state of the environment. Among these, the linear weighted function increases the expressivity of the network, which combines individual agent utilities into a centralized critic or joint action-value function. Additionally, monotonicity ensures the individual global maximum (IGM) shown in Equation (21), i.e., consistency between the argmax of the joint action-value function and the argmax of the individual action-value function [37]:

$$\operatorname*{argmax}_{\boldsymbol{a}} Q_{tot} = (\operatorname*{argmax}_{a_1} Q_1, \ldots, \operatorname*{argmax}_{a_n} Q_n) \tag{21}$$

To evaluate the policy, the critic network is optimized by minimizing the following TD loss:

$$\mathcal{L}_Q(\phi, \omega) = \mathbb{E}_D[(Q_{tot}^{\phi,\omega}(\boldsymbol{\tau}, \boldsymbol{a}, s) - y_{tot})^2] \tag{22}$$

$$y_{tot} = r + \gamma(Q_{tot}^{\overline{\phi},\overline{\omega}}(\boldsymbol{\tau}', a'_1, \ldots, a'_n|_{\pi_{\overline{\theta}_i}(a'_i|\tau'_i)}, s') - \alpha \log(\pi_{\overline{\boldsymbol{\theta}}}(\boldsymbol{a}'|\boldsymbol{\tau}'))) \tag{23}$$

$$\pi_{\overline{\boldsymbol{\theta}}}(\boldsymbol{a}'|\boldsymbol{\tau}') = \prod_{i=1}^{n} \pi_{\overline{\theta}_i}(a'_i|\tau'_i) \tag{24}$$

where Equation (23) is the target critic. $\overline{\phi}$, $\overline{\boldsymbol{\theta}}$, and $\overline{\omega}$ are the parameters of the target critic, actors, and mixing network, respectively. Among these, $\overline{\boldsymbol{\theta}}$ represents the joint target actors, which are similar to those in [37].

5.2. Factored Soft Policy Iteration

Then, we leverage the factored critic to train actors in an end-to-end manner to realize decentralized execution. Here, similar to Equation (2), we propose a soft policy iteration with factored maximum entropy trained as follows:

$$\begin{aligned} \mathcal{L}(\pi_\theta) &= \mathbb{E}_{\pi_\theta}[\alpha \log(\pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{\tau})) - Q_{tot}^{\phi,\omega}(\boldsymbol{\tau}, \boldsymbol{a}, s)] \\ &= -g_\omega(s, \mathbb{E}_{\pi_\theta}[Q_i^{\phi_i}(\tau_i, a_i) - \alpha \log(\pi_{\theta_i}(a_i|\tau_i))]) \end{aligned} \tag{25}$$

where g_ω is a one-layer monotonic mixing network the same as that in Equation (20). The bias is generated from input s. Then, we derive the bias in detail. We adopt the aristocrat utility to assign credits [38], providing the following:

$$\begin{aligned}
&g_\omega(s, \mathbb{E}_{\pi_\theta}[Q_i^{\phi_i}(\tau_i, a_i) - \alpha_i \log(\pi_{\theta_i}(a_i|\tau_i))]) \\
&\quad = \sum_i [\omega_i(s)\mathbb{E}_{\pi_\theta}[Q_i^{\phi_i}(\tau_i, a_i)] + b(s) - \sum_i [\omega_i(s)\mathbb{E}_{\pi_\theta}[\alpha_i \log(\pi_{\theta_i}(a_i|\tau_i))] \\
&\quad\quad (\sum_i [\omega_i(s)\mathbb{E}_{\pi_\theta}[Q_i^{\phi_i}(\tau_i, a_i)] + b(s) = \mathbb{E}_{\pi_\theta}[Q_{tot}^{\phi,\omega}(\boldsymbol{\tau}, \boldsymbol{a}, s)]) \\
&\quad = \mathbb{E}_{\pi_\theta}[Q_{tot}^{\phi,\omega}(\boldsymbol{\tau}, \boldsymbol{a}, s)] - \sum_i \mathbb{E}_{\pi_\theta}[\omega_i(s)\alpha_i \log(\pi_{\theta_i}(a_i|\tau_i))] \\
&\quad\quad (\text{let } \boldsymbol{\alpha} = \omega_i(s)\alpha_i) \\
&\quad = \mathbb{E}_{\pi_\theta}[Q_{tot}^{\phi,\omega}(\boldsymbol{\tau}, \boldsymbol{a}, s)] - \sum_i \mathbb{E}_{\pi_\theta}[\boldsymbol{\alpha} \log(\pi_{\theta_i}(a_i|\tau_i))] \\
&\quad\quad (\text{let } \pi_{\boldsymbol{\theta}} = \prod \pi_{\theta_i}, \text{then } \sum_i \mathbb{E}_{\pi_\theta}\boldsymbol{\alpha} \log(\pi_{\theta_i}(a_i|\tau_i)) = \mathbb{E}_{\pi_\theta}\boldsymbol{\alpha} \log(\pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{\tau}))) \\
&\quad = \mathbb{E}_{\pi_\theta}[Q_{tot}^{\phi,\omega}(\boldsymbol{\tau}, \boldsymbol{a}, s)] - \mathbb{E}_{\pi_\theta}[\boldsymbol{\alpha} \log(\pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{\tau}))] \\
&\quad = -\mathbb{E}_{\pi_\theta}[\boldsymbol{\alpha} \log(\pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{\tau})) - Q_{tot}^{\phi,\omega}(\boldsymbol{\tau}, \boldsymbol{a}, s)]
\end{aligned}$$

which then complies with the original SAC policy update.

Similar to the individual temperature parameter automatic adjustment in [36], we apply an automatically adjusted global temperature parameter by considering this adjustment as a constrained optimization problem. We optimize the temperature with the following loss:

$$J(\alpha) = \mathbb{E}_{a_t \sim \pi_t}[-\alpha_i \log(\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{\tau}_t)) - \boldsymbol{\alpha}\mathcal{H}_0] \tag{26}$$

where $\mathcal{H}_0$ is a minimum entropy constraint to guarantee $-\log(\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{\tau}_t) > \mathcal{H}_0$, which ensures a minimum randomness of the policy. Additionally, α will decrease if the policy satisfies the entropy constraint; otherwise, α will be increased to enforce the policy to explore. Therefore, the entropy constraint encourages exploration and ensures that the policy convergence is optimal.

5.3. FMASAC-Based MTT of the UAV Swarm

Thus, we propose a cooperative multi-target tracking method with the UAV swarm based on the FMASAC algorithm. The overall algorithm architecture is depicted in Figure 3.

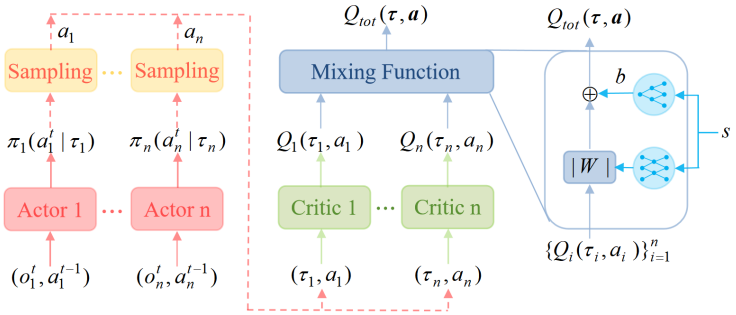


Figure 3. The overall FMASAC architecture. Bold represents joint vector.

For each agent, there is an actor network that selects action a_i^t conditioning based on its local observations $o_i^t = (x_i^t, y_i^t, \varphi_i^t, o_{i,k}^t, c_{i,j}^t)$ and last time action a_i^{t-1}. There is also one critic network that estimates the Q value of its observation-action (τ_i, a_i), which are combined into the joint action-value function $Q_{tot}^{\phi,\omega}$ via a linear monotonic mixing function. $Q_{tot}^{\phi,\omega}$ is optimized via TD learning, according to Equation (22). Then, each optimized $Q_i^{\phi_i}$ is used to help update each actor network via the factored soft policy iteration, according to Equation (25).

The detailed algorithm procedure is summarized as Algorithm 1, which can be divided into three phases: the initialize phase, the experience collection phase, and the network update

phase. First, all parameters are initialized for learning. Then, in the experience collection phase, all UAV agents interact with the environment according to their policies to collect and store experiences. Actor and critic networks are optimized in the network update phase to enable each UAV to learn to cooperate and track the target. After training, each UAV can search and track the target autonomously based on its learned policy, and the UAV swarm can implement cooperative multi-target tracking behavior in an unknown environment.

Algorithm 1: Factored multi-agent soft actor-critic (FMASAC)

Initialize phase
1: Initialize critic networks $Q_i^{\phi_i}$, actor networks π_{θ_i}, and mixing network g_ω with random parameters ϕ, θ_i, ω
2: Initialize target networks: $\overline{\phi} = \phi$, $\overline{\theta}_i = \theta_i$, $\overline{w} = w$
3: Initialize a replay buffer D
4: **for** episode = 1 to *max_train_episodes* N_e **do**
5: Reset environment
Experience collection phase
6: **for** t = 0 to *max_episode_steps* N_t **do**
7: For each agent i, take action $a_i \sim \pi_i(\cdot|\tau_i)$
8: Execute joint action $\boldsymbol{a}$
9: Observe observation $\boldsymbol{\tau}'$, and reward r, done, info
10: Store $(\boldsymbol{\tau}, \boldsymbol{a}, r, \boldsymbol{\tau}')$ in replay buffer D
11: **end for**
Actor and critic network update phase
12: **for** t = 1 to T **do**
13: Sample a minibatch trajectory from D
14: Calculate $\mathcal{L}_Q(\phi, \omega)$
15: Update critic networks and mixing network w.r.t. Equation (22)
16: Update decentralized policies using the gradients $\nabla_{\theta_i} J(\pi_\theta)$ w.r.t. Equation (25)
17: Update temperature parameter α w.r.t. Equation (26)
18: **if** update target network then
19: $\overline{\phi} = \phi$, $\overline{\theta}_i = \theta_i$, $\overline{w} = w$
20: **end if**
21: **end for**
22:**end for**
23:Return π

6. Experiments

In this section, we evaluate the proposed FMASAC algorithm using simulation experiments and compare it with three popular multi-agent actor-critic algorithms. The effectiveness, generalization, and scalability of the FMASAC algorithm are then validated. The simulation results show that the proposed algorithm can realize cooperative multi-target tracking while avoiding collision; the results outperform the baselines in terms of the mean reward and tracking success rate.

6.1. Parameter Setup

For this study, we developed a two-dimensional UAV swarm multi-target tracking simulation environment, which is developed based on commonly used tools, including Linux (Ubuntu 20.04) or MacOS, Pycharm, and Multi-Agent Particle Environment [22]. The user interface can be customized and modified. The number of UAVs and targets and the map size are variable and can be modified according to demand. The positions of the UAVs and targets are initialized randomly for each episode. The environmental parameters are configured as shown in Table 3.

Table 3. Environmental parameters.

Entity	Physical Meaning	Notation	Value
Environment	Size		2000 m × 2000 m
	Scenario boundary length	d^{bound}	2000 m
UAV	Observation range	d_o	200 m
	Communication range	d_c	800 m
	Safe distance	d_s	400 m
	Speed	v_i	60 m/s
	Maximum heading angular rate	$a_{\max}$	$\pi/6$ rad/s
Target	Speed	v_k^T	40 m/s

Next, we test the performance of the proposed FMASAC algorithm in the scenario with three UAVs and three targets and compare it with three baselines: the MASAC [39], multi-agent proximal policy optimization (MAPPO) [40], and multi-agent deep deterministic policy gradient (MADDPG) [22] algorithms. The algorithm hyperparameter settings are shown in Table 4, and the hyperparameters of the baselines are fine-tuned according to the information in [41]. For the sake of fairness, each algorithm is run with five random seeds, and the results are shown with 95% confidence intervals.

Table 4. Hyperparameter settings.

Hyperparameter	Value
Episode limit	100
Max step	1,005,000
Buffer size	5000
Minibatch size	64
Target update interval	200
Actor learning rate	0.0001
Critic learning rate	0.0005
TD lambda	0.6
Gamma	0.99
Minimum policy entropy	−1
Equivalent spatial entropy distance	500

6.2. Effectiveness

First, each algorithm was trained for 1,005,000 steps until reaching convergence. To intuitively understand whether the UAVs learned how to track the targets, the tracking process of each UAV in the FMASAC algorithm is visualized (Figure 4). Figure 4 shows that each UAV learned to autonomously and cooperatively search and track a target.

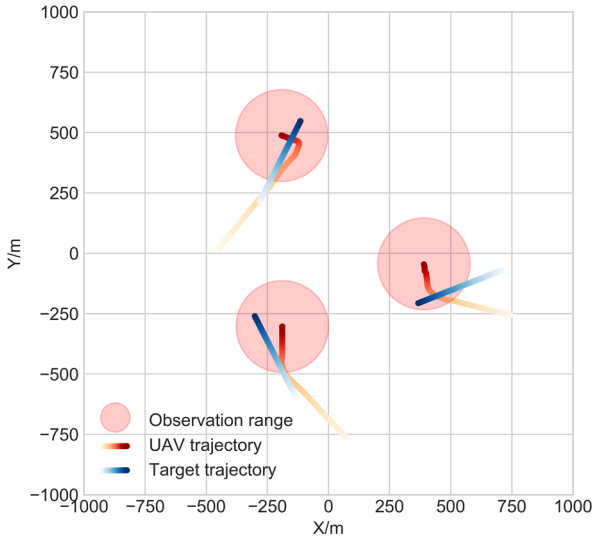


Figure 4. Visualization of the MTT task. X is the horizontal coordinate and Y is the vertical coordinate.

Next, we plotted the mean reward and relative distances between the UAVs and the UAV target and the tracking success rate during training to measure the performance of the four algorithms. Successful tracking is determined by more than 70 tracking steps in an episode. The tracking success rate is defined as the ratio of the number of successful tracking times to the total tracking times every 80 episodes. Figure 5 shows the mean reward learning curve of the four algorithms.

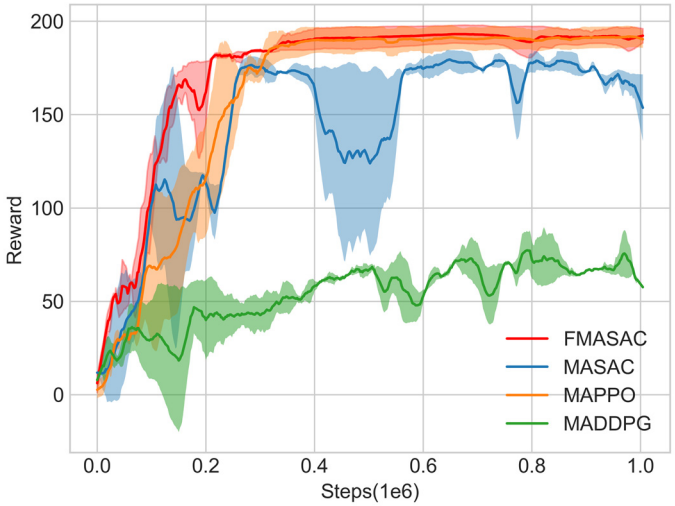


Figure 5. Mean reward for different algorithms.

Figure 5 shows that the performance of FMASAC is better than that of the other three baseline algorithms in terms of the final mean reward and convergence speed, which means that the factored critic and soft policy iterations are efficient. Compared to the vanilla MASAC algorithm, our FMASAC algorithm behaves more stably and has a lower policy variance, leading to improved performance, while the MASAC suffers from policy collapse. It can be inferred that the MASAC satisfies the conflicts in the policy update, as greedily maximizing the team reward may exacerbate the credit assignment issue, and the exploration or sub-optimality of some agents' policies might aggravate the training of other agents through the centralized critic. Therefore, learning the factored critic implicitly realizes multi-agent credit assignment and efficient learning because the individual critic assigns credit information for each agent to improve its policy.

To further demonstrate the effectiveness and contribution of each core component in FMASAC, we conducted ablation studies as shown in Figure 6 to compare FMASAC, FMASAC using a fixed alpha named FMASAC_Fixed_α, and FMASAC without SER, named FMASAC_NoSER.

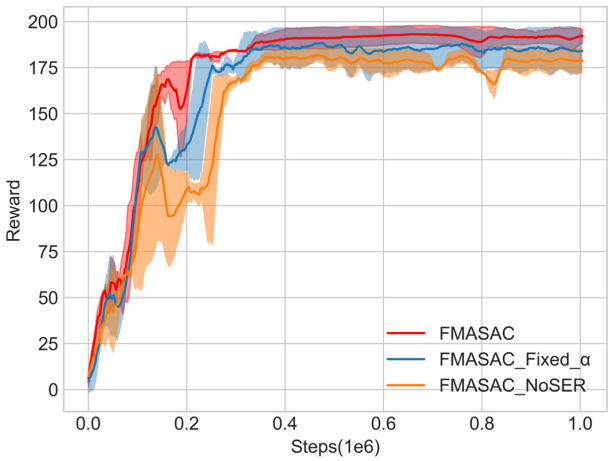
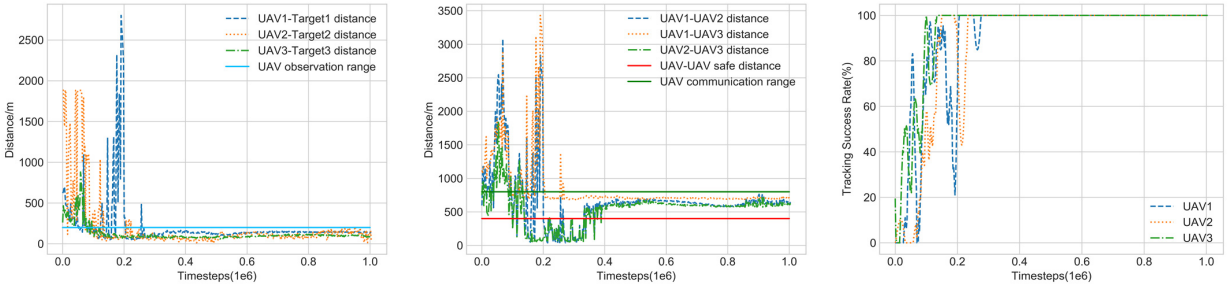


Figure 6. Ablation results comparing FMASAC and its ablated version.

It can be seen from Figure 6 that the performance of the full version FMASAC is better than that of FMASAC_Fixed_α and the no r^{SE} version FMASAC_NoSER. This result illustrates that automatic temperature adjustment parameter α can improve policy performance by balancing exploration and exploitation. Notably, SER significantly increases the stability and final performance of the policy by encouraging the expansion of the detection coverage, which objectively verifies the importance and effectiveness of SER.

Next, we analyze the relative distances between the UAVs and the UAV-target and tracking success rates during training for this tracking task. The results are shown in Figure 7.



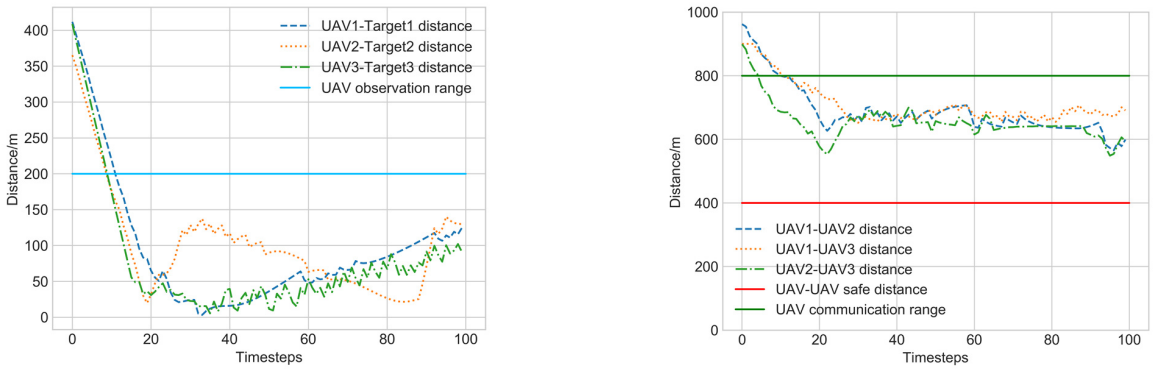
(**a**) (**b**) (**c**)

Figure 7. An example of target tracking trajectory for UAVs: (**a**) The UAV-target relative distances; (**b**) the relative distances between UAVs; (**c**) tracking success rate.

As illustrated in Figure 7a, the relative distances between each UAV and the target are less than the UAV observation range after 0.3 million training steps, which means that each UAV can search and track a target successfully. Figure 7b shows that the relative distances between UAVs are larger than the UAV–UAV safe distance after training. Additionally, the UAVs learn to avoid collisions while tracking, and the distances between UAVs are smaller than the UAV communication range to better communicate mutual information. The most significant indicator in the MTT task, the tracking success rate, is depicted in Figure 7c. Here, the tracking success rate of each UAV during training reaches 1 after about 0.3 million steps. Therefore, three UAVs learned to autonomously search and track targets while avoiding collisions, which demonstrates that our proposed FMASAC algorithm is effective in the MTT task.

6.3. Generalization

Next, we load the trained policy model (actor network) without further fine-tuning to test the generalization of the model in unknown and dynamic changing environments. Here, all positions of the UAVs and targets are randomly initialized. The test results of one episode are shown in Figure 8.



(**a**) (**b**)

Figure 8. Test for the target tracking trajectory of UAVs: (**a**) the relative distances between UAVs and targets; (**b**) the relative distances between UAVs.

As can be seen from Figure 8a, each UAV can quickly track a target after 10 steps in an unknown environment because the UAV policy model has seen hundreds and thousands of scenarios during training and has memorized the neural network parameters. Meanwhile, as shown in Figure 8b, the relative distances between UAVs are larger than the safe distance, while maintaining effective communication. In conclusion, the learned policy model can directly generalize to an unknown environment without fine-tuning or re-training, which offers efficient and real-time results.

Figure 9 shows the relative distances and test tracking success rates of 100 tests. We can see from the violin plot of relative distances that the UAV–target distances are almost less than the UAV observation range. This result indicates that every target is tracked by the trained UAV model, even in an unknown scenario. Moreover, the tracking success rate for each UAV during testing is greater than 98%, which indicates that the trained policy can be generalized to unknown scenarios.

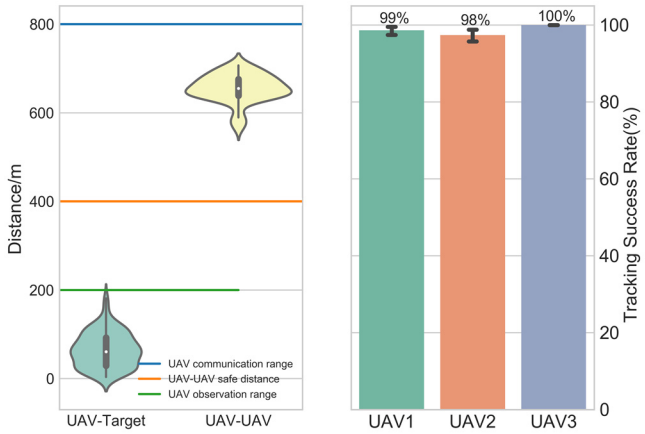


Figure 9. Generalization test.

6.4. Scalability

To further test the scalability of the proposed FMASAC algorithm in large-scale UAV swarm scenarios, we also performed experiments on five UAVs and five targets, as well as ten UAVs and ten targets. Figure 10 shows the change trend of the average tracked targets and tracking success rates of different algorithms as the number of UAVs increases. The detailed results of the different algorithms are shown in Table 5.

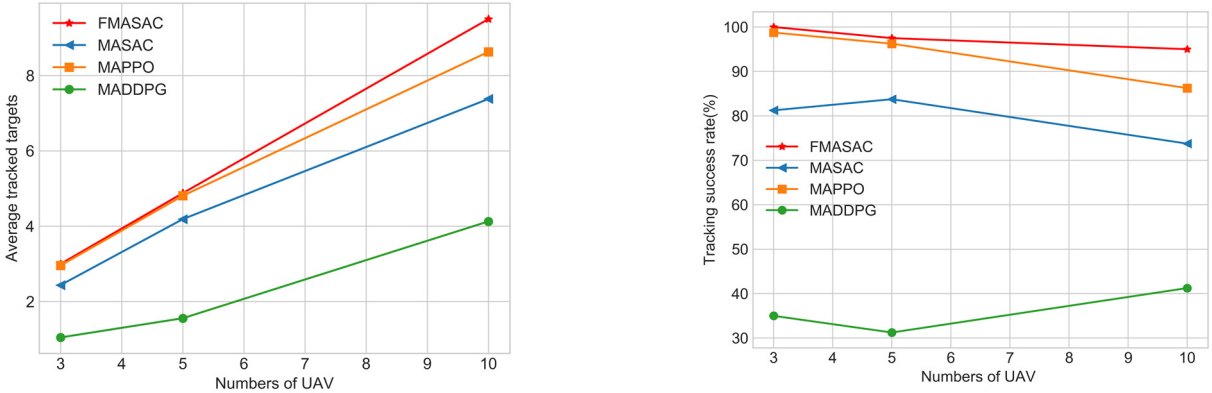


Figure 10. Performance comparison in different scenarios: (**a**) the average tracked targets; (**b**) the tracking success rate.

Table 5. Comparison of results in different scenarios.

Scenarios	Map Size (km)	Indicators	FMASAC	MASAC	MAPPO	MADDPG
3 UAVs and 3 targets	2 × 2	Mean reward	**172.63**	144.54	160.51	54.28
		Reward standard deviation	**1.11**	18.84	1.26	8.07
		Average tracked targets	**3**	2.44	2.96	1.05
		Tracking success rate	**100%**	81.25%	98.75%	35%
5 UAVs and 5 targets	5 × 5	Mean reward	**315.23**	289.72	308.88	124.36
		Reward standard deviation	**1.32**	16.95	1.45	9.87
		Average tracked targets	**4.88**	4.19	4.81	1.56
		Tracking success rate	**97.5%**	83.75%	96.25%	31.25%
10 UAVs and 10 targets	10 × 10	Mean reward	**611.97**	457.42	543.83	277.19
		Reward standard deviation	1.64	23.63	**1.58**	13.42
		Average tracked targets	**9.5**	7.38	8.63	4.13
		Tracking success rate	**95%**	73.75%	86.25%	41.25%

It can be seen from Table 5 that the proposed FMASAC algorithm has the highest mean reward and the lowest standard deviation in almost three scenarios. Additionally, the average tracked targets and tracking success rates during training were better than those of the other baselines. Therefore, the performance of FMASAC is better than that of the other baselines in the three scenarios. This result verified that the factored critic and soft policy iteration reduced the complexity of the joint action-value function and enabled the scalable learning of CTDE paradigms by combining individual utilities conditioned on local information. Therefore, we can conclude that the FMASAC algorithm scales better when the number of agents and joint action spaces increase.

7. Conclusions

This paper studied the problem of multi-target tracking with a UAV swarm in an unknown environment. An end-to-end MARL scheme named FMASAC was proposed to enable UAV swarm cooperative multi-target tracking in a distributed manner. By learning a centralized-but-factored critic and factored soft policy iteration, the algorithm learning efficiency was improved, and the variance in policy updates was reduced. SER increased the detection coverage of the UAV swarm and improved cooperative performance in the task. The simulation results show that the proposed FMASAC algorithm can realize cooperative multi-target tracking and significantly outperform the three popular MARL baselines. Additionally, FMASAC can generalize and scale to unknown and large-scale scenarios.

In the future, we will continue to study the cooperative multi-target tracking method for heterogeneous nonlinear swarms with individual rewards in unknown environments [42,43]. In addition, more complicated real-world scenarios and accurate UAV/HFV models will be considered [44].

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

Funding: This research was funded by the National Natural Science Foundation of China, grant number 62106284, the Natural Science Foundation of Shaanxi Province, China, grant number 2021JQ-370, and the Young Talent Fund of Association for Science and Technology in Shaanxi, China, grant number 20220101.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Zhou, L.Y.; Leng, S.P.; Liu, Q.; Wang, Q. Intelligent UAV Swarm Cooperation for Multiple Targets Tracking. *IEEE Internet Things J.* **2022**, *9*, 743–754. [CrossRef]
2. Chen, Y.; Dong, Q.; Shang, X.Z.; Wu, Z.Y.; Wang, J.Y. Multi-UAV autonomous path planning in reconnaissance missions considering incomplete information: A reinforcement learning method. *Drones* **2022**, *7*, 10. [CrossRef]
3. Shi, W.; Li, J.; Wu, H.; Zhou, C.; Chen, N.; Shen, X. Drone-cell trajectory planning and resource allocation for highly mobile networks: A hierarchical DRL approach. *IEEE Internet Things J.* **2020**, *99*, 9800–9813. [CrossRef]
4. Serna, J.G.; Vanegas, F.; Brar, S.; Sandino, J.; Flannery, D.; Gonzalez, F. UAV4PE: An open-source framework to plan UAV autonomous missions for planetary exploration. *Drones* **2022**, *6*, 391. [CrossRef]
5. Kumar, M.; Mondal, S. Recent developments on target tracking problems: A review. *Ocean Eng.* **2021**, *236*, 109558. [CrossRef]
6. Vo, B.N.; Mallick, M.; Bar-Shalom, Y.; Coraluppi, S.; Osborne, R., III; Mahler, R.; Vo, B.T. Multitarget Tracking. In *Wiley Encyclopedia of Electrical and Electronics Engineering*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2015; pp. 1–25. [CrossRef]
7. Pitre, R.R.; Li, X.R.; Delbalzo, R. UAV route planning for joint search and track missions-an information-value approach. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *48*, 2551–2565. [CrossRef]
8. Jilkov, V.P.; Li, X.R. On fusion of multiple objectives for UAV search and track path optimization. *J. Adv. Inf. Fusion* **2009**, *4*, 27–39.
9. Botts, C.H.; Spall, J.C.; Newman, A.J. Multi-Agent Surveillance and Tracking Using Cyclic Stochastic Gradient. In Proceedings of the 2016 American Control Conference (ACC), Boston, MA, USA, 6–8 July 2016; pp. 270–275.
10. Khan, A.; Rinner, B.; Cavallaro, A. Cooperative robots to observe moving targets: Review. *IEEE Trans. Cybern.* **2018**, *48*, 187–198. [CrossRef]
11. Li, B.; Yang, Z.P.; Chen, D.Q.; Liang, S.Y.; Ma, H. Maneuvering target tracking of UAV based on MN-DDPG and transfer learning. *Def. Technol.* **2021**, *17*, 457–466. [CrossRef]
12. Wang, T.; Qin, R.X.; Chen, Y.; Hichem, S.; Chang, C. A reinforcement learning approach for UAV target searching and tracking. *Multimed. Tools Appl.* **2019**, *78*, 4347–4364. [CrossRef]
13. Rosello, P.; Kochenderfer, M.J. Multi-agent reinforcement learning for multi-object tracking. In Proceedings of the 17th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), Richland, SC, USA, 9–11 July 2018; pp. 1397–1404.
14. Zhou, W.H.; Li, J.; Liu, Z.H.; Shen, L.C. Improving multi-target cooperative tracking guidance for UAV swarms using multi-agent reinforcement learning. *Chin. J. Aeronaut.* **2022**, *35*, 100–112. [CrossRef]
15. Kraemer, L.; Banerjee, B. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing* **2016**, *190*, 82–94. [CrossRef]
16. Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W.M.; Zambaldi, V.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J.Z.; Tuyls, K.; et al. Value-Decomposition Networks for Cooperative Multi-Agent Learning Based on Team Reward. In Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Richland, SC, USA, 9–11 July 2018; pp. 2085–2087.
17. Rashid, T.; Samvelyan, M.; Schroeder, C.; Farquhar, G.; Foerster, J.; Whiteson, S. Qmix: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In Proceedings of the International Conference on Machine Learning (ICML), Stockholm, Sweden, 10–15 July 2018; pp. 1–14.
18. Son, K.; Kim, D.; Kang, W.J.; Hostallero, D.E.; Yi, Y. Qtran: Learning to Factorize with Transformation for Cooperative Multi-Agent Reinforcement Learning. In Proceedings of the International Conference on Machine Learning (ICML), Long Beach, CA, USA, 10–15 June 2019; pp. 1–18.
19. Rashid, T.; Farquhar, G.; Peng, B.; Whiteson, S. Weighted QMIX: Expanding Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Virtual, 6–12 December 2020; pp. 1–20.
20. Yang, Y.; Hao, J.; Liao, B.; Shao, K.; Chen, G.; Liu, W.; Tang, H. Qatten: A general framework for cooperative multiagent reinforcement learning. *arXiv* **2020**, arXiv:2002.03939.
21. Wang, J.H.; Ren, Z.Z.; Liu, T.; Yu, Y.; Zhang, C.J. Qplex: Duplex Dueling Multi-Agent Q-Learning. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual, 3–7 May 2021; pp. 1–27.
22. Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; Mordatch, I. Multi-Agent Actor-Critic for Mixed Cooperative Competitive Environments. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 24–28 January 2018; pp. 1–12.
23. Foerster, J.; Farquhar, G.; Afouras, T.; Nardelli, N.; Whiteson, S. Counterfactual Multi-Agent Policy Gradients. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), New Orleans, LA, USA, 2–7 February 2018; pp. 1–10.
24. Wei, E.; Wicke, D.; Freelan, D.; Luke, S. Multiagent Soft Q-Learning. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), New Orleans, LA, USA, 2–7 February 2018; pp. 1–7.
25. Iqbal, S.; Sha, F. Actor-Attention-Critic for Multi-Agent Reinforcement Learning. In Proceedings of the International Conference on Machine Learning (ICML), Long Beach, CA, USA, 10–15 June 2019; pp. 1–14.
26. Wang, Y.H.; Han, B.N.; Wang, T.H.; Dong, H.; Zhang, C.J. DOP: Off-Policy Multi-Agent Decomposed Policy Gradients. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual, 26 April–1 May 2020; pp. 1–20.

27. Tumer, K.; Agogino, A.K.; Wolpert, D.H. Learning Sequences of Actions in Collectives of Autonomous Agents. In Proceedings of the the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1, Bologna Italy, 15–19 July 2002; pp. 378–385. [CrossRef]
28. Batty, M. Spatial entropy. *Geogr. Anal.* **1974**, *6*, 1–31. [CrossRef]
29. Agogino, A.K.; Tumer, K. Unifying Temporal and Structural Credit Assignment Problems. In Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), New York, NY, USA, 19–23 July 2004; pp. 980–987.
30. Cheriguene, Y.; Bousbaa, F.Z.; Kerrache, C.A.; Djellikh, S.; Lagraa, N.; Lahby, M.; Lakas, A. COCOMA: A resource-optimized cooperative UAVs communication protocol for surveillance and monitoring applications. *Wirel. Netw.* **2022**. [CrossRef]
31. Zhou, W.H.; Li, J.; Zhang, Q.J. Joint communication and action learning in multi-target tracking of UAV swarms with deep reinforcement learning. *Drones* **2022**, *6*, 339. [CrossRef]
32. Mishra, D.; Trotta, A.; Traversi, E.; Felice, M.D.; Natalizio, E. Cooperative cellular UAV-to-Everything (C-U2X) communication based on 5G sidelink for UAV swarms. *Comput. Commun.* **2022**, *192*, 173–184. [CrossRef]
33. Gao, N.; Liang, L.; Cai, D.H.; Li, X.; Jin, S. Coverage control for UAV swarm communication networks: A distributed learning approach. *IEEE Internet Things J.* **2022**, *9*, 19854–19867. [CrossRef]
34. Dibangoye, J.S.; Amato, C.; Buffet, O.; Charpillet, F. Optimally solving dec-POMDPs as continuous-state MDPs. *J. Artif. Intell. Res.* **2016**, *55*, 443–497. [CrossRef]
35. Oliehoek, F.A.; Amato, C. *A Concise Introduction to Decentralized POMDPs*; Springer: Berlin/Heidelberg, Germany, 2016; Volume 1.
36. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In Proceedings of the 35th international conference on machine learning (ICML), Stockholm, Sweden, 10–15 July 2018; pp. 1861–1870.
37. Zhang, T.H.; Li, Y.H.; Wang, C.; Xie, G.M.; Lu, Z.Q. Fop: Factorizing Optimal Joint Policy of Maximum-Entropy Multi-Agent Reinforcement Learning. In Proceedings of the International Conference on Machine Learning (ICML), Virtual, 18–24 July 2021; pp. 12491–12500.
38. Wolpert, D.H.; Tumer, K. Optimal payoff functions for members of collectives. *Adv. Complex Syst.* **2002**, *4*, 355–369.
39. Xia, Z.Y.; Du, J.; Wang, J.J.; Jiang, C.X.; Ren, Y.; Li, G.; Han, Z. Multi-Agent Reinforcement Learning Aided Intelligent UAV Swarm for Target Tracking. *IEEE Trans. Veh. Technol.* **2021**, *71*, 931–945. [CrossRef]
40. Yu, C.; Velu, A.; Vinitsky, E.; Wang, Y.; Bayen, A.; Wu, Y. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. In Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS 2022), New Orleans, LA, USA, 28 November–9 December 2022; pp. 1–30.
41. Yu, C.; Velu, A.; Vinitsky, E.; Wang, Y.; Bayen, A.; Wu, Y. Benchmarking Multi-Agent Deep Reinforcement Learning Algorithms. In Proceedings of the Workshop in Conference on Neural Information Processing Systems (NeurIPS), Virtual, 6–12 December 2020; pp. 1–22.
42. Lv, M.; Yu, W.; Cao, J.; Baldi, S. A separation-based methodology to consensus tracking of switched high-order nonlinear multi-agent systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 5467–5479. [CrossRef] [PubMed]
43. Lv, M.; Schutter, B.D.; Cao, J.; Baldi, S. Adaptive prescribed performance asymptotic tracking for high-order odd-rational-power nonlinear systems. *IEEE Trans. Autom. Control* **2023**, *68*, 1047–1053. [CrossRef]
44. Lv, M.; Schutter, B.D.; Baldi, S. Non-recursive control for formation-containment of HFV swarms with dynamic event-triggered communication. *IEEE Trans. Ind. Inform.* 2022, *early access*. [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.