Chinese Journal of Aeronautics, (2021), xxx(xx): xxx–xxx

Chinese Society of Aeronautics and Astronautics
& Beihang University

Chinese Journal of Aeronautics

cja@buaa.edu.cn
www.sciencedirect.com

Improving multi-target cooperative tracking guidance for UAV swarms using multi-agent reinforcement learning

Wenhong ZHOU, Jie LI *, Zhihong LIU, Lincheng SHEN

College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China

Received 24 March 2021; revised 27 September 2021; accepted 27 September 2021

KEYWORDS

Decentralized cooperation;
Maximum reciprocal reward;
Multi-agent actor-critic;
Pointwise mutual information;
Reinforcement learning

Abstract Multi-Target Tracking Guidance (MTTG) in unknown environments has great potential values in applications for Unmanned Aerial Vehicle (UAV) swarms. Although Multi-Agent Deep Reinforcement Learning (MADRL) is a promising technique for learning cooperation, most of the existing methods cannot scale well to decentralized UAV swarms due to their computational complexity or global information requirement. This paper proposes a decentralized MADRL method using the maximum reciprocal reward to learn cooperative tracking policies for UAV swarms. This method reshapes each UAV's reward with a regularization term that is defined as the dot product of the reward vector of all neighbor UAVs and the corresponding dependency vector between the UAV and the neighbors. And the dependence between UAVs can be directly captured by the Pointwise Mutual Information (PMI) neural network without complicated aggregation statistics. Then, the experience sharing Reciprocal Reward Multi-Agent Actor-Critic (MAAC-R) algorithm is proposed to learn the cooperative sharing policy for all homogeneous UAVs. Experiments demonstrate that the proposed algorithm can improve the UAVs' cooperation more effectively than the baseline algorithms, and can stimulate a rich form of cooperative tracking behaviors of UAV swarms. Besides, the learned policy can better scale to other scenarios with more UAVs and targets.

1. Introduction

With the development of Unmanned Aerial Vehicle (UAV) technology, the cooperation of UAV swarms has become a research hotspot. Through close cooperation, UAV swarms can show superior coordination, intelligence, and autonomy than traditional multi-UAV systems. At the same time, Multi-Target Tracking Guidance (MTTG) in unknown environments has also become an important application direction

* Corresponding author.
E-mail address: lijie09@nudt.edu.cn (J. LI).
Peer review under responsibility of Editorial Committee of CJA.

Production and hosting by Elsevier

https://doi.org/10.1016/j.cja.2021.09.008
1000-9361 © 2021 Production and hosting by Elsevier Ltd. on behalf of Chinese Society of Aeronautics and Astronautics.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Please cite this article in press as: ZHOU W et al. Improving multi-target cooperative tracking guidance for UAV swarms using multi-agent reinforcement learning, *Chin J Aeronaut* (2021), https://doi.org/10.1016/j.cja.2021.09.008

for UAV swarms. In our MTTG problem, numerous small fixed-wing UAVs are deployed in the mission area to cooperatively track the perceived targets and search the unknown targets. But the cooperation for UAV swarms is not easy. Due to the curse of dimensionality, the computational complexity of the joint decision for all the UAVs increases exponentially with the increase in the UAVs' number. And the UAVs are partially observable, they can only communicate with the neighboring UAVs and perceive the targets locally, but cannot directly obtain the global information of the entire environment for global cooperation. Therefore, centralized cooperation that requires global information and coordinates all UAVs' actions at a central node is not feasible, and how to achieve decentralized cooperation of large-scale UAV swarms is still a huge challenge.

Due to the similarity between UAV swarms and biological flocking, several cooperation methods based on the natural flocking phenomenon were proposed, such as bionic imitation methods,[1,2] consensus-based methods[3,4] and graphy-theory-based methods,[5,6] etc. However, most methods simplify the complexity of the problem, such as assuming that the complex environment model is known or can be established, the UAVs can communicate globally, or the UAVs follow the designed rules, etc., which cannot meet the intrinsic characteristics of the MTTG Problem for UAV swarms. Therefore, these methods still have great difficulties in practicality.

The breakthrough progress of Multi-Agent Deep Reinforcement Learning (MADRL) technology in the game field[7,8] verified that MADRL technology can empower agents the ability to learn coordinating behaviors[9]. In MADRL, agents learn how to behave to maximize their rewards through repeated interactions with the environment and other agents, and learn potential coordination relationships between agents, including cooperation, competition, etc. Benefits from

these, many MTTG methods based on MADRL have been proposed, such as partially observable Monte-Carlo (MC) planning,[10] simultaneous target assignment and path planning,[11] and other methods.[12,13] However, it is challenging to extend these methods directly to the UAV swarms, because the cooperation or coordination process in most of the existing methods is centralized or requires access to global information,[14] which is incompatible with the distributed characteristics of the swarm systems.

In the evolution of cooperation theory, reciprocal altruism is an important mechanism.[15] This mechanism describes that when an agent interacts with other agents, its action can not only make itself rewarded, but also enable other agents to obtain a certain benefit. Inspired by this, we assume that the interactions between cooperative UAVs are also reciprocal, which means that each UAV should not only consider maximizing its reward when making an action decision, but also consider the impact of the action on other UAVs, and avoid adversely affecting their rewards. In this way, the cooperation between UAVs can be improved and the system performance can be more effective.

In this paper, we propose a decentralized MADRL method using the maximum reciprocal reward to learn cooperative tracking policies for small fixed-wing UAV swarms. Firstly, based on the MTTG problem description, we formalize this problem into a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) setting. Then, inspired by the reciprocity mechanism, we define the reciprocal rewards of UAVs and proposed the calculation method of reciprocal rewards. On this basis, because of the homogeneity of UAV swarms, the experience sharing Reciprocal Reward Multi-Agent Actor-Critic (MAAC-R) algorithm based on the experience sharing training mechanism is proposed to learn a shared cooperative policy for homogeneous UAVs. Experimental results demonstrate that the proposed maximum reciprocal reward method can improve the cooperation between UAVs more effectively than the baseline algorithms, and excite the UAV swarms to emerge various cooperative behaviors. Besides, the learned policy can well scale to other cooperative scenarios with more UAVs and targets.

The major contributions of this paper are as follows:

(1) The decentralized maximum reciprocal reward method based on the reciprocity mechanism is proposed to enhance the cooperation of large-scale UAV swarms. The reciprocal reward is defined as the dot product of the reward vector of all neighbor UAVs and the dependency vector between the UAV and its neighbors. Then the reciprocal reward is used to regularize the UAV's private reward, and the reshaped reward is used to learn the UAV's cooperative policy using MADRL algorithms.
(2) Pointwise Mutual Information (PMI) is used to capture the immediate dependence between UAVs, where PMI can be estimated by a neural network and calculated directly without complex sampling and statistics.
(3) Based on the experience replay sharing mechanism, the MAAC-R algorithm combining the maximum reciprocal reward method and PMI estimation is proposed to learn the cooperative policy for all homogeneous UAVs.

The rest of this paper is organized as follows. Section 2 summarizes related work about the MTTG problem and cooperative learning in MADRL. Section 3 gives a brief introduction about Dec-POMDP and PMI. Section 4 describes the MTTG mission models and Section 5 details the proposed method and MAAC-R algorithm. The experimental results and discussion are presented in Section 6. Finally, Section 7 concludes this paper.

For the sake of simplicity, only those uncommon symbols are summarized in Table 1, and the definition of the default symbols in MADRL is omitted.

2. Related work

2.1. Multi-UAV tracking multi-target

Swarming is an emerging direction of multi-UAV research. There are relatively few works about UAV swarms tracking multi-target, so we can look at it from a wider perspective, multi-UAV tracking multi-target, which has been extensively studied. Therefore, here we summarize the related literature on the multi-UAV MTTG problem. This problem can be subdivided into different subtopics,[16] such as cooperative perception, simultaneous cover and track, and multi-robot pursuit-evasion, etc. Here, we summarize the general methods from two perspectives of traditional optimization and Reinforcement Learning (RL).

Please cite this article in press as: ZHOU W et al. Improving multi-target cooperative tracking guidance for UAV swarms using multi-agent reinforcement learning, *Chin J Aeronaut* (2021), https://doi.org/10.1016/j.cja.2021.09.008

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.
$(x_{\mathrm{U}}^{(i)}, y_{\mathrm{U}}^{(i)}), v_{\mathrm{U}}^{(i)}, \theta_{\mathrm{U}}^{(i)}$	The position, velocity, and heading of UAV i.
$(x_{\mathrm{T}}^{(k)}, y_{\mathrm{T}}^{(k)}), v_{\mathrm{T}}^{(k)}, \theta_{\mathrm{T}}^{(k)}$	The position, velocity, and heading of target k.
$\dot{\theta}_{\max}, a^{(i)}$	The maximum heading angular rate and the action of UAV i.
$N_{\mathrm{a}}, n_{\mathrm{a}}$	The UAV's cardinality of the discrete action space and the corresponding index of its discrete action.
$d_{\mathrm{c}}, d_{\mathrm{p}}$	The maximum communication distance and maximum perception distance.
$d^{(i,j)}, d^{(i,k)}$	The distance between UAV i and UAV j, the distance between the ground projection point of UAV i and target k.
$c^{i,j}, o^{(i,k)}$	UAV i's communication information from neighbor j and perception information about target k.
$r_{\mathrm{e}}^{(i)}, r_{\mathrm{c}}^{(i)}, r_{\mathrm{tar}}^{(i)}, r_{\mathrm{rt}}^{(i)}, r_{\mathrm{bound}}^{(i)}$	UAV i's environment reward, reciprocal reward, target-tracking reward, redundant tracking punishment, and boundary punishment.
$d^{(ij)}$	The dependence index between UAV i and j.
$\mathcal{N}^{(\mathrm{i})}$	The set of UAV i's neighbors.
$b^{(i)}$	The boundary information of UAV i.
$s^{(i)}$	UAV i's state information.
$l^{(i)} = (c^{(i)}, o^{(i)}, b^{(i)}, s^{(i)})$	The set of local information.
X_1, X_2 and x_1, x_2	The random proxy variables and the corresponding random proxy events.

A lot of methods based on traditional optimization have been published. Jilkov and Li[17] adopted Fisher Information Matrix to design a Multi-Objective Optimization (MOO) framework to formulate the MTTG problem and presented a MOO algorithm to solve this problem. Pitre et al.[18] also formulated this mission similarly, but they solved it using a modified particle swarm optimization algorithm. A Kalman filter was adopted in literature[19] to track targets' traces and a Consensus-Based Bundle Algorithm (CBBA) was proposed to dynamically switch each UAV's task between searching and tracking. Peterson[20] grouped the UAVs when their decision spaces intersect, he also designed a reward function using information-based measures, then a receding horizon control algorithm was designed to plan the best routes for all UAVs. Botts et al.[21] formulated a cyclic stochastic optimization algorithm for the stochastic multi-agent and multi-target surveillance mission, it was demonstrated that each agent can search a region for targets and track all discovered targets. Beyond those methods, more methods based on traditional optimization theories were detailed in the review.[22]

With the development of RL in recent years, some studies also tried to use RL methods to solve this problem. Assuming that each UAV has access to the global state and all UAVs' joint action, Wang et al.[13] adopted a centralized RL method for UAV fleets to learn the optimal routes to maximize the perceived probability of the targets. Rosello and Kochenderfer[12] formulated the MTTG problem as a motion planning problem at the motion primitive level and used an RL method to learn each UAV's macro action that determines to create, propagate, or terminate a target-tracking task. Qie et al.[11] implemented the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm to solve the multi-UAV simultaneous target assignment and path planning problem. However, it is assumed that each UAV can only track a single target and each target can only be tracked by a single UAV, which is not flexible and efficient for UAV swarms.

In our problem, both the UAVs and targets are homogeneous. UAVs use perception to track targets but may not be able to recognize and distinguish the identity of each target. Therefore, there is no clear allocation between UAVs and targets. Each UAV can track any perceived target, and a target can also be tracked by one or more UAVs. This creates a more flexible and extensive form of cooperation between UAVs. However, the curse of dimensionality caused by the increase in the scale of UAVs prevents most existing MADRL methods from being well-scaled to more complex scenarios.

2.2. Learning cooperation in MADRL

How to cooperate better is an important issue in MADRL. To learn the cooperation policies of agents, many methods have been proposed, including communication learning, value function factorization, centralized critic, reward modification, etc.

Communication learning. Assume that there are explicit communication channels between agents, and agents can receive communication information from others as part of their input variables. On the other hand, when making decisions, agents can not only output their actions but also output communication information sent to other agents. Reinforced Inter-Agent Learning and Differentiable Inter-Agent Learning algorithms were proposed[23] to learn discrete communication and continuous communication information between agents, respectively, to achieve cooperation. To ensure the effectiveness and scalability of the communication method, BiCNet[24] was introduced to learn a bidirectionally-coordinated network between agents. CommNet[25] was proposed to learn a common communication model for all agents in a centralized manner.

Value-function factorization. For an environment where all agents share a common reward, methods such as QMIX,[26] QTRAN,[27] and VDN[28] were proposed to decompose the joint Q function of all agents into the sum of individual Q functions to capture the complex cooperative relationship between agents.

Centralized critic: As the name suggests, a centralized critic network is proposed to evaluate the agent's policy, where the input of the network is the relevant information of all agents (such as joint observation and joint action). COMA[29] compared the actual reward and counterfactual reward to evaluate the impact of an agent's action on the global reward. And MADDPG[30] introduced a critic network that can receive all agents' information to guide the training of the actor-network.

Reward regularization. This refers to quantifying the cooperation between agents and using it as an additional reward. Based on the following assumption: the actions between cooperative agents are highly correlated, Kim et al.[31] and Cuervo

and Alzate[32] reshaped global reward with Mutual Information (MI) that captures the dependence between agents' policies, then agents can learn cooperative policies by maximizing the reshaped reward. Wang et al.[33] proposed EITI that used MI to capture the interactions between the transition functions of the agents. Then they also proposed EDTI to capture the influence of one agent's behavior on the expected rewards of other agents. It has been proved that optimization by using EITI or EDTI as a regularization term can encourage agents to learn cooperative policies. Except for MI, Kullback-Leibler (KL) divergence can also be used for regularized rewards. Jaques et al.[34] used the intrinsic social influence to reward agents for causal influence over another agent's actions. There is an assumption that the influencer takes an action, then the influencee takes the action as part of its input variable and uses counterfactual reasoning to evaluate the causal influence of the action. Thus, the influence process between agents is nonsymmetric. Moreover, counterfactual reasoning needs to know the influencee's policy and input variables. The MI and KL divergence used in the above regularization methods are aggregate statistics, so their calculation requires multiple sampling, and they cannot evaluate the dependence between two specific random events. Barton et al.[35–37] implemented a technique called Convergent Cross Mapping[38] (CCM) to measure the caused influence between agents' specific events (such as actions or states). However, how to choose the embedding dimension and delay time in the phase space reconstruction process of high-dimensional input variables is still a huge challenge.

In addition, some of the latest cooperative policy learning methods using MADRL were proposed for UAV swarms.

Baldazo et al.[39] presented a MADRL method to learn the cooperative policies for fixed-wing UAV swarms to monitor floods in a decentralized fashion. But it assumes that each UAV can concatenate all UAVs' information, this is incompatible with the partial observability of the UAV swarms, and greatly limits the scalability of the proposed method. Wang et al.[40] developed a MADRL algorithm to learn a sharing policy for a UAV swarm, in which each UAV only directly cooperates with the nearest two neighbors on its left side and right side. However, the protocol that only captures the fixed two collaborating neighbors is too sloppy, because cooperation emergence depends on topological[41]. Khan et al.[42] presented a MADRL method by employing a Graph Neural Network (GNN) to learn cooperative formation flying policies for a UAV swarm. Although the GNN can extract local information of individuals, how to perform the graph convolutional operation for UAVs whose exact number is unknown and maybe changing is still a challenge. Venturini et al.[43] proposed a distributed RL approach that scales to larger swarms without modifications in monitoring and remote area surveillance applications. However, they assume that the system environment consists of square grids and each UAV can directly share the observation with others.

The research goals of these works are similar to that of this paper, but the scales of the UAVs are not large (only a few dozen). Besides, these methods make over-idealized assumptions that ignore some inherent characteristics of UAV swarms, such as global communication, fully observable, etc. Hence, we believe that these methods may have practical issues for large-scale UAV swarms. This paper considers the large-scale, partial observability, distributed decision-making, homogeneity, interchangeability, and other characteristics of the MTTG problem for UAV swarms, which makes the model more versatile and feasible. Moreover, we propose a maximum reciprocal reward method to further improve the cooperation between UAVs. Compared with the existing work, our task is more challenging, in which the UAV swarms can handle different maps and group sizes, different numbers of the targets, etc.

3. Preliminary

3.1. Dec-POMDP setting

The decision process of UAV swarms can be defined using Dec-POMDP[44]. In Dec-POMDP, each agent can only get local observation information (including perception and communication), but cannot obtain the global state. At every step t, each agent makes its action decision based on its local observation information, and all agents execute their joint action to refresh the environment. Except for special definition, the time subscript t of variables is omitted for convenience, and the joint variables over all agents are bold.

Assuming there are n agents, the Dec-POMDP is defined as:

$$(N, S, \boldsymbol{A}, T, \boldsymbol{R}, \boldsymbol{O}, Z, \gamma),$$

where N is the set of n agents; S is the state space, and state $s \in S$; $\boldsymbol{A}: \{A^{(1)}, A^{(2)}, \cdots A^{(n)}\}$ denotes all agents' joint action space, $a^{(i)} \in A^{(i)}$; the probability $P(s'|s,a) \to [0,1]$ denotes the transition probability model from state s to next state $s' \in S$ after executing the joint action $a: \{a^{(1)}, a^{(2)}, \cdots, a^{(n)}\}$; $\boldsymbol{R}(s,a): \{r^{(1)}, r^{(2)}, \cdots, r^{(n)}\}$ is the joint reward function by executing the joint action a given state s; $\boldsymbol{O}: \{O^{(1)}, O^{(2)}, \cdots O^{(n)}\}$ denotes all agents' joint observation space, $o: \{o^{(1)}, o^{(2)}, \cdots, o^{(n)}\}$ and $o^{(i)} \in O^{(i)}$; $Z: o^{(i)} = Z(s,i)$ denotes the individual observation model of each agent given state s; $\gamma \in [0,1]$ is a constant discount factor.

In Dec-POMDP, the reward function describes the coordinating relationship between agents. For example, there are two agents, if $r^{(1)} = r^{(2)}$, they are fully cooperative; $r^{(1)} = -r^{(2)}$, they are fully competitive; otherwise, they are mixed cooperative-competitive. Each agent makes its action decision based on its local observation information, the policy of the agent i is $\pi^{(i)}: O^{(i)} \to A^{(i)}$, and the joint policy of all agents is denoted as $\pi: \{\pi^{(1)}, \pi^{(2)}, \cdots, \pi^{(n)}\}$. Given the joint observation o and the joint policy π, $V_{\pi}^{(i)}(o) = E_{\pi}[\sum_{t=0}^{\infty} \gamma^t r_t^{(i)} | o_{t=0} = o]$ denotes the state-value function of agent i, and $Q_{\pi}^{(i)}(o,a) = E_{\pi}[\sum_{t=0}^{\infty} \gamma^t r_t^{(i)} | (o,a)_{t=0} = (o,a)]$ is the action-value function executing the joint action a.

3.2. Mutual information and pointwise mutual information

Let the discrete random variable pair be (X, Y), their joint distribution is $P(X, Y)$, the marginal distributions are $P(X)$ and $P(Y)$ respectively, and $P(X)P(Y)$ is the product of the marginal distributions, then the MI of (X, Y) is denoted by:

$$I(X;Y) = D_{KL}(P(X,Y), P(X)P(Y)) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)},$$

where D_{KL} is the KL divergence, and log is a logarithm to the base of the constant e. The greater the dependence between X

and Y, the greater $I(X;Y)$. If X and Y are independent, $I(X;Y)=0$.

For specific random event pair (x,y), PMI is used to measure their dependence:

$$PMI(x;y)=\log\frac{p(x,y)}{p(x)p(y)}=\log\frac{p(x|y)}{p(x)}=\log\frac{p(y|x)}{p(y)}.$$

Both MI and PMI are symmetrical, and $I(X;Y)$ is the weighted sum of PMI$(x;y)$ for all possible (x,y). To avoid $\log 0=-\infty$, define Positive PMI[45] (PPMI) as:

$$\mathrm{PPMI}(x,y)=\max\left(\log\frac{P(x,y)}{P(x)P(y)},0\right).$$

4. Problem formalization

4.1. Problem description

Suppose a large number of homogeneous small fixed-wing UAVs and homogeneous targets are scattered in the mission scenario. The UAVs are flying at a constant speed in a two-dimensional plane, their actions are the heading angular rates. The UAVs can perform emergency maneuvers, such as temporarily staggering the flying height to avoid collisions. The targets are randomly walk in the environment, and the UAVs cannot get the prior information of the targets in advance.

As shown in Fig. 1, each UAV uses onboard sensors that look down to perceive targets. When a target is within the perception range, the UAV can perceive and track the target, but the UAV cannot distinguish the target's specific identity. Meanwhile, the UAV can receive the local communication information from its neighboring UAVs. And it can also obtain local information relative to the boundary of the mission area. Then the UAV makes its action decision based on the local information.

Due to the limited perception range of the UAVs and the uncontrollable movement of the targets, the UAVs may lose the tracked targets. And since there is no explicit target assignment, a single UAV may track multiple aggregated targets at the same time, or multiple UAVs may cooperate to track the single or multiple targets. Therefore, the UAVs should keep the targets within their field of view persistently, and cooperate in a decentralized manner to track as many targets as possible. Moreover, the UAVs should also be able to avoid collisions and fly off the boundaries, and satisfy safe constraints.

4.2. Models

Based on the overall description of the problem above, we establish the relevant models based on the Dec-POMDP paradigm as follows:

Kinematic model. There are n UAVs and m targets in the mission scenario, and they are indexed by $i\in[1,n]$ and $k\in[1,m]$, respectively. Each UAV flies with a constant linear speed v_U, and the air-frame orientation coincides with its heading θ_{U}. The control variable a is the bounded heading angular rate $\dot{\theta}_{\mathrm{U}}$. Then the kinematic model of each UAV is defined by its position and heading $[x_{\mathrm{U}}^{(i)},y_{\mathrm{U}}^{(i)},\theta_{\mathrm{U}}^{(i)}]$ as follows:

$$\begin{cases} x_{\mathrm{U},t+1}^{(i)}=x_{\mathrm{U},t}^{(i)}+\Delta t v_{\mathrm{U}}^{(i)}\cos\theta_{\mathrm{U},t}^{(i)}, & 0\leqslant x_{\mathrm{U},t}^{(i)}\leqslant x_{\max} \\ y_{\mathrm{U},t+1}^{(i)}=y_{\mathrm{U},t}^{(i)}+\Delta t v_{\mathrm{U}}^{(i)}\sin\theta_{\mathrm{U},t}^{(i)}, & 0\leqslant y_{\mathrm{U},t}^{(i)}\leqslant y_{\max} \\ \theta_{\mathrm{U},t+1}^{(i)}=\theta_{\mathrm{U},t}^{(i)}+\Delta t a_t^{(i)}, & -\dot{\theta}_{\max}\leqslant a_t^{(i)}\leqslant\dot{\theta}_{\max} \end{cases} \quad (1)$$

where Δt is the simulation time step, $a_t^{(i)}$ is the action of UAV i, and $\dot{\theta}_{\max}$ is the UAV's maximum heading angular rate. And all the state variables are bounded. Similarly, for a target $k,k\in[1,m]$, its kinematic model is also defined by its location and heading, $[x_{\mathrm{T}}^{(k)},y_{\mathrm{T}}^{(k)},\theta_{\mathrm{T}}^{(k)}]$, but its heading angular velocity is a bounded random variable.

Action space. Ideally, the action of each UAV should be a continuous random variable. However, we find that learning in continuous action space is much more difficult than learning in discrete space. In this paper, we focus on learning the cooperative policies of UAV swarms, rather than the precise motion of a single UAV. To reduce the learning difficulty, we discretize each UAV's action space into a finite number of symmetrical action primitives. Suppose that the cardinality of the discrete action set is N_{a}, then the action space is discretized as:

$$a=\frac{2n_{\mathrm{a}}-N_{\mathrm{a}}-1}{N_{\mathrm{a}}-1}\dot{\theta}_{\max},n_{\mathrm{a}}\in[1,N_{\mathrm{a}}]. \quad (2)$$

Local communication and observation models. The local communication and observation models of each UAV are illustrated in Fig. 2, where the maximum communication dis-

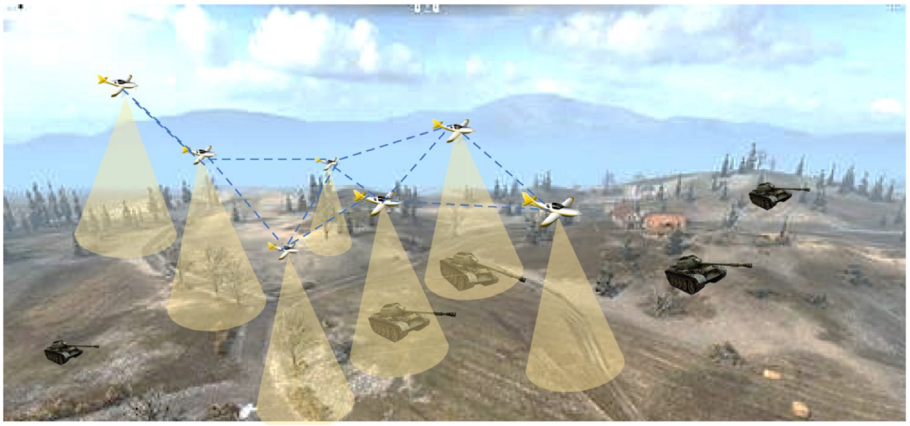


Fig. 1 The scenario of UAV swarms tracking multi-target.

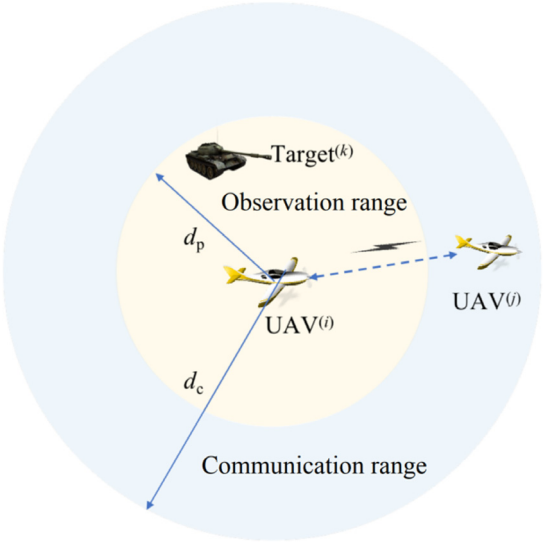


Fig. 2 The communication and observation diagrams of each UAV.

tance is d_c and the maximum perception range is d_p . If the relative distance between UAV i and UAV j is less than d_c , UAV j is a neighbor of UAV i , and UAV i can receive the communication information from the neighbor j . The communication information is denoted as $c^{(i,j)} = (x_U^{(j)}, y_U^{(j)}, v_{x_U}^{(j)}, v_{y_U}^{(j)}, a_{-1}^{(j)})$, where $(v_{x_U}^{(j)}, v_{y_U}^{(j)})$ is the linear velocity of UAV j , and $a_{-1}^{(j)}$ is its last action. Similarly, UAV i can only sense the targets in the circular area with a radius d_p directly below it, but cannot recognize their specific identities. Then, UAV i 's perception information of target k is denoted as $o^{(i,k)} = (x_T^{(k)}, y_T^{(k)}, v_{x_T}^{(k)}, v_{y_T}^{(k)})$, and $(v_{x_T}^{(k)}, v_{y_T}^{(k)})$ is the linear velocity of target k . Since each UAV in the swarm is partially observable, the communication and perception information should be converted from the global coordination to the UAV's local coordination for further normalization.

4.3. Reward shaping

To encourage the UAVs to learn cooperative policies, each UAV is supposed to not only track the already perceived targets well but also avoid overlapping perception to maximize the total perception range. Besides, they should avoid flying out of the mission scenario and ensure a safe flight. These expectations are represented by the reward shaping potentially. Thus, it is necessary to design reasonable rewards for UAVs to guide the learning process.

Tracking targets: Both the targets and UAVs are moving dynamically, and the perceived targets may move out of the UAVs' sensing range. A naïve idea is that the closer the UAV is to a target, the better the tracking effect it is. Therefore, UAV i 's reward for tracking a target is as follows:

$$r_{tar}^{(i,k)} = \begin{cases} 1 + (d_p - d^{(i,k)})/d_p, & d^{(i,k)} \leqslant d_p \\ 0, & \text{Else} \end{cases} \tag{3}$$

where $d^{(i,k)}$ is the relative distance between UAV i and target k , then the reward of UAV i for tracking multiple targets is $r_{tar}^{(i)} = \sum_{k=1}^{m} r_{tar}^{(i,k)}$. There is a bias in the reward shaping to encourage to track more targets instead of focusing on a single one. For example, if there are two targets in the sight of UAV i , the reward is greater than 2, while the reward of focusing on only one does not exceed 2.

Punishing duplicate tracking. Duplicate tracking means that a target is tracked by multiple UAVs at the same time, which would cause a waste of resources and increase the risk of collision. And it is not conducive to improving system efficiency. Then, some of them are supposed to fly away to search for unknown targets. When the relative distance between two UAVs is greater than twice the perception radius, duplicate tracking would not occur. Thus, the punishment for repetitive tracking between UAV i and j is as follows:

$$r_{rt}^{(i,j)} = \begin{cases} -0.5\exp((2d_p - d^{(i,j)})/(2d_p)), & d^{(i,j)} \leqslant 2d_p \\ 0, & \text{Else} \end{cases} \tag{4}$$

where $d^{(i,j)}$ is the relative distance between UAV i and j , and $r_{rt}^{(i,j)} = r_{rt}^{(j,i)}$. Then the duplicate tracking punishment is $r_{rt}^{(i)} = \sum_{j=1}^{n} r_{rt}^{(i,j)}$. The punishments of UAVs drive them to not only minimize the overlap of perception areas to reduce the possibility of repetitive tracking but also keep a safe distance between the UAVs.

Boundary punishment. Seriously, flying out of the borders would cause security risks, so each UAV is supposed to fly within the mission area. When the UAV is too close to the boundary, part of its perception range may fall outside the mission area, and the perception of the non-mission area may be useless. The minimal distance to the boundaries is defined as d_{min} , we add a boundary punishment:

$$r_{bound}^{(i)} = \begin{cases} -0.5(d_p - d_{min})/d_p, & d_{min} < d_p \\ 0. & \text{Else} \end{cases} \tag{5}$$

Hereby, the reward of UAV i at each step is shaped as:

$$r^{(i)} = r_{tar}^{(i)} + r_{rt}^{(i)} + r_{bound}^{(i)}. \tag{6}$$

5. Maximum reciprocal reward method

Inspired by the mechanism of reciprocal altruism in the evolution of intelligence theory, if an agent is beneficial to other agents, the agent should be rewarded. The intuition behind this is that if an agent knows the impact of itself on other agents, it can cooperate with other agents to achieve win–win teamwork.

In the MTTG problem, we assume that if a UAV can deliberately lead to cooperation with neighbors so that they can get better rewards, the UAV should be appropriately encouraged; otherwise, when the neighbor gets negative rewards, the UAV should take part of the responsibility, and therefore would be punished. Specifically, we modify each UAV's immediate reward as:

$$r^{(i)} = (1 - \alpha) r_e^{(i)} + \alpha r_c^{(i)}, \tag{7}$$

where $r_e^{(i)}$ is the private reward obtained from the environment, and $r_c^{(i)}$ is the reciprocal reward from other UAVs. The weight coefficient α balances between private reward and reciprocal reward. When $\alpha = 0$, the UAV is fully selfish, which is the naïve setting of RL; while $\alpha = 1$, the UAV is fully selfless. When $0 < \alpha < 1$, maximizing the expected discount com-

pound reward to learn a cooperative policy that not only maximizes its interest but also encourages cooperation with other UAVs.

5.1. Reciprocal reward

For each UAV, its reciprocal reward exists only when it cooperates with other UAVs, at which time there is a dependence between them. And the greater the degree of cooperation, the greater the dependence. Thus, the reciprocal reward is not only related to the others' rewards but also related to their dependence. Then, the reciprocal reward of UAV i is denoted as:

$$r_{\mathrm{c}}^{(i)} = \frac{1}{d^{(i)}} \sum_{j \in \mathcal{N}^{(i)}} d^{(ij)} r_{\mathrm{e}}^{(j)}, \tag{8}$$

where $\mathcal{N}^{(i)}$ denotes the set of UAV i's neighbors and $d^{(ij)}$ measures the dependence between UAV i and j, $d^{(i)} = \sum_{j \in \mathcal{N}^{(i)}} d^{(ij)}$ and $1/d^{(i)}$ is a normalized coefficient.

Generally, the dependence index $d^{(ij)} \geqslant 0$, and if there is no cooperation between UAV i and j, $d^{(ij)} = d^{(ji)} = 0$, otherwise, $d^{(ij)} > 0$. We hope that the dependence between UAVs can encourage them to produce positive and effective cooperation, that is, $d^{(ij)} r_{\mathrm{e}}^{(j)} > 0$. Note that when $d^{(ij)}$ is great, $d^{(ij)} r_{\mathrm{e}}^{(j)}$ may be small or even negative, because highly coupled actions may not have a positive impact, and could even cause unfavorable conflicts. Besides, even if $d^{(ij)} r_{\mathrm{e}}^{(j)}$ is small, $r_{\mathrm{e}}^{(j)}$ may be large (it is bounded usually), because independent agents may also work effectively[46].

5.2. Pointwise mutual information estimation

The environment reward of each UAV is given, then a major obstacle in the calculation of reciprocal reward is the dependence indexes between every two cooperative UAVs. In UAV swarms, each UAV is partially observable and makes the action decision based on its local information, including communication with neighbors, observation about targets, boundary information, and state information, etc. Since all the UAVs coexist in one environment and the global state is unknown, the local information and actions between two neighboring UAVs may not be independent. We use PMI to capture the dependence between UAVs. Then the dependence index between UAV i and j is denoted as:

$$\begin{aligned} d^{(ij)} &= \mathrm{PMI}(l^{(i)}, a^{(i)}; l^{(j)}, a^{(j)}) \\ &= \log \frac{p(l^{(i)}, a^{(i)}, l^{(j)}, a^{(j)})}{p(l^{(i)}, a^{(i)}) p(l^{(j)}, a^{(j)})}, \end{aligned} \tag{9}$$

where l represents the local information of each UAV, $l^{(i)} = (c^{(i)}, o^{(i)}, b^{(i)}, s^{(i)})$, $b^{(i)}$ is UAV i's boundary information, and $s^{(i)}$ is its state information.

Usually, to calculate PMI, it needs to estimate the joint probability distribution $p(l^{(i)}, a^{(i)}, l^{(j)}, a^{(j)})$ and the marginal probability distributions $p(l^{(i)}, a^{(i)})$ and $p(l^{(j)}, a^{(j)})$ firstly. In a simple environment that the local information space and action space of each UAV are small, they can be estimated via MC sampling. For example, $p(l^{(i)}, a^{(i)}, l^{(j)}, a^{(j)})$ can be approximated from the ratio of the visited frequencies:

$$p(l^{(i)}, a^{(i)}, l^{(j)}, a^{(j)}) = \frac{N(l^{(i)}, a^{(i)}, l^{(j)}, a^{(j)})}{N(L^{(i)}, A^{(i)}, L^{(j)}, A^{(j)})}, \tag{10}$$

where $N(\cdot)$ is a counter, $N(l^{(i)}, a^{(i)}, l^{(j)}, a^{(j)})$ is the number of occurrences of $(l^{(i)}, a^{(i)}, l^{(j)}, a^{(j)})$ during the sampling process, and $N(L^{(i)}, A^{(i)}, L^{(j)}, A^{(j)})$ is the sum of the numbers of all possible joint information-action pairs. $p(o^{(i)}, a^{(i)})$ and $p(o^{(j)}, a^{(j)})$ can also be estimated similarly. However, in a complex environment that its local information or action space is large or even continuous, MC sampling may be infeasible since the time and memory consuming.

Fortunately, we can get the optimal PMI estimation by Maximizing Mutual Information (MMI). MI can be expressed as the expectation of the divergence between the joint probability distribution and the product of marginal probability distributions. According to different measurement methods, many works that maximize the variational lower bound of MI are proposed, such as Ref.[47–50]. Among them, I_{JS} estimates the MI using Jensen-Shannon (JS) divergence, which is more stable than the other methods. Inspired by mutual information neural estimation,[49] we can also estimate PMI using a neural network. For brevity and convenience of derivation, we use x_1 and x_2 as the proxies of $(l^{(i)}, a^{(i)})$ and $(l^{(j)}, a^{(j)})$, respectively. Then we can estimate PMI using the following lemma.

Lemma 1. For random variables X_1 and X_2, their JS MI is defined as:

$$\begin{aligned} &I_{JS}(X_1; X_2) = D_{JS}(P(X_1, X_2) || P(X_1)P(X_2)) \\ &= \tfrac{1}{2} KL(P(X_1, X_2) || M) + \tfrac{1}{2} KL(P(X_1)P(X_2) || M), \end{aligned} \tag{11}$$

where $M = \frac{P(X_1, X_2) + P(X_1)P(X_2)}{2}$. Then the variational lower bound of $I_{\mathrm{JS}}(X_1, X_2)$ is:

$$\begin{aligned} I_{\mathrm{JS}}(X_1; X_2) \geqslant \sup_{f_\omega} & E_{P(X_1, X_2)}[-\mathrm{sp}(-f_\omega(x_1, x_2))] \\ & - E_{P(X_1) \times P(X_2)}[\mathrm{sp}(f_\omega(x_1, x_2))], \end{aligned} \tag{12}$$

where $f_\omega(x_1, x_2)$ is a fitting function parameterized with ω, $\mathrm{sp}(x) = \log(1 + \mathrm{e}^x)$. When the variational lower bound is maximum, we can find an optimal fitting function:

$$f_\omega^*(x_1, x_2) = \log \frac{p(x_1, x_2)}{p(x_1)p(x_2)} = \mathrm{PMI}(x_1; x_2). \tag{13}$$

Proof: Define $I_{\mathrm{JS}}(X_1; X_2) = E_{P(X_1, X_2)}[-\mathrm{sp}(-f_\omega(x_1, x_2))] - E_{P(X_1) \times P(X_2)}[\mathrm{sp}(f_\omega(x_1, x_2))]$, according to Ref.[51], its first-order derivative concerning $f_\omega(x_1, x_2)$ is:

$$\begin{aligned} \frac{\partial I_{\mathrm{JS}}}{\partial f_\omega(x_1, x_2)} = & \frac{\mathrm{e}^{-f_\omega(x_1, x_2)}}{1 + \mathrm{e}^{-f_\omega(x_1, x_2)}} \mathrm{d}P(X_1, X_2) \\ & - \frac{\mathrm{e}^{f_\omega(x_1, x_2)}}{1 + \mathrm{e}^{f_\omega(x_1, x_2)}} \mathrm{d}P(X_1)P(X_2). \end{aligned} \tag{14}$$

When $\frac{\partial I_{\mathrm{JS}}}{\partial f_\omega(x_1, x_2)} = 0$,

$$\frac{\mathrm{d}P(X_1, X_2)}{\mathrm{d}P(X_1)P(X_2)} = \frac{p(x_1, x_2)}{p(x_1)p(x_2)} = \mathrm{e}^{f_\omega(x_1, x_2)}. \tag{15}$$

Furthermore, the second-order derivative of I_{JS} concerning $f_\omega(x_1, x_2)$ is:

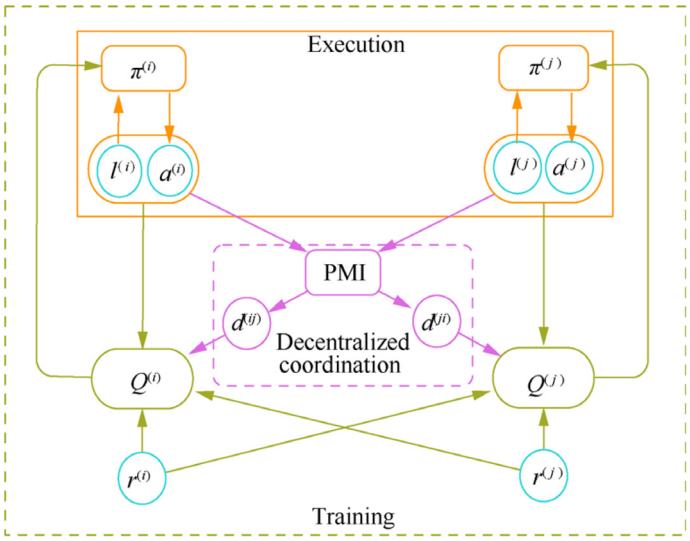


Fig. 3 Overview of the proposed algorithm.

$$\frac{\partial^2 I_{\mathrm{JS}}}{\partial f_\omega^2(x_1,x_2)} = -\frac{\mathrm{e}^{-f_\omega(x_1,x_2)}}{\left(1+\mathrm{e}^{-f_\omega(x_1,x_2)}\right)^2}\mathrm{d}P(X_1,X_2) - \frac{\mathrm{e}^{f_\omega(x_1,x_2)}}{\left(1+\mathrm{e}^{f_\omega(x_1,x_2)}\right)^2}\mathrm{d}P(X_1)P(X_2) < 0, \tag{16}$$

So I_{JS} is a convex function concerning f_ω , and it has a unique maximum value when $f_\omega^*(x_1,x_2) = \log\frac{p(x_1,x_2)}{p(x_1)p(x_2)}$, and $f_\omega^*(x_1,x_2)$ is the PMI of the random event pair (x_1,x_2) .

Therefore, we can use the neural network $f_\omega(l^{(i)},a^{(i)},l^{(j)},a^{(j)})$ to estimate $\mathrm{PMI}(l^{(i)},a^{(i)};l^{(j)},a^{(j)})$ via maximizing $I_{\mathrm{JS}}(L^{(i)},A^{(i)};L^{(j)},A^{(j)})$. In practice, the minimum PMI is 0 when there is no dependence between two random events, but the output of f_ω maybe less than 0, so we take $\mathrm{PPMI}(l^{(i)},a^{(i)};l^{(j)},a^{(j)})$ instead.

5.3. Algorithm construction

Due to the additional reciprocal rewards, the reshaped immediate reward of each UAV is not only related to the environment but also related to the UAV's neighbors. In formal way, replacing the local observation in the Dec-POMDP with the local information of each UAV, and replacing the environmental reward with the reshaped reward in Eq. (7), will not change the Dec-POMDP paradigm, so the proposed method can be easily combined with the existing MADRL algorithms.

In this article, we modify the vanilla Multi-Agent Actor-Critic (MAAC) algorithm and add a step to calculate each UAV's reciprocal rewards with its neighbors. Since all the UAVs are homogeneous and the PMI is symmetrical, the output of the PMI network should be identity irrelevant. Thus, the UAVs could share a common PMI network. When performing a cooperative mission, the homogeneity of UAVs means that their policies should also be similar. Therefore, their policies could also be shared. Using reciprocal reward to adapt the vanilla MAAC algorithm, the overview of the proposed algorithm is shown in Fig. 3.

Combined with the experience replay mechanism[52], we propose our experience sharing MAAC-R algorithm, as shown in Algorithm 1. In this algorithm, all UAVs' experiences are collected to train the shared actor-critic network and PMI network, which significantly improves the diversity of the training data and benefits to improve the training efficiency and the generalization ability of the networks. Then the trained networks are published to all UAVs to execute in a decentralized fashion.

In the process of training, we assume that each UAV can immediately obtain the environmental rewards and local information of neighboring UAVs. And, it uses the PMI network to respectively calculate the dependency index with each neighbor. The dot product of these dependence indexes and the received environmental immediate rewards is used to reshape the UAV's environment reward. Then we can perform the update step of the MADRL algorithm using the reshaped reward, which is consistent with the update step of the vanilla actor-critic algorithm. During execution process, each UAV makes its action decision using the shared policy in a decentralized manner. The input variable of the policy is each UAV's local information, and the output variable is the index of the UAV's action decision, which is identity irrelevant.

Please note that during both the training and execution processes, each drone only obtains the local information directly related to it, not the global information of the environment. Thus, the proposed MAAC-R algorithm can scale well to the scenarios whose population is large and variable to achieve cooperation, which is infeasible in the methods that require all agents' information, such as MADDPG[30] and QMIX[26], etc.

Please cite this article in press as: ZHOU W et al. Improving multi-target cooperative tracking guidance for UAV swarms using multi-agent reinforcement learning, *Chin J Aeronaut* (2021), https://doi.org/10.1016/j.cja.2021.09.008

Algorithm 1. Experience sharing reciprocal reward MAAC (MAAC-R)

Initialize. Actor network parameters θ , evaluation network parameters ϕ , target network parameters ϕ^- , AC experience replay buffer D_1 , PMI network parameters ω , PMI experience replay buffer D_2
1. **for** episodes $= 1 : M$ **do**
2. Reset environment
3. Receive all UAVs' joint local information l
4. **for** $t = 1 : T$ **do**
//Collect experience
5. For each UAV i , select action $a^{(i)} \sim \pi_\theta(l^{(i)})$ w.r.t. the shared policy and exploration
6. Execute joint action a to receive joint environmental immediate reward $\boldsymbol{r}_e$ and refreshed joint local information l'
7. **for** $i = 1 : N$ **do**
8. For UAV i , get its neighbors $\mathcal{N}^{(i)}$
9. **for** $j = 1 : |\mathcal{N}^{(i)}|$ **do**
10. Calculate the cooperation index $d^{(ij)} = f_\omega(l^{(i)}, a^{(i)}; l^{(j)}, a^{(j)})$
11. Store $(l^{(i)}, a^{(i)}; l^{(j)}, a^{(j)})$ into buffer D_2
12. **end for**
13. Reshape $r^{(i)}$ according to Eqs. (7) and (8)
14. Store $(l^{(i)}, a^{(i)}, r^{(i)}, l'^{(i)})$ into buffer D_1
15. **end for**
//Update actor-critic network
16. Randomly sample B_1 samples (l_k, a_k, r_k, l'_k) from buffer D_1
17. Set $y_k = r_k + \gamma Q^-_{\phi^-}(l'_k, a'_k)|_{a'_k \sim \pi_\theta(l'_k)}$
18. Update ϕ by minimizing $\frac{1}{B_1}\sum_k (y_k - Q_\phi(l_k, a_k))^2$
19. Update θ by minimizing $-\frac{1}{B_1}\sum_k \log(\pi_\theta(a_k|o_k))Q_\phi(l_k, a_k)$
20. **if** update target network **then**
21. $\phi^- \leftarrow \tau\phi + (1-\tau)\phi^-$
22. **end if**
//Update PMI network
23. Randomly sample B_2 samples $(l_k^1, a_k^1; l_k^2, a_k^2)$ from buffer D_2
24. Randomly sample B_2 samples $(\bar{l}_k^2, \bar{a}_k^2)$ from buffer D_2
25. Update ω by minimizing $\frac{1}{B_2}\sum_k \log(1 + \exp(-f_\omega(l_k^1, a_k^1, l_k^2, a_k^2))) + \log(1+ \exp(f_\omega(l_k^1, a_k^1, \bar{l}_k^2, \bar{a}_k^2)))$
26. **end for**
27. **end for**

6. Numerical experiment

In this section, we evaluate the proposed algorithm in the established MTTG problem. And we hope that the UAVs can learn to cooperate with their neighbors in a decentralized way, but the methods that required global information are not satisfied, such as MADDPG[30], QMIX[26], and influence of state transition[33], etc. Thus, the baseline algorithms include the vanilla MAAC algorithm and an ablated version of MAAC in which UAVs can receive the global reward as their common reward (named as MAAC-G).

The objective is to answer the following questions: (A) Can the reciprocal reward method improve the cooperation and performance of UAVs in the MTTG problem? (B) What did the UAVs learn using the proposed algorithm? (C) How scalable are the learned policy?

6.1. Parameters setting

According to the established models, we developed a UAV swarm tracking multi-target simulation platform. All the algorithms are trained using the same environment configuration, as shown in Table 2. And the hyperparameters are configured in Table 3. During all algorithms' training processes, we use an annealing exploration probability $\beta \in [0, 1]$ to balance the exploration and exploitation issue. Specifically, at the beginning of the training process, the UAVs are supposed to fully explore the environment, so we set $\beta = 1$ that encourage the UAVs to randomly select different actions in their action spaces to collect various experiences. As the training episode increase, β gradually anneals to 0, and the UAVs gradually apply the learned policy to make their action decisions.

6.2. Validity verification

According to the goal of the MTTG problem, we count the number of targets that a single UAV can track and the number of targets tracked by all UAVs respectively at each step to evaluate the performance of the algorithms. The weight parameter in MAAC-R is set as $\alpha = 0.3$.

Fig. 4 shows the training results of the algorithms. It can be seen from Fig. 4(a) and Fig. 4(b) that the performance of MAAC-R is better than the other two algorithms in terms of the average targets and the collective targets. The advantage of MAAC-R over the vanilla MAAC shows that in decentralized cooperation, using reciprocal rewards from neighbors to reshape the UAVs' original rewards can effectively improve the cooperation between UAVs. Interestingly, when using MAAC-G, the numbers of the tracked targets in both subfigures first increase quickly, but then decrease immediately. It can be inferred that in UAV swarms, greedily maximizing the global reward may have a certain effect in a certain period, but the enhancement of greed may also cause conflicts between UAVs, which is not conducive to cooperation. Therefore, the proposed maximum reciprocal reward method can improve the cooperation and performance of UAV swarms in the MTTG problem.

6.3. Performance test

To better evaluate the policies learned by these algorithms, we reload and execute the trained actor networks without further tuning. And the reciprocal reward is no longer required in the execution process. The parameters of the testing environment are identical to the training one, and the environment is randomly initialized before each execution episode.

Each actor-network is executed 100 episodes and the average statistical indicators are shown in Table 4. In these tests, all the statistical indicators of MAAC-R obviously outperform those of other methods. Especially, the comparison of environmental reward indicates that the introduction of the weighted reciprocal reward can improve UAVs' environmental reward. Moreover, it can be inferred that the enhancement of cooperation between UAVs may also improve the shortcomings of

Table 2 Environment parameter settings.

Entity	Variable	Value
Environment	Shape	Square
	Size	2 km × 2 km
UAV	Total number	10
	Communication range (m)	500
	Perception range (m)	200
	Speed (m/s)	20
	Max heading angular rate ((°)/s)	30
Target	Total number	10
	Speed (m/s)	5

Table 3 Hyperparameter configurations.

Hyperparameter	Value
Max episode	1000
Max step	200
Replay buffer	1e5
Batch size	128
Discount factor	0.95
Critic learning rate	5×10^{-4}
Actor learning rate	1×10^{-4}

the original reward shaping, which can further improve the performance of the system while increasing individual rewards. This also illustrates the significance of cooperation for UAV swarms.

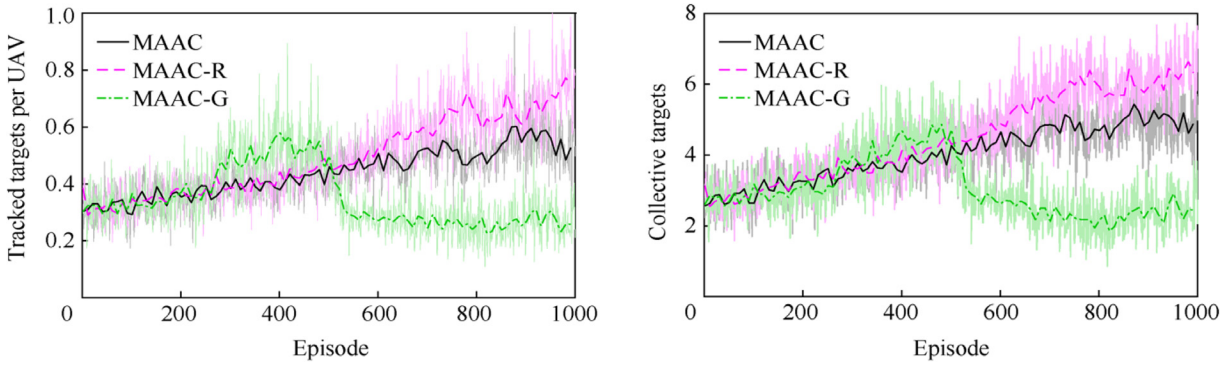
To intuitively understand whether the UAVs have learned how to track targets, Fig. 5 shows the visualization of the tracking process in which UAVs execute the policy learned by MAAC-R. In Fig. 5(a), all the UAVs and targets are randomly initialized, and there are only 3 targets covered by the UAVs. Then, UAVs track more targets cooperatively by executing the learned policy and various cooperation forms between the UAVs emerge, including (A) formation: neighboring UAVs that are not tracking any targets may automatically form a formation and maintain maximum coverage (no repeated perceptions); (B) n/n : multi-UAV track multi-target cooperatively; (C) $1/n$: a single UAV tracks multi-target simultaneously; (D) $n/1$: multi-UAV track a single target cooperatively; (E) $1/1$: one UAV tracks a single target; (F) exchange: UAVs exchange each other's tracking targets.

Surprisingly, even if there is no explicit formation motivation in reward shaping, the UAVs in the red box in Fig. 5(b) form a formation. The previous experiments have verified that maximizing reciprocal rewards can encourage the UAVs to actively cooperate, and the cooperation only exists between the neighboring UAVs, so they have the motivation to approach each other. However, when the UAVs get too close, their observation areas may overlap, which is punished in reward shaping. Therefore, there may also be repulsive motives between the UAVs. It can be seen from the figure that only the communication information is valid among the input information of the 3 UAVs, and each UAV moves under the drive of these two motivations. Therefore, each UAV stays away from other UAVs that are trying to approach, while chasing other UAVs that are trying to stay away. When these motivations among the UAVs are balanced with each other, the UAVs form a formation. Therefore, the visualization demonstrates that the UAVs can emerge a variety of flexible cooperation behaviors using the proposed MAAC-R algorithm.

6.4. Scalability test

In the proposed MAAC-R algorithm, each UAV only interacts directly with its neighbors during both training and execution processes. The above experiments are implemented in the same scenario (2 km × 2 km, 10 UAVs, and 10 targets). Next, we verify the scalability of the previously learned policies in different scenarios. Similarly, the saved actor-networks are reloaded and published to all UAVs for execution. And the learned policies are executed 100 episodes in each scenario. The statistical indicators of the test results are shown in Table 5.

To normalize the tracking performance metrics in different scenarios, the tracking ratio is defined as the proportion of tracked targets to all targets. It can be obtained from the comparison of statistical indicators in all the test scenarios, the performance of MAAC-R is better than the other two methods. This demonstrates that the policy learned using MAAC-R can maintain better generalization ability and scalability in different scenarios. The possible reason is that when calculating the reciprocal reward of each UAV, only those neighboring UAVs that interact directly are considered, which makes the reciprocal reward free from the curse of dimensionality. And,



(a) Average number of targets tracked by per UAV

(b) Collective number of targets tracked by all UAVs

Fig. 4 Training curves of different algorithms.

Table 4 Test results statistical indicators.

Statistical indicator	MAAC	MAAC-R	MAAC-G
Average environment reward per UAV	−0.1818	**0.1310**	−0.6812
Average tracked targets per UAV	0.4169	**0.7553**	0.2756
Average collective tracked targets	0.3827	**0.6315**	0.2624

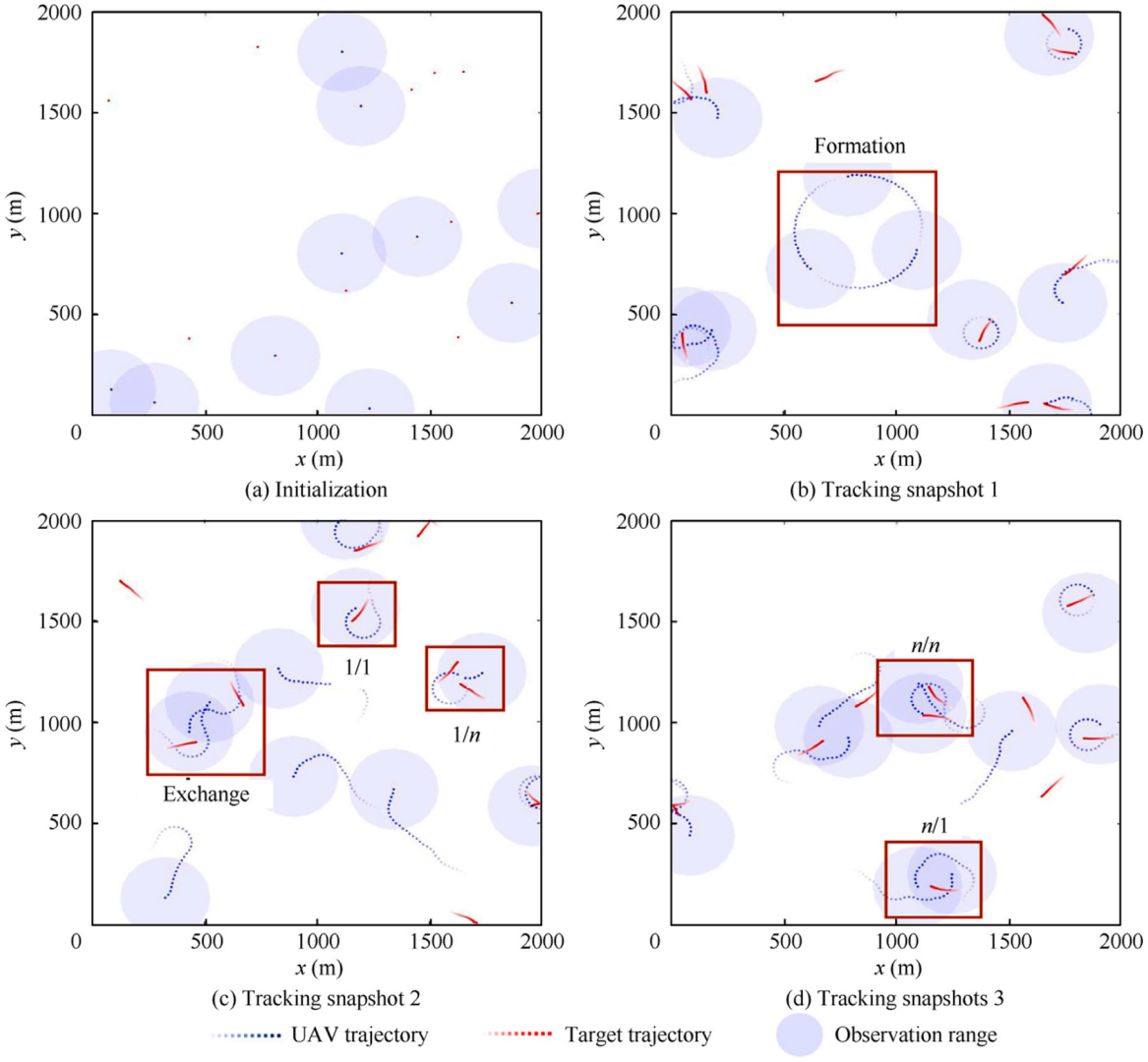


Fig. 5 Visualization of multi-target tracking by UAVs.

the introduction of PMI can identify those necessary and highly relevant interactions and eliminate unnecessary and weakly relevant interactions. Therefore, the decentralized cooperative policy learned in MAAC-R can be executed well in scenarios where the number of UAVs is larger. Consequently, we believe that the proposed MAAC-R algorithm has the potential to be widely scaled to other collaborative scenarios with more UAVs and targets.

6.5. Computational complexity analysis

There are N homogeneous UAVs and the cardinal number of the individual discrete action space is M . Consider the most extreme case that each UAV can communicate with all other $N-1$ UAVs. To compute the reciprocal reward for each UAV, its computational complexity is $O(N-1)$ since it only needs to calculate the PMI of N-1 random event pairs. How-

Table 5 Statistical indicators of execution results in different scenarios.

Mapsize(m)	$\frac{\text{UAVs'number}}{\text{Targets'number}}$	Statistical indicator	MAAC	MAAC-R	MAAC-G
2000	5/5	Average environment reward per UAV	0.0069	**0.2908**	−0.5177
		Average tracked targets per UAV	0.2318	**0.5159**	0.1463
		Tracking ratio	0.2241	**0.4863**	0.1415
2000	10/10	Average environment reward per UAV	−0.1759	**0.1249**	−0.6812
		Average tracked targets per UAV	0.4042	**0.7681**	0.2756
		Tracking ratio	0.3792	**0.6420**	0.2624
2000	20/20	Average environment reward per UAV	−0.9762	**−0.6487**	−1.3348
		Average tracked targets per UAV	0.7801	**1.2162**	0.5212
		Tracking ratio	0.6155	**0.8037**	0.4447
2000	50/50	Average environment reward per UAV	−3.9922	**−3.7682**	−4.0998
		Average tracked targets per UAV	1.8226	**2.3769**	1.3104
		Tracking ratio	0.9020	**0.9363**	0.7310
5000	100/100	Average environment reward per UAV	−0.7829	**−0.4595**	−1.2182
		Average tracked targets per UAV	0.6568	**1.1208**	0.4902
		Tracking ratio	0.5495	**0.7767**	0.4277
5000	200/200	Average environment reward per UAV	−2.4423	**−2.3646**	−2.9769
		Average tracked targets per UAV	1.2534	**1.7713**	0.9811
		Tracking ratio	0.8014	**0.8921**	0.6633
10,000	1000/1000	Average environment reward per UAV	−3.5977	**−3.5681**	−4.3600
		Average tracked targets per UAV	1.5793	**2.1836**	1.2972
		Tracking ratio	0.8694	**0.9233**	0.7514

ever, if we use the MI to capture the dependence between UAVs' policies, the computational complexity is $O((N-1)M^2)$; and if we use the social influence model to compute the causal influence of a UAV's action on other UAVs' policies, the computational complexity is $O(2(N-1)M)$ when the causal influence is assessed using counterfactual reasoning. Moreover, both MI and social influence are aggregate statistics between random variables, and their calculation requires knowing or estimating the action policies of other UAVs. Therefore, these are much more complicated than directly calculating PMI.

7. Conclusions

This work studies the MTTG problem for UAV swarms in an unknown environment using the MADRL technique. We propose the maximum reciprocal reward method to enable large-scale homogeneous UAVs to learn cooperative policies in a decentralized manner. Specifically, the reciprocal reward of each UAV is defined as the dot product of the environment reward vector of all neighboring UAVs and the dependency vector between the UAV and its neighbors, where the dependence can be estimated using a PMI neural network. Further, the reciprocal reward is used as a regularization term to reshape the UAV's original reward. Maximizing the reshaped reward can not only maximize the UAV's reward but also maximize the rewards of the neighboring UAVs, which realizes decentralized cooperation between UAVs. Combined with the maximum reciprocal reward method and PMI estimation, we propose the MAAC-R algorithm based on the experience replay sharing mechanism to learn the collaborative sharing policies for UAV swarms. Numerical experiments demonstrate that the proposed MAAC-R algorithm can better improve the cooperation between UAVs than the baseline algorithms, and excite the UAV swarms to emerge a rich form of cooperative behaviors. Also, the learned policy can well scale to other collaborative scenarios with more UAVs and targets.

Although this paper only focuses on improving the cooperation of homogeneous UAV swarms, in the future, we will adapt the maximum reciprocal reward method to heterogeneous UAV swarms to improve their cooperation in a decentralized manner.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work is funded by the Science and Technology Innovation 2030-Key Project of "New Generation Artificial Intelligence", China (No. 2020AAA0108200) and the National Natural Science Foundation of China (No. 61906209).

References

1. Roberge V, Tarbouchi M, Labonte G. Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. *IEEE Trans Ind Inform* 2013;**9**(1):132–41.
2. Kulkarni RV, Venayagamoorthy GK. Bio-inspired algorithms for autonomous deployment and localization of sensor nodes. *IEEE Trans Syst Man Cybern C Appl Rev* 2010;**40**(6):663–75.
3. Kuriki Y, Namerikawa T. Formation control with collision avoidance for a multi-UAV system using decentralized MPC and consensus-based control. 2015 European control conference (ECC); 2015 July 15-17; Linz, Austria. Piscataway: IEEE Press; 2015.p.3079–84.
4. Wu XL, Yang ZC, Huo JN, et al. UAV formation control based on consistency. 2015 7th international conference on modelling, identification and control (ICMIC); 2015 December 18-20; Sousse, Tunisia. Piscataway: IEEE Press; 2015.p.1–5.
5. Yao P, Wang H, Su Z. Cooperative path planning with applications to target tracking and obstacle avoidance for multi-UAVs. *Aerosp Sci Technol* 2016;**54**:10–22.

6. He B, Liu G, Yan JZ, et al. A UAV route planning method based on voronoi diagram and quantum genetic algorithm. Electron Opt Control 2013;20(1):5–8,18 [Chinese].
7. Crandall JW, Goodrich MA. Learning to compete, coordinate, and cooperate in repeated games using reinforcement learning. *Mach Learn* 2011;**82**(3):281–314.
8. Jaderberg M, Czarnecki WM, Dunning I, et al. Human-level performance in 3D multiplayer games with population-based reinforcement learning. *Science* 2019;**364**(6443):859–65.
9. Rizk Y, Awad M, Tunstel EW. Decision making in multiagent systems: a survey. *IEEE Trans Cogn Dev Syst* 2018;**10**(3):514–29.
10. Goldhoorn A, Garrell A, Alquézar R, Sanfeliu A. Searching and tracking people with cooperative mobile robots. *Auton Robots* 2018;**42**(4):739–59.
11. Qie H, Shi D, Shen T, Xu X, Li Y, Wang L. Joint optimization of multi-UAV target assignment and path planning based on multi-agent reinforcement learning. *IEEE Access* 2019;**7**:146264–72.
12. Rosello P, Kochenderfer MJ. Multi-agent reinforcement learning for multi-object tracking. *Proceedings of the 17th international conference on autonomous agents and multi-agent systems*, 2018. p.1397–404..
13. Wang T, Qin R, Chen Y, Snoussi H, Choi C. A reinforcement learning approach for UAV target searching and tracking. *Multimed Tools Appl* 2019;**78**(4):4347–64.
14. Zhu P, Dai W, Yao W, Ma J, Zeng Z, Lu H. Multi-robot flocking control based on deep reinforcement learning. *IEEE Access* 2020;**8**:150397–406.
15. Nowak MA. Five rules for the evolution of cooperation. *Science* 2006;**314**(5805):1560–3.
16. Senanayake M, Senthooran I, Barca JC, Chung H, Kamruzzaman M, Murshed M. Search and tracking algorithms for swarms of robots: a survey. *Robotics Auton Syst* 2016;**75**:422–34.
17. Jilkov VP, Li XR. On fusion of multiple objectives for UAV search & track path optimization. *J Adv Information Fusion* 2009;**4**(1):27–39.
18. Pitre RR, Li XR, Delbalzo R. UAV route planning for joint search and track missions—an information-value approach. *IEEE Trans Aerosp Electron Syst* 2012;**48**(3):2551–65.
19. Choi HL, Brunet L, How JP. Consensus-based decentralized auctions for robust task allocation. *IEEE Trans Robotics* 2009;**25**(4):912–26.
20. Peterson CK. Dynamic grouping of cooperating vehicles using a receding horizon controller for ground target search and track missions. 2017 IEEE conference on control technology and applications (CCTA); 2017 August 27-30; Maui, HI, USA. Piscataway: IEEE Press; 2017.p.1855–60.
21. Botts CH, Spall JC, Newman AJ. Multi-agent surveillance and tracking using cyclic stochastic gradient. 2016 American control conference (ACC); 2016 July 6-8; Boston, MA. Piscataway: IEEE Press; 2016.p.270–5.
22. Khan A, Rinner B, Cavallaro A. Cooperative robots to observe moving targets: review. *IEEE Trans Cybern* 2018;**48**(1):187–98.
23. Mao HY, Gong ZB, Ni Y, et al. ACCNet: Actor-coordinator-critic net for "learning-to-communicate" with deep multi-agent reinforcement learning. 2017:arXiv: 1706.03235[cs.AI]. https://arxiv.org/abs/1706.03235
24. Peng P, Wen Y, Yang YD, et al. Multiagent bidirectionally-coordinated nets: emergence of human-level coordination in learning to play StarCraft combat games. 2017: arXiv: 1703.10069[cs.AI]. https://arxiv.org/abs/1703.10069
25. Sukhbaatar S, Szlam A, Fergus R. Learning Multiagent Communication with Backpropagation*Proceedings of the 30th international conference on neural information processing systems*. p. 2252–60.
26. Rashid T, Samvelyan M, Schroeder C, et al. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning*Proceedings of the 35th international conference on machine learning*. p. 4295–304.
27. Son K, Kim D, Kang WJ, et al. QTRAN: learning to factorize with transformation for cooperative multi-agent reinforcement learning. Proceedings of the 36th international conference on machine learning; 2019.p.10329-46.
28. Sunehag P, Lever Guy, Gruslys A, et al. Value-decomposition networks for cooperative multi-agent learning based on team reward. Proceedings of the 17th international conference on autonomous agents and multiagent systems; 2018.p.2085–87.
29. Foerster J N, Farquhar G, Afouras T, et al. Counterfactual multi-agent policy gradients. Proceedings of the AAAI conference on artificial intelligence; 2018.p.2974–82.
30. Lowe R, Harb J, Tamar A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments. 31st conference on neural information processing systems; 2017.p.6379-90.
31. Kim W, Jung W, Cho M, et al. A maximum mutual information framework for multi-agent reinforcement learning. 2020:arXiv: 2006.02732[cs.MA]. https://arxiv.org/abs/2006.02732
32. Cuervo S, Alzate M. Emergent cooperation through mutual information maximization. 2020:arXiv: 2006.11769[cs.AI]. https://arxiv.org/abs/2006.11769
33. Wang TH, Wang JH, Wu Y, et al. Influence-based multi-agent exploration. 2019:arXiv: 1910.05512[cs.AI]. https://arxiv.org/abs/1910.05512
34. Jaques N, Lazaridou A, Hughes E, et al. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. 2018:arXiv: 1810.08647[cs.LG]. https://arxiv.org/abs/1810.08647
35. Barton SL, Waytowich NR, Asher DE. Coordination-driven learning in multi-agent problem spaces. 2018:arXiv: 1809.04918[cs.MA]. https://arxiv.org/abs/1809.04918
36. Barton SL, Waytowich NR, Zaroukian E, et al. Measuring collaborative emergent behavior in multi-agent reinforcement learning. Human systems engineering and design. Cham: Springer International Publishing, 2018.p.422–7.
37. Barton SL, Zaroukian E, Asher DE, et al. Evaluating the coordination of agents in multi-agent reinforcement learning. Advances in intelligent systems and computing. Cham: Springer International Publishing, 2019.p.765–70.
38. Sugihara G, May R, Ye H, et al. Detecting causality in complex ecosystems. *Science* 2012;**338**(6106):496–500.
39. Baldazo D, Parras J, Zazo S. Decentralized multi-agent deep reinforcement learning in swarms of drones for flood monitoring. *27th European Signal Processing Conference;* 2019.p.1–5.
40. Wang C, Wang J, Zhang XD. A deep reinforcement learning approach to flocking and navigation of uavs in large-scale complex environments. 2018 IEEE global conference on signal and information processing (GlobalSIP); 2018 November 26-29; Anaheim, CA. Piscataway: IEEE Press; 2018.p.1228–32.
41. Ballerini M, Cabibbo N, Candelier R, et al. Interaction ruling animal collective behavior depends on topological rather than metric distance: evidence from a field study. *PNAS* 2008;**105**(4):1232–7.
42. Khan A, Kumar V, Ribeiro A. Large scale distributed collaborative unlabeled motion planning with graph policy gradients. *IEEE Robotics Autom Lett* 2021;**6**(3):5340–7.
43. Venturini F, Mason F, Pase F, et al. Distributed reinforcement learning for flexible and efficient UAV swarm control. *IEEE Trans Cogn Commun Netw* 2021;**7**(3):955–69.
44. Dibangoye JS, Amato C, Buffet O, Charpillet F. Optimally solving dec-POMDPs as continuous-state MDPs. *Jair* 2016;**55**:443–97.
45. Takayama J, Arase Y. Relevant and informative response generation using pointwise mutual information. Proceedings of the first workshop on NLP for conversational AI; 2019.p.133-8.
46. Tampuu A, Matiisen T, Kodelja D, et al. Multiagent cooperation and competition with deep reinforcement learning. 2015:arXiv: 1511.08779[cs.AI]. https://arxiv.org/abs/1511.08779.
47. Oord AVD, Li YZ, Vinyals O. Representation learning with contrastive predictive coding. 2018:arXiv: 1807.03748[cs.LG]. https://arxiv.org/abs/1807.03748.

Please cite this article in press as: ZHOU W et al. Improving multi-target cooperative tracking guidance for UAV swarms using multi-agent reinforcement learning, *Chin J Aeronaut* (2021), https://doi.org/10.1016/j.cja.2021.09.008

48. Nguyen X, Wainwright MJ, Jordan MI. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Trans Inf Theory* 2010;**56**(11):5847–61.
49. Belghazi MI, Baratin A, Rajeswar S, et al. Mutual information neural estimation. Proceedings of the 35th international conference on machine learning; 2018.p.864–73.
50. Poolel B, Ozair S, Van DOA, et al. On variational bounds of mutual information. Proceedings of the 36th international conference on machine learning; 2019.p.9036–49.
51. Tsai YHH, Zhao H, Yamada M, et al. Neural methods for pointwise dependency estimation. 2020:arXiv: 2006.05553 [cs.LG]. https://arxiv.org/abs/2006.05553.
52. Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. *Nature* 2015;**518**(7540):529–33.

Please cite this article in press as: ZHOU W et al. Improving multi-target cooperative tracking guidance for UAV swarms using multi-agent reinforcement learning, *Chin J Aeronaut* (2021), https://doi.org/10.1016/j.cja.2021.09.008