drones

Article

UAV Swarm Rounding Strategy Based on Deep Reinforcement Learning Goal Consistency with Multi-Head Soft Attention Algorithm

Zhaotian Wei [1,2] **and Ruixuan Wei** [2,*]

[1] Graduate School, Air Force Engineering University, Xi'an 710051, China
[2] Aviation Engineering School, Air Force Engineering University, Xi'an 710038, China
[*] Correspondence: rxwei369@sohu.com

Abstract: Aiming at the problem of target rounding by UAV swarms in complex environments, this paper proposes a goal consistency reinforcement learning approach based on multi-head soft attention (GCMSA). Firstly, in order to make the model closer to reality, the reward function when the target is at different positions and the target escape strategy are set, respectively. Then, the Multi-head soft attention module is used to promote the information cognition of the target among the UAVs, so that the UAVs can complete the target roundup more smoothly. Finally, in the training phase, this paper introduces cognitive dissonance loss to improve the sample utilization. Simulation experiments show that GCMSA is able to obtain a higher task success rate and is significantly better than MADDPG in terms of algorithm performance.

Keywords: UAV swarms; roundup strategy; deep reinforcement learning; GCMSA

1. Introduction

With the rapid development of artificial intelligence technology, Multi-Agent Reinforcement Learning (MARL) has been widely used in many fields, such as games, robots, communication and so on. As an important research field of MARL, cooperative reinforcement learning plays an important role in solving UAV swarm cooperative decision-making problems in complex environments. In the face of the partial observability and non-stationarity of the environment, how to set cooperative behavior among agents to maximize team rewards is a huge challenge.

UAV swarm cooperative target roundup is the study of how to guide swarms with autonomous decision-making ability to roundup a single target or a set of targets through mutual cooperation. A good decision scheme can reduce the task cost and improve the efficiency, so the problem of UAV swarm cooperative roundup has been widely examined by scholars. The difficulty of dealing with this problem is how to establish a cooperative relationship between UAVs and the design of a cooperative roundup strategy.

In view of the cooperative relationship between UAV swarms, some communication-based multi-agent reinforcement learning methods are used to solve the problem [1], which also brings problems such as the selection of interaction information and communication bandwidth. Modeling the relationship between agents as a graph is an effective method to deal with the information interaction of agents [2]. This way of simulating human cooperation can promote cooperation between agents. However, the method of modeling agents only as a graph does not consider the influence of complex environments, so the effect of collaborative decision-making in complex environments is poor. The attention mechanism enables the model to selectively focus on the information that is most important for the current task and ignore other less important information. It is a good idea to utilize this advantage to deal with synergistic relationships between agents.

check for updates

Citation: Wei, Z.; Wei, R. UAV Swarm Rounding Strategy Based on Deep Reinforcement Learning Goal Consistency with Multi-Head Soft Attention Algorithm. *Drones* **2024**, *8*, 731. https://doi.org/10.3390/drones8120731

Academic Editor: Xiwang Dong

Received: 29 October 2024
Revised: 26 November 2024
Accepted: 26 November 2024
Published: 3 December 2024

For the design of a cooperative roundup strategy, paper [3] designed an individual self-organizing motion controller by decomposing roundup behavior and proposed a self-organizing cooperative roundup strategy based on the Loose-Preference Rule (LP Rule). Some scholars design roundup strategies by simulating group behavior in nature. Paper [4] summarized the roundup strategies of wolves by observing and calculating the rules and main characteristics of the roundup behavior of northern gray wolves. In recent years, reinforcement learning methods have also achieved good results in solving the roundup decision problem. From the perspective of multi-agent deep reinforcement learning, paper [5] constructed the scenario of agent cooperation and competition and proposed the multi-agent deep deterministic policy gradient (MADDPG) algorithm. A framework of centralized training and decentralized execution was adopted to achieve a cooperative strategy among groups. However, there is a common problem in the current research: only the roundup strategy of agents is considered, and the escape strategy of the target is ignored. At the same time, some task environments ignore the task boundary, which leads to the description of the problem being far from reality.

To solve the above problems, based on the Centralized Training with Decentralized Execution (CTDE) learning framework [6], this paper proposes a multi-agent reinforcement learning model with goal consistency by multi-head soft attention (GCMSA). We use the multi-head soft attention mechanism to determine the importance of information, promote the consistency of target cognition, and improve the decision-making efficiency. At the same time, in order to increase the authenticity of the roundup scene, the escape strategy of the target is imported and the task boundary of the agents is set.

We evaluated our approach in the established UAV swarm roundup single-target model. The experimental results show that our method can effectively complete the cooperative roundup tasks, and the UAVs have efficient information interaction behavior. In addition, we compared our method with the current main MARL method; the results showed that our proposed method had obvious advantages in convergence and cooperativity.

The major contributions and innovations of this paper include the following:

1. Different from the traditional simplified task model, this paper increases the escape strategy of the target and, considering the boundary of the task scenario, the reward functions when the target is in different positions are set, respectively, which makes the task scenario closer to reality.
2. The observation vectors of a UAV are parameterized into low-level cognitive vectors. Then, an MSA module based on the multi-head attention mechanism is designed to determine the importance level of the input information, which realizes the efficient selection of the target information by the intelligent body and promotes decision-making efficiency.
3. We propose the GCMSA algorithm based on the CTDE paradigm. In the centralized training phase, the global observation–action history can be accessed, and the KL divergence is used to calculate the team cognitive dissonance loss (TCD-loss) in the global perspective, and in the distributed execution phase the agents make strategy choices based on the local information, and the KL divergence is used to calculate the self cognitive dissonance loss (SCD-loss) in the local perspective. This approach can increase the cognitive consistency among agents.

This paper is organized as follows. Section 2 summarizes the related works. Section 3 provides a preliminary outline of Dec-POMDP and MADRL. Section 4 analyzes the target roundup problem and establishes the mathematical models. Section 5 details the proposed GCMSA methods. Experimental results and analyses are presented in Section 6. Finally, Section 7 concludes this paper.

2. Related Work

2.1. UAV Swarm Roundup Target

For the problem of the UAV swarm roundup target, scholars firstly used traditional control and optimization methods to solve it. Based on the distributed control, paper [7] used the consensus method to design multi-UAV convergence towards the target. After fully analyzing the group behavior of the biological world, some scholars have found that it can provide good learning strategies for UAVs by simulating the cooperative roundup behavior of biological groups. Based on this, paper [8] proposed a roundup method that could deal with a complex confrontation environment by analyzing the behavior of a gray wolf pack, and successfully solved the problem of target roundup. By observing the behavioral mechanism of whale roundup prey, paper [9] proposed an improved whale optimization algorithm, which predicted the target trajectory through polynomial fitting and allocated ideal roundup points by a two-way negotiation method to improve the efficiency of target roundup. Paper [10] referred to the cooperative mechanism of Harris eagle roundup prey and proposed a hybrid control strategy of multiple switching of "charging" and "relay attack" to increase the success rate of the roundup task.

With the development of reinforcement learning methods, scholars began to try to solve the problem of UAV swarm cooperative target roundup by reinforcement learning methods.The methods mainly guide UAVs to hunt targets by setting the judgment condition of successful roundup as a reward function. Among them, Fan [11] combined model control and reinforcement learning to design a target roundup model guided by potential energy, and divided the roundup strategy into tracking roundup and circular roundup. The tracking roundup mainly tracked and captured the target, and the circular roundup mainly ensured the stable convergence of the UAV to the roundup circle to prevent the target from escaping. Paper [12,13] applied the boundary analysis method to the constraint analysis of the single target roundup task, and determined the required number of UAVs and the speed relationship between the two sides of the confrontation.

In this paper, we focus on the problem of UAV swarm cooperation to round up a single target. An escape strategy of the target is introduced to improve the authenticity of the task environment, and a reward function is designed to promote UAV cooperation in multi-stage tasks.

2.2. Collaborative Reinforcement Learning

How to better carry out cooperation is an important issue in MADRL, and it is also one of the relevant research areas. In order to learn cooperative strategies between agents, several methods have been proposed, including communication learning-based methods, collaborative learning-based methods, reward shaping-based methods, etc.

The methods based on communication learning mainly exchange information through communication between agents, so that agents can combine their own local observation information and the information from other agents to make decisions, which alleviates the partial observability problem of the environment to a large extent. Among them, the CommNet algorithm [14] used continuous communication to complete the collaborative task, and learned the communication and strategy of multi-agents together. However, due to the use of a global reward for training, there is a problem of credit allocation. The BiCNet algorithm [15] used Bi-RNN, a bidirectional recurrent neural network, for communication. In order to enable agents to maintain their internal states during communication, Bi-RNN was used as the local memory of agents. The IC3Net algorithm [16] was based on an individual controlled continuous communication model, which used multiple network controllers with shared parameters composed of LSTM networks, so that each Agent can use different individual rewards for training, and then effectively improved the credit assignment problem of the CommNet algorithm. The I2C algorithm [17] greatly reduced the communication load by introducing a causal inference module to decide the unique communication object in the agent domain.

The method based on collaborative learning achieves global cooperation through an implicit communication method: that is, each agent considers the local observations and strategies of other agents when making decisions. This kind of method can be considered from three perspectives: based on "value function decomposition", "value function sharing" and reward shaping.

The value function decomposition approach is to train a joint value function network (Q-network), which is the aggregation of the individual networks of each agent. VDN [18] was an early method based on the decomposition value function, which used LSTM as a Q-network and approximately estimated the joint value function as the sum of all decomposition value functions, but this simple sum method had certain limitations. As an extension of VDN, QMIX [19] solved the problem of belief assignment faced by VDN in dealing with problems by using a nonlinear Mixing network instead of simple addition. The QTRAN [20] algorithm was a further improvement of VDN and QMIX, it mapped the original global value function to a new value function, so that the optimal joint action of the two functions was equivalent. QTRAN achieved good results in dealing with non-monotonic tasks.

The method of value function sharing is usually based on the learning framework of the actor–critic algorithm, which uses global information to learn the critic, and learns an independent actor for each agent based on the global critic. In MADDPG [5], the critic policy of each agent was learned based on global information, so it could effectively solve the problem of a non-stationary environment. COMA [21] aimed to solve the multi-agent credit assignment problem in Dec-POMDP. The MAAC [22] algorithm was an extension of MADDPG by replacing the DDPG algorithm adopted in MADDPG with the SAC (soft actor–critic) algorithm and introducing the counterfactual benchmark proposed in COMA. The MAAC algorithm also introduced an attention mechanism into the construction of the value function, giving different attention to the agent, so as to make full use of the local observations and actions of the agent.

The method based on reward shaping takes setting a perfect reward function as the breakthrough point to promote closer cooperation between agents. In the process of task execution, the agent may pursue its own short-term interests and damage the long-term interests. This problem also exists in some optimization problems. The cooperative multi-agent reinforcement learning algorithm based on reward shaping tries to achieve better cooperation between agents in this problem. Paper [23] solved the simple two-agent problem. The concept of a prosocial agent was proposed, and it was embodied by a cost function (reward function) of prosocial agents. The reward function of different agents was balanced by means of weights to promote the cooperation of agents. Paper [24] introduced the theory of Inequity Aversion to solve the problem of social dilemma, and can solve complex video games based on this method.

In this paper, we pay more attention to the dynamic information interaction between agent and agent, as well as agent and environment, and guide efficient exploration by means of importance information sampling. Our approach improves the efficiency of information interaction and facilitates collaborative search among UAVs.

3. Preliminary Outline

3.1. Dec-POMDP

The UAV swarm target rounding up problem is often locally observable and the reward signal is shared globally. Generally, Dec-POMDP [25] is used to describe such a decision-making problem. A Dec-POMDP process consists of the following tuples: $G = \langle I, S, A, P, R, \Omega, O, n, \gamma \rangle$. Here, I represents a finite set of n agents, $s \in S$ is the global state of the environment and A is a collection of actions shared by the agents. $P(s'|s,a) : S \times A \times S \rightarrow [0,1]$ represents the state transition function of the environment, where a is the joint action of all agents. $r = R(s,a)$ represents the global reward of the agents. $o_i \in \Omega$ is the local observation of the agents; it is determined by the observation function $O(s,i)$. $\gamma \in [0,1)$ represents the discount factor, which measures long-term and short-term rewards.

The process of target rounding up is a fully cooperative decision-making process. For each state s, agent i will choose an action a_i based on local observations o_i. These actions form a joint action $a \in A$, and through the probability $P(s'|s,a)$ transfer the system to the next state s'. According to this, the environment will feedback a global reward signal

$r = r(s,a)$. Each agent has a local action–observation history $\tau_i \in T \equiv (\Omega \times A)^*$. Each agent's goal is to learn a policy $\pi_i(a_i|o_i)$ to maximize the total reward $E\left(\sum_{t=0}^{H} \gamma^t r_i^t\right)$.

3.2. Multi-Agent Deep Reinforcement Learning (MADRL)

Reinforcement learning (RL) methods are commonly used to solve the special DEC-POMDP problem with N = 1. With the development of deep learning technology, MARL has shifted from table-based methods to deep learning-based methods. Examples include DQN, MADDPG and MAAC.

DQN [26]. In order to improve the sample efficiency, DQN uses a target network and experience replay mechanism. The parameter ω is updated by minimizing the square of the TD error δ:

$$L(\omega) = E_{(s,a,r,s')\sim D}\left[(\delta)^2\right] \tag{1}$$

$$\delta = r + \gamma Q(s', a'; \omega^-) - Q(s, a; \omega) \tag{2}$$

where D represents the experience buffer pool containing the experience group (s, a, r, s') and $Q(s', a'; \omega^-)$ is the target network.

MADDPG [5]. MADDPG uses a centralized training and decentralized execution framework, which allows policies to use additional information when training. It is a simple extension of the actor–critic policy gradient approach, where the critic adds additional information about other agents and the actor only receives local information.

Therefore, the network update formula of the critic is as follows:

$$L(\omega_i) = \mathbb{E}_{x,a,r,x'}\left[\left(Q_i^{\mu}(x, a_1, \cdots, a_N) - y\right)^2\right] \tag{3}$$

$$y = r_i + \gamma Q_i^{\mu'}(x', a_1{}', \cdots, a_N{}')|_{a_j{}'=\mu_j{}'(o_j)} \tag{4}$$

The network update formula of the actor is as follows:

$$\nabla_{\theta_i} J(\mu_i) = \mathbb{E}_{x,a\sim D}\left[\nabla_{\theta_i}\mu_i(a_i|o_i)\nabla_{a_i} Q_i^{\mu}(x, a_1, \cdots, a_N)|_{a_i=\mu_i(o_i)}\right] \tag{5}$$

MAAC [22]. The biggest highlight of MAAC is that it uses an attention mechanism to improve the scalability of the critic input in MADDPG, which increases exponentially with the number of agents. It learns a focused critic with a soft attention mechanism that is able to dynamically select which agents can participate at each time step.

Since the critics of different agents use shared parameters, MAAC proposes to train each critic with a joint loss function, which is expressed as follows:

$$L_Q(\psi) = \sum_{i=1}^{N} \mathbb{E}_{(o,a,r,o')\sim D}\left[\left(Q_i^{\psi}(o,a) - y_i\right)^2\right] \tag{6}$$

$$y_i = r_i + \gamma \mathbb{E}_{a'\sim\pi_{\bar{\theta}}(o')}\left\{Q_i^{\bar{\psi}}(o', a') - \alpha \log\left[\pi_{\bar{\theta}_i}(a_i{}'|o'{}_i)\right]\right\} \tag{7}$$

The actor policies for each agent are updated as follows:

$$\nabla_{\theta_i} J(\pi_\theta) = \mathbb{E}_{o\sim D, a\sim pi}\left\{\nabla_{\theta_i}\log(\pi_{\theta_i}(a_i|o_i))\left[-\alpha\log(\pi_{\theta_i}(a_i|o_i)) + Q_i^{\psi}(o,a) - b\left(o, a_{\setminus i}\right)\right]\right\} \tag{8}$$

Attention Mechanism. Soft attention is a widely used attention mechanism, which can change the information transmission from focusing on all information to focusing on key information. The mathematical description of soft attention is as follows:

$$e_k = f(T, S_k) \tag{9}$$

$$\omega_k = \frac{\exp(e_k)}{\sum_{i=1}^{K}\exp[f(T, S_i)]} \tag{10}$$

$$C = \sum_{k=1}^{n} \omega_k S_k \tag{11}$$

The specific process is shown in Figure 1, which involves entering into a set of state vectors $[S_1, S_2, \cdots, S_K]$ and a target vector T. The importance of the vector $S_k$is measured by the custom function $f(T, S_k)$, and the importance information is normalized by Formula (10) and encoded into the context vector C.

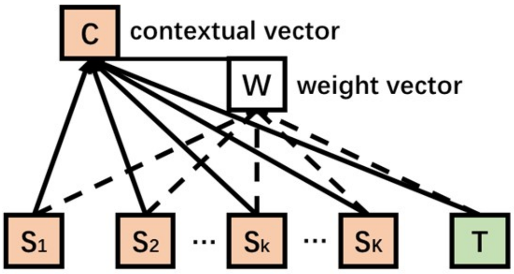

Figure 1. Soft attention.

4. Problem Formalization

4.1. Problem Description

As shown in Figure 2, suppose that in a task space containing fixed obstacles, both the UAV and the target are set to a circle with a radius of 3 and the obstacle to a circle with a radius of 5. There is a cluster consisting of N UAVs($N \geq 3$) flying at constant speed in a two-dimensional plane, and UAVs take emergency maneuvers to avoid obstacles when they encounter them. The target moves randomly in the environment and UAVs cannot obtain prior information about the target in advance. All UAVs are regarded as particles. In this paper, we consider the task of complete cooperation, and the rounding up UAVs form a closed circle through cooperation to complete the rounding up, and the location of the target is unknown to the UAVs before the rounding up.

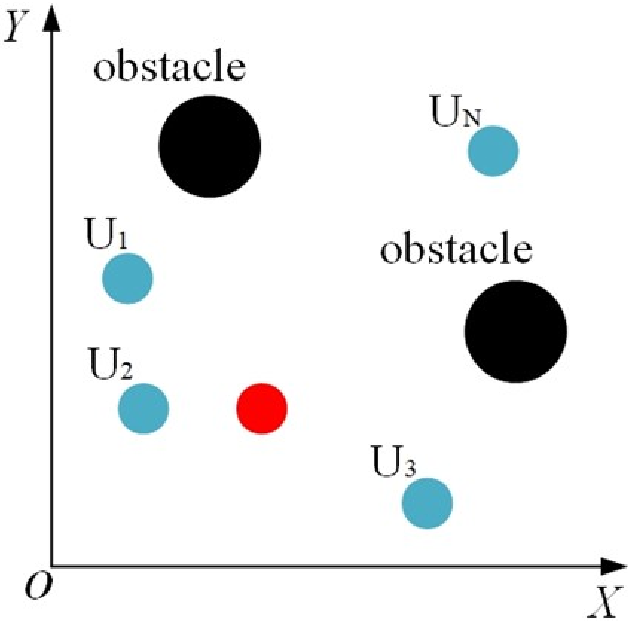

Figure 2. Roundup problem diagram.

The UAVs have certain detection ranges and the target has certain perception range, specifically manifested as a circular area of a certain radius with the center of the UAV or the target as the center. The specific radius is given in Parameter Setting. After the task begins, UAVs explore the location of the target, and when the UAV finds the target, it will immediately transmit the location information of the target to the other n UAVs in a cooperative manner. The UAV that performs the pursuit task gradually approaches the target and completes the location deployment in the process. If the UAVs are close to the target, they will be found by the target, and then the target will escape from the UAVs with a certain escape strategy.

As shown in Figure 3, when three or more UAVs set up an encircling circle around the target, and the maximum distance between the surrounded UAVs and the target is less than the detection range of the target, it is regarded as a successful encircling.

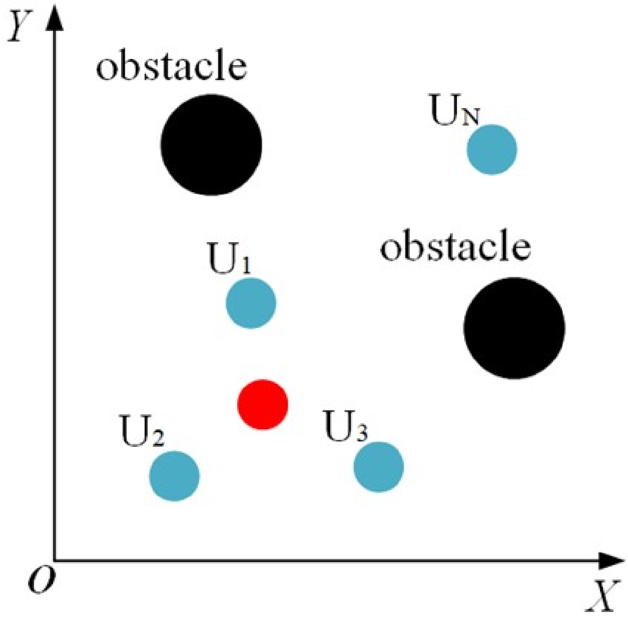

Figure 3. Roundup success diagram.

In order to improve the adaptability of UAVs to the environment, it is necessary to comprehensively consider the constraints of the task environment and consider the situation that the UAV exceeds the task boundary. Three typical task scenarios are shown in Figure 4.

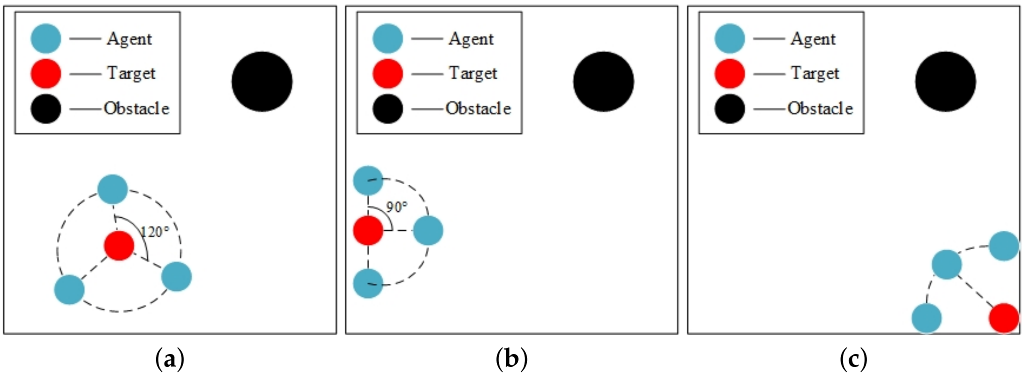

Figure 4. Three typical roundup scenarios. (**a**) The scene when the target is in an arbitrary area, (**b**) the scene when the target is at the edge of the area, and (**c**) the scene with the target in the corner.

Figure 4 shows that when the target is in any position, UAVs will form a specific rounding up formation to ensure that the target cannot escape in any direction. When the target is at the edge of the task area, UAVs only need to ensure that the target cannot escape within the direction of 180°. When the target is in the corner of the task area, UAVs only need to form a rounding up formation in the 90° direction around the target to achieve the successful hunting of the target. Through the consideration of the above three cases, although the difficulty of solving is increased, the problem can be made more in line with the needs of the actual situation and increase the authenticity of the task environment.

In addition, an obstacle is also involved in the task process of target roundup. As shown in Figure 5, when the UAVs and the obstacles together form an encirclement of the target, it can also be judged as the completion of the target roundup task.

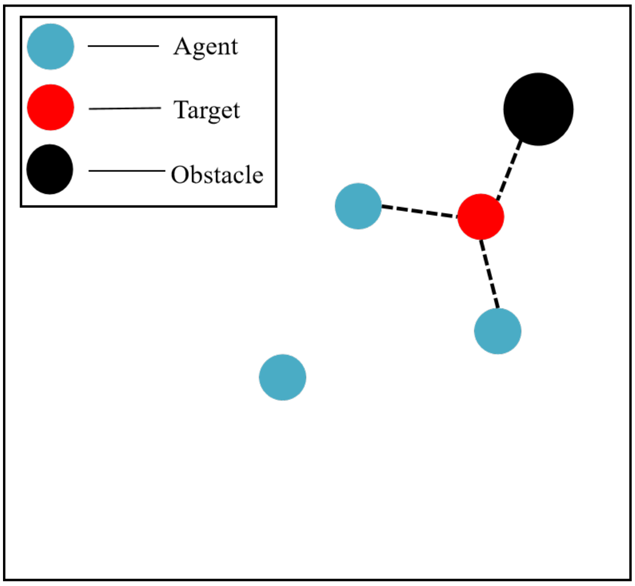

Figure 5. The scene when obstacles are involved in the roundup process.

4.2. Models

Based on the above overall description of the problem, we establish the relevant model as follows:

Kinematic model. In our paper, the UAV is regarded as a particle, and the position vector of the UAV at time t is assumed to be $\left[x_i^t, y_i^t\right]^T$. The motion model of the UAV is described as follows:

$$\begin{cases} \dot{x}_i^t = v \cos \theta_i^t \\ \dot{y}_i^t = v \sin \theta_i^t \end{cases} \tag{12}$$

where v is the movement speed of the UAV, $\dot{x}_i^t$ is the horizontal component of UAV i's speed at the time of t and $\dot{y}_i^t$ is the vertical component of UAV i's speed at the time of t. In this paper, angles of attack and sideslip are ignored, and θ_i^t denotes the heading angle of UAV i at the time of t.

After time Δt, the position and heading angle of UAV i are updated by the following formula:

$$\begin{cases} x_i^{t+1} = x_i^t + \dot{x}_i^t \cdot \Delta t, \ 0 \leq x_i^t \leq x_{\max} \\ y_i^{t+1} = y_i^t + \dot{y}_i^t \cdot \Delta t, \ 0 \leq y_i^t \leq y_{\max} \\ \theta_i^{t+1} = \theta_i^t + a_i^t \cdot \Delta t, \ -\dot{\theta}_{\max} \leq a_i^t \leq \dot{\theta}_{\max} \end{cases} \tag{13}$$

Here, $(x_{\max}, y_{\max})$ represents the coordinate boundaries of the task area, a_i^t is the action of UAV i and $\dot{\theta}_{\max}$ is the maximum heading angular velocity of UAV i.

Similarly, for the target, the kinematic model is defined by the position and heading angle, denoted as $\left[x_j, y_j, \theta_j\right]$, but the heading angle of the target is a bounded random variable.

4.3. Reward Shaping

In order to make the reward function more realistic, in this paper, we fully consider the conditions of successful roundup of the target, not only to ensure that the target is within the detection range of UAVs, and the UAVs and the target form a certain angle of roundup, but also to ensure that UAVs are outside the detection range of the target, to avoid the target escape phenomenon. In addition, UAVs need to avoid obstacles and leave the task area to ensure the safe execution of the task. Specifically, the following reward function is designed to guide the learning process.

Rounding up the target. The UAVs and the target are in the process of dynamic movement, and the distance between the UAVs and the target d_{ij} is used to determine the rounding up situation. Here, UAVs have a certain detection distance D_a, and only when the distance between the UAVs and the target is less than D_a can the UAVs detect the target and round up.

However, the target also has a certain sensing distance D_t. When the distance between the UAVs and the target is less than D_t, the target will sense the UAVs and escape. Therefore, the best distance of the UAVs should be kept between D_a and D_t; too long or too short a distance will cause some reward loss. The reward function based on distance is set as follows:

$$r_d = \begin{cases} -1/d_{ij},\ d_{ij} \le D_t \\ 0,\ D_t \le d_{ij} \le D_a \\ -\ d_{ij},\ D_a \le d_{ij} \end{cases} \tag{14}$$

In addition, UAVs should also meet the requirements of rounding up angles when rounding up targets. Considering the three kinds of rounding up angles designed in Figure 4, the rounding up reward function based on the angle is designed as follows [27]:

$$r_{ang} = -(\psi - \psi^*)^2 - (\varphi - \varphi^*)^2 \tag{15}$$

where ψ and φ represent the angles between the target and two adjacent UAVs, respectively, and ψ^* and φ^* represent the optimal angle. In Figure 4a, $\psi^* = \varphi^* = 2 * \pi / N$, in Figure 4b, $\psi^* = \varphi^* = \pi / N$ and in Figure 4c, $\psi^* = \varphi^* = \pi / (2 * N)$.

Obstacle Punishment. In order to avoid collision between UAVs and obstacles, a reward function regarding obstacle avoidance is designed, and the formula is expressed as follows:

$$r_{ob} = -1/d^{io}_{\min} \tag{16}$$

where $d^{io}_{\min}$ represents the shortest straight-line distance between the UAV and the obstacle, $d^{io}_{\min} = \sqrt{(x_i - x_o)^2 + (y_i - y_o)^2}$. (x_i, y_i) represents the position of UAV i. (x_o, y_o) represents the position of the obstacle.

Boundary Punishment. When UAVs move outside the boundary, there will be some security risks. Therefore, constraining UAVs to move in the task area contributes to the movement safety of UAVs. The shortest distance to the boundary is defined as $d^{ib}_{\min}$, and the reward function based on the boundary punishment is expressed as follows:

$$r_{bou} = \begin{cases} -0.5\left(d_a - d^{ib}_{\min}\right)/d_a,\ d^{ib}_{\min} < d_a \\ 0,\ else \end{cases} \tag{17}$$

The judgment mark of successful roundup is set, and the formula is expressed as follows:

$$flag = \begin{cases} 1,\ success \\ 0,\ fail \end{cases} \tag{18}$$

In summary, the reward function of UAV swarm target rounding up is expressed as follows:

$$r = \alpha \cdot r_d + \beta \cdot r_{ang} + \gamma \cdot r_{ob} + \delta \cdot r_{bou} + flag \cdot 10 \tag{19}$$

5. Method

In this section, we propose a novel algorithm, goal consistency with multi-head soft attention (GCMSA), to facilitate efficient target rounding up by UAV swarms. Based on the actor–critic framework, we adopt the training mode of Centralized Training Distributed Execution (CTDE). In the local perspective, the multi-head soft attention (MSA) module is used to complete the cognitive process between each UAV and the target, and then the important target information is determined. In the global perspective, the critic will use the information of all UAVs, complete the information cognitive process between UAVs, and output the observation–action value function.

5.1. Strategy Formation Process from Local Perspective

The overall model structure is shown in Figure 6. Specifically, in order to obtain an effective target exploration strategy, the method transmits the input observation information to the MSA module to filter the target information, and then updates the strategy through the MIP module.

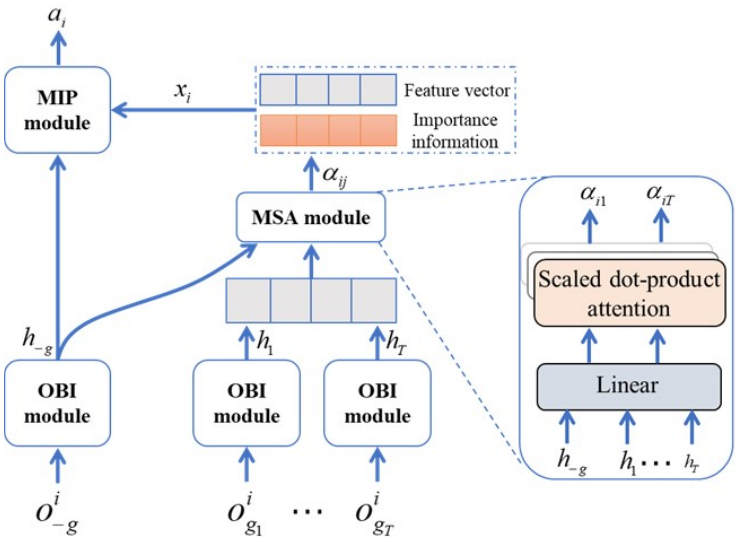

Figure 6. Policy network framework from local perspective.

Observation inputting (OBI) module. The local observations of UAVs are obtained, and the observation information is a vector represented and encoded as a low-level cognitive vector h_i.

Multi-head soft attention (MSA) module. In order to obtain effective target information, the MSA module is introduced to achieve efficient target selection by considering the importance of the target.

Mix policy (MIP) module. The decision module is used to estimate the state value, which is helpful for the policy update of UAVs.

5.2. The Specifics of GCMSA

5.2.1. Specific Operation Process

We set UAV i's observation information, represented as $o^i = \left\{ o^i_{-g}, o^i_{g_1}, \cdots, o^i_{g_T} \right\}$. Here, $o^i_{g_j}$ represents UAV i's observation information about goal g_j. o^i_{-g} is all non-target observation information of UAV i, such as environment information, etc. The observation information of UAV i is passed into the OBI module with ReLU activation function f_{tr}, which is transformed into cognitive feature vector h_j, calculated as $h_j = f_{tr}\left(o^i_{g_j}\right)$. UAV i's non-target observation information o^i_{-g}'s cognition of the characteristic vector is represented as h_{-g}. Then, the generated cognitive feature vectors h_j and h_{-g} are passed into the MSA module based on the multi-head soft attention mechanism to identify and filter out the important information received by UAV i. The importance of UAV j for UAV i can be expressed as

$$\alpha_{ij} = \frac{\exp\left[\text{attention}\left(wh_{-g}, wh_j\right)\right]}{\sum_{j\in T} \exp\left[\text{attention}\left(wh_{-g}, wh_j\right)\right]} \tag{20}$$

After k attentions, the vector generated in the last layer is

$$h_j' = \sigma\left(\frac{1}{K}\sum_{k=1}^{K} \sum\nolimits_{j\in T} \alpha_{ij}^k w^k h_j\right) \tag{21}$$

Through the MSA module process, the UAVs will pay more attention to the information with higher importance and choose actions according to the feature vector and their own environment observation information:

$$a_i = f_i(h_{-g}, x_i) \tag{22}$$

where $x_i = \sum\limits_{j=1}^{T} \alpha_{ij} h_j'$. f_i indicates the fully connected layer. In this way, the UAV learns to dynamically attend to observable target information at each stage and take actions based on its local view of the cognition of the target.

5.2.2. Centralized Training Process in Global Perspective

During centralized training, the critic uses information from all UAVs, learns a global target cognition, and evaluates the value of observation–action pairs. In the global perspective, the softmax function completes the information cognition between UAVs, where the characteristics of the global non-target information are expressed as $h^{-g}_{global} = f_i\left(o^i_{-g}, o^{-i}, a\right)$, where o^{-i} represents the observation without UAV i and a indicates the joint action. In the global perspective, the importance of target j to UAV i can be expressed as

$$\alpha^{ij}_{global} = \frac{\exp\left(h^{-g}_{global} h^{j}_{global}\right)}{\sum_{j=1}^{T} \exp\left(h^{-g}_{global} h^{j}_{global}\right)} \tag{23}$$

Therefore, the observation–action value function of UAV i can be expressed as follows:

$$Q^i(o, a; \omega_i) = f_i\left(h^i_{global}, x^i_{global}\right) \tag{24}$$

where h^i_{global} represents the global cognitive feature vector of UAV i, $x^i_{global} = \sum_{j=1}^{T} \alpha^{ij}_{global} h^j_{global}$, and ω_i is a parameter of the critic network. This method can better interact with the environment and choose a better rounding up strategy.

5.3. *Training of GCMSA*

The training process is shown in Figure 7. Similar to the actor–critic framework, UAV i observes state o^t_i at each time step t, and according to the policy network π_{θ_i} chooses an action a^t_i. After executing a^t_i, UAV i can observe a new observation o'^t_i and obtain a reward of r^t_i. Then, an experience tuple $\left(o_t, a_t, o'_t, r_t\right)$ is stored in experience replay buffer D. During training, the experience tuple is sampled from the UAV's experience replay buffer and the critic network is updated by minimizing a loss function.

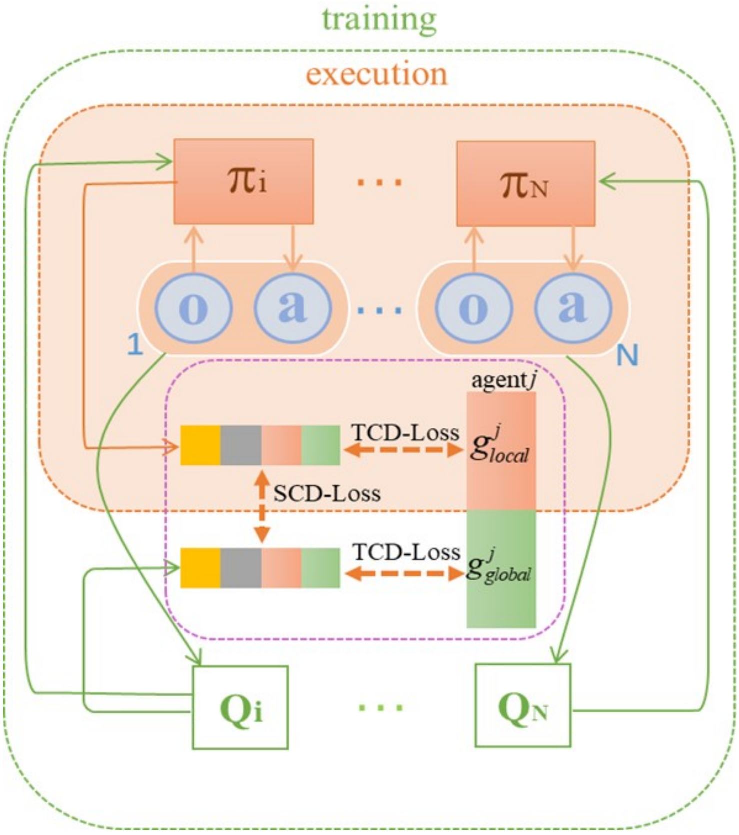

Figure 7. Training method process diagram.

Ideally, when all UAVs achieve cognitive consistency at time t, there exists conditional probability distribution $p(s_t|g_t)$ between the global state s_t and the real goal g_t. At this point, the probability distribution $q\left(g^i_{t,local}|o^i_t\right)$ of UAV i's local cognition $g^i_{t,local}$ of the target j needs to be consistent with $p(s_t|g_t)$. As a commonly used method to compare the relationship between two distributions, KL divergence has obvious advantages in judging the similarity between two distributions. Therefore, in this paper, the consistency between objectives is achieved by minimizing the KL divergence between distributions:

$$\min KL\left[q\left(g^i_{t,local}|o^i_t\right) \parallel p(s_t|g_t)\right] \tag{25}$$

5.3.1. Critic Network Training Process in Global Perspective

As shown in Figure 5, during the training of the global critic network, we update the loss network by minimizing a linear combination of temporal difference loss (TD-loss) and the team cognitive dissonance loss (TCD-loss) from a global perspective. In the calculation of TCD-loss, as shown in Equation (25), the true probability distribution cannot be obtained directly. In order to solve this problem, drawing on the idea of paper [28], in the global perspective, we use the combination of all UAVs' target observation $o_t = \left\{o^1_t, \cdots, o^1_t\right\}$ and joint action $a_t = \left\{a^1_t, \cdots, a^N_t\right\}$ to generate the probability distribution $p\left(g^i_{t,global}|o_t, a_t\right)$. Then, the real target probability distribution $p(s_t|g_t)$ can be approximated. Therefore, in the global perspective, TDD-loss among UAVs can be expressed as follows:

$$L_{tcd}(\omega_i) = \sum_{k\neq i} KL\left[q\left(g^i_{global}|o, a; \omega_i\right) \parallel p\left(g^k_{global}|o, a; \omega_k\right)\right] \tag{26}$$

where ω represents network parameters. The team cognitive dissonance loss (TCD-loss) can be expressed as follows:

$$L_{td}(\omega_i) = \mathbb{E}_{o,a,r_i,\hat{o}\sim D}\left[(Q_i(o, a; \omega_i) - y_i)^2\right] \tag{27}$$

where $y_i = r_i + \gamma Q_i(\hat{o}, \hat{a}; \hat{\omega}_i)|_{\hat{a}_j = \pi(\hat{o}_j; \hat{\theta}_j)}$. Combining Equations (26) and (27), the update formula of the critic network can be expressed as

$$L_{total}(\omega_i) = \alpha L_{tcd}(\omega_i) + L_{td}(\omega_i) \tag{28}$$

where α represents the cognitive ability between UAVs.

5.3.2. Actor Training Process in Local Perspective

For the update of the actor network, in addition to the policy gradient in the traditional MADDPG algorithm, the self cognitive dissonance loss (SCD-loss) and the TCD-loss between UAVs from a local perspective are also considered.

Among them, the policy gradient loss of SCD-loss can be expressed as follows:

$$J_{scd}(\theta_i) = KL\left[q\left(g^i_{local}|o_i; \theta_i\right) \parallel p\left(g^i_{global}|o, a; \theta_i\right)\right] \tag{29}$$

where θ_i represents the actor network parameter of UAV i.

Similarly, the policy gradient loss of TCD-loss can be expressed as

$$J_{tcd}(\theta_i) = \sum_{k\neq i} KL\left[q\left(g^i_{local}|o, a; \theta_i\right) \parallel p\left(g^k_{local}|o, a; \theta_k\right)\right] \tag{30}$$

Finally, the policy gradient of the expected return of UAVi can be expressed as

$$\nabla_{\theta_i} J(\theta_i) = \mathbb{E}_{o,a\sim D}\left[\nabla_{\theta_i}\pi_i(o_i; \theta_i)\nabla_{\theta_i} Q_i(o, a; \omega_i)|_{a_i=\pi_i(o_i;\theta_i)}\right] \tag{31}$$

Combining Equations (29)–(31), the update of the actor network in the local perspective can be expressed as

$$\nabla_{\theta_i} J_{total}(\theta_i) = \nabla_{\theta_i} J(\theta_i) + \lambda J_{scd}(\theta_i) + \beta J_{tcd}(\theta_i) \tag{32}$$

6. Experiment

In this section, we evaluate the proposed GCMSA algorithm in the established UAV swarm collaborative target rounding model and compare it with the widely used baseline algorithm. The goal of the experiment is to answer three questions: (A) Can the GCMSA algorithm improve target rounding? (B) How stable and scalable is the learned policy? (C) What are the advantages of the proposed algorithm in this paper compared to other benchmark algorithms?

6.1. Parameters Setting

In this study, we develop a simulation environment for a two-dimensional target rounding task based on a Multi-agent Particle Environment (MPE), and the configuration of the environment parameters is shown in Table 1. The initial positions of the UAVs, targets, and obstacles are randomly reset before the start of each episode, and the number of each indicator and the size of the map can be changed.

Table 1. Environment parameter settings.

Entity	Variable	Value
Environment	Size	2.5 km × 2.5 km
	Scenario boundary length	2.5 km
	Obstacles number	1
UAVs	Total number	3
	Detection range (D_a)	10 m
	Speed (v_i)	5 m/s
Target	Total number	1
	Perception range (D_t)	5 m
	Speed (v)	5.5 m/s

Then, we test the performance of the proposed GCMSA algorithm in a three-UAV roundup one-target scenario. The hyperparameters of the algorithm are shown in Table 2.

Table 2. Hyperparameter configurations.

Hyperparameter	Value
Max episode	2000
Max episode length	200
Replay buffer	5000
Discount factor	0.95
Learning rate	0.001
Weight parameter α	0.5
Weight parameter λ	0.2
Weight parameter β	0.8

6.2. Algorithm Effectiveness

First, we intercept the position information of the UAVs and target at different task steps. Figure 8 depicts the detailed process of rounding up; in the figure, the blue circle indicates the UAV, the red circle indicates the target and the black circle indicates the obstacle. At the initial moment, the UAVs and the target are randomly distributed in the task space, and as the time step advances continuously, the UAVs keep moving towards the target, while the target moves away from the UAVs. At time step = 40 and step = 60, the UAVs search for the target and achieve task information consistency with neighbouring UAVs. At the end, when step = 78, the three UAVs successfully complete the rounding up of the target and, in the process, no collision problem occur.

In order to verify the effectiveness of the algorithm in different task scenarios, we visualize the task execution trajectories when the target is at the edge and corner. As shown in Figures 9 and 10, in both cases, the UAVs successfully complete the rounding up task and successfully bypass the obstacles. This is because we set the reward function based on the angle when the target is in different positions, which ensures the efficient and successful completion of the task.

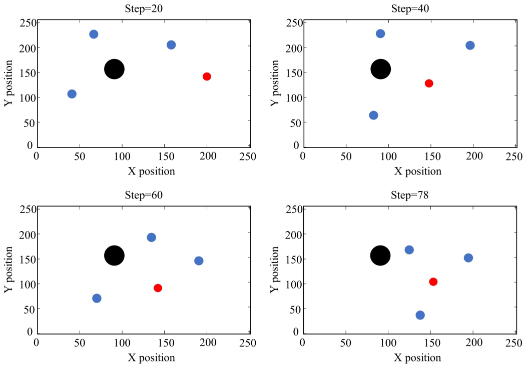

Figure 8. Detailed view of the roundup.

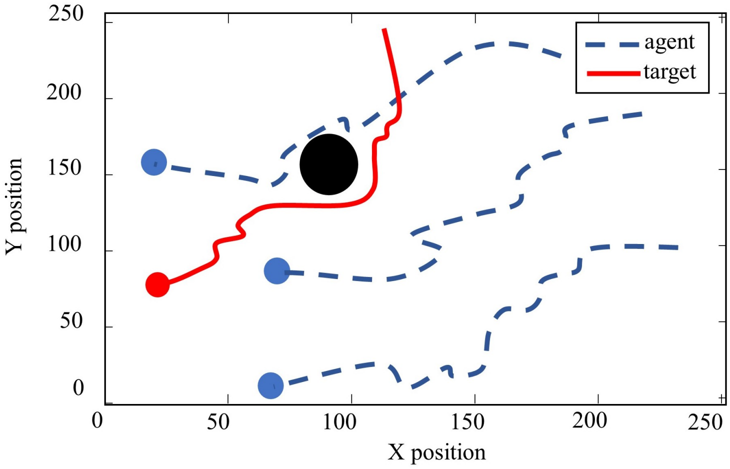

Figure 9. Detailed view of the roundup.

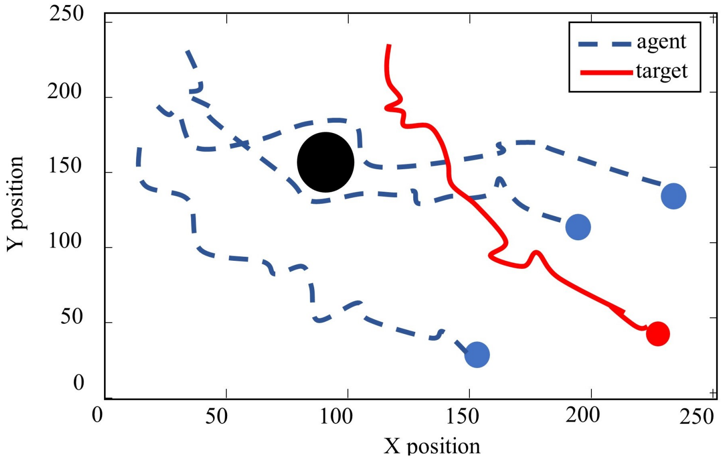

Figure 10. Detailed view of the roundup.

To further explore the convergence effect of the designed network, we plot the loss curves of each UAV, and the result is shown in Figure 11. It can be seen that, at the initial moment, the network loss is large for all three UAVs, which decreases rapidly with training episodes and reaches convergence in all of them at around 900 episodes, and the convergence process is more stable. This shows that the network designed in this paper has a good learning effect.

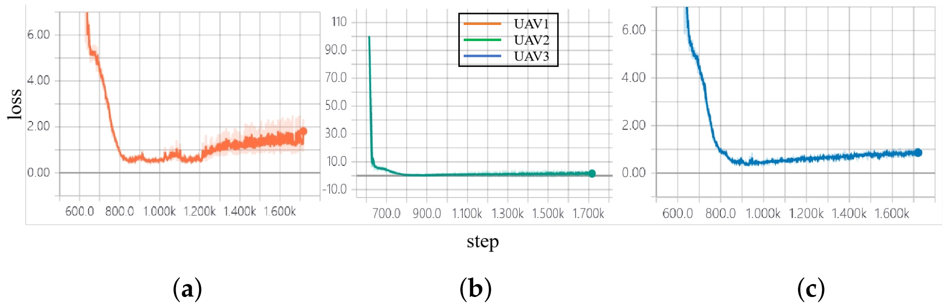

Figure 11. Schematic diagram of the training process. (**a**) Training process for UAV1, (**b**) training process for UAV2, and (**c**) training process for UAV3.

6.3. Algorithm Performance Comparison

In this section, we further validate the advantages of GCMSA by comparing the performance of GCMSA with the benchmark algorithm MADDPG in performing the tasks. In the interest of fairness, each algorithm was run using five random seeds and the results are shown with 95% confidence intervals.

Firstly, we plot the reward curves of the UAVs during training. The horizontal coordinate of the curve is the number of training episodes, and the vertical coordinate is the reward value, specifically the sum of the reward values of the episodes obtained by the three UAVs. As shown in Figure 12, it can be seen that in the initial stage, the UAVs are less cognitive about the environment, resulting in a lower reward function value. With the increasing number of episodes, the UAVs keep exploring, the reward value increases, and finally, the GCMSA algorithm is able to converge at a higher level. This shows that compared with MADDPG, our algorithm is able to obtain a more reasonable fencing strategy.

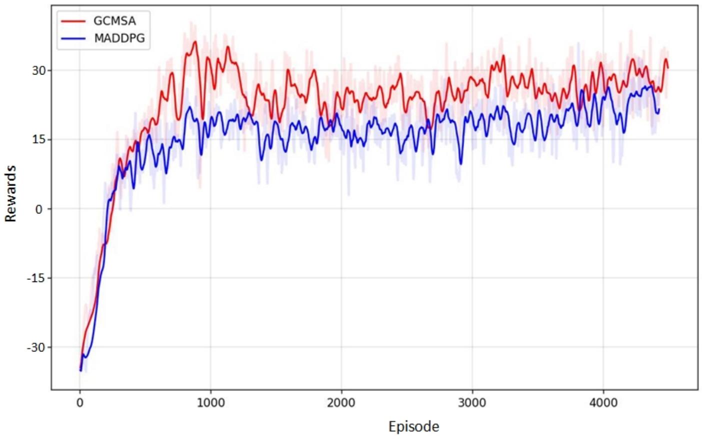

Figure 12. Total reward curve for rounding up.

Then, we compare the performance of different algorithms in terms of rounding success rate. As shown in Figure 13, blue and yellow colours indicate the rounding success rates of the GCMSA algorithm and the MADDPG algorithm, respectively. The rounding success rate of the GCMSA algorithm can be maintained at around 75% after 3000 episodes, compared to the MADDPG algorithm, whose rounding success rate only converges to around 45%. The reason for this is that the algorithm proposed in this paper realizes the information sharing between the UAVs, so that each UAV obtains more information about the task, which in turn enables it to explore a more reasonable strategy and improve the success rate of rounding up.

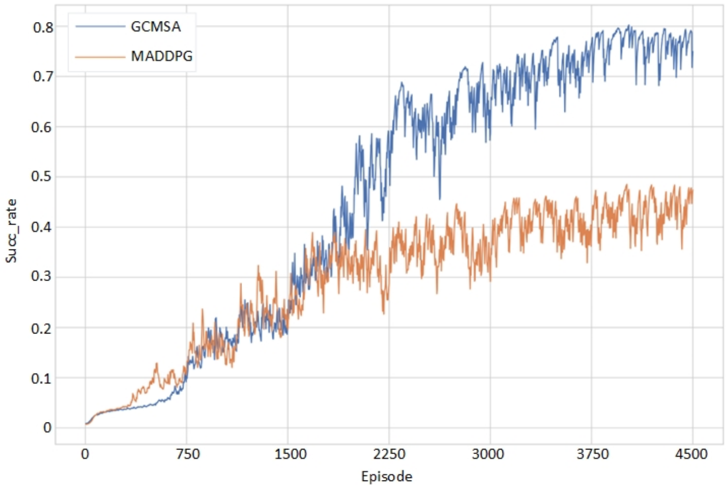

Figure 13. Comparative curve of roundup success rate.

6.4. Scalability Test

In order to further test the scalability of the proposed GCMSA algorithm, simulation experiments were conducted in a larger scale mission scenario. We consider three scenarios of three UAVs rounding up one target, four UAVs rounding up one target, six UAVs rounding up two targets, and eight UAVs rounding up two targets, respectively. In each scenario, the learned policy was executed for 100 rounds. The statistical metrics of the test results are shown in Table 3.

Table 3. Statistical indicators of execution results in different scenarios.

Mapsize (km)	Scenarios	Indicators	GCMSA	MADDPG
2.5 × 2.5	3 UAVs and 1 target	Mean reward Rounding up success rate	**28.47** **0.78**	27.52 0.41
2.5 × 2.5	4 UAVs and 1 target	Mean reward Rounding up success rate	**27.83** **0.72**	26.54 0.39
4.0 × 4.0	6 UAVs and 2 targets	Mean reward Rounding up success rate	**32.67** **0.73**	29.24 0.40
4.0 × 4.0	8 UAVs and 2 targets	Mean reward Rounding up success rate	**30.86** **0.69**	27.14 0.37

It can be seen from Table 3 that the proposed GCMSA algorithm also exhibits the highest average reward and the highest task success rate in all four task scenarios. Therefore, the GCMSA algorithm outperforms the baseline in all four scenarios. This result validates the idea that the GCMSA algorithm is able to improve the cognitive abilities of individuals in the cluster and enables scalable learning of the CTDE paradigm in the form of combining importance information. It is worth noting that the mission success rate of rounding up a single target using four UAVs is lower than rounding up a single target using three UAVs. The reason for this is that the more UAVs that are involved in the rounding up task, the greater the increase in complexity of the network, which may lead to a reduction in the convergence speed; in addition, when the size of the UAVs increases, the risk of collision between the individuals increases, and the UAVs will be distracted from obstacle avoidance. Therefore, it can be seen that a balance between the number of UAVs and targets should be achieved when deploying the task.

From the above analysis, we can conclude that the GCMSA algorithm has better scalability when the number of UAVs and the joint action space increases.

7. Conclusions

In this paper, we study the multi-UAV cooperative roundup mission in a complex environment using the MADRL method. In order to fully consider the reality of the mission environment, three roundup positions and a target escape strategy are introduced. We propose the GCMSA algorithm to promote cooperation between UAVs, and use the multi-head soft attention module to filter out the target information with higher importance, which in turn promotes target cognition between UAVs and achieves efficient cooperation between UAVs.

Simulation experiments show that we successfully complete the roundup task with the proposed GCMSA algorithm. Compared with the baseline algorithm MADDPG, GCMSA has obvious advantages in terms of task success rate and convergence.

Author Contributions: Conceptualization, investigation, methodology, writing—original draft preparation, resources, software, visualization, and validation, Z.W.; formal analysis and writing—review and editing, R.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data are unavailable.

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

References

1. Zhu, C.; Dastani, M.; Wang, S. A survey of multi-agent deep reinforcement learning with communication. In Proceedings of the AAMAS '24: 23rd International Conference on Autonomous Agents and Multiagent Systems, Auckland, New Zealand 6–10 May 2024; Volume 38. [CrossRef]
2. Xu, Q.; Geng, H.; Chen, S.; Yuan, B.; Zhuo, C.; Kang, Y.; Wen, X. GoodFloorplan: Graph Convolutional Network and Reinforcement Learning-Based Floorplanning. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2022**, *41*, 3492–3502. [CrossRef]
3. Huang, T.; Xuebo Cen, W.X. Self-Organizing collaborative capture of swarm robot systems based on loose preference rules. *IEEE/CAA J. Autom. Sin.* **2013**, *39*, 57–68. [CrossRef]
4. Muro, C.; Escobedo, R.; Escobedo, S.; Coppinger, R. Wolf-pack (*Canis lupus*) hunting strategies emerge from simple rules in computational simulations. *Behav. Processes* **2011**, *88*, 192–197. [CrossRef] [PubMed]
5. Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; Mordatch, I. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. *arXiv* **2017**, arXiv:1706.02275.
6. Tang, W.; Gao, M.; Gao, Q.; Peng, X. Distributed Training Decentralized Execution Framework of Multi-agent Learning for Cooperative Edge Caching. In Proceedings of the 2023 5th International Conference on Frontiers Technology of Information and Computer (ICFTIC), Qiangdao, China, 17–19 November 2023; pp. 993–998. [CrossRef]
7. Olfati-Saber, R. Flocking for Multi-agent Dynamic Systems: Algorithms and Theory. *IEEE Trans. Autom. Control* **2006**, *51*, 401–420. [CrossRef]
8. Liu, Y.; Yu, H. Space Target Hunting Method of Satellite Cluster based on Wolf Swarm Optimization. *J. Beijing Univ. Aeronaut. Astronaut.* **2023**, *48*.
9. Wan, R.; Wang, X.; Ma, Z. UAV route planning based on improved whale optimization algorithm and dynamic artificial potential field method. In Proceedings of the 2023 6th International Symposium on Autonomous Systems (ISAS), Nanjing, China, 23–25 June 2023; pp. 1–8. [CrossRef]
10. Tong, B.; Liu, J.; Duan, H. Multi-UAV Interception Inspired by Harris Hawks Cooperative Hunting Behavior. In Proceedings of the 2021 IEEE International Conference on Robotics and Biomimetics (ROBIO), Sanya, China, 27–31 December 2021; pp. 1656–1661. [CrossRef]
11. Fan, Z.; Yang, H. Target Hunting Control for Multi-Agent Systems Based on Reinforcement Learning. *Chin. J. Aeronaut.* **2023**, *44*.
12. Zha, W.; Chen, J.; Peng, Z.; Gu, D. Construction of Barrier in a Fishing Game With Point Capture. *IEEE Trans. Cybern.* **2017**, *47*, 1409–1422. [CrossRef] [PubMed]
13. Chen, J.; Zha, W.; Peng, Z.; Gu, D. Multi-player pursuit-evasion games with one superior evader. *Automatica* **2016**, *71*, 24–32. [CrossRef]
14. Sukhbaatar, S.; Szlam, A.; Fergus, R. Learning Multiagent Communication with Backpropagation. *arXiv* **2016**, arXiv:1605.07736.
15. Peng, P.; Wen, Y.; Yang, Y.; Yuan, Q.; Tang, Z.; Long, H.; Wang, J. Multiagent Bidirectionally-Coordinated Nets: Emergence of Human-level Coordination in Learning to Play StarCraft Combat Games. *arXiv* **2017**, arXiv:1703.10069.
16. Singh, A.; Jain, T.; Sukhbaatar, S. Learning when to Communicate at Scale in Multiagent Cooperative and Competitive Tasks. *arXiv* **2018**, arXiv:1812.09755.
17. Ding, Z.; Huang, T.; Lu, Z. Learning Individually Inferred Communication for Multi-Agent Cooperation. *arXiv* **2020**, arXiv:2006.06455.
18. Shi, K.; Liu, J.; Wang, X.; Xie, L. Joint Optimization of Multi-UAV-Assisted Data Collection and Energy Replenishment via Transfer Learning aided Deep Reinforcement Learning. In Proceedings of the 2023 IEEE 23rd International Conference on Communication Technology (ICCT), Wuxi, China, 20–22 October 2023; pp. 967–972.
19. Rashid, T.; Farquhar, G.; Peng, B.; Whiteson, S. Weighted QMIX: Expanding Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. *arXiv* **2020**, arXiv:2006.10800.

20. Son, K.; Ahn, S.; Reyes, R.D.; Shin, J.; Yi, Y. QTRAN++: Improved Value Transformation for Cooperative Multi-Agent Reinforcement Learning. *arXiv* **2020**, arXiv:2006.12010.
21. Chhablani, C.; Kash, I.A. An Analysis of Connections Between Regret Minimization and Actor Critic Methods in Cooperative Settings. In Proceedings of the Adaptive Agents and Multi-Agent Systems, London, UK, 29 May–2 June 2023.
22. Liu, T.; Chen, H.; Hu, J.; Yang, Z.; Yu, B.; Du, X.; Miao, Y.; Chang, Y. Generalized multi-agent competitive reinforcement learning with differential augmentation. *Expert Syst. Appl.* **2024**, *238*, 121760. [CrossRef]
23. Peysakhovich, A.; Lerer, A. Prosocial learning agents solve generalized Stag Hunts better than selfish ones. *arXiv* **2017**, arXiv:1709.02865.
24. Hughes, E.; Leibo, J.Z.; Phillips, M.; Tuyls, K.; Duéñez-Guzmán, E.A.; Castañeda, A.G.; Dunning, I.; Zhu, T.; McKee, K.R.; Koster, R.; et al. Inequity aversion improves cooperation in intertemporal social dilemmas. In Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, QC, Canada, 3–8 December 2018.
25. Guicheng, S.; Yang, W. Review on Dec-POMDP Model for MARL Algorithms. In *Smart Communications, Intelligent Algorithms and Interactive Methods*; 2022.
26. Sumarudin, A.; Sutisna, N.; Syafalni, I.; Trilaksono, B.R.; Adiono, T. DQN Algorithm Design for Fast Efficient Shortest Path System. In Proceedings of the 2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Taipei, Taiwan, 31 October–3 November 2023; pp. 254–260. [CrossRef]
27. Jiang, L.; Wei, R.; Wang, D. UAVs rounding up inspired by communication multi-agent depth deterministic policy gradient. *Appl. Intell.* **2022**, *53*, 11474–11489. [CrossRef]
28. Chen, X.; Liu, X.; Zhang, S.; Ding, B.; Li, K. Goal Consistency: An Effective Multi-Agent Cooperative Method for Multistage Tasks. In Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence IJCAI-22, Vienna, Austria, 23–29 July 2022; pp. 172–178. [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.