applied sciences

MDPI

Article

Multi-Agent Collaborative Target Search Based on the Multi-Agent Deep Deterministic Policy Gradient with Emotional Intrinsic Motivation

Xiaoping Zhang [1,2], Yuanpeng Zheng [1,*], Li Wang [1], Arsen Abdulali [2] and Fumiya Iida [2]

[1] School of Electrical and Control Engineering, North China University of Technology, Beijing 100144, China; zhangxiaoping369@163.com (X.Z.); wangli939@ncut.edu.cn (L.W.)
[2] Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, UK; aa2335@cam.ac.uk (A.A.); fi224@cam.ac.uk (F.I.)
* Correspondence: zyp1416@163.com

Abstract: Multi-agent collaborative target search is one of the main challenges in the multi-agent field, and deep reinforcement learning (DRL) is a good way to learn such a task. However, DRL always faces the problem of sparse reward, which to some extent reduces its efficiency in task learning. Introducing intrinsic motivation has proved to be a useful way to make the sparse reward in DRL. So, based on the multi-agent deep deterministic policy gradient (MADDPG) structure, a new MADDPG algorithm with the emotional intrinsic motivation name MADDPG-E is proposed in this paper for the multi-agent collaborative target search. In MADDPG-E, a new emotional intrinsic motivation module with three emotions, joy, sadness, and fear, is designed. The three emotions are defined by corresponding psychological knowledge to the multi-agent embodied situations in an environment. An emotional steady-state variable function H is then designed to help judge the goodness of the emotions. Based on H, an emotion-based intrinsic reward function is finally proposed. With the designed emotional intrinsic motivation module, the multi-agent system always tries to make itself joy, which means it always learns to search the target. To show the effectiveness of the proposed MADDPG-E algorithm, two kinds of simulation experiments with a determined initial position and random initial position, respectively, are carried out, and comparisons are performed with MADDPG as well as MADDPG-ICM (MADDPG with an intrinsic curiosity module). The results show that with the designed emotional intrinsic motivation module, MADDPG-E has a higher learning speed and better learning stability, and the advantage is more obvious when facing complex situations.

Keywords: multi-agent collaboration; intrinsic motivation; MADDPG; emotion; deep reinforcement learning

Citation: Zhang, X.; Zheng, Y.; Wang, L.; Abdulali, A.; Iida, F. Multi-Agent Collaborative Target Search Based on the Multi-Agent Deep Deterministic Policy Gradient with Emotional Intrinsic Motivation. *Appl. Sci.* **2023**, *13*, 11951. https://doi.org/10.3390/app132111951

Academic Editors: Andrea Omicini and Stefano Mariani

Received: 12 September 2023
Revised: 14 October 2023
Accepted: 18 October 2023
Published: 1 November 2023

1. Introduction

A multi-agent system is composed of multiple agents. Through communication, cooperation, or competition between agents, the multi-agent system can complete a large number of complex tasks that cannot be completed by a single agent [1]. Multi-agent collaborative control, with its advantages of high efficiency, high fault tolerance, and inherent parallelism [2], has been widely used in formation [3], unmanned systems [4], network resource allocation [5], multi-robot cooperative motion planning [6], target search [7], and other fields.

Among the applications described above, target search has attracted wide attention because of its wide application scenarios and practicability. Most of the early studies focused on single-agent target search in a static environment, and the search methods usually used random search or rule search, which greatly reduced the search efficiency when faced with dynamic environments and dynamic targets. In recent years, target search became an application direction of swarm intelligence technology, which can be used for

search and rescue, environmental detection, warehouse handling, target roundup, etc. [8]. In the above application scenarios, the single agent can no longer search the target well, so researchers gradually pay attention to multi-agent collaborative target search. A wrong strategy will reduce the efficiency of the agent search, so the cooperative strategy between multi-agents is the key of the multi-agent collaborative target search. Hazra et al. [9] introduced the Shapley function and fuzzy Shapley function to facilitate target search in a two-dimensional region with time constraints by minimizing the mission time and fuel usage. Cooper et al. [10] presented a method for analyzing the upper bound for time to find a target under the potential field guidance algorithm assuming a radially expanding search area. Tang et al. [11] proposed an adaptive robotic bat algorithm (ARBA) for multi-robot target searching in unknown environments. The obstacle avoidance problem and the mechanism of jumping out of the local optimum are considered. The idea of a method of cooperative strategy between multi-agents is to transform it into a dynamic programming problem, such as Lion Swarm Optimization (LSO) [12], the greedy algorithm [13], model predictive control [14], and other methods, to learn the optimal strategy of agents. Traditional methods usually require complex mathematical calculations to solve multi-agent cooperative target search problems and are prone to easily falling into the local optimum. With the great success of deep learning in the field of artificial intelligence, more and more researchers use the method of deep reinforcement learning (DRL) to find the optimal search strategy.

Deep reinforcement learning has been a hot topic in multi-agent target search in recent years [7,15–18]. DRL combines the perception ability of deep learning and the decision-making ability of reinforcement learning [19], and it provides a solution for the perception and decision-making problems of complex multi-agent systems. In 2017, Tampuu et al. [20] first extended the Deep Q-Learning framework in multi-agent environments between two learning agents to play the the Atari Pong game. The result indicates that DQN can be extended for the learning of multi-agent systems. Lowe et al. [21] proposed the multi-agent deep deterministic policy gradient (MADDPG), which effectively solved the problem of a non-stationary environment, and achieved good results in various environments, such as cooperative, competitive, and mixed. In 2018, a new multi-agent policy gradient algorithm [22] was proposed, which solved the high variance gradient estimation problem and could be used to imitate complex behaviors in high-dimensional environments with multiple cooperative or competing agents. Compared with other algorithms, the MADDPG algorithm can be applied to multiple task scenarios such as competition and cooperation between multiple agents. Meanwhile, it can use the observation information from other agents for centralized training, so as to improve the efficiency of the algorithm. Deep reinforcement learning adopts an end-to-end strategy, which is more targeted than traditional methods, but for the problem of sparse rewards in multi-agent target search scenarios, the algorithm stability is still poor.

In the multi-agent target search process based on deep reinforcement learning, however, directly using sparse reward samples will cause neural network training to diverge or even fail to improve the strategy. A straightforward approach to address this problem is to use artificially designed dense rewards. However, such a method has certain limitations, for example, the convergence of the agent's strategy is easy to fall into a local optimum, which has a negative impact on the agent's learning [23]. Another way to make the sparse reward is adding goals, uncertainty measures, or intrinsic motivation inside the deep reinforcement learning exploration. Compared with adding the goal and uncertainty measures, the deep reinforcement learning method based on intrinsic motivation [7,24–26] formalizes a variety of heuristic concepts derived from cognitive psychology into intrinsic reward signals to drive the agent to independently and efficiently explore the environment. On the other hand, the internal reward system and motivation are believed to act to differentiate an intelligent being from an unintelligent one [27]. Intrinsic motivation can combine with deep reinforcement learning methods based on value functions or policy gradients [7,15] to form a strong heuristic exploration strategy. As early as 2004, Barto et al. [28] studied the

use of sophisticated reinforcement learning techniques on a simple novelty-based intrinsic motivation system. Inspired by their work, in [29], thinking that the critic in reinforcement learning can be part of the agent itself [30], Oudeyer et al. presented an intrinsic motivation system named Intelligent Adaptive Curiosity (IAC), which tended to push a robot toward situations in which it maximized its learning progress, and pointed out that any existing reinforcement learning technique could be associated with the IAC drive. Pathak et al. [31] proposed an intrinsic curiosity module (ICM) and formulated curiosity as the error in an agent's ability to predict the consequence of its own actions in a visual feature space learned by a self-supervised inverse dynamics model.

Another important intrinsic motivation is emotion. From the perspective of the embodiment view of the mind, it is assumed that cognition is situated [32], and as an important factor of cognition, emotion is also embodied. A mental representation of emotions emerges during the interaction of the agent's body state, as well as its awareness of the stimulus from the environment in which that state is observed [33]. Furthermore, it is believed that emotional responses are characterized by changes in the body state, e.g., behavior [34]. During the agents' sensorimotor learning in tasks, they not only observe the current state of the environment but also experience their emotion changes, which further affects the learning process. Researchers in the fields of neuroscience and psychology also demonstrated that emotion is an important part of decision making [35], and both positive and negative emotions can lead to changes in learning motivation. Feldmaier et al. [36] proposed a framework to incorporate an emotional model into the decision-making process of a machine learning agent and used a hierarchical structure to combine reinforcement learning with a dimensional emotional model. Fang et al. [37] proposed an algorithm of pursuit task allocation based on an emotional contagion to study the interaction between affective robots in multi-agent cooperative systems. Guzzi et al. [38] proposed a model for adaptation and implicit coordination in multi-robot systems based on the definition of artificial emotions. Emotions are defined as the robot's situation, and emotions are classified as neutral, fear, frustration, urgency, and confusion. Achiam et al. [39] used the surprise emotion as intrinsic motivation and enabled the agent to succeed in a wide range of environments with high-dimensional state spaces and very sparse rewards. Loyola et al. [25] used the boredom emotion as an intrinsic motivation to generate routes in scenarios where rewards were absent and to facilitate the robot navigation toward the goals. However, in most of these works, emotional intrinsic motivation is applied to a single-agent learning task and is rarely applied to multi-agent collaborative target search.

In this work, considering the multi-agent collaborative target search problem, and based on the MADDPG algorithm [21], firstly, an emotion intrinsic motivation module is designed so as to provide intrinsic rewards, and then a new MADDPG algorithm with emotion intrinsic motivation named MADDPG-E is proposed. In the emotion intrinsic motivation module, according to the multi-agent target search task, three emotions are introduced and defined, which are joy, sadness, and fear. An emotional steady-state variable function is designed to judge the goodness of the emotions and decides the final value of the proposed emotion-based intrinsic reward function. With the introduced emotion intrinsic motivation module, it is hoped that the multi-agent system could always lean toward positive emotions, so as to speed up the task learning speed and optimize the learning performance.

The remainder of this paper is structured as follows. Section 2 introduces some background information. The proposed algorithm as well as the emotional intrinsic motivation design are introduced in Section 3. The simulation experiment results and discussions are presented in Section 4. Section 5 concludes this paper and envisages some future work.

2. Background

During multi-agent training, the state of each agent changes, and the environment is unstable from the point of view of other agents. Therefore, traditional reinforcement learning methods, such as Q-Learning or policy gradient, are not suitable for multi-agent

environments. For this reason, Lowe et al. [21] extended DDPG, a single-agent actor–critic method of deep reinforcement learning, to MADDPG and made it suitable for the multi-agent environment. Its algorithm framework is shown in Figure 1. The most core part of the MADDPG algorithm is that the *Critic* part of each agent can obtain the action information of all the other agents for centralized training and decentralized execution. This means, during the training, the *Critic* can observe the whole situation and can be introduced to guide the training of the *Actor*. When testing, the algorithm only uses the *Actor* network with local observations to take action.

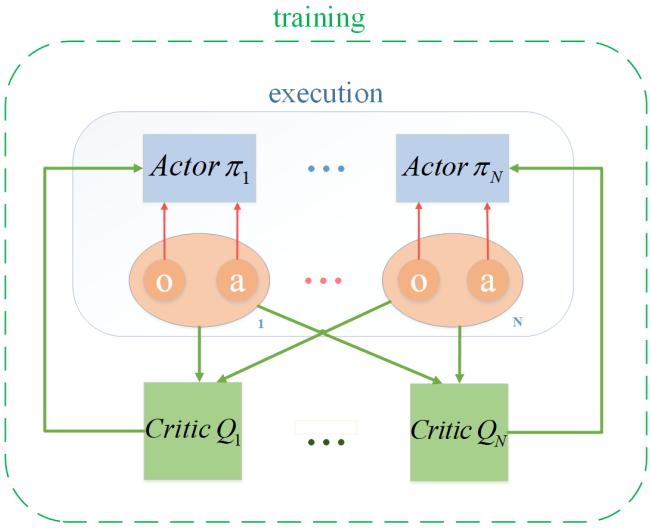


Figure 1. The MADDPG algorithmic framework.

In the MADDPG algorithm, the policy set of all the agents is $\pi_i = \{\pi_1, \ldots, \pi_N\}$, and the gradient of the expected return $J(\theta_i) = \mathbb{E}[R_i]$ for agent i is

$$\nabla_{\theta_i} J(\theta_i) = \mathbb{E}_{s\sim p^{\mu}, a_i \sim \pi_i}\left[\nabla_{\theta_i} \log \pi_i(a_i|o_i) Q_i^{\pi}(x, a_1, \ldots, a_N)\right] \tag{1}$$

where p^{μ} is the state distribution. $Q_i^{\pi}(x, a_1, \ldots, a_N)$ is a centralized action-value function; it takes the actions $(a_1, \ldots, a_N)$ of all the agents and the observed value of all the agents $x = (o_1, \ldots, o_N)$ as the input, and the output is the Q value of the agent i. For an agent in one state, it may have different actions as choices; however, according to the Q value calculated by the *Critic* network, the agent would like to choose the action with the highest Q value so as to obtain the maximum reward. The strategy gradient here in the MADDPG system increases the selective probability of actions with a high Q value and decreases the selective probability of actions with a low Q value.

Lowe et al. considered N continuous policies μ_i and extended them to deterministic strategies. The gradient is written as

$$\nabla_{\theta_i} J(\mu_i) = \mathbb{E}_{x,a\sim D}\left[\nabla_{\theta_i} \mu(a_i|o_i) \nabla_{a_i} Q_i^{\mu}(x, a_1, \ldots, a_N)|_{a_i}\right] \tag{2}$$

where D is the experience replay buffer, which records the experiences of all the agents. o_i is the observed state of agent i. The centralized action-value function Q_i^{μ} is updated as

$$\begin{cases} \mathcal{L}(\theta_i) = \mathbb{E}_{x,a,r,x'}\left[\left(Q_i^{\mu}(x, a_1, \ldots, a_N) - y\right)^2\right] \\ y = r_i + \gamma Q_i^{\mu'}(x', a'_1, \ldots, a'_N)\Big|_{a'_j = \mu'_j(o_j)} \end{cases} \tag{3}$$

where $\mu' = \{\mu_{\theta'_1}, \ldots, \mu_{\theta'_N}\}$ is the set of target policies with delayed parameters θ_i'. γ is the discount factor.

The MADDPG algorithm has the following three characteristics:

- The optimal policy obtained by learning only needs to use the local information to take the optimal action.
- The environment and special communication requirements are not needed.
- The algorithm can be used not only in a cooperative environment but also in a competitive environment.

3. Methods

3.1. The MADDPG-E Algorithm Framework

The basic idea of MADDPG is its framework of centralized training with decentralized execution. That is, during the training process, the *Critic* network of each agent collects the state and action information of all the agents, but during the training phase, decisions are made only by each agent's *Actor* network based on local information for the agent's own actions and states. Due to problems such as a sparse reward and unstable environment, the agent is not motivated to explore, resulting in a low reward and insufficient model convergence. Therefore, the MADDPG algorithm with emotional intrinsic motivation named MADDPG-E is proposed in this paper, and its algorithm framework is shown in Figure 2. In MADDPG-E, a new emotional intrinsic motivation module is designed. The emotional intrinsic motivation reward at each time can be generated according to the environmental stimuli and cognitive states, and such an intrinsic reward as well as the environmental rewards of the agent can then be used as the overall reward of the agent's search process. In this way, not only can the problem of reward sparsity be effectively solved but also collisions can be better avoided.

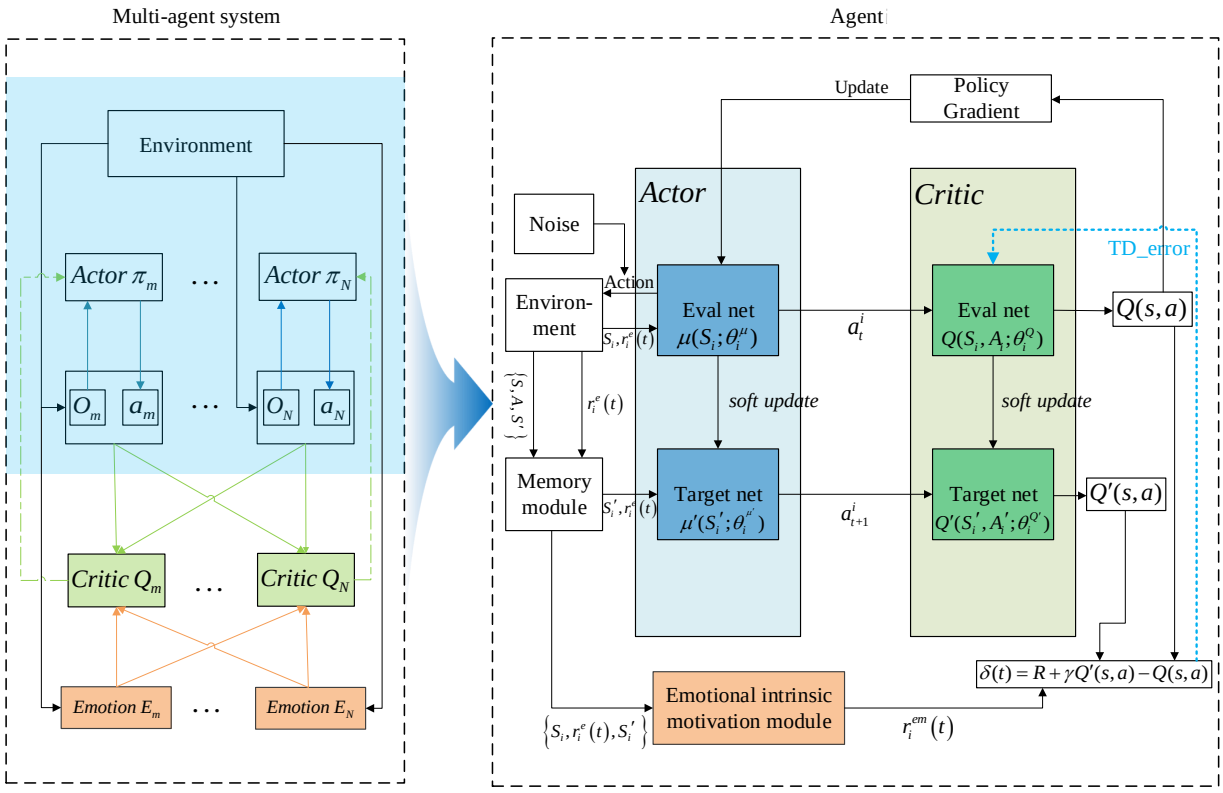


Figure 2. The MADDPG-E algorithmic framework.

The algorithm proposed in this paper consists of six elements: $\langle S, O, A, r^e(t), R, M\rangle$. The specific meaning of each element is as follow:

- $S = \{s_k | k = 1, 2, \ldots, m\}$ is the set of the state space, s_k represents the k-th state of the agent, and m represents the state number of the agent. All agents share the same state space.

- $O = \{O_1, O_2, \ldots, O_N\}$ is the set of the multi-agent observation space, and N is the agent's number. In each episode, every agent observes the state of the environment through perception, and the agent can obtain the position of obstacles, its own position, the position of other agents, and the position of the target within its detectable range.
- $A = \{A_1, A_2, \ldots, A_N\}$ is the set of the multi-agent action space, A_N is the action of the agent N, and it is mainly related to speed and direction. Then, the action of the agent N at time $t+1$ is expressed as

$$\begin{cases} a_{t+1}^N = (\alpha(t+1), v(t+1)) \\ \alpha(t+1) = \alpha(t) + \beta \\ v(t+1) = v(t) + v' \end{cases} \tag{4}$$

where $\alpha(t+1)$ is the movement angle of the agent at time $t+1$, β is the change rate of the agent's motion angle, $v(t+1)$ is the movement speed of the agent at time $t+1$, and v' is the acceleration.

- $r^e(t) = \left\{r^1(t), r^2(t), r^3(t), r^4(t)\right\}$ is the set of environmental rewards. All agents share the same environment rewards set. The agent is rewarded for moving to the target location and punished for colliding with the obstacles or bounds. A dynamic penalty function is set for the collision between agents, which can prevent the occurrence of unsafe states to the greatest extent. By setting environmental rewards, agents can learn to move toward the direction with the largest reward value and adopt the search strategy with the largest cumulative reward to help agents search for the target faster. At each time step, the agent changes its state and receives a reward from the environment. The environmental reward function $r^e(t)$ in this paper is designed as follows:

  $r^1(t) = 10$, if the agent has searched the target.
  $r^2(t) = -2$, if the agent collides with an obstacle.
  $r^3(t) = -\lambda \sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} \left[\left\|(x_i(t), y_i(t)) - \left(x_j(t), y_j(t)\right)\right\|\right]^{-1}$, if an agent collides with another agent. λ is the collision penalty factor, and $(x_i(t), y_i(t))$ is the positions of the agent i at time t.
  $r^4(t) = -10 \times (\max(|x_i(t)|, |y_i(t)|) - 0.9)$, if $max(|x_i(t)|, |y_i(t)|) \geq 0.9$, which represents the agent colliding with the bounds.

- $R(t)$ is the average value of the rewards obtained by N agents at the time t and is the overall reward of the multi-agent system in the collaborative target search process. The formula is as follows:

$$\begin{cases} R(t) = \dfrac{1}{N} \sum_{i}^{N} [-0.1 \times d_i + r_i^{em}(t) + r_i^e(t)], \\ d_i = \left(\sqrt{(x_i(t) - x_{tar}(t))^2 + (y_i(t) - y_{tar}(t))^2}\right) \end{cases} \tag{5}$$

where $(x_{tar}(t), y_{tar}(t))$ is the location of the target at the time t. $r_i^{em}(t)$ is the emotional intrinsic motivation reward of agent i. $r_i^e(t)$ is the environmental reward of agent i at the time t.

- M stands for the memory module, which stores the collected experience with an experience playback array, each of which is a quadruplet $\{s(t), a(t), s(t+1), R(t)\}$ as follows:

$$\begin{cases} s(t) = \left[o^1(t), \ldots, o^N(t)\right], \\ a(t) = \left[a^1(t), \ldots, a^N(t)\right], \\ s(t+1) = \left[o^1(t+1), \ldots, o^N(t+1)\right], \\ R(t) = [R_1(t), \ldots, R_N(t)] \end{cases} \tag{6}$$

where $o^N(t)$ is the observed state of agent N at time t. $a^N(t)$ is the action chosen by the agent N at time t.

In our algorithm, for N agents in the system, each agent i has an *Actor* network $\mu\left(S_i;\theta_i^{\mu}\right)$ and a *Critic* network $Q\left(S_i, A_i;\theta_i^{Q}\right)$. The *Actor* network is deterministic, and for the deterministic inputs (S_i, O_i), the output action a^i is deterministic. The input of the *Critic* network is the global state and the actions of all the agents, and the output is a real number. It indicates the degree to which action a is performed based on the state s. The *Critic* networks are used to evaluate all the actions and guide the *Actor* networks to make improvements.

The role of the *Actor* network is to increase the average value of the *Critic* network by improving the parameters θ_i^{μ} through training. The gradient of the expected return for agent i is as follows:

$$\nabla_{\theta^i} J(\theta^i) = \mathbb{E}_{o^i, a\sim M}\left[\nabla_{a^i} Q^i\left(o, a^1, \ldots, a^N\right) \nabla_{\theta^i}\theta_i^{\mu}\left(o^i\right)\middle| a^i\right] \tag{7}$$

where Q^i corresponds to the centralized critic of agent i. Its input consists of the agent joint observation $o = \left(o^1, o^2, \ldots, o^N\right)$ as well as the chosen specific actions $\left(a^1, a^2, \ldots, a^N\right)$ of all the agents.

The role of the *Critic* network is to conduct centralized training in joint observation and action.The policy input solely consists of the individual observation o^i to choose action a^i. Centralized critics for deterministic action policies are optimized with respect to the following loss function. Update the i-th value network with a TD error so that the value network better fits the value function $Q(s, a)$.

The update of the *Critic* network is as follows:

$$\begin{cases} \mathcal{L}(\theta_i) = \mathbb{E}_{o,a,R,o',M}\left[\left(y - Q^i\left(o, a^1, \ldots, a^N\right)\right)^2\right] \\ y = R + \gamma Q^{i'}\left(o', a^{1'}, \ldots, a^{N'}\right)\Big|_{a^{i'}=\mu^{i'}\left(o^i\right)} \end{cases} \tag{8}$$

where $\mu^{i'}$ corresponds to the deterministic policy of agent i. And $Q^{i'}$ represents the critic value with delayed parameters for agent i.

Both the *Critic* target network and the *Actor* target network use soft update methods for the parameter:

$$soft\ update: \begin{cases} \theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'} \\ \theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'} \end{cases} \tag{9}$$

Our MADDPG-E algorithm training process takes a centralized training and decentralized execution approach. That is, each agent obtains the actions performed in the current state according to its own strategy and interacts with the environment to obtain the experience stored in its own memory module. After all the agents interact with the environment, each agent randomly draws experience from the pool of experiences to train their own neural network; so as to output the optimal action at each moment to speed up the learning process of an agent, the input to the *Critic* network includes the observed state and actions taken by other agents by minimizing the loss to more *Critic* network parameters. The parameters of the updated action network are then calculated based on the gradient descent method.

3.2. Emotional Intrinsic Motivation Module

Emotion has the function of driving behavioral adaptation, and the emotion changes can generate learning changes [40]. At present, the choice of emotion mainly focuses on the six basic emotions proposed by Ekman: anger, disgust, fear, joy, sadness, and surprise [41]. Each basic emotion can be considered as an elementary response pattern, or action tendency [42]. Among them, in the robot emotional system, fear is mainly to

avoid danger, joy is mainly positive reinforcement behavior, and sadness is dominated by negative reinforcement behavior.

In the MADDPG-E algorithm, three agent states are defined in the process of multi-agent target search:

- State 1: It means that the agents search the target;
- State 2: It means that the agents do not search the target;
- State 3: It means that the agents are in danger, such as a collision.

It is hoped that the agents reach state 1 more often, which means the agents can search for the target more often, and at the same time, fewer collisions are expected. Therefore, in MADDPG-E, related to the three states defined above, three emotional motivations of joy, sadness, and fear are introduced as shown in Table 1. Among the three emotional motivations, joy belongs to the positive emotional motivation, and sadness and fear belong to the negative emotional motivation. Both positive and negative emotional motivation will generate the corresponding learning motivation and then affect the action choice of the agents. Positive emotional motivation makes the agent move toward the target, while negative emotional motivation prevents the agent from reaching bad states, such as states 2 and 3. Thus, the convergence process of the MADDPG-E algorithm is accelerated.

Table 1. Emotional motivations.

State	Emotional Motivation
The agents search the target	Joy
The agents do not search the target	Sadness
The agents are in danger, such as a collision	Fear

Aiming at the multi-agent search process with three emotional motivations, an emotional steady-state variable function H is defined:

$$H_m = H_{m-1} + H_m^{change} \tag{10}$$

m represents the m-th learning episode of the agent. H_m is the steady-state variable function of the emotional motivation change inside the agent at the learning episode m, and H_m^{change} represents the emotional motivation change value within the agent in the learning episode m, which can be calculated by the following formula,

$$H_m^{change} = \varphi K_m^{joy} r^{joy} + \delta K_m^{fear} r^{fear} + \omega K_m^{sad} r^{sad} \tag{11}$$

φ, δ, and ω are the weight parameters of the internal changes in the three emotional motivations of joy, fear, and sadness, respectively, which determine the degree of influence of the external environmental information on internal emotional changes, and the value is in the range $[0, 1]$. K_m^{joy}, K_m^{sad}, and K_m^{fear} are the numbers of the agents reaching state 1, state 2, and state 3. r^{joy}, r^{sad}, and r^{fear} represent the rewards that the agents obtain from the environment when the agents reach state 1, state 2, and state 3, respectively.

In two adjacent learning episodes, the difference in the value of the agent's emotional steady-state variable function will lead to the generation of emotional changes in the agent. Inspired by the one-dimensional emotional model, we only consider one emotional dimension and define the emotional function E as follows:

$$E = \begin{cases} 0 & \text{if } H_m \leqslant H_{m-1} \\ 1 & \text{if } H_m > H_{m-1} \end{cases} \tag{12}$$

Emotional intrinsic motivation affects the learning efficiency of the agent because it is an indirect mapping of the information in the learning process of the agent. Therefore, the emotion-based intrinsic reward function is as follows:

$$r^{em}(m) = \begin{cases} r^{e}(m) & m = 1 \\ Cr^{e}(m) & m = 2, 3, \ldots, T \end{cases} \tag{13}$$

where T is the maximum time episode, and C is the emotion coefficient, defined as follows:

$$C = \begin{cases} ke^{\frac{H_{m-1}}{H_m}} & E = 1 \\ \frac{H_{m-1}}{H_m} & E = 0 \end{cases} \tag{14}$$

where k is the emotional motivation reward parameter, and the value is in the range of $[0, 1]$.

Emotional intrinsic motivation can generate the agents different emotions and act as an intrinsic reward according to the environmental stimulus and cognitive state, which together with the environmental reward serves as the agent's overall reward. In our algorithm, it can be seen from formula (8) that adding emotional intrinsic motivation rewards can better evaluate deterministic strategic actions and guide the agent's action selection. The emotional function E can promote the agent to move in the direction that can generate positive emotion.

4. Simulation Experiment

The multi-agent collaborative target search problem refers to multi-agents cooperating with each other and avoiding collisions and obstacles to search for targets. Agents can communicate with each other to avoid collisions by sharing location information. In order to verify the effectiveness of our MADDPG-E algorithm, we used the Multi-agent Particle Environment (MPE) [43] provided by OpenAI. The multi-agent cooperative search scenario is shown in Figure 3. In the two-dimensional plane, the red balls represent the three agents, and the speed is relatively slow. The green ball represents the target to be searched for, and its speed is relatively high. Because the target moves faster, it is difficult for a single agent to search for it, so multi-agents need to cooperate to fulfill the task. The black balls represent the two obstacles.

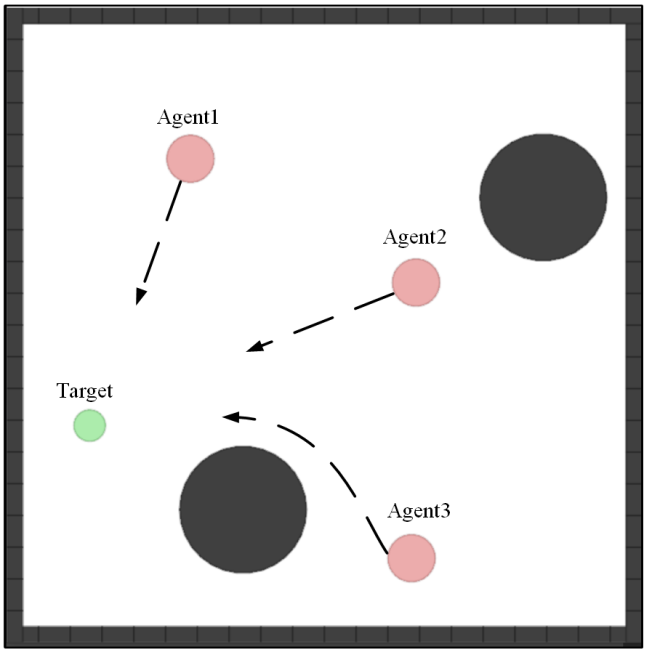


Figure 3. The multi-agent collaborative target search scenario.

The multi-agent collaborative target search task is for the agents to avoid collision and search for the target. Therefore, the distance between the agents, the distance between the agents and obstacles, and the distance between the agents and targets are needed and used as evaluation indicators to measure the multi-agent search strategy. The multi-agent search strategy evaluation index is as follows:

$$\begin{cases} \|D_i - D_n\| \geqslant R_i + R_n \\ and\ \|D_i - D_o\| \geqslant R_i + R_o \quad , n \in \{1, 2, \ldots, N\} \\ and\ \|D_i - D_t\| \leqslant R_i - R_t \end{cases} \tag{15}$$

D_i, D_n, D_o, and D_t are the central coordinates of the agent i, the n-th agent, the obstacle, and the target, respectively. R_i, R_n, R_o, and R_t are the the radius of the agent i, the n-th agent, the obstacle, and the target, respectively. Through the quality evaluation index, the status of the agent in the current position and the quality of the current search strategy can be judged.

The hyperparameters in the experiment are shown in Table 2.

Table 2. The training parameters.

Training Parameter	Values
Total number of episodes	30,000
Maximum episode length	50
Discount factor γ	0.98
Actor network learning rate α_a	0.01
Critic network learning rate α_c	0.01
Batch size	1024
Size of replay buffer M	10^6

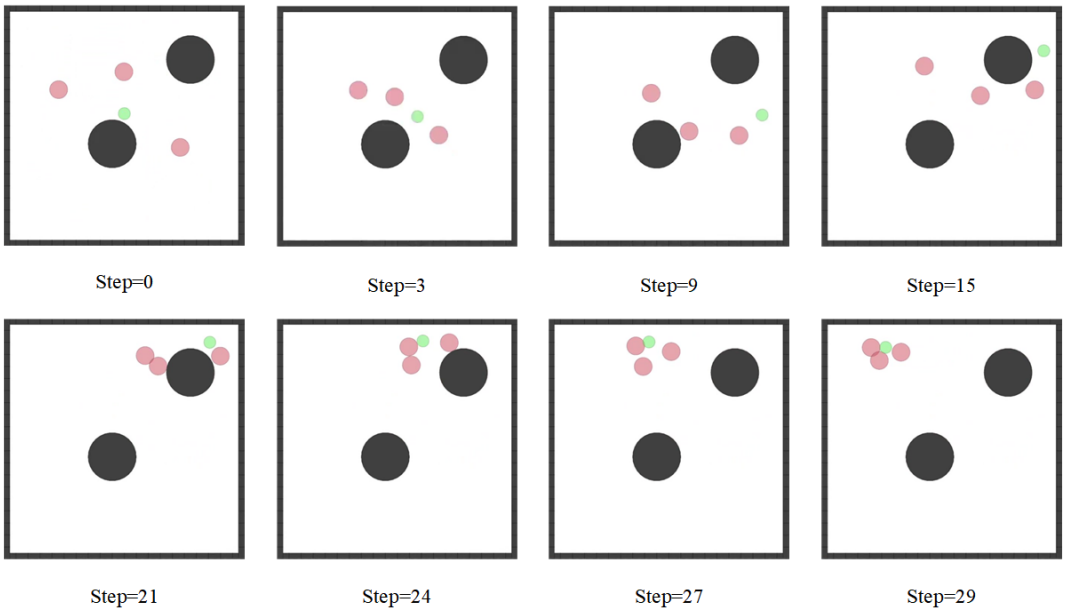
Two experiments are designed in this paper, which are the fixed initial position and random initial position.

4.1. Experiment 1: Fixed Initial Position

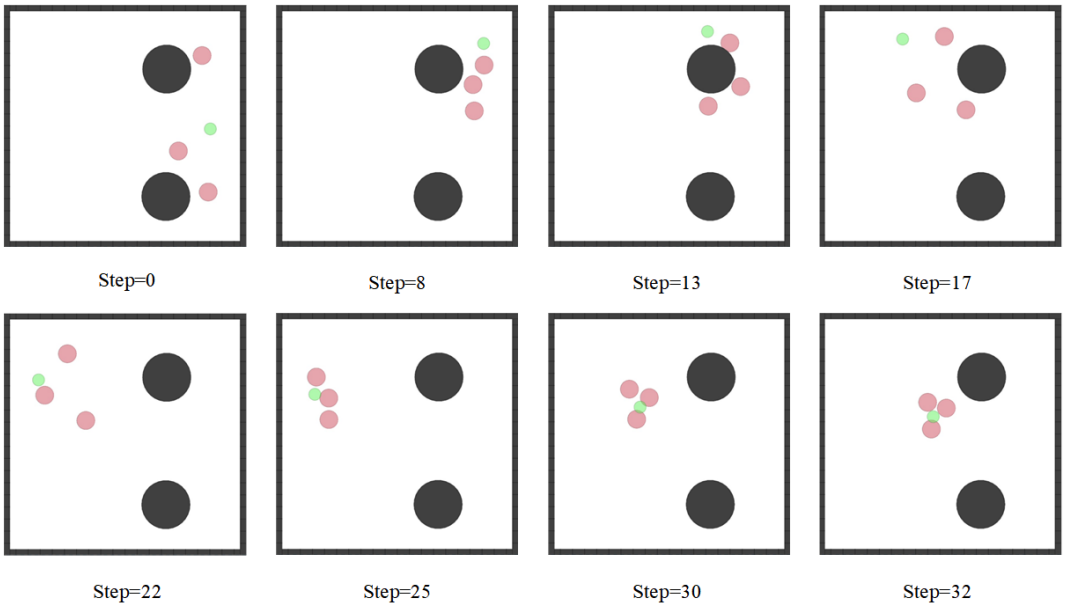
The fixed initial position experiment means that for every episode of training, it will start with the same positions of the agents, the obstacles, and the target. To show that the proposed MADDPG-E algorithm can help the multi-agent system achieve the objective in any situation, two fixed initial position experiments with different initial positions (as shown in Table 3) are performed. Figure 4a shows the multi-agent target search process under initial positions 1, and Figure 4b shows the results with initial positions 2. The figure shows that the agents find the target at 29 and 32 steps, respectively. It can be seen that the multi-agents can learn good cooperative search strategies through training. As it can be seen from the process of the agent searching for the target, that when facing a target moving faster, agents can use obstacles, boundaries, and other environmental factors to form an encircling strategy for the target so as to achieve the purpose of searching for the target.

Table 3. Initial position for the agents, targets, and obstacles.

Elements	Initial Positions 1	Initial Positions 2
Agent1	[−0.60, 0.30]	[0.45, −0.15]
Agent2	[0.00, 0.50]	[0.60, 0.60]
Agent3	[0.50, −0.20]	[0.65, −0.52]
Target	[0.00, 0.10]	[0.65, 0.00]
Obstacle1	[−0.10, −0.15]	[0.40, 0.45]
Obstacles2	[0.55, 0.55]	[0.40, −0.55]



(**a**) Fixed initial positions experiment 1



(**b**) Fixed initial positions experiment 2

Figure 4. Multi-agent target search experiment with fixed initial positions.

Two indicators are usually used to evaluate different deep reinforcement learning models: the convergence speed and the reward value after convergence. To show the effectiveness of the emotional intrinsic motivation module in MADDPG-E, we test its performance with the MADDPG algorithm and MADDPG-ICM algorithm (MADDPG algorithm with curiosity intrinsic motivation). The average reward is as in Figure 5. It can be found that, firstly, compared with MADDPG (blue line), both MADDPG-E (green line) and MADDPG-ICM (red line) finally converge to a significantly higher reward value. The larger the reward value after convergence, the more times the agents search for the target, which demonstrates the effectiveness of the intrinsic motivation. Secondly, although with a similar reward value, we can finally see that MADDPG-E begins to converge at about 6000 episodes, while MADDPG-ICM begins to converge at about 7500 episodes. This proves that MADDPG-E converges faster than MADDPG-ICM, which shows the

MADDPG-E has a faster learning speed, and the emotional intrinsic module here is useful. Figure 6 shows the mean reward values and standard deviations of the multi-agent system in 6000 episodes of training. We can see that the standard deviation of the MADDPG-E algorithm is the smallest, and the smaller the standard deviation means the better the stability of the algorithm model. This means MADDPG-E in this paper has better learning stability compared with the other two algorithms.

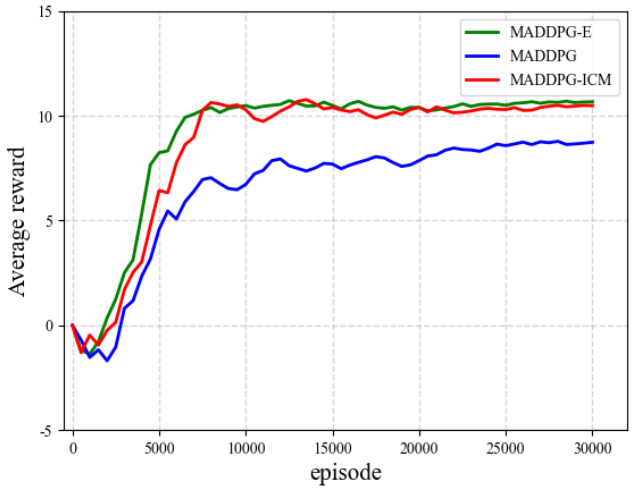


Figure 5. Average reward of MADDPG-E, MADDPG, and MADDPG-ICM in experiment 1.

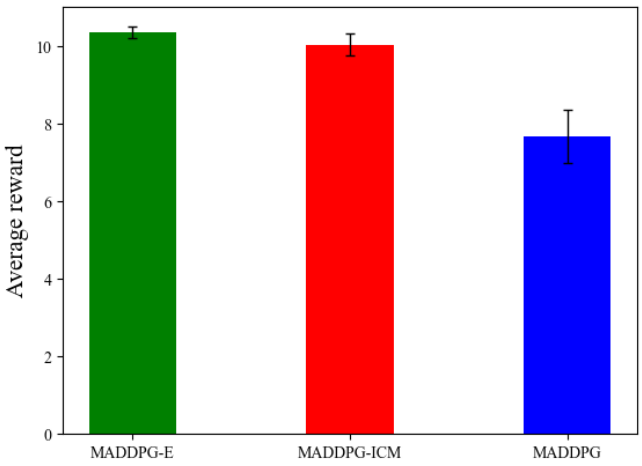


Figure 6. Average reward value and standard deviation in experiment 1.

In addition, in order to test the performance of the algorithm models during the learning process, a score evaluation index is defined here. In each episode of training, the agents get five points if they find the target and no score if they fail to find the target. The average scores are shown in Figure 7. We can see that the convergence score value of the MADDPG algorithm is significantly lower than the other two algorithms, which means that, during the same learning times, it searches the target fewer times. Let the multi-agent system learn more, and Table 4 gives more details including the average reward, average score, and target search time of the three algorithms in 30,000 episodes. The results show that the MADDPG-E algorithm has a shorter average target search time, higher model efficiency, faster target search, and more successful target search times than the other two algorithms. It shows that the algorithm proposed in this paper has great advantages both in the search speed and the number of successful searches.

Table 4. Average reward value, average score, and average target search time in 30,000 episodes.

Algorithm	Average Reward	Average Score	Target Search Time
MADDPG-E	8.94	3.67	3.84
MADDPG-ICM	8.55	3.59	4.28
MADDPG	6.45	2.68	4.73

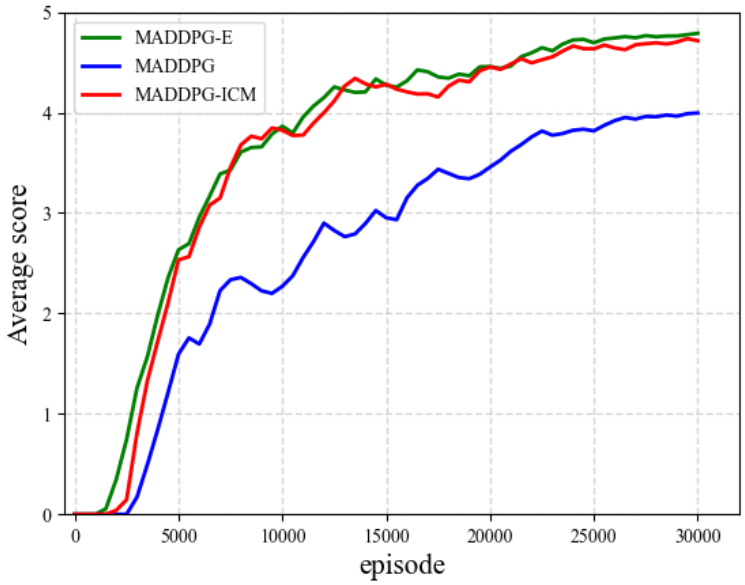


Figure 7. Average score of MADDPG-E, MADDPG, and MADDPG-ICM in experiment 1.

4.2. Experiment 2: Random Initial Position

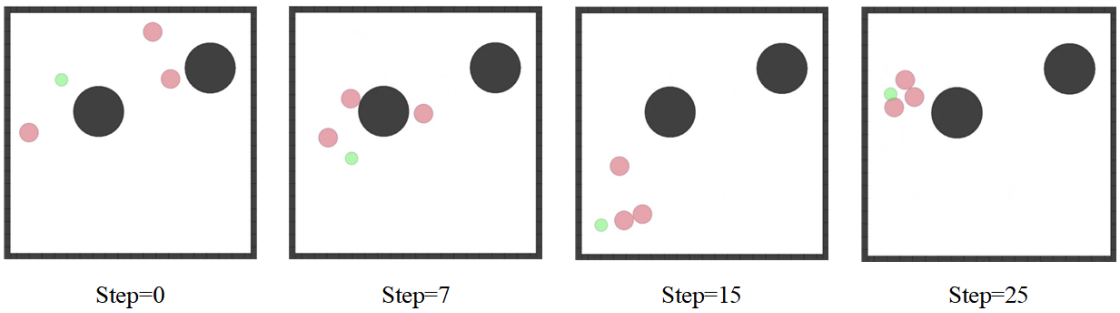
To show the robustness of the proposed MADDPG-E algorithm, here, the random initial position experiments are performed, which means for every episode of training, the positions of the agents, the obstacles, and the target are randomly generated. Figure 8 shows three groups in the multi-agent target search process. It can be seen that although the system starts to learn from different situations, it can still search the target, and the agents learn a good collaborative target search strategy, which proves the strong learning ability of the MADDPG-E algorithm.

Figure 9 shows the average reward during the training process. It can be seen that in the early stage (before about 6000 episodes of training), the reward curves of all three algorithms fluctuate heavily because of the excessive exploration. However, the MADDPG-E algorithm begins to converge after about 10,000 episodes of training, which is faster than the other two algorithms. Because MADDPG-E adds emotional intrinsic motivation, it can increase the reward that the multi-agent system obtains and can accelerate the learning speed as well as the convergence speed. What is more, compared with MADDPG-ICM and MADDPG, the reward line of MADDPG-E seems smoother, which means its learning result is more stable and less oscillation happens. Figure 10 shows the mean reward value and standard deviation of the system after 5000 episodes of training in experiment 2. What can be seen is that after convergence, the average standard deviation of the reward of the MADDPG-E algorithm is smaller than that of the other two algorithms, indicating that the MADDPG-E model has better stability. The curves of the average score of the three algorithms in experiment 2 are shown in Figure 11. It shows that the MADDPG-E algorithm has a significantly higher average score than the other two algorithms, and the agents successfully search for the target more times. What is more, in the early stage of exploration, MADDPG-E is more stable. Table 5 shows the values of the agent's average reward, average score, and average target search time in 30,000 rounds of training. Compared with the other two algorithms, the MADDPG-E algorithm in this paper searches for the target in a

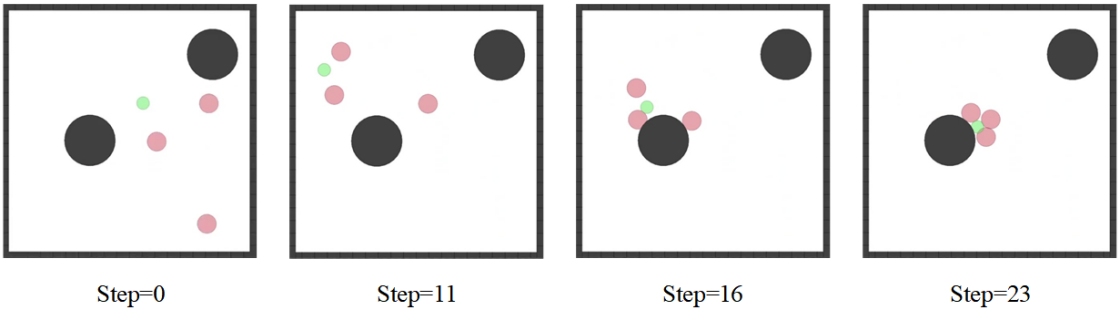
shorter time and more times, indicating that the MADDPG-E algorithm is more efficient in task learning.

Table 5. Average reward value, average score, and average target search time in 30,000 episodes.

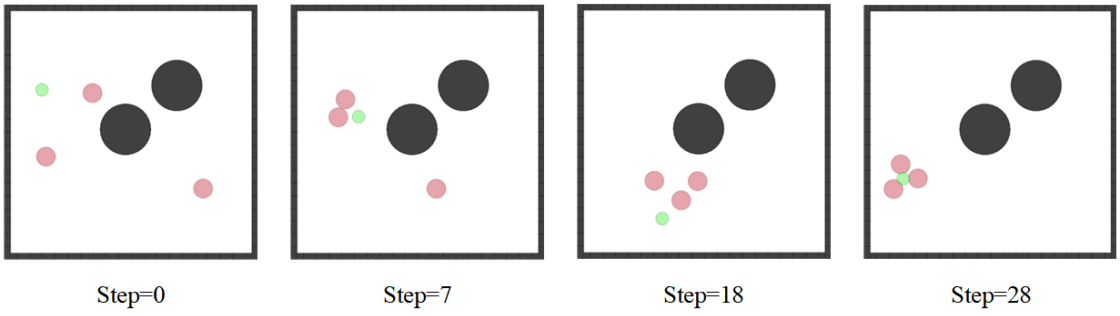
Algorithm	Average Reward	Average Score	Target Search Time
MADDPG-E	9.26	3.79	3.68
MADDPG-ICM	8.63	3.46	4.12
MADDPG	6.83	2.54	4.57



(**a**) Randomized initial position experiment 1



(**b**) Randomized initial position experiment 2



(**c**) Randomized initial position experiment 3

Figure 8. Multi-agent target search experiment with randomized initial positions.

Compared with the experimental results of the fixed initial position, the MADDPG-E algorithm proposed in this paper has more obvious advantages over the other two algorithms in the complex random initial position experiments, whether it is the size of the reward value after convergence or the convergence speed and stationarity, which indicates that the proposed MADDPG-E algorithm can better adapt to complex and unknown situations.

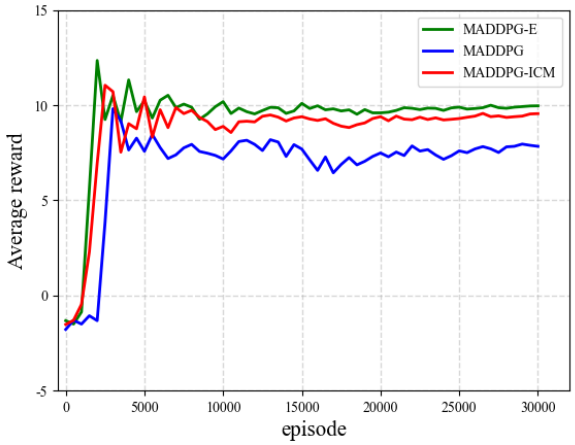


Figure 9. Average reward of MADDPG-E, MADDPG, and MADDPG-ICM in experiment 2.

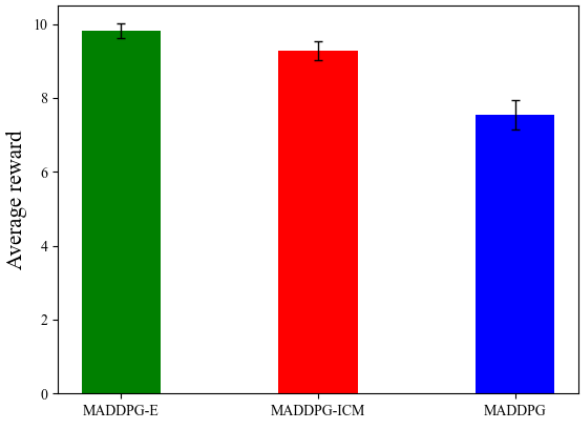


Figure 10. Average reward value and standard deviation in experiment 2.

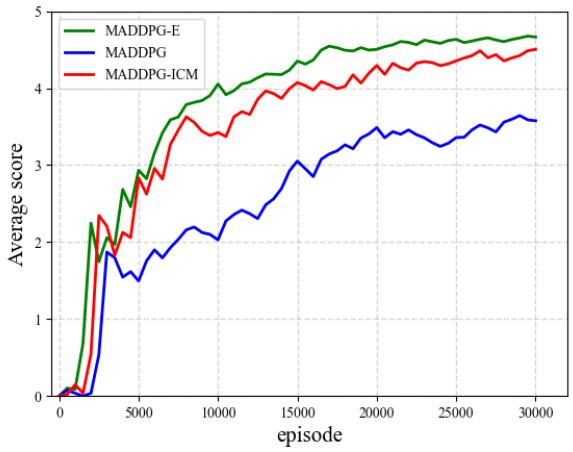


Figure 11. Average score of MADDPG-E, MADDPG, and MADDPG-ICM in experiment 2.

5. Conclusions

In this paper, an improved MADDPG algorithm is proposed to solve the sparse reward problem in a multi-agent collaborative target search. Under the framework of the MADDPG algorithm, a new module of emotional intrinsic motivation is added, and three kinds of emotional motivation, including joy, sadness, and fear, are introduced. The emotional intrinsic motivation module can generate corresponding intrinsic rewards according to different states of the multi-agent collaborative target search process to accelerate the learning speed as well as optimize the learning process. Two kinds of simulation experiments are then

carried out in this paper, and the results show that the proposed MADDPG-E algorithm can learn a good search strategy and has higher search efficiency and better stability.

At present, the proposed MADDPG-E algorithm has a good effect on the multi-agent target search task. However, a limitation of this study is that we do not make role distinctions for agents, which can lead to increased training time, especially in some real-world complex scenarios. Therefore, in future work, we will distinguish the roles of the multi-agents and improve our emotional intrinsic motivation module to make our method more suitable for the complex environment and improve the success rate of the target search.

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

Funding: The work was supported by the Natural Science Foundation of China (grant no. 61903006), Beijing Natural Science Foundation (grant no. 4202022, 4204096), R&D Program of Beijing Municipal Education Commission (KM202210009012), Beijing Municipal Great Wall Scholar Program (grant no. CIT&TCD 20190304), China Scholarship Council and Beijing Association for Science and Technology Young Talent Promotion Project, Beijing Urban Governance Research Base of North China University of Technology, and North China University of Technology unveiling project (2023YZZKYO3).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Amirkhani, A.; Barshooi, A.H. Consensus in multi-agent systems: A review. *Artif. Intell. Rev.* **2022**, *55*, 3897–3935. [CrossRef]
2. Li, Y.; Xu, F.; Xie, G.; Huang, X. Survey of development and application of multi-agent technology. *Comput. Eng. Appl.* **2018**, *54*, 13–21.
3. Cai, Y.; Shen, Y. An integrated localization and control framework for multi-agent formation. *IEEE Trans. Signal Process.* **2019**, *67*, 1941–1956. [CrossRef]
4. Han, W.; Zhang, B.; Wang, Q.; Luo, J.; Ran, W.; Xu, Y. A multi-agent based intelligent training system for unmanned surface vehicles. *Appl. Sci.* **2019**, *9*, 1089. [CrossRef]
5. Liu, X.; Yu, J.; Feng, Z.; Gao, Y. Multi-agent reinforcement learning for resource allocation in IoT networks with edge computing. *China Commun.* **2020**, *17*, 220–236. [CrossRef]
6. He, Z.; Dong, L.; Song, C.; Sun, C. Multiagent Soft Actor-Critic Based Hybrid Motion Planner for Mobile Robots. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**. [CrossRef]
7. Zhou, X.; Zhou, S.; Mou, X.; He, Y. Multirobot Collaborative Pursuit Target Robot by Improved MADDPG. *Comput. Intell. Neurosci.* **2022**, *2022*, 4757394. [CrossRef]
8. Senanayake, M.; Senthooran, I.; Barca, J.C.; Chung, H.; Kamruzzaman, J.; Murshed, M. Search and tracking algorithms for swarms of robots: A survey. *Robot. Auton. Syst.* **2016**, *75*, 422–434. [CrossRef]
9. Hazra, T.; Kumar, C.S.; Nene, M. Multi-agent target searching with time constraints using game-theoretic approaches. *Kybernetes* **2017**, *46*, 1278–1302. [CrossRef]
10. Cooper, J.R. Optimal multi-agent search and rescue using potential field theory. In Proceedings of the AIAA Scitech 2020 Forum, Orlando, FL, USA, 6–10 January 2020; p. 0879.
11. Tang, H.; Sun, W.; Yu, H.; Lin, A.; Xue, M. A multirobot target searching method based on bat algorithm in unknown environments. *Expert Syst. Appl.* **2020**, *141*, 112945. [CrossRef]
12. Wu, Z.; Xie, Z. A multi-objective lion swarm optimization based on multi-agent. *J. Ind. Manag. Optim.* **2023**, *19*, 1447–1458. [CrossRef]
13. Shapero, S.A.; Hughes, H.; Tuuk, P. Adaptive semi-greedy search for multidimensional track assignment. In Proceedings of the 2016 19th International Conference on Information Fusion (FUSION), Heidelberg, Germany, 5–8 July 2016; pp. 409–415.
14. Teatro, T.A.; Eklund, J.M.; Milman, R. Nonlinear model predictive control for omnidirectional robot motion planning and tracking with avoidance of moving obstacles. *Can. J. Electr. Comput. Eng.* **2014**, *37*, 151–156. [CrossRef]

15. Sun, L.; Chang, Y.C.; Lyu, C.; Shi, Y.; Shi, Y.; Lin, C.T. Toward multi-target self-organizing pursuit in a partially observable Markov game. *arXiv* **2022**, arXiv:2206.12330.
16. Wang, G.; Wei, F.; Jiang, Y.; Zhao, M.; Wang, K.; Qi, H. A Multi-AUV Maritime Target Search Method for Moving and Invisible Objects Based on Multi-Agent Deep Reinforcement Learning. *Sensors* **2022**, *22*, 8562. [CrossRef]
17. Gupta, J.K.; Egorov, M.; Kochenderfer, M. Cooperative multi-agent control using deep reinforcement learning. In Proceedings of the Autonomous Agents and Multiagent Systems: AAMAS 2017 Workshops, Best Papers, São Paulo, Brazil, 8–12 May 2017; Revised Selected Papers 16; Springer: Berlin/Heidelberg, Germany, 2017; pp. 66–83.
18. Cao, X.; Lu, T.; Cai, Y. Intrinsic Motivation for Deep Deterministic Policy Gradient in Multi-Agent Environments. In Proceedings of the 2020 Chinese Automation Congress (CAC), Shanghai, China, 6–8 November 2020; pp. 1628–1633.
19. Li, Y. Deep reinforcement learning: An overview. *arXiv* **2017**, arXiv:1701.07274.
20. Tampuu, A.; Matiisen, T.; Kodelja, D.; Kuzovkin, I.; Korjus, K.; Aru, J.; Aru, J.; Vicente, R. Multiagent cooperation and competition with deep reinforcement learning. *PLoS ONE* **2017**, *12*, e0172395. [CrossRef] [PubMed]
21. Lowe, R.; Wu, Y.I.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6379–6390.
22. Song, J.; Ren, H.; Sadigh, D.; Ermon, S. Multi-Agent Generative Adversarial Imitation Learning. *arXiv* **2018**, arXiv:1807.09936.
23. Parisi, S.; Tateo, D.; Hensel, M.; D'eramo, C.; Peters, J.; Pajarinen, J. Long-Term Visitation Value for Deep Exploration in Sparse-Reward Reinforcement Learning. *Algorithms* **2022**, *15*, 81. [CrossRef]
24. Perovic, G.; Li, N. Curiosity driven deep reinforcement learning for motion planning in multi-agent environment. In Proceedings of the 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dali, China, 6–8 December 2019; pp. 375–380.
25. Loyola, O.; Kern, J.; Urrea, C. Novel Algorithm for Agent Navigation Based on Intrinsic Motivation Due to Boredom. *Inf. Technol. Control.* **2021**, *50*, 485–494. [CrossRef]
26. Sequeira, P.; Melo, F.S.; Paiva, A. Emotion-based intrinsic motivation for reinforcement learning agents. In Proceedings of the Affective Computing and Intelligent Interaction: 4th International Conference, ACII 2011, Memphis, TN, USA, 9–12 October 2011; Proceedings, Part I 4; Springer: Berlin/Heidelberg, Germany, 2011; pp. 326–336.
27. Starzyk, J.A. *Motivation in Embodied Intelligence*; INTECH Open Access Publisher: London, UK, 2008.
28. Barto, A.G.; Singh, S.; Chentanez, N. Intrinsically motivated learning of hierarchical collections of skills. In Proceedings of the 3rd International Conference on Development and Learning, La Jolla, CA, 20–22 October 2004; Volume 112, p. 19.
29. Oudeyer, P.Y.; Kaplan, F.; Hafner, V.V. Intrinsic motivation systems for autonomous mental development. *IEEE Trans. Evol. Comput.* **2007**, *11*, 265–286. [CrossRef]
30. Sutton, R.; Barto, A. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 1998.
31. Pathak, D.; Agrawal, P.; Efros, A.A.; Darrell, T. Curiosity-driven exploration by self-supervised prediction. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 2778–2787.
32. Barrett, L.F.; Lindquist, K.A. The embodiment of emotion. In *Embodied Grounding: Social, Cognitive, Affective, and Neuroscientific Approaches*; Cambridge University Press: Cambridge, UK, 2008; pp. 237–262.
33. Duffy, E. Is emotion a mere term of convenience? *Psychol. Rev.* **1934**, *41*, 103. [CrossRef]
34. Young, P.T. *Emotion in Man and Animal; Its Nature and Relation to Attitude and Motive*; APA PsycInfo: Washington, DC, USA, 1943.
35. Huang, X.; Wu, W.; Qiao, H. Computational modeling of emotion-motivated decisions for continuous control of mobile robots. *IEEE Trans. Cogn. Dev. Syst.* **2020**, *13*, 31–44. [CrossRef]
36. Feldmaier, J.; Diepold, K. Path-finding using reinforcement learning and affective states. In Proceedings of the The 23rd IEEE International Symposium on Robot and Human Interactive Communication, Edinburgh, Scotland, 25–29 August 2014; pp. 543–548.
37. Fang, B.; Guo, X.; Wang, Z.; Li, Y.; Elhoseny, M.; Yuan, X. Collaborative task assignment of interconnected, affective robots towards autonomous healthcare assistant. *Future Gener. Comput. Syst.* **2019**, *92*, 241–251. [CrossRef]
38. Guzzi, J.; Giusti, A.; Gambardella, L.M.; Di Caro, G.A. Artificial emotions as dynamic modulators of individual and group behavior in multi-robot system. In Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, Stockholm, Sweden, 10–15 July 2018; pp. 2189–2191.
39. Achiam, J.; Sastry, S. Surprise-based intrinsic motivation for deep reinforcement learning. *arXiv* **2017**, arXiv:1703.01732.
40. Yu, H.; Yang, P. An emotion-based approach to reinforcement learning reward design. In Proceedings of the 2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC), Banff, AB, Canada, 9–11 May 2019; pp. 346–351.
41. Ekman, P. Basic emotions. In *Handbook of Cognition and Emotion*; John Wiley & Sons: Hoboken, NJ, USA, 1999; Volume 98, p. 16.
42. Frijda, N.H.; Kuipers, P.; Ter Schure, E. Relations among emotion, appraisal, and emotional action readiness. *J. Personal. Soc. Psychol.* **1989**, *57*, 212. [CrossRef]
43. pzhokhov. Multiagent-Particle-Envs. 2017. Available online: https://github.com/openai/multiagent-particle-envs (accessed on 11 September 2023).