Multi-Robot Cooperative Target Encirclement through Learning Distributed Transferable Policy

Tianle Zhang*† Zhen Liu*† Shiguang Wu*† Zhiqiang Pu*† Jianqiang Yi*†
*University of Chinese Academy of Sciences, Beijing, 100049, China
†Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
(tianle-zhang@outlook.com, liuzhen@ia.ac.cn,
shiguang.wu@outlook.com, zhiqiang.pu@ia.ac.cn, jianqiang.yi@ia.ac.cn)

***Abstract*—Making efficient motion decisions for a multi-robot system is a challenging problem in target encirclement with collision avoidance. Specifically, each robot with local communication has to consider cooperative target encirclement and collision avoidance simultaneously. In this paper, a distributed transferable policy network framework based on deep reinforcement learning is proposed to solve the problem of multi-robot cooperative target encirclement with collision avoidance. The proposed policy network framework is able to process the information of uncertain number of robots and obstacles, which is a desirable property for multi-robot systems. In particular, graph attention communication mechanism is adopted to model multi-robot interactions as a graph and extract cooperative information from the graph. Long short-term memory is used to accept the states of uncertain number of obstacles. In addition, a compound reward is designed to lead the training of the behavior of target encirclement with collision avoidance. Curriculum learning is implemented to speed up the process of this training. Simulation results validate the effectiveness of the proposed algorithm. Moreover, we further show that the learned policy can directly transfer to different scenarios along with good generalization.**

***Index Terms*—multi-robot, target encirclement, collision avoidance, deep reinforcement learning, curriculum learning**

I. Introduction

In recent years, target encirclement of multi-robot systems has attracted more and more attention among researchers due to the promising broad applications, such as escorting [1], capture of the enemy target [2], reconnaissance and surveillance [3]. The key problem of these applications is to control a multi-robot system to cooperatively encircle a specific target with an expected formation through appropriate method. In particular, each robot with local communication has to not only encircle a target, but also avoid both inter-robot collisions and obstacle collisions. Moreover, the target may have a highly intelligent escape strategy. Hence, the problem remains challenging.

In the related works of target encirclement, the early researches focus on encircling a stationary target [4]. Subsequently, several recent works investigate the problem of moving target. Several control algorithms are proposed to address the problem of distributed multi-robot cooperative target encirclement [5]–[8]. However, these algorithms require that all robots are initially placed using a predefined stand-off distance between the robots and the target. To avoid the initial setup, some studies only require the positions of all robot to meet the certain conditions for successful encirclement [9], [10]. But these works based on control theory mostly depend on the precise control models, which are not easy to obtain in practical applications. Besides, most of existing works do not consider collision avoidance in the process of encircling the target, which put the multi-robot team in an unsafe situation.

Recent studies have shown the potentialities of the deep reinforcement learning (DRL) methods on multi-robot systems. Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [11] uses a framework of centralized training with decentralized execution to enable multi-robot teams to learn a cooperative or competitive behavior. Counterfactual Multi-Agent (COMA) [12] is proposed to solve the problem of multi-agent credit assignment. However, MADDPG and COMA share all information to train a policy for each robot, which is not available in practice. To deal with this limitation, Mean Field [13] adopts the state and mean action of neighboring agents to make decision, which ignores different impacts of neighboring agents. Considering this shortcoming, Attentional Communication (ATOC) [14] uses the attentional communication model to interact with neighboring agents. However, the communication topology graph for interactions among robots is not considered. Furthermore, the aforementioned works do not directly address the problem of target encirclement. Subsequently, Ma et al. [15] designs a DRL method to enable the multi-robot system to encircle a target and avoid collisions at the same time. Unfortunately, it does not consider the interactions between robots and is unable to transfer the learned policy to different environments. Therefore, this method is infeasible in large-scale robot teams.

Motivated by these problems, we propose a distributed transferable policy network framework based on DRL to solve the problem of multi-robot cooperative target encirclement with collision avoidance. Specifically, graph attention communication mechanism (GACM) is used to model the interactions of multiple robots as a graph. Robots form nodes in the graph, and the edge exists between two communicating robots. The cooperative information among robots is obtained from the graph. Long short-term memory (LSTM) [16] is adopted to process the information of uncertain number of obstacles. In addition, a compound reward is designed to guide a multi-robot system to learn to cooperatively encircle a target and

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

avoid collisions at the same time. Meanwhile, curriculum learning is implemented to speed up the learning process. In summary, the main contributions in this paper are:

- A distributed transferable policy network framework, with GACM of completing interactions among robots and LSTM for processing the information of uncertain number of obstacles, for solving the problem of multi-robot cooperative target encirclement with collision avoidance.
- A special design of the compound reward function for target encirclement and collision avoidance at the same time, considering the encirclement formation size (radius), the encirclement formation shape (the distance among neighboring robots), and avoiding collisions (other robots and obstacles).
- Simulation results that show the effectiveness and generalization of the proposed algorithm in various test scenarios, where different number of robots equally encircle a stationary or moving target in uncertain number of obstacle environments.

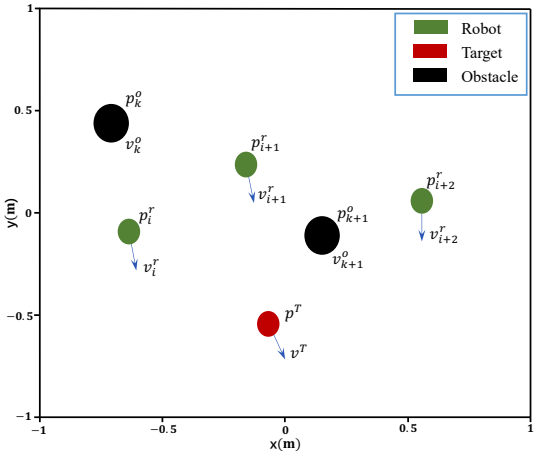


Fig. 1. Multi-robot cooperative target encirclement. The red circle, green circles and black circles represent the target, the robots and obstacles at the current time respectively, whose initial positions are randomly set. The blue arrows represent their velocities.

II. Preliminaries

In this section, the problem formulation of target encirclement will be presented in detail, and then, the policy-based learning method will be introduced.

A. Problem Formulation

The target encirclement task of a multi-robot system is to form a specific formation that can encircle a target in 2D space. To achieve this task, the multi-robot formation should satisfy the following conditions as much as possible [15]:

- The formation should be a convex polygon that encircles the target, and the distance between adjacent vertices should be the same as possible in the convex polygon.
- In formation, each vertex of the convex polygon can be occupied by any robot.

Based on the above conditions, the problem of multi-robot cooperative target encirclement is defined as follows.

Definition 1: A target position $\boldsymbol{p}^T(t) = [p_x^T(t), p_y^T(t)]$ is equally encircled by $n(n > 1)$ robots of the complete position distribution $\boldsymbol{p}_i^r(t) = [p_{xi}^r(t), p_{yi}^r(t)]$, $i \in \mathbb{U}$, $\mathbb{U} = 1, 2, ..., n$, if

$$\begin{aligned} &lim_{t\to+\infty}||\boldsymbol{p}_i^r(t) - \boldsymbol{p}^T(t)|| = \rho, \\ &lim_{t\to+\infty}||\boldsymbol{p}_i^r(t) - \boldsymbol{p}_j^r(t)|| \geq d, \quad i \neq j \in \mathbb{U}, \end{aligned} \tag{1}$$

where $\rho > 0$ is the radius of encirclement formation, and $d = 2\rho sin(\pi/n)$. For every two adjacent robots, denoting $\hbar$ and ℓ, the property (1) together with the geometric constraints, implies $lim_{t\to+\infty}||\boldsymbol{p}_\hbar^r(t) - \boldsymbol{p}_\ell^r(t)|| = d$ [10].

As shown in Fig. 1, a target encirclement task where n robots need to cooperatively encircle one stationary or moving target in an environment with m obstacles is investigated in this paper. This can be expressed as a sequential decision-making problem in the framework of reinforcement learning (RL). Each robot is controlled by the distributed control policy learned through RL. This policy enables each robot to cooperatively encircle the target and avoid collisions with other robots and obstacles at the same time. In general, the key point of the generation of appropriate control policy is designing a proper policy network framework and learning method.

In this paper, we assume that each robot can observe the velocity and position of obstacles in its field of vision and communicate with its neighboring robots. Meanwhile, the position and velocity of the target can be obtained by each robot. Let $\boldsymbol{v}_i^r(t) = [v_{xi}^r, v_{yi}^r]$, $\boldsymbol{v}^T(t) = [v_x^T, v_y^T]$ denote the velocities of robot i and target respectively. The velocity and position of obstacle k $(k = 1, 2, ..., m)$ are represented as $\boldsymbol{p}_k^o(t) = [p_{xk}^o(t), p_{yk}^o(t)]$, $\boldsymbol{v}_k^o(t) = [v_{xk}^o, v_{yk}^o]$ respectively. Furthermore, $\boldsymbol{s}^T(t) = [v_x^T(t), v_y^T(t), p_x^T(t), p_y^T(t)]$, $\boldsymbol{s}_k^o(t) = [v_{xk}^o(t), v_{yk}^o(t), p_{xk}^o(t), p_{yk}^o(t)]$ represent the state of the target and obstacle k at time t respectively, and $\boldsymbol{s}_i^r(t) = [v_{xi}^r(t), v_{yi}^r(t), p_{xi}^r(t), p_{yi}^r(t), d_i^T(t)]$, where $d_i^T(t)$ is the distance from the robot to the target, represents the state of robot i at time t, while $\boldsymbol{s}^o(t) = [\boldsymbol{s}_1^o(t), \boldsymbol{s}_2^o(t), ..., \boldsymbol{s}_m^o(t)]$ represents the states of all obstacles at time t. In addition, the dynamic model of each robot is modeled as a double integrator model, and the action of robot i denotes $\boldsymbol{a}_i(t) = [F_{xi}(t), F_{yi}(t)]$, which represents the force applied to the robot in both directions. The goal of this paper is to design a behavior policy for each robot i, $\pi_i : \boldsymbol{s}_i(t) \to \boldsymbol{a}_i(t)$, $\boldsymbol{s}_i(t) = [\boldsymbol{s}^T(t), \boldsymbol{s}_i^r(t), \boldsymbol{s}^o(t)]$, to select an appropriate action for target encirclement and collision avoidance. The behavior policy of each robot is generally approximated by a policy network. In addition, in order to guide the generation of this behavior policy, we design a compound reward $R_i(\boldsymbol{s}^{all}(t), \boldsymbol{a}(t))$, where $\boldsymbol{s}^{all}(t) = [\boldsymbol{s}^T(t), \boldsymbol{s}^r(t), \boldsymbol{s}^o(t)]$, $\boldsymbol{s}^r(t) = [\boldsymbol{s}_1^r(t), \boldsymbol{s}_2^r(t), ..., \boldsymbol{s}_n^r(t)]$, and $\boldsymbol{a}(t) = [\boldsymbol{a}_1(t), \boldsymbol{a}_2(t), ..., \boldsymbol{a}_n(t)]$. The optimal behavior policy of robot i is obtained by training the policy network through maximizing cumulative reward using RL. Meanwhile, All learnable parameters of the policy network are shared to each robot, that is, all robots share the policy network. These learnable parameters are trained through policy-based learning

method.

B. Policy-Based Learning

According to the above, this paper considers RL framework which generates a behavior policy that a robot can execute. We adopt an actor-critic algorithm called proximal policy optimization (PPO) [17] to produce the behavior policy. The PPO uses a single deep neural network (DNN) to approximate both the value (critic) and policy (actor) functions, and the DNN is trained with two loss terms,

$$l_{v_i} = (Q_i(t) - V_i(\boldsymbol{s}_i(t))^2, \tag{2}$$

$$\begin{aligned} l_{\pi_i} =& min(\frac{\pi_i(\boldsymbol{a}_i(t)|\boldsymbol{s}_i(t))}{\pi_i^{old}(\boldsymbol{a}_i(t)|\boldsymbol{s}_i(t))}(Q_i(t) - V(\boldsymbol{s}_i(t))), \\ & clip(\frac{\pi_i(\boldsymbol{a}_i(t)|\boldsymbol{s}_i(t))}{\pi_i^{old}(\boldsymbol{a}_i(t)|\boldsymbol{s}_i(t))}, 1-\epsilon, 1+\epsilon)(Q_i(t) - V(\boldsymbol{s}_i(t))), \end{aligned} \tag{3}$$

where (2) trains the DNN's value output $V_i(\boldsymbol{s}_i(t))$, which can be used to evaluate the policy π_i. The value need to match the future discount reward estimate, $Q_i(t) = R_i(\boldsymbol{s}^{all}(t), \boldsymbol{a}(t)) + \gamma V_i(\boldsymbol{s}_i(t+1))$, where $\gamma \in (0,1)$ is a discount factor. For the policy output in (3), the importance sampling is implemented to convert the training process of on-policy to off-policy, that is, we can fully sample through the old policy, i.e., $\pi_i^{old}(\boldsymbol{s}_i(t))$, and then improve the new policy, i.e., $\pi_i(\boldsymbol{s}_i(t))$. The $clip$ operation limits the value of $\frac{\pi_i(\boldsymbol{s}_i(t))}{\pi_i^{old}(\boldsymbol{s}_i(t))}$ to $(1-\epsilon, 1+\epsilon)$ where $\epsilon = 0.2$ is a hyper parameter, which makes the difference between $\pi_i(\boldsymbol{s}_i(t))$ and $\pi_i^{old}(\boldsymbol{s}_i(t))$ not too big. This ensures the rationality of the importance sampling.

In this work, we open multiple threads to simulate the interactions between the robots and the environment in parallel, and the robots share a policy network. Based on PPO, the policy network is trained using the fusion experiences of the robots.

III. Approach

In this section, GACM is introduced in details, and LSTM module handling uncertain number of obstacles is given. Next, the compound reward function is designed. Then, the distributed transferable policy network framework is established to solve the problem of multi-robot cooperative target encirclement with collision avoidance. In addition, the curriculum learning method is presented.

A. Graph Attention Communication Mechanism

In reality, the communication topology of a multi-robot system is always represented as a graph, and each edge of the graph conveys a different message. In this paper, GACM based on graph attention network [18] is implemented to describe multi-robot interactions as a graph and then extract cooperative information from the graph with attention.

We define a graph $\mathbb{G} := (\mathbb{V}, \mathbb{E})$, where each node $r \in \mathbb{V}$ denotes a robot, and there exists an edge $e \in \mathbb{E}$ between two nodes if the nodes can communicate with each other. The robots can exchange messages along the edges of the graph. In this work, communication between the robots is local, that is, two robots can communicate with each other only if the distance between them is less than a pre-defined threshold. This is an important setting which deploys multi-robot teams to the real world. In fact, the closer the robots are to each other, the greater the impact is on each other. This is key information that should be embedded in the graph, which greatly facilitates cooperation between the robots. Therefore, the attention mechanism in GACM is used to enable the robots to selectively attend to messages coming from their neighbors.

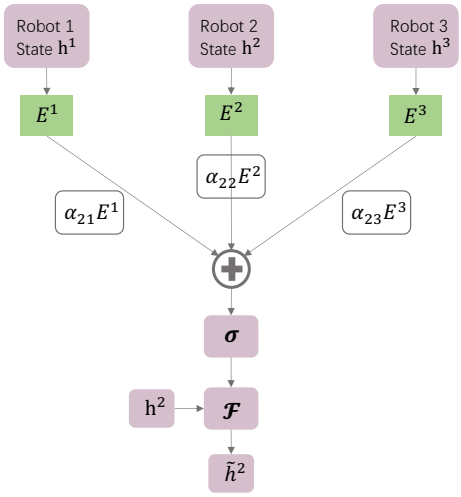


Fig. 2. Graph attention communication mechanism.

In the following, GACM is introduced in details through a simple task where a team of 3 robots is required to cooperatively encircle a target. As shown in Fig. 2, robot 2 communicates with its neighboring robots, robot 1 and robot 3. The communicating information of robot i is encoded as h^i, which represents the robot's understanding of its own state and the environment. This encoding does not contain any cooperative information about the multi-robot team. Firstly, the communicating encoding of robot i is transformed to transition encoding, i.e. $E^i = \boldsymbol{W}h^i$, where $\boldsymbol{W}$ is a learnable parameter matrix. Based on E^i, the attention weights computed can be expressed as:

$$\alpha_{ij} = \begin{cases} \frac{exp(LeakyReLU(\boldsymbol{a}^T[E^i||E^j]))}{\sum_{q\in N_i} exp(LeakyReLU(\boldsymbol{a}^T[E^i||E^q]))} & if \quad A_{ij} = 1 \\ 0 & otherwise \end{cases}, \tag{4}$$

where $\boldsymbol{a}^T$ is a learnable parameter vector, $\boldsymbol{A} = \{A_{ij}\}$ is the adjacency matrix of the graph, $||$ is the concatenation operation, $LeakyReLU$ is a nonlinear activation function. If two robots can communicate with each other, that is $A_{ij} = 1$, α_{ij} for nodes $j \in N_i$, where N_i is some neighborhood of node r_i in the graph, is obtained by calculating. Otherwise, $\alpha_{ij} = 0$. Then, all communicative messages are aggregated through computing a weighted sum of its neighbors' transition encoding, i.e. $E^i_{com} = \sigma(\sum \alpha_{ij} E^j)$, where σ is a nonlinear activation function. Finally, the robot i updates its state in-

formation as $\tilde{h}^i$ by a non-linear transformation of its current state information h^i concatenated with E^i_{com} through using a neural network $\mathcal{F}$. This $\tilde{h}^i$ implicitly encodes cooperation information between the robots. Furthermore, we set K-hop communication [19] to enlarge the receptive field of the robots. The K-hop communication is simply expressed as $h^i(1) \rightarrow GACM \rightarrow \tilde{h}^i(1) \rightarrow GACM \rightarrow \tilde{h}^i(2) \rightarrow ... \rightarrow \tilde{h}^i(K)$, which enables each robot to cooperate with more robots that are not in its communication range.

B. Handling Uncertain Number of Obstacles with LSTM

Recall that the RL training process aims to find an optimal policy, $\pi_i : \boldsymbol{s}_i(t) \mapsto \boldsymbol{a}_i(t)$, which maps from available sates of robot i to a probability distribution across actions. The available states contain the state of the target, the state of the robot itself and the states of obstacles in the environment. Specially the number of obstacles is uncertain in the environment, which has a great impact on the behavior of the robots. To solve this problem, LSTM is adopted to process the information of uncertain number of obstacles [20]. As shown in Fig. 3, the states of obstacles is fed into LSTM in reverse order sorted by distance to the robot, which means that the closest obstacle has the biggest impact on the robot. The final hidden state of LSTM, $\tilde{\boldsymbol{s}}^o$, is used as the output of LSTM, which is a fixed-length vector. Moreover, the output of LSTM implicitly contains the encoded information of all obstacles.

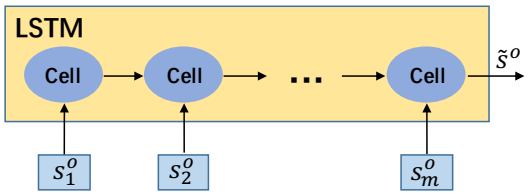


Fig. 3. LSTM module.

C. Reward Function

The design of the reward function is an especially important step in RL. It should be designed according to the specific task, and an appropriate reward function ensures the task to be completed well. Recall that the problem of multi-robot cooperative target encirclement is defined as **Definition 1**. There are specific constraints shown in (1) in this problem. Specifically, a multi-robot team needs to not only satisfy these constraints about target encirclement, but also avoid collisions. Meanwhile, the cooperation of the multi-robot team need to be promoted. Therefore, we design a compound reward function, $R_i(\boldsymbol{s}^{all}(t), \boldsymbol{a}(t))$, which contains three parts.

Firstly, the collision avoidance reward for each robot i is defined as:

$$R_i^c = \begin{cases} -2 & if \quad d_{min} < 0 \\ k_1 \cdot (d_{min} - D) & if \quad d_{min} < D \\ 0 & otherwise \end{cases}, \tag{5}$$

where k_1 is a hyper parameter, D is the threshold of an uncomfortable distance between the robots, or between the robots and obstacles, d_{min} is the distance closest to other entity (other robots and obstacles). Next, we use the following function to define the target encirclement radius reward, i.e.,

$$R^\rho = -clip(\frac{1}{n}\sum_{i=1}^{n}(||\boldsymbol{p}_i^r - \boldsymbol{p}^T|| - \rho), 0, 2), \tag{6}$$

where the $clip$ operation makes this reward in the range of 0 to 2. This radius reward can enable the multi-robot team satisfy the first equation in (1). Finally, we define the reward function R^d that indicates the distance difference between neighboring robots,

$$R^d = -clip(\frac{1}{n(n-1)}\sum_{i}^{n}\sum_{j}^{n}(||\boldsymbol{p}_i^r - \boldsymbol{p}_j^r|| - d), 0, 2), \tag{7}$$

where $i \neq j \in \mathbb{U}$. This reward can enable the multi-robot team to satisfy the second constraint in (1). Therefore, the compound reward function is obtained as follows:

$$R_i = R_i^c + R^\rho + R^d. \tag{8}$$

D. Distributed Transferable Policy Network framework

Based on GACM described in Section III-A, LSTM module given in Section III-B and the reward function designed in Section III-C, the distributed transferable policy network framework is designed as Fig. 4. The extract network is used to extract all information of robot i and its environment, including the target, obstacles, and cooperation with its neighboring robots. Specifically, the state of the target $\boldsymbol{s}^T$ is fed into the Encoder module which is a fully-connected (FC) layer, and the output of the Encoder module is $\tilde{\boldsymbol{s}}^T$ which implicitly encodes the future state of the target. Next, $\boldsymbol{s}_i^{jn} = [\boldsymbol{s}_i^r, \tilde{\boldsymbol{s}}^T, \tilde{\boldsymbol{s}}^o]$ is fed into the Process module which is a FC layer, and the output of the Process module is $\tilde{\boldsymbol{s}}_i^{jn}$ which represents the state obtained by processing all information of the robot. Then, $\tilde{\boldsymbol{s}}_i^{jn}$ is fed into the Broadcast module which is a FC layer, and the output is h^i which represents the communicating information of robot i. Meanwhile, h^i is used as the input of GACM, and $\tilde{h}^i$ containing cooperative information is obtained in the output of the GCAM. Finally, the $\tilde{s}_i^{jn}$ is concatenated with $\tilde{h}^i$ to form $\boldsymbol{s}_i^l$, which contains the knowledge of the information of the robot and its environment. Subsequently, $\boldsymbol{s}_i^l$ is used as the input of the actor network and the critic network. The output of the actor network is a discrete probability distribution over actions, and the output of the critic network is a scalar value. In addition, the critic network only works in the training stage, and it is used to judge the behavior of the actor. Furthermore, the designed reward function is used to lead the training of the behavior of target encirclement with collision avoidance.

In this paper, the multi-robot system shares all the learnable parameters including the actor network, critic network and extract network. Since each robot receives different states containing its own state, the states of obstacles, and the information of cooperation with its neighboring robots, sharing

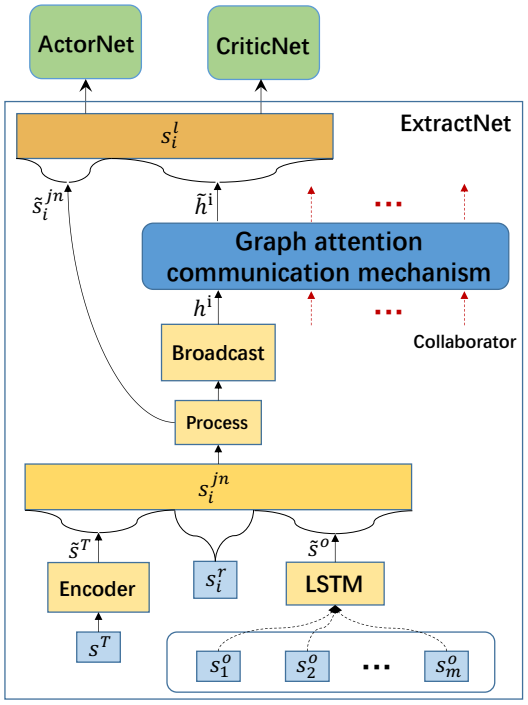


Fig. 4. Distributed Transferable Policy Network Framework.

parameters does not stop them from behaving differently. Moreover, The proposed policy network is trained and executed both in a decentralized manner. Furthermore, the proposed policy network framework is invariant to the number of the robots and obstacles, and is able to transfer to environments with different number of the robots and obstacles.

E. Curriculum Learning

Robots learn much better when examples are not randomly presented but organized in a meaningful order which illustrates gradually more concepts, and gradually more complex ones. Formalizing such training strategies calls curriculum learning [21]. In fact, an appropriate curriculum strategy acts to help the training process (faster convergence to better solutions) and enable the policies of the robots to have good generalization. In this paper, our proposed policy network framework can be applied to tasks with arbitrary number of robots and obstacles, and the robots share the network parameters. Therefore, this enables us to directly use a policy π trained for a task $\mathcal{Q}$ with N robots and M obstacles to a different task $\mathcal{Q}^{'}$ with $N^{'}$ robots and $M^{'}$ obstacles [19]. The policy π with a good initialization for task $\mathcal{Q}^{'}$ can be improved further to achieve the task. the robots firstly learn to cooperatively encircle a target in a small team, and then learn to achieve this goal in a large team with the addition of new members. In other words, the robots apply their previous knowledge to a new scenario and gradually learn complex cooperative policies in a large team. Moreover, the speed of the moving target is gradually improved in the process of the curriculum learning.

IV. Simulations

A. Simulation Settings

In order to verify the effectiveness and generalization of the proposed algorithm, we conduct two tasks for multi-robot teams to accomplish it: 1) encircle a stationary target; 2) encircle a moving target. The difficulty of these tasks is from low to high. We have implemented them in a target encirclement simulation environment designed based on the multi-agent particle environment [11]. The simulation environment is set to be in 2D space where a multi-robot team with a double integrator dynamics model can move to encircle a stationary or moving target while avoiding randomly placed static obstacles. Specifically, each robot is controlled by our proposed algorithm. The action space of each robot is discrete, and the robot can accelerate and decelerate in X and Y directions. The maximum velocity of each robot is set to 1.0 $(m \cdot s^{-1})$. In addition, the initial position of the target is randomly placed. For the stationary target, its position is not changed, while the moving target has its escaping strategy. As show in Fig. 5, green circles represent robots, and a red circle represents a target. Robots, R_1, R_2, R_3, R_4, cooperatively encircle the target T. By calculating the angle between the neighboring robots and itself, the target chooses the middle position $(R_{2,3})$ of the encircling robots corresponding to the largest angle θ as the escape direction (T to $R_{2,3}$). Moreover, if there are multiple middle positions (multiple the same largest angles), the target randomly selects one of them.

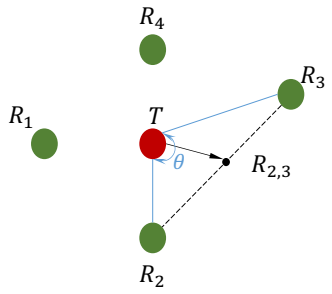


Fig. 5. Illustration of the target escaping strategy.

B. Implementation Specifications

In the design of the distributed network (see Fig. 4), the LSTM module takes as input the 4-dim obstacle state $(\boldsymbol{s}_k^o)$ and outputs a 10-dim embedding $(\tilde{\boldsymbol{s}}^o)$. The Encoder module inputs the 4-dim target state $(\boldsymbol{s}^T)$ and outputs a 20-dim embedding $(\tilde{\boldsymbol{s}}^T)$. The Process and Broadcast modules take as input a 35-dim vector $(\boldsymbol{s}_i^{jn})$ and a 32-dim embedding $(\tilde{\boldsymbol{s}}_i^{jn})$ respectively, and output a 32-dim embedding and a 128-dim embedding (h^i) respectively. Meanwhile, in GACM, we set the learnable parameter matrix $\boldsymbol{W} \in \mathcal{R}^{128X128}$ and the learnable parameter vector $\boldsymbol{a} \in \mathcal{R}^{256X1}$. Therefore, $\tilde{h}^i$ is a 128-dim vector. Finally, the actor and critic networks are both two FC networks, and output a 4-dim action probability distribution and a scalar value respectively. We use $K = 3$ communication

hops between the robots. For restricted communication, the communication distance between the robots is set to 1 m. The simulation environments are 2×2 square meter in size. In addition, for the reward function, the encirclement radius ρ is set to 0.24 m, and $D = 0.04$, $k_1 = 15$.

In the training phase, each episode lasts up to 50 steps. each network parameter update is performed through PPO after accumulating experience for total 4096 steps (128 steps on 32 parallel process). Evaluation is implemented on 100 episodes after every 50 updates. The learned policy is tested on 500 episodes in new seed. Each robot performs greedy action selection in evaluation and test [19].

C. Results

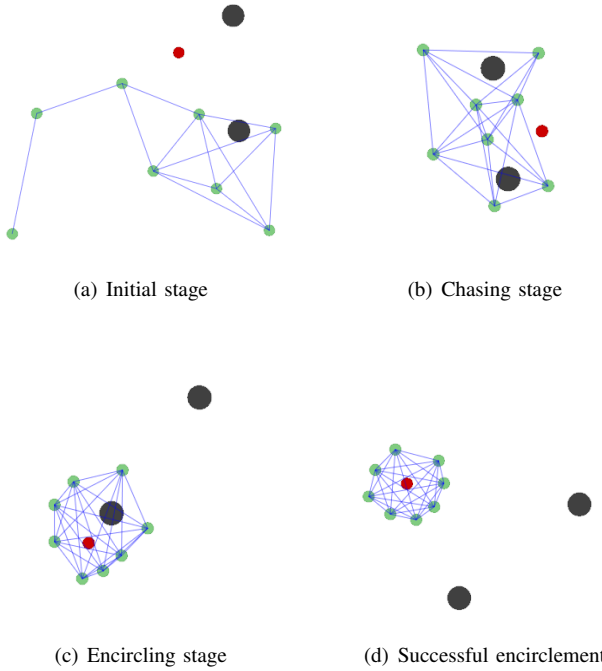
To fully evaluate the effectiveness of the proposed method, we conduct a task where an eight-robot team ($n = 8$) to cooperatively encircle one stationary or moving target in two static obstacles environments ($m = 2$). The initial positions of robots, target and obstacles are all random. Since it's difficult for the robots to learn this task directly, curriculum learning is adopted to enable the team to learn to complete this task. Meanwhile, we set up several evaluation metrics:

- Success rate (S%): Percentage of this task completed in evaluation or test episodes.
- Mean per-step reward (MPR): Mean of average rewards for each step of the robots in evaluation or test episodes.
- Mean episode length (MEL): Mean of successful episode length in evaluation or test episodes.

In addition, the number of the policy network updates (UN) also needs to be noted.

Curriculum learning and test for a stationary target encirclement are shown in Table. I. We design a curriculum with the increasing number of robots. A policy is first trained with a two-robot team ($n = 2$). Once the team reaches a threshold of success rate (90%), the learned policy is transferred to a team with $n + 1$ robots to continue training. The process is repeated until an eight-robot team has learned to cooperatively encircle a stationary target. After that, we test the policy learned through the curriculum learning. As we expected, the team cooperatively encircles a stationary target with a high success rate while avoiding collisions.

Curriculum learning and test for a moving target encirclement are shown in Table. II. A training curriculum is designed with the increase of the number of robots and the speed of the moving target. Compared to encircling a stationary target, it is a very difficult and complex task for a multi-robot team to encircle a moving target that has escaping strategy. Through this curriculum, a policy is obtained for a eight robots team encircling the target with a speed of 0.6. Meanwhile, the test results show that this policy enables the team to successfully encircle a target that is escaping at a high speed. In addition, the process of target encirclement for an eight robots team is shown in Fig. 6. Firstly, the initial stage of target encirclement is shown in Fig. 6.(a). In this stage, although there is less communication between the robots, an interaction graph has been formed. It enables this team to develop a sense of cooperation. Then, the chasing stage is shown in Fig. 6.(b). The team that is gradually closing to the target is flexibly avoiding collisions with obstacles. Communication in the team is gradually strengthened. Next, the encircling stage is shown in Fig. 6.(c) and Fig. 6.(d). The former shows that the team encircles the target and obstacle when the target hides near an obstacle. It can reflect the high intelligence and cooperation of the team. The latter indicates that the team has successfully encircled the target. Based on the above simulation results, the effectiveness of the proposed algorithm is fully verified.



(a) Initial stage (b) Chasing stage

(c) Encircling stage (d) Successful encirclement

Fig. 6. The illustration of an eight robots team encircling a moving target that its speed is 0.6 in an environment with two obstacles. The green circles, red circle and black circles are the robots, the target and obstacles at the current time respectively. The blue line connecting the two robots represents the communication between the two robots.

D. Method Generalization

In order to verify the generalization of the proposed algorithm, we evaluate a policy trained for five robots and two obstacles without any fine-tuning on different robot numbers and obstacle numbers. The generalization results is shown in Table. III. As we expected, the learned policy shows satisfactory generalization success rate. Since the policy is obtained through the curriculum learning with the addition of new robots (2 to 5), teams with less than five robots have better success rates than teams with more than five robots. Meanwhile, the learned policy has good adaptability in environments with different obstacle numbers. Furthermore, the results show that our proposed method is transferable and enables the multi-robot team to cooperatively accomplish a complex encirclement task. This owes to our proposed distributed transferable policy network framework.

TABLE I
CURRICULUM LEARNING AND TEST FOR A STATIONARY TARGET ENCIRCLEMENT.

phase	n = 2		n = 3		n = 4		n = 5		n = 6		n = 7		n = 8	
Curriculum learning	S%	UN	S%	UN	S%	UN	S%	UN	S%	UN	S%	UN	S%	UN
	97	400	96	900	93	150	93	1750	91	150	91	950	**92**	**7600**
Test	n = 8, target speed = 0.0													
	S%				MPR						MEL			
	95.4				**-0.223**						**16.88**			

TABLE II
CURRICULUM LEARNING AND TEST FOR A MOVING TARGET ENCIRCLEMENT.

phase		n = 2		n = 3		n = 4		n = 5		n = 6		n = 7		n = 8	
Curriculum learning	target speed	S%	UN	S%	UN	S%	UN	S%	UN	S%	UN	S%	UN	S%	UN
	0.0	97	400	96	900	93	150	93	1750	91	150	91	950	92	4600
	0.2	100	50	100	50	99	50	98	0	93	0	93	50	97	200
	0.4	100	50	96	50	96	50	94	50	94	100	91	50	97	600
	0.6	99	50	97	100	95	50	96	50	93	50	96	50	**100**	**12200**
Test	n = 8, target speed = 0.6														
	S%				MPR						MEL				
	99.4				**-0.244**						**27.28**				

TABLE III
GENERALIZATION RESULTS. THE MOVING TARGET SPEED IS 0.4.

S% Ro[1] / Ob[2]	n = 3	n = 4	**n = 5**	n = 6	n = 7
2	89.0	99.8	**99.8**	96.0	83.8
3	86.0	98.8	99.6	95.2	83.0
4	84.2	99.6	99.8	94.0	80.0

[1] Number of robots in the team.
[2] Number of obstacles.

V. CONCLUSION

In this work, we have presented a distributed transferable policy network framework to tackle the problem of multi-robot cooperative target encirclement with collision avoidance. Specifically, GACM is adopted to describe multi-robot interactions as a graph and extract cooperative information from the graph. LSTM is used to process the information of uncertain number of obstacles. In addition, we design a compound reward for target encirclement and collision avoidance. Under the guidance of this reward function, the distributed policy network is trained through curriculum learning, which speeds up the process of the training. Simulation results validate the effectiveness and generalization of the proposed algorithm. Meanwhile, various simulations show that our proposed policy network has the property of transferability and enables multi-robot systems to have intelligent cooperative behaviors.

ACKNOWLEDGMENT

This work was supported by the National Key Research and Development Program of China under Grant 2018AAA0101005, 2018AAA0102404 and Innovation Academy for Light-duty Gas Turbine, Chinese Academy of Sciences, No. CXYJJ19-ZD-02.

REFERENCES

[1] G. Antonelli, F. Arrichiello, and S. Chiaverini, "The entrapment/escorting mission for a multi-robot system: Theory and experiments," in *2007 IEEE/ASME international conference on advanced intelligent mechatronics*, Sep. 2007, pp. 1–6.

[2] A. Hafez, M. Iskandarani, S. Givigi, S. Yousefi, and A. Beaulieu, "Uavs in formation and dynamic encirclement via model predictive control," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 1241 – 1246, 2014, 19th IFAC World Congress.

[3] A. Sarwal, D. Agrawal, and S. Chaudhary, "Surveillance in an open environment by co-operative tracking amongst sensor enabled robots," in *2007 International Conference on Information Acquisition*, July 2007, pp. 345–349.

[4] T.-H. Kim, S. Hara, and Y. Hori, "Cooperative control of multi-agent dynamical systems in target-enclosing operations using cyclic pursuit strategy," *International Journal of Control*, vol. 83, no. 10, pp. 2040–2052, 2010.

[5] Y. Shi, R. Li, and K. Teo, "Cooperative enclosing control for multiple moving targets by a group of agents," *International Journal of Control*, vol. 88, no. 1, pp. 80–89, 2015.

[6] H. Shen, N. Li, S. Rojas, and L. Zhang, "Multi-robot cooperative hunting," in *2016 International Conference on Collaboration Technologies and Systems (CTS)*, Oct 2016, pp. 349–353.

[7] C. Wang, G. Xie, and M. Cao, "Controlling anonymous mobile agents with unidirectional locomotion to form formations on a circle," *Automatica*, vol. 50, no. 4, pp. 1100 – 1108, 2014.

[8] F. Chen, W. Ren, and Y. Cao, "Surrounding control in cooperative agent networks," *Systems & Control Letters*, vol. 59, no. 11, pp. 704 – 712, 2010.

[9] Y. J. Shi, R. Li, and K. L. Teo, "Rotary enclosing control of second-order multi-agent systems for a group of targets," *International Journal of Systems Science*, vol. 48, no. 1, pp. 13–21, 2017.

[10] B. Liu, Z. Chen, H. Zhang, X. Wang, T. Geng, H. Su, and J. Zhao, "Collective dynamics and control for multiple unmanned surface vessels," *CoRR*, vol. abs/1905.01215, 2019.

[11] R. Lowe, Y. WU, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 6379–6390.

[12] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," *AAAI Conference on Artificial Intelligence*, 2018.

[13] Y. Yang, R. Luo, M. Li, M. Zhou, W. Zhang, and J. Wang, "Mean field multi-agent reinforcement learning," *CoRR*, vol. abs/1802.05438, 2018.
[14] J. Jiang and Z. Lu, "Learning attentional communication for multi-agent cooperation," in *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018, pp. 7254–7264.
[15] J. Ma, H. Lu, J. Xiao, Z. Zeng, and Z. Zheng, "Multi-robot target encirclement control with collision avoidance via deep reinforcement learning," *Journal of Intelligent & Robotic Systems*, Nov 2019.
[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[17] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *ArXiv*, vol. abs/1707.06347, 2017.
[18] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," *ArXiv*, vol. abs/1710.10903, 2017.
[19] A. Agarwal, S. Kumar, and K. P. Sycara, "Learning transferable cooperative behavior in multi-agent teams," *ArXiv*, vol. abs/1906.01202, 2019.
[20] Z. Sui, Z. Pu, J. Yi, and T. Xiong, "Formation control with collision avoidance through deep reinforcement learning," in *2019 International Joint Conference on Neural Networks (IJCNN)*, July 2019, pp. 1–8.
[21] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. New York, NY, USA: ACM, 2009, pp. 41–48.