2022 IEEE International Conference on Robotics and Automation (ICRA)
May 23-27, 2022. Philadelphia, PA, USA

Multi-Target Encirclement with Collision Avoidance via Deep Reinforcement Learning using Relational Graphs*

Tianle Zhang[1,2], Zhen Liu[1,2](✉), Zhiqiang Pu[1,2] and Jianqiang Yi[1,2]

Abstract— **In this paper, we propose a novel decentralized method based on deep reinforcement learning using robot-level and target-level relational graphs, to solve the problem of multi-target encirclement with collision avoidance (MECA). Specifically, the robot-level relational graphs, composed of three heterogeneous relational graphs between each robot and other robots, targets and obstacles, are modeled and learned through using graph attention networks (GATs) for extracting different spatial relational representations. Moreover, for each target within the observation of each robot, a target-level relational graph is built with GAT to construct spatial relations from the robot. Furthermore, the movement of each target is modeled by the target-level relational graph and learned through supervised learning for predicting the trajectory of the target. In addition, a knowledge-embedded compound reward function is defined to solve the multi-objective problem in MECA, and guide the policy learning for deriving the behavior of MECA. An actor-critic training algorithm based on the centralized training and decentralized execution framework is adopted to train the policy network. Simulation and real-world experiment results demonstrate the effectiveness and generalization of our method.**

I. INTRODUCTION

Multi-robot systems have received increasing attention from researchers owing to their broad applications, such as collaborative patrolling, formation control, robot navigation, autonomous search and rescue [1]–[4]. Among research topics of multi-robot systems, encirclement control has attracted the interest of many researchers because of its promising applications in civil and military fields. The applications include collaborative escorting, capture of the enemy target, reconnaissance and surveillance, patrolling and hunting with unmanned surface vessels [5]–[7], etc. The core problem of these applications is how to control a multi-robot system to cooperatively encircle multiple targets with circular formations while avoiding collisions. Specifically, the multi-robot system needs to form multiple robot groups to encircle all targets while avoiding collisions, where each target is encircled by a robot group with a circular formation. Since the multi-target encirclement with collision avoidance (MECA) problem involves multi-target assignment, target encirclement and collisions avoidance subproblems simultaneously, it remains great challenges, especially for decentralized multi-robot systems (DMSs).

The existing methods for multi-robot target encirclement can be roughly classified into two categories, control theory based methods and learning based methods. The former mostly focus on single-target scenarios [8], [9]. There are some works on multi-target scenarios [10], [11]. But, these works only treat all targets as one target by estimating the center of all the targets. Ma et al. [12] integrates task allocation, encirclement control and artificial potential fields methods to solve the MECA problem. However, this simple integration approach cannot generate optimal solutions due to lacking overall coordinate manners. Besides, most control theory based methods do not consider the collision avoidance problem, and highly depend on the precise control model, which are not always available in practice.

Due to the limitation of the control theory based methods, learning based methods show great potential by introducing deep reinforcement learning (DRL) [13]–[15] to address the target encirclement problem. Recently, Ma et al. [16] designs a centralized DRL method to enable a multi-robot system to encircle a target and avoid collisions at the same time. Unfortunately, the centralized method lacks adaptability to new environments, and is infeasible in large-scale robot teams. Zhang et al. [17] proposes a distributed transferable policy network framework based on DRL, which adopts a graph communication mechanism for robot interaction, to solve the problem of single-target encirclement with collision avoidance. But, communications among robots are not always feasible in practice due to communication delay and limitations. Besides, most learning based methods focus on single-target scenarios, and cannot be implemented on multi-target scenarios due to lack of multi-target assignment. Meanwhile, they do not consider predicting the target's moving trajectories, which will lead to the failure of the task when the target has a strong escape strategy. Moreover, in these methods, each robot receives information in the form of a lumped vector by stacking everything, without utilizing the natural spatial structures of environments.

In reality, a multi-robot system can be naturally described as a graph structure. Moreover, in the MECA problem, there are three types of entities, i.e., robots, targets and obstacles. Different entities have different influence relations on each robot, e.g., avoiding obstacles, encircling targets and cooperating with other robots. Therefore, each robot and the entities belonging to the same type should be modeled as a graph to represent their special spatial influence relations.

*This work was supported by the National Key Research and Development Program of China (No. 2018AAA0102402 and No. 2018AAA0101005), the Strategic Priority Research Program of Chinese Academy of Sciences (No. XDA27030403), the National Natural Science Foundation of China (No. 62073323), the External cooperation key project of Chinese Academy Sciences (No. 173211KYSB20200002), and the Science and Technology Development Fund of Macau (No.0025/2019/AKP).

[1]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China.

[2]Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: tianle-zhang@outlook.com, liuzhen@ia.ac.cn, zhiqiang.pu@ia.ac.cn, jianqiang.yi@ia.ac.cn).

2022 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9681-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICRA46639.2022.9812151

Motivated by the aforementioned discussions, we propose a new decentralized method based on DRL using robot-level and target-level relational graphs (RGs), named MECA-DRL-RG, to address the problem of MECA for DMSs. Specifically, to extract different spatial relational representations, the robot-level RGs, composed of three heterogeneous relational graphs between each robot and other robots, targets and obstacles, are modeled and learned through using graph attention networks (GATs). Moreover, the target-level RG is built with GAT to construct spatial relations from the robots to each target. Furthermore, to predict the trajectory of each target. the movement of the target is modeled by the target-level RG and learned through supervised learning. Besides, a knowledge-embedded compound reward function is defined to tackle the multi-objective problem in MECA. In this paper, the main contributions are listed as follows:

1): Differing from simply integrating the methods of independent subproblems, a novel decentralized method based DRL using robot-level and target-level RGs is proposed to solve the MECA problem. **2):** The robot-level RGs are modeled and learned by using GATs for extracting different spatial relational representations, instead of roughly stacking the observed information. **3):** The movement of each target is modeled by the target-level RG constructed with GAT and learned through supervised learning, for predicting the target' trajectory. **4):** A knowledge-embedded reward function is defined to solve the multi-objective problem of MECA.

II. Problem Formulation

As shown in Fig. 1, a MECA task, where a multi-robot system with N robots (green circles) need to cooperatively encircle K ($1 < K < N$) stationary or moving targets (red circles) in an environment with L static obstacles (black circles), is investigated in this paper. For simplicity, we model the geometry of the robots, targets, and obstacles as a disc with a radius. Mathematically, $p_k^a = [p_{kx}^a, p_{ky}^a]$, $p_i^r = [p_{xi}^r, p_{yi}^r]$, $p_l^b = [p_{xl}^b, v_{yl}]$ denote the positions of target k, robot i and obstacle l respectively, and $v_k^a = [v_{xk}^a, v_{yk}^a]$, $v_i^r = [v_{xi}^r, v_{yi}^r]$ denote the velocities of target k and robot i respectively, where $i = 1, ..., N$, $k = 1, ..., K$ and $l = 1, ..., L$. Meanwhile, each robot can only observe the positions of the targets, other robots and obstacles within its visual area with radius D^O, and the communication among the robots is prohibited. In the MECA task, all robots in the multi-robot system need to automatically form multiple groups to encircle all targets, in which each group is required to form a circular formation to encircle one independent target while avoiding collisions. This involves the following three subproblems.

1) Dynamic Multi-Target Assignment and Forming Groups:

In the MECA task, each robot can only encircle one target at a time, and one target need to be encircled by multiple different robots. To efficiently complete the task, each target should be reasonably assigned to robots at each moment. The dynamic multi-target assignment subproblem [18] can be addressed by maximizing the objective $\sum_{i=1}^{N}\sum_{k=1}^{K} U_{ik}X_{ik}$ under the constraints: 1) $\sum_{i=1}^{N} X_{ik} \geq B, k = 1, ..., K$; 2) $\sum_{k=1}^{K} X_{ik} = 1, i = 1, ..., L$; 3) $X_{ik} \in \{0, 1\}$, where U_{ik} represents the utility of assigning target k to robot i, B represents the minimum number of robots required to encircle one target, $X_{ik} = 1$ if target k is assigned to robot i, and 0 otherwise. In this paper, we assume that $B = 3$ and $U_{ik} = -\|p_i^r - p_k^a\|$. Finally, through the dynamic multi-target assignment, each target can be assigned to at least three robots, and the robots assigned to the same target automatically form a group.

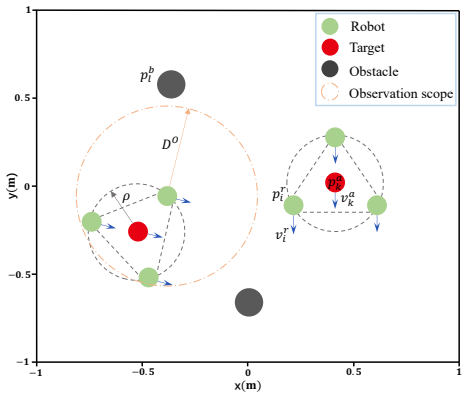


Fig. 1. Illustration of MECA for a decentralized multi-robot system.

2) Target Encirclement of Each Group:

To realize the target encirclement of each group, the circular formation should satisfy two requirements as much as possible [19]: 1) The formation should be a convex polygon surrounding the target, and the distance between adjacent vertices in the convex polygon should be the same as possible; 2) In the formation, each vertex of the convex polygon can be occupied by any robot. Based on the above requirements, the target encirclement subproblem of each group is defined as follow [20],

Definition 1: The position $p_k^a(t)$ of target k is equally encircled by h-th group $\mathcal{G}_h$ composed of n ($n \geq B$) robots of the complete position distribution $p_i^r(t)$, $i \in \mathcal{G}_h$, if

$$\begin{aligned} &lim_{t\to+\infty}\|\boldsymbol{p}_i^r(t) - \boldsymbol{p}_k^a(t)\| = \rho, \\ &lim_{t\to+\infty}\|\boldsymbol{p}_i^r(t) - \boldsymbol{p}_j^r(t)\| \geq d, \quad j \in \mathcal{G}_h \ and \ j \neq i, \end{aligned} \tag{1}$$

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

3) Collision Avoidance:

In the MECA task, each robot needs to not only avoid collisions with obstacles, but also avoid collisions with targets and other robots.

This MECA task can be formulated as a partially observable Markov decision process [21] in a reinforcement learning framework. At each timestep, robot i can obtain a partial observation $o_i = [s_i^s, s_i^o]$ composed of itself state $s_i^s = [p_i^r, v_i^r]$ and an observation state $s_i^o = [s_i^{ao}, s_i^{ro}, s_i^{bo}]$, where $s_i^{ao} = \{p_k^a | k \in \mathcal{N}_i\}$, $s_i^{ro} = \{p_j^r | j \in \mathcal{N}_i\}$, $s_i^{bo} = \{p_l^b | l \in \mathcal{N}_i\}$ represent the observed states of the targets, other robots and obstacles within the visual range of robot i respectively, $j = 1, ..., N$ and $j \neq i$, N_i is some neighborhood within the visual area of robot i. The dynamics of each robot is modeled as a double integrator, and the action of robot i

denotes $a_i = [F_{xi}, F_{yi}]$, where F_{xi}, F_{yi} represent the force applied to the robot in x and y directions respectively. This paper aims to design an optimal policy $\pi_i : o_i \rightarrow a_i$ for robot i to complete the MECA task which simultaneously involves the three subproblems.

III. METHOD

A. Overall Structure

The overall structure of the proposed method mainly consists of three components as shown in Fig. 2: 1) a designed decentralized policy network structure using robot-level and target-level RGs, in which heterogeneous relational graph learning using robot-level RGs is presented to extract different spatial relational representations, and target strategy modeling using target-level RGs is given to predict the targets' trajectories; 2) a defined knowledge-embedded compound reward function, which guides the policy learning of robots to derive the behavior of MECA; 3) an actor-critic training algorithm, which trains the policy network for completing the MECA task.

B. Decentralized Policy Network Structure using Relational Graphs

The decentralized policy network structure is a coupled actor-critic structure. The actor parameterized by θ, $\pi_i^\theta : o_i \mapsto a_i$, composed of state representation and policy head networks, takes partial observation of robot i as input and outputs action values of robot i for making decisions. The critic parameterized ϕ, $v_i^\phi : o \times \mathcal{D} \mapsto \mathbb{R}$, composed of state representation and value head networks, takes partial observations $o = [o_i, ..., o_N]$ of all robots and the identity $d_i \in \mathcal{D}$ of robot i as inputs and outputs a scalar value for the actor training. Especially, the state representation module using RGs is specially designed to represent the partial observation of each robot from the environment. The state representation module consists of heterogeneous relational graph learning using robot-level RGs and target strategy modeling using target-level RGs.

1) Heterogeneous Relational Graph Learning using Robot-Level Relational Graphs: In the MECA task, there are three different types of entities in the observation scope of robot i, i.e., other robots, targets and obstacles. Different types of entities have different spatial influence relations on robot i. Therefore, to represent these different relations, we build three relational graphs for robot i : 1) robot-target RG $\mathcal{G}_i^a := (V_i^a, E_i^a)$; 2) robot-robot RG $\mathcal{G}_i^r := (V_i^r, E_i^r)$; 3) robot-obstacle RG $\mathcal{G}_i^b := (V_i^b, E_i^b)$, where each node in $\mathcal{G}_i^a$, $\mathcal{G}_i^r$, $\mathcal{G}_i^b$ is the target, the other robot and the obstacle within the visual area of robot i respectively (or robot i), and their edges are pointed to robot i from the targets, the other robots and the obstacles respectively. These RGs are used to construct the spatial influence relations of other entities on the robot, called robot-Level RGs. The three heterogeneous relational graphs are modeled and learned through the following three stages.

Encoding: At each timestep, each robot can obtain a partial observation $o_i = [s_i^s, s_i^o]$, $s_i^o = [s_i^{ao}, s_i^{ro}, s_i^{bo}]$. We can divide

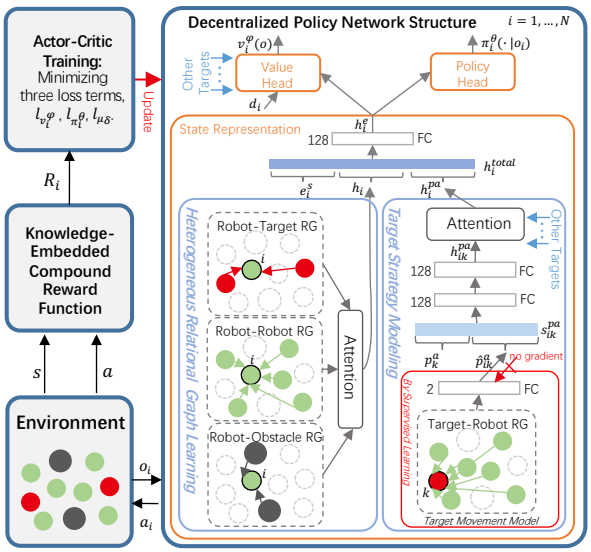


Fig. 2. Overall structure of MECA-DRL-RG.

the partial observation into three categories, i.e., $\psi_i^a, \psi_i^r, \psi_i^b$, which represent the target set, other robot set and obstacle set. Firstly, these observed states from different entities are encoded as different embeddings by using four different fully-connected (FC) layer networks, i.e., $e_i^s = f_1(s_i^s)$, $e_i^k = f_2(p_k^a)$, $e_i^j = f_3(p_j^r)$, $e_i^l = f_b(p_l^b)$, where $k \in \psi_i^a, j \in \psi_i^r, l \in \psi_i^b$.

Graph Modeling: Next, based on these embeddings, the three RGs are modeled through three graph attention networks [22], which are the same structure but different parameters. We take robot-target RG as an example to illustrate the modeling process. Firstly, in the robot-target RG, the weight coefficient of target k to robot i is calculated as,

$$\alpha_i^k = \frac{exp(\sigma_0(\boldsymbol{a}_a^T[\boldsymbol{W}_a e_i^s \| \boldsymbol{W}_a e_i^k]))}{\sum_{q \in \psi_i^a} exp(\sigma_0(\boldsymbol{a}_a^T[\boldsymbol{W}_a e_i^s \| \boldsymbol{W}_a e_i^q))}, \tag{2}$$

where $\boldsymbol{a}_a$ is a learnable parameter vector, $\boldsymbol{W}_a$ is a learnable parameter matrix, and σ_0 is *LeakyReLU* activation function. Then, according to these coefficients, relational embedding h_i^a of all neighbor targets on robot i can be obtained, $h_i^a = \sigma(\sum_{q \in \psi_i^a} \alpha_i^q * \boldsymbol{W}_a e_i^q)$, where σ is *ReLU* activation function. Similarly, in the robot-robot RG and the robot-obstacle RG, the relational embeddings h_i^r, h_i^b of all other robot and obstacles on robot i can be calculated.

Attention Aggregation: To enable robot i to selectively centralize these RGs, an attention network $\mathcal{F}_1$ with two FC layers is used to aggregate these relational embeddings. An attention coefficient for the robot-target relational embedding is obtained, $\alpha_i^{ha} = softmax(\mathcal{F}_2(e_i^s \| h_i^a)))$. Similarly, the attention coefficients $\alpha_i^{hr}, \alpha_i^{hb}$ of robot-robot and robot-obstacle embeddings can be obtained. Finally, the total relational embedding h_i is obtained by computing a weighted sum of all the relational embeddings, $h_i = \alpha_i^{ha} h_i^a + \alpha_i^{hr} h_i^r + \alpha_i^{hb} h_i^b$, which implicitly encodes the spatial influence relation of the environment on robot i.

2) Target Strategy Modeling using Target-Level Relational Graphs: The target strategy modeling mainly contains target

movement modeling and target trajectory predicting.

Target Movement Modeling: Note that the robots and targets mentioned below are in the partial observation o_i of robot i. Firstly, a target movement model network parameterized by δ, $\mu_\delta(\cdot) : o_i \mapsto \mathbb{R}^2$, is built to predict the positions of targets at next timestep. In the movement network, GAT is utilized to model a target-robot RG $\mathcal{G}^r_{ik} := (V^r_{ik}, E^r_{ik})$, where each node denotes target k or a robot, and there exists an edge between target k and each robot. This RG constructs spatial relations from the robots to each target, called target-level RG. The specific modeling process is as follows. At first, the states of the robots and target k are recoded as embeddings through two different FC layer networks, $e^{st}_j = f_5(p^r_j)$, $e^{at}_{ik} = f_6(p^a_k)$, $j \in \psi^r_i \cup i$. Then, based on the embeddings, the relational embedding h^{ra}_{ik} of the robots on target k can be calculated,

$$h^{ra}_{ik} = \sigma\Big(\sum_{j\in\psi^r_i\cup\{i\}} \frac{exp(\sigma_0(\boldsymbol{a}^T_{ra}[\boldsymbol{W}_{ra}e^{at}_{ik}\|\boldsymbol{W}_{ra}e^{st}_j]))}{\sum_{q\in\psi^r_i\cup\{i\}} exp(\sigma_0(\boldsymbol{a}^T_{ra}[\boldsymbol{W}_{ra}h^{at}_{ik}\|\boldsymbol{W}_{ra}e^{st}_q))} * \boldsymbol{W}_{ra}e^{st}_j\Big), \tag{3}$$

where $\boldsymbol{a}_{ra}$ is another learnable parameter vector, $\boldsymbol{W}_{ra}$ is another learnable parameter matrix. The target-robot relational embedding h^{ra}_{ik} implicitly represents the spatial relation and impact of the robots on target k. Finally, h^{ra}_{ik} is fed into a FC-layer network, which outputs a predicted position $\hat{p}^a_{ik}$ of target k at next timestep. The target movement model network is trained separately by supervised learning. Meanwhile, we store $\{o_i(t), p^a_k(t+1)\}$ as a sample of training data set $\mathbb{S}$ for the target movement model network.

Target Trajectory Predicting: After obtaining the predicted position $\hat{p}^a_{ik}$, we concatenate the predicted position $\hat{p}^a_{ik}$ and current position p^a_k of target k as a position state s^{pa}_{ik}. The position state is fed into a two-FC-layer network, which outputs the predicted embedding h^{pa}_{ik} that implicitly represents the predicted trajectory of target k. Besides, since there may be multiple targets in the observation of robot i, an attention network $\mathcal{F}_2$ with two FC layers is adopted to selectively aggregate the predicted embeddings. The attention coefficient of each predicted embedding can be computed, $\alpha^{pa}_{ik} = softmax(\mathcal{F}_2(e^{st}_i\|h^{pa}_{ik}))$. The total predicted embedding h^{pa}_i can be obtained, $h^{pa}_i = \sum_{k\in\psi^a_i} \alpha^{pa}_{ik} * h^{pa}_{ik}$, which contains information about the trajectory predictions of all the targets in the observation of robot i.

After obtaining the total predicted embedding h^{pa}_i and total relational embedding h_i, a total embedding h^{total}_i is represented by concatenating h^{pa}_i, h_i and e^s_i. Then the total embedding is fed into a FC layer, which outputs an environmental embedding h^e_i that is a state representation of the surrounding environment of robot i.

Recall that the policy network structure is a coupled actor-critic structure. In the actor, the environmental embedding h^e_i is fed into a policy head network with two FC layers, which outputs action values of robot i. In the critic, the environmental embeddings of all the robots and the ID of robot i are fed into a value head network with two FC layers, which outputs a scale value.

C. Knowledge-Embedded Compound Reward Function

According to the three subproblems in the problem formulation of MECA (dynamic multi-target assignment and forming groups, target encirclement of each group and collision avoidance), a knowledge-embedded compound reward function is designed to guide the multi-robot system to complete the MECA task. Specifically, the calculation of the reward function consists of three parts corresponding to the three subproblems respectively.

Firstly, according to the description of dynamic multi-target assignment and forming groups, we adopt a two-phase way to address it. Phase one is to find all possible combinations about the number of robots encircling each target, e.g., in a task with 2 targets and 7 robots, there two possible combinations, i.e., $\{3,4\}, \{4,3\}$, where $\{3,4\}$ denotes that the first and second targets require three and four robots respectively. The phase two is that find the minimum utility of each possible combination through Hungarian algorithm [23]. In all possible combinations, the assignment result of the combination with the minimum utility is regarded as the solution of the subproblem, in which the robots assigned to the same target automatically form a group, e.g., $\mathcal{G}_k$, where denotes k-th group consisting of robots assigned to target k.

Next, based on the assignment result, each group needs to form a circular formation to encircle the specified target. According to the definition in (1), we firstly define a target encirclement radius reward, i.e.,

$$R^\rho_{\mathcal{G}_k} = -clip(\frac{1}{|\mathcal{G}_k|}\sum_{i\in\mathcal{G}_k}(\|\boldsymbol{p}^r_i - \boldsymbol{p}^a_k\| - \rho), 0, 10), \tag{4}$$

This radius reward can enable k-th group to satisfy the first equation in (1). Then, we define a reward function $R^d_{\mathcal{G}_k}$ that indicates the average distance among neighboring robots belonging to k-th group,

$$R^d_{\mathcal{G}_k} = -clip(\frac{1}{|\mathcal{G}_k|(|\mathcal{G}_k|-1)}\sum_{i\in\mathcal{G}_k}\sum_{j\in\mathcal{G}_k and j\neq i}(\|\boldsymbol{p}^r_i - \boldsymbol{p}^r_j\| - d), 0, 10), \tag{5}$$

This reward can enable k-th group to satisfy the second constraint in (1).

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

$$R^c_i = \begin{cases} -2 & if \quad d^i_{min} < 0 \\ k_1 \cdot (d^i_{min} - D) & if \quad d^i_{min} < D \\ 0 & otherwise \end{cases}, \tag{6}$$

where k_1 is a hyper parameter, D is the threshold of the uncomfortable distance, d^i_{min} is the distance closest to other entity (other robots, obstacles and targets) for robot i.

Therefore, the knowledge-embedded compound reward function can be obtained, $R_i = R^c_i + \sum_{k=1}^K (R^\rho_{\mathcal{G}_k} + R^d_{\mathcal{G}_k})$.

D. Actor-Critic Training Algorithm

In this paper, the centralized training and decentralized execution framework [24] is utilized to learn a centralized critic to update the decentralized policy network of robot i during training. Meanwhile, the proximal policy

TABLE I

PERFORMANCE OF OUR METHOD AND BASELINE METHODS IN THE MECA TASK

Methods	Target Max Speed = 0.0	Target Max Speed = 0.4	Target Max Speed = 0.8	Target Max Speed = 1.2
	S(%) / MER / MEL	S(%) / MER / MEL	S(%) / MER / MEL	S(%) / MER / MEL
MECA-NODRL	87.5/-9.76/20.42	30.4/-20.77/37.20	16.0/-29.06/40.28	14.2/-31.21/40.27
MECA-DRL-CommC	99.3/-8.22/16.85	96.8/-11.31/23.18	19.1/-24.26/38.25)	0.4/-31.78/40.62
MECA-DRL-CommD	89.2/-11.62/24.06	72.7/-17.53/31.23	0.1/-36.08/37.00	0.0/-46.37/ –
MECA-DRL-RG (ours)	**99.8/-7.60/16.09**	**99.1/-10.37/21.22**	**71.3/-19.10/33.98**	**63.2/-21.57/35.02**

TABLE II

GENERALIZATION RESULTS OF OUR METHOD AND BASELINE METHODS IN THE MECA TASK

Methods	N=6,K=2,L=4	N=6,K=2,L=6	N=8,K=2,L=2	N=9, K=3, L=2
	S(%) / MER / MEL	S(%) / MER / MEL	S(%) / MER / MEL	S(%) / MER / MEL
MECA-NODRL	29.6/-21.70/37.84	27.1/-22.75/37.00	36.3/-20.96/38.31	24.6/-24.60/37.95
MECA-DRL-CommC	93.4/-12.24/25.07	89.0/-13.14/26.38	32.0/-16.93/27.64	1.5/-38.91/34.28
MECA-DRL-CommD	61.6/-19.38/32.46	54.8/-20.52/32.45	17.3/-20.58/35.37	0.4/-30.45/39.93
MECA-DRL-RG (ours)	**97.0/-10.92/22.28**	**93.9/-11.44/22.89**	**84.1/-11.56/26.47**	**74.4/-15.01/27.46**

optimization (PPO) [25] algorithm based on actor-critic style is implemented to update the parameter of the policy network composed of the actor and critic through minimizing two loss terms, $l_{v_i^\phi} = \mathbb{E}[(y_i - v_i^\phi(o, d_i))^2]$, $l_{\pi_i^\theta} = \mathbb{E}[min(\frac{\pi_i^\theta(\cdot|o_i)}{\pi_i^{\theta_{old}}(\cdot|o_i)}A_i(o,a), clip(\frac{\pi_i^\theta(\cdot|o_i)}{\pi_i^{\theta_{old}}(\cdot|o_i)}, 1-\epsilon, 1+\epsilon)A_i(o,a)]$, where $\pi_i^{\theta_{old}}(\cdot|o_i)$ is the actor before the update or the sampling actor, $y_i = R_i + \gamma v_i^\phi(o', d_i)$ is the temporal-difference (TD) target, $\epsilon = 0.2$ is a hyper parameter and $A_i(o,a)$ is an advantage function, which is estimated through the generalized advantage estimator (GAE) method [26]. Besides, the target movement network μ_δ is trained with supervised learning by minimizing the loss, $l_{\mu_\delta} = \mathbb{E}_{o_i(t), p_k^a(t+1) \sim \mathbb{S}}[(p_k^a(t+1) - \mu_\delta(o_i(t)))^2]$.

IV. SIMULATION AND EXPERIMENT RESULTS

A. Simulation Settings

A simulation environment for MECA is built based on the particle-world environment (MAPE) [24], where all robots, targets, and obstacles are initially randomly positioned on a 2×2 m^2 square area. At each timestep, each robot makes decisions based on its own partial observation, and selects one of the following actions: accelerate to north, south, east or west, and no acceleration. Besides, the targets adopt a fixed Voronoi escape strategy [27], and are limited in 3×3 m^2 square area. The maximum speed of the robots is set to 1 m/s, and S_{max}^a denotes the maximum speed of the targets. The greater the maximum speed of the targets, the more difficult the MECA task for the robots is. The task ends when the robots successfully encircle all the targets or the running time exceeds a fixed period $T_{max} = 50$. The condition for successfully encircling target k is defined: $\|\boldsymbol{p}_i^r - \boldsymbol{p}_k^a\| < D_{thred} + \rho$ and $\|\boldsymbol{p}_i^r - \boldsymbol{p}_j^r\| < D_{thred} + d$, $i \in \mathcal{G}_k$, where $D_{thred} = 0.05$ is the distance threshold of completing the task.

Our proposed method MECA-DRL-RG and the following baseline methods are implemented for performance evaluation.

- MECA-NODRL: This is not a DRL-based centralized method, and integrates the designed multi-target assignment algorithm in this paper, the pre-desired encirclement position approach for the robots and the optimal reciprocal collision avoidance (ORCA) algorithm [28];
- MECA-DRL-CommC: the centralized method based on DRL adopts a graph communication mechanism, and uses LSTM to process the information of the obstacles and targets [17];
- MECA-DRL-CommD: Except that global information cannot be used, the distributed method is the same as MECA-DRL-CommC.

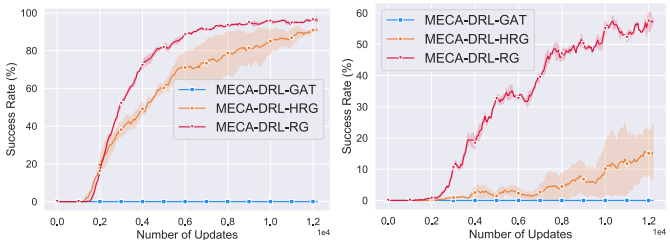
Besides, three metrics are set for evaluating the performance of different methods. The performance metrics of each method are obtained by testing 500 episodes with 5 different random seeds, 1) Success rate (S%): Percentage of the MECA task completed in all test episodes; 2) Mean episode reward (MER): Mean of episode rewards in all test episodes; 3) Mean episode length (MEL): Mean of successful episode length in all test episodes.

B. Implementation Specifications

In the heterogeneous relational graph learning and target strategy modeling module, all FC layer networks output a 128-dim embedding, except that the FC network in the target movement model outputs a 2-dim embedding, i.e., $\hat{p}_k^a$. Meanwhile, $\boldsymbol{W}_a, \boldsymbol{W}_{ra} \in \mathcal{R}^{128X128}$ and $\boldsymbol{a}_a, \boldsymbol{a}_{ra} \in \mathcal{R}^{256X1}$. The value head and policy head networks output a scalar value and a 5-dim action value vector respectively. The visual radius D^O of each robot is set as 1.6. The radius of the robots, targets, and obstacles are set as 0.05, 0.05, 0.08, respectively. Besides, for the reward function, $\rho = 0.2$, $D = 0.05$, $k_1 = 15$.

C. Simulation Results

1) Effectiveness: To fully evaluate the effectiveness of the proposed method, we conduct four tasks where $N = 6$ robots cooperatively encircle $K = 2$ stationary or moving targets in an environment with $L = 2$ static obstacles. In the four tasks, the maximum speed of the targets is set to 0.0, 0.4, 0.8 and 1.2 m/s respectively. Moreover, curriculum learning based on model reload [29] is implemented to speed up the process of the policy training for the robots. A curriculum is designed with the increasing of targets' maximum speed



(a) A MECA task with $S^a_{max} = 0.4$ (b) A MECA task with $S^a_{max} = 0.6$

Fig. 3. Training curves (success rate vs. number of updates) of ablation methods and our method in the MECA tasks where $N = 6, K = 2, L = 2$.

($S^a_{max} = 0.0, 0.4, 0.8, 1.2$). Firstly, the policy is leaned in the task with $S^a_{max} = 0.0$, then the learned policy is transferred to the task with $S^a_{max} = 0.4$ to continue training, until the task with $S^a_{max} = 1.2$.

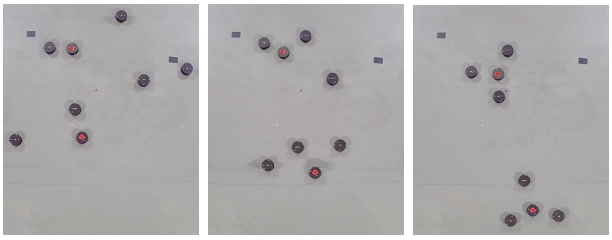
The test results of our method and the baseline methods are shown in Table I. The results show that the proposed method has the best performance than the baseline methods in terms of success rate, episode reward and episode length. On the contrary, MECA-NODRL fails when the targets move. Besides, the MECA-DRL-CommC and MECA-DRL-CommD methods based on graph communication mechanism have poor performance when the targets is fast, e.g., $S^a_{max} = 0.8$ or 1.2. This may be because although there is communication between the robots, the robots lack the understanding of their surrounding and the prediction of the targets. By contrast, the MECA-DRL-RG method enables the robots to learn heterogeneous spatial relational graphs from their surroundings and predict the targets' trajectories, which promotes each robot to understand and predict its surrounding. In general, the results fully demonstrate the effectiveness of the proposed method.

2) Generalization: In order to verify the generalization of the proposed method, the learned policy is evaluated in new tasks without any fine-tuning. The learned policy is obtained in the task which has $N = 6$ robots, $K = 2$ moving targets with $S^a_{max} = 0.4$, and $L = 2$ static obstacles. Four new tasks with $S^a_{max} = 0.4$ are selected, including $N = 6, K = 2, L = 4$ task, $N = 6, K = 2, L = 6$ task, $N = 8, K = 2, L = 2$ task and $N = 9, K = 3, L = 2$ task. The generalization results of our method and the baseline methods is shown in Table. II. As we expected, the proposed method outperforms the baseline methods in terms of success rate, episode reward and episode length, and has good generalization performance in the new tasks. No matter the number of robots, obstacles, or targets increase, the proposed method can achieve better generalization performance than baseline methods.

3) Ablation Analysis: To further investigate the effectiveness of key components of the proposed method, we specially design an ablation simulation. Meanwhile, two ablation methods are designed, including MECA-DRL-GAT and MECA-DRL-HRG. MECA-DRL-GAT adopts GAT to process all the information of targets and obstacles, without the heterogeneous relational graph learning and target strategy modeling modules. MECA-DRL-HRG use the heterogeneous relational graph learning module, without target strategy modeling. The proposed method and the ablation methods are performed in two tasks, i.e., $N = 6, K = 2, L = 2, S^a_{max} = 0.4$ task and $N = 6, K = 2, L = 2, S^a_{max} = 0.6$ task. The training curves (success rate vs. number of updates) of the ablation methods and our method are shown in Fig. 3. The results show that the proposed method has better superiority than the ablation methods in terms of success and convergence rates. The greater the maximum speed of the targets, the more obvious the superiority. By contrast, MECA-DRL-GAT fails in the tasks. Meanwhile, MECA-DRL-HRG outperforms MECA-DRL-GAT, which is due to the heterogeneous relational graph learning using robot-level RGs. Moreover, MECA-DRL-RG outperforms MECA-DRL-HRG, which is due to target strategy modeling using target-level RGs. Therefore, the ablation simulation fully verifies the effectiveness of the heterogeneous relational graph learning and target strategy modeling.

D. Real-World Experiment Results

Aside from the simulations above, we also examine the trained policy in real-world experiments on an experimental platform based on omnidirectional robots. In the experimental platform, each robot is equipped with an onboard computer (Nvidia Jetson TX2) to improve the real-time performance of policy implement. Moreover, the positions and velocities of the robots are provided by the NOKOV motion capture system. The three snapshots of 6 robots encircling 2 targets in 2 obstacle environment are presented in Fig. 4. These actual results show the robots can successfully complete the MECA task, which demonstrates the effectiveness and practicability of the proposed method.



(a) Initial stage (b) Encircling stage (c) Successful stage

Fig. 4. Snapshots of 6 robots encircling 2 targets in 2 obstacle environment

V. CONCLUSION

In this paper, MECA-DRL-RG is proposed to tackle the problem of MECA by using robot-level and target-level relational graphs. Specifically, the robot-level relational graphs are modeled and learned by using GATs for extracting different spatial relational representations. Moreover, to predict the trajectories of targets, the target-level relational graphs are built with GAT. Moreover, the movement of each target is modeled by the target-level RG and learned through supervised learning. Besides, a knowledge-embedded compound reward function is defined to address the multi-objective problem in MECA. Both simulation and real-world experiment results fully demonstrate the effectiveness and generalization of the proposed method.

References

[1] A. Farinelli, L. Iocchi, and D. Nardi, "Distributed on-line dynamic task assignment for multi-robot patrolling," *Autonomous Robots*, vol. 41, no. 6, pp. 1321–1345, 2017.

[2] Z. Sui, Z. Pu, J. Yi, and S. Wu, "Formation control with collision avoidance through deep reinforcement learning using model-guided demonstration," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 6, pp. 2358–2372, 2020.

[3] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3052–3059.

[4] A. Macwan, J. Vilela, G. Nejat, and B. Benhabib, "A multirobot path-planning strategy for autonomous wilderness search and rescue," *IEEE transactions on cybernetics*, vol. 45, no. 9, pp. 1784–1797, 2014.

[5] 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.

[6] 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.

[7] 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.

[8] M. Zhang and H. H. Liu, "Game-theoretical persistent tracking of a moving target using a unicycle-type mobile vehicle," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 11, pp. 6222–6233, 2014.

[9] X. Yu and L. Liu, "Distributed circular formation control of ring-networked nonholonomic vehicles," *Automatica*, vol. 68, pp. 92–99, 2016.

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

[11] B.-B. Hu, H.-T. Zhang, and J. Wang, "Multiple-target surrounding and collision avoidance with second-order nonlinear multiagent systems," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 8, pp. 7454–7463, 2020.

[12] J. Ma, W. Yao, W. Dai, H. Lu, J. Xiao, and Z. Zheng, "Cooperative encirclement control for a group of targets by decentralized robots with collision avoidance," in *2018 37th Chinese Control Conference (CCC)*. IEEE, 2018, pp. 6848–6853.

[13] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[14] 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.

[15] 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.

[16] 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.

[17] T. Zhang, Z. Liu, S. Wu, Z. Pu, and J. Yi, "Multi-robot cooperative target encirclement through learning distributed transferable policy," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[18] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International journal of robotics research*, vol. 23, no. 9, pp. 939–954, 2004.

[19] 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*, vol. 99, no. 2, pp. 371–386, 2020.

[20] 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.

[21] M. T. Spaan, "Partially observable markov decision processes," in *Reinforcement Learning*. Springer, 2012, pp. 387–414.

[22] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," *ArXiv*, vol. abs/1710.10903, 2017.

[23] R. Jonker and T. Volgenant, "Improving the hungarian assignment algorithm," *Operations Research Letters*, vol. 5, no. 4, pp. 171–175, 1986.

[24] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in neural information processing systems*, 2017, pp. 6379–6390.

[25] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[26] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[27] Z. Zhou, W. Zhang, J. Ding, H. Huang, D. M. Stipanović, and C. J. Tomlin, "Cooperative pursuit with voronoi partitions," *Automatica*, vol. 72, pp. 64–72, 2016.

[28] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research*, C. Pradalier, R. Siegwart, and G. Hirzinger, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 3–19.

[29] A. Agarwal, S. Kumar, and K. Sycara, "Learning transferable cooperative behavior in multi-agent teams," *arXiv preprint arXiv:1906.01202*, 2019.