Neighborhood curiosity-based exploration in multi-agent reinforcement learning

Shike Yang, Ziming He, Jingchen Li, Haobin Shi*, Qingbing Ji*, Kao-Shing Hwang *Senior Member, IEEE* and Xianshan Li

***Abstract*—Efficient exploration in cooperative multi-agent reinforcement learning is still tricky in complex tasks. In this paper, we propose a novel multi-agent collaborative exploration method called Neighborhood Curiosity-based Exploration (NCE), by which agents can explore not only novel states but also new partnerships. Concretely, we use the attention mechanism in graph convolutional networks to perform a weighted summation of features from neighbors. The calculated attention weights can be regarded as an embodiment of the relationship among agents. Then we use the prediction errors of the aggregated features as intrinsic rewards to facilitate exploration. When agents encounter novel states or new partnerships, NCE will produce large prediction errors, resulting in large intrinsic rewards. In addition, agents are more influenced by their neighbors and only interact directly with them in multi-agent systems. Exploring partnerships between agents and their neighbors can enable agents to capture the most important cooperative relations with other agents. Therefore, NCE can effectively promote collaborative exploration even in environments with a large number of agents. Our experimental results show that NCE achieves significant performance improvements on the challenging StarCraft II Micromanagement (SMAC) benchmark.**

***Index Terms*—Multi-agent reinforcement learning, Machine learning, Multi-agent system.**

I. Introduction

In recent years, cooperative multi-agent reinforcement learning (MARL) has attracted a great deal of attention because it can be effectively applied to many complex tasks [1], [2], [3], [4], such as robot swarm control [5] and autonomous driving [6]. The development of multi-agent reinforcement learning benefits from a popular paradigm called centralized training and decentralized execution (CTDE) [7]. CTDE learns a fully centralized value function based on global information and then uses it to guide the optimization of decentralized policies.

This work is supported in part by Major Research Project of National Natural Science Foundation of China under Grant 92267110, National Natural Science Foundation of China under Grant 62476225, 62076202, National Key R&D Program of China under Grant 2023YFF0905604, Shaanxi Province Key Research and Development Program of China under Grant 2023-YBGY-354, and Hebei Province Central Leading Local Science and Technology Development Project under Grant 246Z1817G. (Corresponding author: Haobin Shi and Qingbing Ji, e-mail:shihaobin@nwpu.edu.cn, jqbdxy@163.com.)

Shike Yang is with the School of Cybersecurity, Northwestern Polytechnical University, Xi'an 710072, China and Twentieth Research Institute, China Electronic Technology Group Corporation, Xi'an 710018, China.

Ziming He, Jingchen Li and Haobin Shi are with the School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China.

Qingbing Ji is with the Key laboratories for confidential communications, Thirtieth Research Institute of CETC Corporation, Chengdu, China.

Kao-Shing Hwang is with the Department of Electrical Engineering, National Sun Yat-sen University, Taiwan 80424, China.

Xianshan Li is with the School of Information Science and Engineering, Yanshan University, Qinghuangdao 066004, China.

Many popular MARL methods have been proposed based on the CTDE paradigm, such as VDN [8], QMIX [9], QTRAN [10], and QPLEX [11]. These methods are based on value decomposition, i.e., using neural networks to represent joint state-action values as a function of individual utility functions. Although these methods are excellent in many tasks, they struggle to solve complex tasks requiring efficient exploration because they only use simple ϵ-greedy exploration strategy [12]. Exploration has been widely investigated in the field of single-agent reinforcement learning, including the following three main categories: counted-based [13], [14], curiosity-based [15], [16], [17], and information gain-based methods [18]. Unfortunately, these techniques cannot be directly applied to multi-agent systems. Naively applying them to each agent in a multi-agent system will lead to a lack of collaboration among agents. At present, the study on exploration for MARL is roughly at the preliminary stage. Only a few works [12], [19], [20] have been proposed for multi-agent collaborative exploration.

As mentioned above, curiosity is an important way to facilitate exploration in the single-agent domain. It usually calculates intrinsic rewards based on the prediction errors of states, encouraging the agent to explore states with large prediction errors. There are two straightforward ways to apply curiosity to MARL. The first is individual curiosity, where each agent independently computes its intrinsic reward based on local observation. The second is global curiosity, in which the global intrinsic reward is calculated through joint observation of all agents. Individual curiosity leads agents to explore independently rather than coordinate with others. Global curiosity does not work when the number of agents is large, because the joint action-observation space grows exponentially with the number of agents.

In recent years, the concept of intrinsic motivation has become increasingly central in the development of curiosity-driven learning algorithms, particularly within the domain of reinforcement learning. Traditional approaches often rely heavily on prediction error as a measure of curiosity, which, while effective, may not fully encapsulate the breadth of learning behaviors that can be inspired by intrinsic motivations. Acknowledging this, several novel measures have been proposed to enhance the exploration strategies in reinforcement learning environments. Notably, measures based on episodic curiosity through reachability, as explored by Reference [21], offer a unique angle by incentivizing agents to explore previously unreachable states, thereby enhancing learning efficiency and environmental interaction. Similarly, the concept of behavior

matching, as discussed by Hafez et al. [22], introduces a framework where agents learn by aligning their actions with observed behaviors, fostering a more dynamic and adaptive learning process. Further, the approach of employing a Curious Meta-Controller [23], represents a significant shift towards adaptive learning strategies, alternating between model-based and model-free control to optimize learning progress. Lastly, the idea of planning to explore using self-supervised world models [24], opens new avenues for exploration by enabling agents to generate hypotheses about unobserved parts of the environment. These methodologies signify a substantial shift from traditional metrics, offering new perspectives on how intrinsic motivation can be conceptualized and implemented to improve exploration in reinforcement learning.

In this paper, we propose a new exploration method called Neighborhood Curiosity-based Exploration (NCE), which compromises individual and global curiosity. The intuition is that agents are more influenced by their neighbors and only interact with them directly [25], [26], [27], [28]. It is worth noting that agents must collaborate to achieve tasks in multi-agent systems. How they cooperate will impact the completion of tasks. Therefore, in order to achieve more effective exploration, agents should explore not only new states but also new partnerships. We use a graph convolutional network [26] to aggregate features from neighbors. By using the attention mechanism to perform a weighted summation of features from neighbors, graph convolution can extract relational representations among neighboring nodes and aggregate features in the neighborhood. When agents encounter novel states or are in new cooperative relationships, the neural network has a large prediction error for aggregated features because it has not been trained with similar input data. Therefore, calculating intrinsic reward based on the prediction error of aggregated features will motivate agents to explore novel states and discover new partnerships. In addition, with whom to cooperate and how to cooperate should be consistent and stable for at least a short period. Using neighborhood curiosity may induce agents to frequently change cooperative relationships [26]. Therefore, we have added temporal relationship regularization to the NCE to ensure that agents maintain stable relationships over the short term.

As far as we know, we are the first to explicitly put forward the idea of exploring cooperative relationships among agents. To evaluate the effectiveness of our proposed NCE method, we conduct experiments on the challenging SMAC benchmark [29]. The experimental results show that NCE outperforms other MARL baselines.

II. Background

A. DEC-POMDP

A cooperative multi-agent task can be modeled as a decentralized partially observable Markov decision process (Dec-POMDP) [30], which is defined by a tuple $G = \langle \mathcal{N}, \mathcal{S}, \mathcal{A}, P, R, \Omega, O, n, \gamma \rangle$. Here $\mathcal{N} \equiv \{1, \ldots, n\}$ denote the finite set of agents and $s \in \mathcal{S}$ is a finite set of global states. Due to the partial observability, each agent $i \in \mathcal{N}$ receives an individual partial observation $o_i \in \Omega$ according to the observation function $O(s, i) : \mathcal{S} \times \mathcal{N} \to \Omega$. At each time step, each agent $i \in \mathcal{N}$ chooses an action $a_i \in \mathcal{A} \equiv \left\{\mathcal{A}^{(1)}, \ldots, \mathcal{A}^{(|\mathcal{A}|)}\right\}$, forming a joint action $\boldsymbol{a} \equiv [a_i]_{i=1}^{n} \in \mathcal{A} \equiv \mathcal{A}^N$, leading to a shared reward $r = R(s, \boldsymbol{a})$ and a transition to the next global state $s' \sim P(\cdot \mid s, \boldsymbol{a})$. $\gamma \in [0, 1)$ is a discount factor. In addition, each agent i has an action-observation history $\tau_i \in \mathcal{T} \equiv (\Omega \times \mathcal{A})^*$, which can be used to construct its individual policy $\pi^i(a_i \mid \tau_i)$. We use $\tau \in \boldsymbol{\mathcal{T}} \equiv \mathcal{T}^n$ to denote joint action-observation history. The formal objective function is to learn a joint policy $\pi = \langle \pi_1, \ldots, \pi_n \rangle$ to maximize the joint value function $V^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s = s_0, \pi\right]$, or the joint action-value function $Q^{\pi}(s, a) = r(s, a) + \gamma E_{s'}\left[V^{\pi}(s')\right]$.

B. QMIX

Following the paradigm of centralized training and decentralized execution, QMIX [9] is built on the concept of value decomposition. QMIX employs a hybrid network to non-linearly sum the utility functions of each agent: $Q_{tot}(\tau, \boldsymbol{a}, s) = f_s\left(Q_1(\tau_1, a_1), \ldots, Q_n(\tau_n, a_n)\right)$. Moreover, QMIX needs to satisfy the monotonicity constraint:

$$\frac{\partial Q_{\text{tot}}}{\partial Q_i} \geq 0, \forall i. \tag{1}$$

The monotonicity constraint ensures that a global argmax performed on Q_{tot} yields the same result as a set of individual argmax operations performed on each Q_i. The weights of the hybrid network are computed by a hypernetwork considering the global state information. Furthermore, the monotonicity constraint can be implemented by restricting the hybrid network to have non-negative weights. QMIX is trained end-to-end to minimize the following squared TD error:

$$L(\theta) = \sum_{i=1}^{b} \left[\left(y_i^{tot} - Q_{tot}(\tau, \boldsymbol{a}, s; \theta)\right)^2\right], \tag{2}$$

where b is the batch size of transitions sampled from the replay buffer, $y_i^{\text{tot}} = r + \gamma \max_{\boldsymbol{a}'} Q_{tot}(\tau', \boldsymbol{a}', s'; \theta^-)$ are the TD targets, and θ^- are the parameters of a target network that are periodically copied from θ.

C. RND

Random network distillation (RND) [16] can measure the occurrence frequency of continuous or non-enumerable inputs. The random network distillation module converts the counting process into a supervised learning task using two neural networks: a prediction network and a target network with the same structure. The weights of the target network are fixed and randomly initialized. When sampling $[s_t, a_t, r_t, s_{t+1}]$ from the replay buffer, the target network generates the target value $f(s_{t+1})$ and the output of the prediction network is $\hat{f}(s_{t+1}; \theta)$, which tries to predict the target value and is trained to minimize the MSE:

$$r_t^{int} = \left\| \hat{f}(s_{t+1}; \theta) - f(s_{t+1}) \right\|^2, \tag{3}$$

where $\hat{f}$ is parameterized by θ. This idea comes from the fact that when an agent repeatedly encounters a particular state, it becomes easier to predict the state's output, resulting in a minimal prediction error. The prediction error will be bigger

for novel states since the prediction network has not been trained with similar states. As a result, the intrinsic reward r_t^{int} can quantify the novelty of states by establishing a link with their occurrence frequency, thus encouraging the agent to visit novel states.

III. Related Work

Single-agent Exploration Single-agent exploration methods have been well studied and can be divided into three main categories: counted-based methods [31], [32], curiosity-based methods [33], [34], and information gain-based methods [35], [36]. Counted-based methods measure whether a state is novel or often appeared by counting the number of times an agent has visited it. However, it is difficult to perform this operation in high-dimensional state space. Therefore, researchers have proposed many techniques to approximate the number of visits to a state, such as density models [13], hash functions [14], etc. Curiosity-based approaches compute intrinsic rewards through prediction error or uncertainty about the consequence of an agent's action. ICM [15] eliminates factors irrelevant to the agent's behavior through an inverse model and takes the difference between the predicted and actual next state in the learned hidden space as its intrinsic reward. RND [16] utilizes a fixed and randomly initialized neural network as the target network and calculates its intrinsic reward by the difference between the prediction network and the target network. Information gain-based methods compute intrinsic rewards based on the reduction of uncertainty about environmental dynamics. VIME [18] measures information gain using variational inference, where the environmental dynamics are approximated by a Bayesian neural network (BNN) [37]. Therefore, the intrinsic reward is calculated as the uncertainty reduction of the BNN weights.

In recent years, the construction of intrinsic reward in different ways to enhance the exploration of agents in the environment has become a research hotspot. Kim et al. [38] introduce DISCO-DANCE, an unsupervised skill discovery algorithm that selects a guide skill to direct other skills, enhancing exploration and maximizing state discriminability. Liu et al. [39] propose ComSD, which employs a contrastive multi-objective reward to balance exploration and diversity, achieving state-of-the-art adaptation in complex multi-joint robot tasks. Sener et al. [40] develop an exploration mechanism that integrates action, object, and outcome representations into a latent space, enabling robots to efficiently learn predictive models with developmental features akin to human infants. Bai et al. [41] present the CeSD framework, an unsupervised reinforcement learning approach that uses an ensemble of skills and partitioned exploration to maximize state coverage without external rewards. Ying et al. [42] propose PEAC, a Cross-Embodiment Unsupervised Reinforcement Learning (CEURL) algorithm, leveraging an intrinsic reward function to enhance cross-embodiment knowledge transfer, improving adaptation and generalization. Xu et al. [43] introduce EDEE, an exploration method for procedurally generated environments that combines episode-decayed exploration scores with imitation learning to improve diversity and stability in DRL agents. Yang et al. [44] present Behavior Contrastive Learning (BeCL), which uses contrastive learning among behaviors to drive diverse and far-reaching skill development without extrinsic rewards.

Unsupervised Reinforcement Learning In order to deal with the challenge that it is difficult to train agents in sparse reward environments, a promising research direction is to improve the exploration ability of agents and enhance the sample efficiency of reinforcement learning. There is a lot of current work on environment exploration in reinforcement learning [31], [45], [13], and a part of this work defines the key to exploration as accurately estimating the uncertainty encountered, such methods [16], [15] have been introduced in the previous paragraph. The main improvement in recent research to boost the exploration of sparse-reward environments is to learn prior information and represent it as options or skills, and this class of methods is collectively referred to as unsupervised reinforcement learning. The idea of generating intrinsic rewards to drive exploration is similar to previous work.

According to the nature of intrinsic rewards, the URL methods can be classified into three types: competence-based [46], [45], knowledge-based [16], [15], and data-based algorithms [47]. Knowledge-based methods define self-supervised tasks by making predictions about some aspect of the environment, while data-based methods maximize the entropy of state access to explore the environment. Competence-based algorithms are the mainstream approach, which forces agents to discover and learn skills in the process of controlling the environment. The DIAYN method proposed by Eysenbach et al. [46], computes a variational approximation of the mutual information decomposition and trains a discriminator network to measure the conditional entropy of this objective. Liu et al. introduced the APT method [47], which explores the environment by maximizing a non-parameter-based particle entropy in the state representation space. The APS method proposed by Liu et al. [45] estimates the state entropy through the particle entropy estimator, and uses the successor features to approximate the conditional entropy. Other studies look at trajectory-based approaches such as IBOL [48], EDL [49], and other algorithms that encode sampled trajectories as skill indications. The original intention of these methods is to scale learnable skills in a low-dimensional feature state space. However, some researchers are keenly aware of the impact of under-exploration on skill diversity in high-dimensional state space [49]. For example, DADS [50] needs to estimate multiple conditional probability densities on the state space, which may lead to poor scalability when dealing with high-dimensional environments. Beyond that, the exploration improvements that these unsupervised reinforcement learning methods can bring when extended to multi-agent scenarios remain to be explored.

Multi-agent Exploration Although single-agent exploration has been extensively studied, there are few studies on multi-agent exploration. Multi-agent exploration methods must take into account interactions between agents. Jaques et al. [51] define intrinsic rewards as "social influence" to motivate agents to choose actions that potentially impact the actions of others. Wang et al. [19] use mutual information

(MI) to measure the impact of an agent on the transition function (EITI) and reward structure (EDTI) of other agents, encouraging agents to visit interaction points. However, "social influence" and mutual information need to be calculated among all agents, which increases computational complexity when there are numerous agents. MAVEN [12] is the state-of-the-art exploration method in MARL. It utilizes a hierarchical strategy to generate shared latent variables, according to which agents adapt their behaviors to perform committed exploration. Since the latent variables can be viewed as exploring the space of joint behaviors [12], MAVEN is inefficient in complex tasks with large state-action space. Gao et al. [52] propose GCEN, a multiagent deep reinforcement learning method that addresses incomplete and noisy observations through grouped cognitive feature representation, enhancing performance and generalization in cooperative tasks. Chen et al. [53] introduce EHCAMA, an entropy-enhanced MARL approach that combines hierarchical graph attention networks with gated recurrent units to develop stable continuous policies, excelling in large-scale multiagent scenarios. Li et al. [54] develop 2ReCom, a decentralized communication framework for MARL that uses a dual-level recurrence mechanism to improve communication efficiency and fairness, outperforming existing methods in both partially and fully observable environments.

In efforts related to curiosity-driven work for multi-agent systems, the focus is predominantly on the distinctions between global and individual differences induced by value estimation [55]. However, the premise of such algorithms is that the multi-agent environment only possesses a global centralized reward. For decentralized rewards, these algorithms cannot operate within scenarios due to the unknown nature of other agents' rewards [56].

IV. Neighborhood Curiosity-based Exploration (NCE)

In this section, we introduce NCE, an effective multi-agent exploration method based on the novelty of neighborhood features. NCE uses a graph convolutional network to aggregate the feature vectors of agents in the neighborhood and an RND network to calculate the intrinsic reward. The intrinsic reward calculated in this way considers the cooperative relationship of agents in the neighborhood, guiding agents to explore collaboratively with their neighbors. We define the neighborhood of agents as their sight range. Thus, when an agent's neighborhood includes all the other agents, it may need to interact with all of them. Neighborhood curiosity can be regarded as a global curiosity to make all agents explore collaboratively. When an agent's neighborhood only includes itself, it does not need to interact with other agents in this region. Neighborhood curiosity can be viewed as an individual curiosity to make it explore alone.

A. Intrinsic Reward of NCE

It is known that in reinforcement learning, the agent in state s^t can make the state transition to s^{t+1} by taking an action a^t. RND uses s^{t+1} to compute the intrinsic reward at time t. The more novel s^{t+1} is, the bigger the intrinsic reward for the transition from s^t to s^{t+1}, thus encouraging the agent to visit novel states. In this paper, we use a similar approach to calculate the intrinsic reward. Figure 1 shows the process of calculating the intrinsic reward of agent i at time t with the neighborhood curiosity module, assuming that the global observations at time $t+1$ are represented by $o^{t+1} = \left[o_1^{t+1}, o_2^{t+1}, \ldots, o_N^{t+1}\right]$, the neighbors of i are $B_i^{t+1} = \{1, \ldots, k\}$, and the adjacency matrix of i are M_i^{t+1}. In defining neighboring agents, we adopt a method similar to other works, assuming that multiple agents exist within an Euclidean space, with each agent having only partial observations [57]. The set of other agents within the local observational range of the i-th agent is defined as its neighboring agents. The adjacency matrix is populated with mutual observability information; that is, if the i-th and j-th agents can observe each other, then the entries in the i-th row and j-th column, as well as the j-th row and i-th column of the matrix, are set to 1; otherwise, they are set to 0. Therefore, the adjacency matrix also takes the form of a diagonal matrix. The neighborhood curiosity module is shared among all agents and contains two main components: the graph convolution module and the RND module.

Graph Convolution Module We use B_{+i}^{t+1} to denote agent i and its neighbors B_i^{t+1} at time $t+1$. The local observations of all agents in B_{+i}^{t+1} are first encoded as feature vectors through MLP. Then, these feature vectors are integrated by an attention layer to generate the neighborhood feature $h_i'^{t+1}$. In the attention layer, the input features of each agent are projected to query, key, and value representations. The attention weights between i and $j \in B_{+i}^{t+1}$ at timestep $t+1$ are computed as:

$$\alpha_{ij}^{t+1} = \frac{\exp\left(W_q h_i^{t+1} \cdot \left(W_k h_j^{t+1}\right)^T / \sqrt{d}\right)}{\sum_{l \in B_{+i}^{t+1}} \exp\left(W_q h_i^{t+1} \cdot \left(W_k h_l^{t+1}\right)^T / \sqrt{d}\right)}, \quad (4)$$

where d is a normalization factor that scales the result. Attention weights reflect cooperative relationships among agents—the bigger the attention weights, the closer the agents' collaboration. Then the value representations of all input features are weighted by the attention weights and summed together to generate the aggregated neighborhood feature of agent i:

$$h_i'^{t+1} = \sum_{j \in B_{+i}^{t+1}} \alpha_{ij}^{t+1} \cdot W_v \cdot h_j^{t+1}. \quad (5)$$

RND Module Intrinsic rewards are calculated using random network distillation (RND) [16], which is an efficient method for the agent to explore new states in the single-agent domain. We use $\phi\left(h_i'^{t+1}\right)$ and $\psi\left(h_i'^{t+1}\right)$ to denote the neighborhood features generated by the prediction network and the target network, respectively. Then a distance function is used to calculate the distance between $\phi\left(h_i'^{t+1}\right)$ and $\psi\left(h_i'^{t+1}\right)$, e.g., the L2 distance. After calculating the intrinsic rewards $\left[r_i^{int}\right]_{i=1}^N$ for all agents separately, take their sum as the total intrinsic reward. That is, the intrinsic reward at time t is generated by the following formula:

$$r^{int} = \sum_{i=1}^{N} \left\|\phi\left(h_i'^{t+1}\right) - \psi\left(h_i'^{t+1}\right)\right\|_2, \quad (6)$$

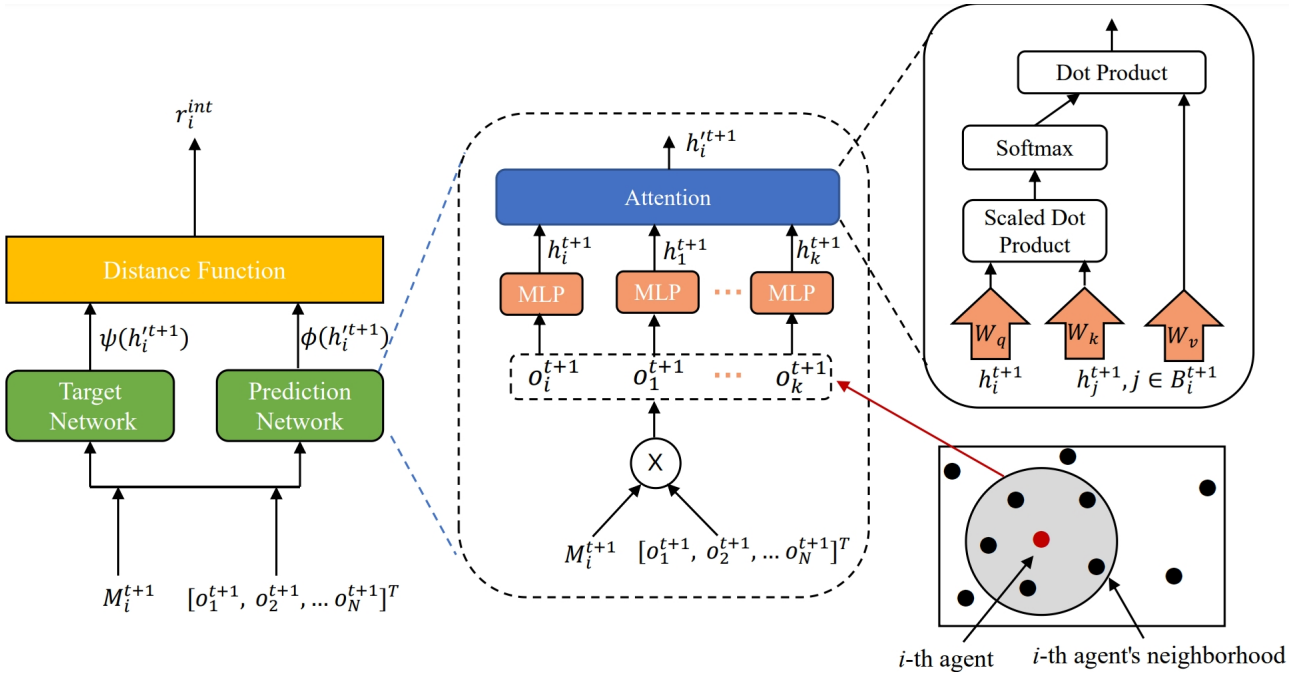


Fig. 1. Architecture of the neighborhood curiosity module. The left is the RND module, which calculates the intrinsic reward of agent i. The target network in RND uses the same network structure as the prediction network, and both use the graph convolution module to aggregate neighborhood features. Note that the difference between o_N^{t+1} and o_k^{t+1} is that o_N^{t+1} represents the observation of the N-th agent at time $t+1$, while o_k^{t+1} represents the observation of the k-th neighbor at time $t+1$ after processing the adjacency matrix M_i^{t+1}. It is necessary to make a distinction between the two.

Calculating the intrinsic reward in this way may cause agents to change cooperative relationships frequently to get more intrinsic rewards, so it is necessary to add temporal relation regularization to make agents maintain stable cooperative relationships in a short time. Specifically, the KL divergence of the attention weight distribution of agent i at timestep t and timestep $t+1$ (represented by $\mathcal{G}\left(O_{B_{+i}}^{t}\right)$ and $\mathcal{G}\left(O_{B_{+i}}^{t+1}\right)$, respectively) is used as the regularization term of the loss function. Both $\mathcal{G}\left(O_{B_{+i}}^{t}\right)$ and $\mathcal{G}\left(O_{B_{+i}}^{t+1}\right)$ are computed by the graph convolution module of the prediction network. We parametrize the neighborhood curiosity module by η. During training, the prediction network is updated by minimizing the following loss function:

$$L_p(\eta) = \frac{1}{b}\sum_{b}\frac{1}{N}\sum_{i=1}^{N}\left(\left\|\phi\left(h_i'^{t+1}\right) - \psi\left(h_i'^{t+1}\right)\right\|_2 + \lambda D_{KL}\left(\mathcal{G}\left(O_{B_{+i}}^{t}\right)\|\mathcal{G}\left(O_{B_{+i}}^{t+1}\right)\right)\right), \tag{7}$$

where λ is the coefficient of the regularization term, and b is the batch size.

B. Learning for NCE

In fact, the NCE method can be applied to almost all the multi-agent value decomposition methods. This paper uses QMIX to implement NCE (see Algorithm 1). In QMIX, each agent i has an individual utility function Q_i. The utilities of all agents are monotonically mixed into Q_{tot} through a mixing network. During training, each agent selects actions through the ϵ-greedy strategy. This simple exploration strategy makes it difficult for agents to explore effectively in complex tasks. Therefore, the NCE method can be added to QMIX to make agents explore effectively and collaboratively. Our NCE+QMIX method changes the TD targets in QMIX to the following form:

$$y_i^{tot} = r^{ext} + \beta r^{int} + \gamma \max_{\boldsymbol{a}'} Q_{tot}\left(\tau', \boldsymbol{a}', s'; \theta^{-}\right), \tag{8}$$

where r^{ext} is the extrinsic reward given by the environment and β is the weighting term to balance the extrinsic and intrinsic reward. Then update the Q_{tot} and Q_i networks by minimizing the following TD loss:

$$L(\theta) = \sum_{i=1}^{b}\left[\left(y_i^{\text{tot}} - Q_{tot}(\tau, \boldsymbol{a}, s; \theta)\right)^2\right]. \tag{9}$$

Derived from prediction errors of the aggregated neighborhood features, we encourage agents to explore novel states and form new collaborative relationships. This mechanism is particularly effective in complex multi-agent settings where traditional exploration strategies may fail. The novelty-based intrinsic reward incentivizes agents to venture into less familiar areas of the state space, promoting a more thorough exploration.

V. Experiments

A. StarCraft II Micromanagement Benchmark

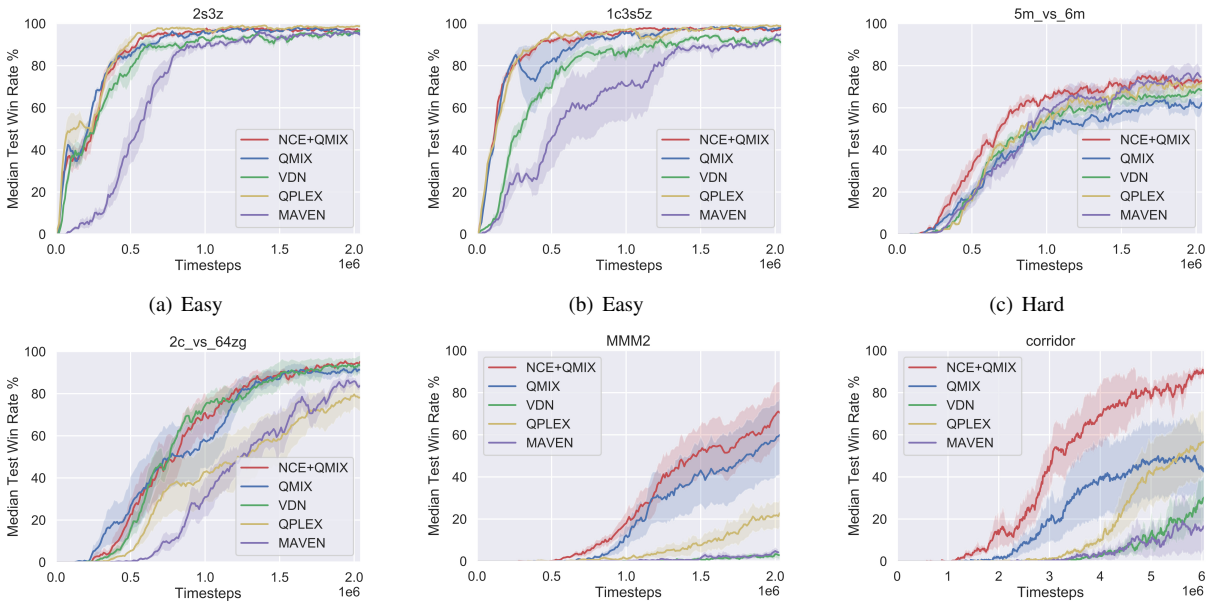
In this section, we conduct experiments on the StarCraft II micromanagement (SMAC) benchmark and compare it with the following value-based MARL algorithms: VDN [8], QMIX [9], MAVEN [12], and QPLEX [11], a novel method that achieves the state-of-the-art performance on the SMAC benchmark.

TABLE I
SMAC MAPS USED IN OUR PAPER.

Map Name	Ally Units	Enemy Units
2s3z	2 Stalkers & 3 Zealots	2 Stalkers & 3 Zealots
1c3s5z	1 Colossus, 3 Stalkers & 5 Zealots	1 Colossus, 3 Stalkers & 5 Zealots
5m_vs_6m	5 Marines	6 Marines
2c_vs_64zg	2 Colossi	64 Zerglings
MMM2	1 Medivac, 2 Marauders & 7 Marines	1 Medivac, 3 Marauders & 8 Marines
corridor	6 Zealots	24 Zerglings

TABLE II
HYPER-PARAMETERS OF NCE. WE SET $\beta = 15$ IN THE CORRIDOR MAP AND $\beta = 1$ IN THE OTHER MAPS.

Name	Value
The number of layers in MLP	1
Unit number in MLP	64
The number of layers in the query, key, and value embedding layer	1
Unit number in the query, key, and value embedding layer	64
optimizer of neighborhood curiosity module	Adam
learning rate	4e-5
weighting term λ of regularization loss	0.2
weighting term β of intrinsic reward	1 or 15



(a) Easy (b) Easy (c) Hard

(d) Hard (e) Super Hard (f) Super Hard

Fig. 2. Median test win rate % in six different SMAC maps: (a) 2s3z (easy), (b) 1c3s5z (easy), (c) 5m_vs_6m (hard), (d) 2c_vs_64zg (hard), (e) MMM2 (super hard), and (f) corridor (super hard). Our method (NCE+QMIX) does not lose its advancement on easy and hard maps, but achieves absolute leading performance on super-hard map. The reason is that other methods are limited in their ability to explore super-hard map.

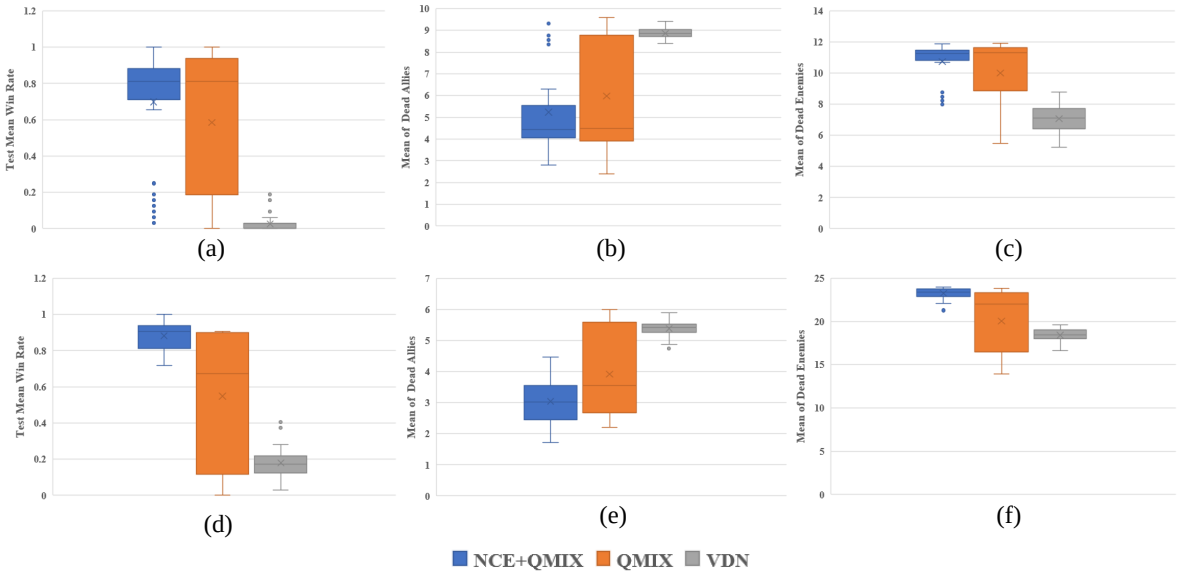


Fig. 3. Comparison of numerical results on super-hard level maps. (a) The average win rate on the MMM2 map. (b) The average number of friendly deaths on the MMM2 map. (c) The average number of enemy kills on the MMM2 map. (d) The average win rate on the Corridor map. (e) The average number of friendly deaths on the Corridor map. (f) The average number of enemy kills on the Corridor map. These box plots are generated from the last 10 episodes of five random tests.

SMAC is a popular benchmark for evaluating the performance of MARL methods. In SMAC, each learning agent controls a separate allied unit, and the built-in StarCraft II artificial intelligence controls enemy units. Agents can only receive local observations within their vision, so the environment is partially observable to them. Combat requires proper micro-management of allies to maximize damage to enemies while minimizing damage received, and thus requires a variety of skills such as focusing fire and avoiding overkill. Learning these different cooperative behaviors under partial observation is challenging. SMAC maps have three difficulty levels: easy, hard, and super hard. To verify the effectiveness of our method, we select two maps from each of these three difficulty levels for experiments. We pause the training every 10,000 timesteps and perform 32 evaluations using a decentralized greedy action selection to calculate the average test win rate. All results are averaged over five runs using different seeds.

We follow the default environment settings of SMAC and use Version SC2.4.10. Each agent chooses actions from a discrete space, including the following actions: move [direction], attack [enemy id], stop, and no-op. Medivacs use heal [agent id] action rather than attack [enemy id] because they are healers. Dead agents can only take no-op action, while living agents cannot. By taking action, agents move and attack in continuous maps. At each time step, all agents receive a global reward equivalent to the total damage inflicted on the enemies. Besides, agents receive a bonus of 10 points for each enemy killed and an extra 200 points for victory. The maximum return of an episode is approximately 20 since rewards are scaled. We briefly introduce the SMAC maps used in our paper in Table I.

All the baselines are implemented through the source codes provided by their authors without changing the hyper-parameters, including the following algorithms: VDN [8], QMIX [9], MAVEN [12], and QPLEX [11]. Our approach is implemented based on QMIX, and the hyper-parameters specific to NCE are listed in Table II.

As can be seen from Figure 2, other methods cannot maintain good performance on all maps, while our approach consistently achieves almost the best performance. In the easy maps, NCE+QMIX and QPLEX converge faster than other algorithms. In the hard map 5m_vs_6m, NCE significantly improves the win rate of QMIX. The main advantage of our method is reflected in the results of super hard maps, where NCE+QMIX significantly outperforms other baselines. This is because super-hard maps require more efficient and coordinated exploration than other maps. MMM2 is a complex environment with many unit types and numbers. It consists of 1 Medivac, 2 Marauders, and 7 Marines. Medics should hide behind allied units to escape sacrifice because they can heal damaged units. Damaged units should retreat to avoid the opponent's focused fire, while healthy units should advance to take fire. NCE can explore the cooperative relationship among agents, allowing agents to learn this complex strategy more quickly. Corridor is a task that requires active state space exploration [29]. The best strategy requires the allied Zealots to move to a choke point on the map to avoid being surrounded by enemy forces. In the absence of adequate exploration, agents will merely damage enemy units for rewards rather than moving to the choke point first and then attacking. Applying the NCE method to this map will help agents quickly discover appropriate unit positioning, thus improving performance.

We further measured three key metrics, namely the average win rate, the average number of friendly deaths, and the average number of enemy kills. The results of the five random tests are presented in Figure 3. It is obvious that our proposed method is more stable and exhibits a more compact data distribution. Although there are a small number of outliers,

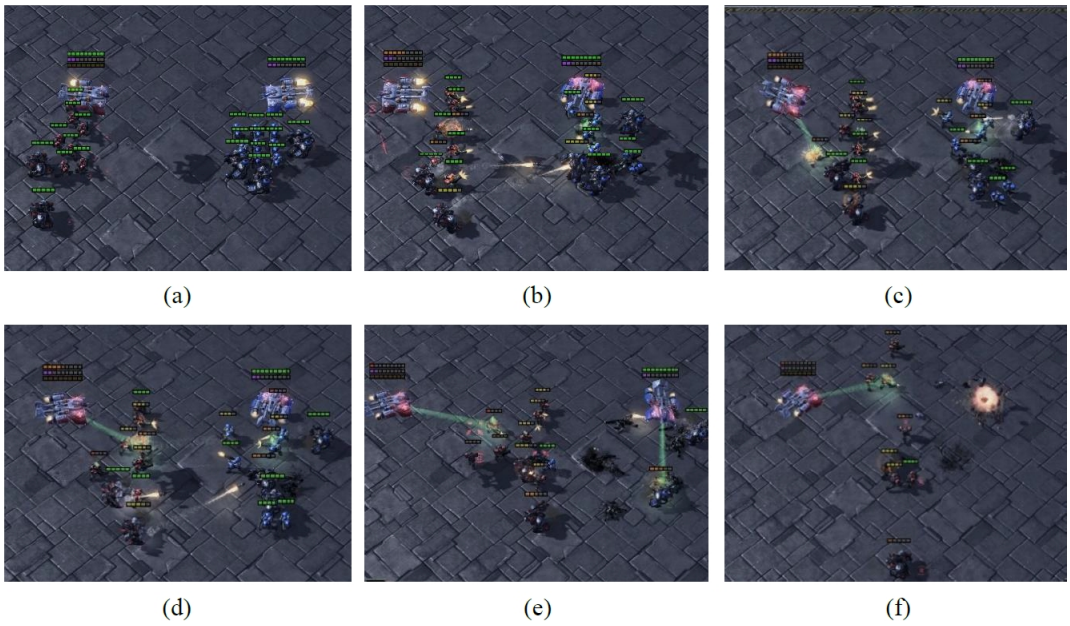
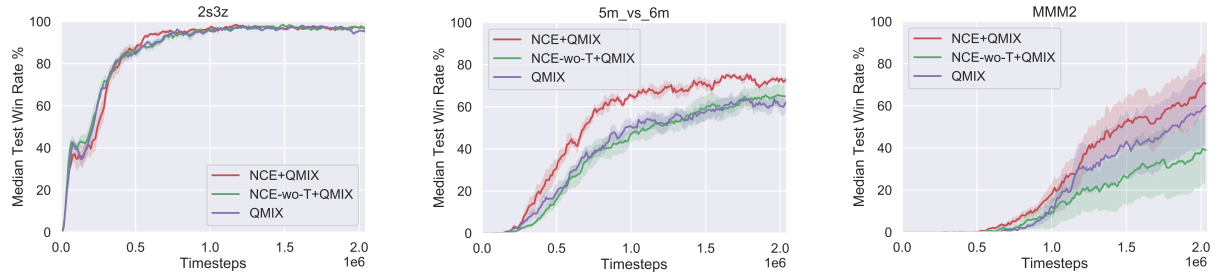


Fig. 4. The battle process using the NCE+QMIX method on the MMM2 map. (a) At the beginning all agents move together towards the enemies. (b) Medic hides behind the team to avoid damage. (c) The agent with the least health backs off and hides at the end to be healed by the medic. (d) Agents cooperate to gather fire and destroy enemy low-health units. (e) The healthy agents take damage in front of the low-health agents. (f) At the end of the battle, we cooperate to annihilate all enemy units.



(a) Easy (b) Hard (c) Super Hard

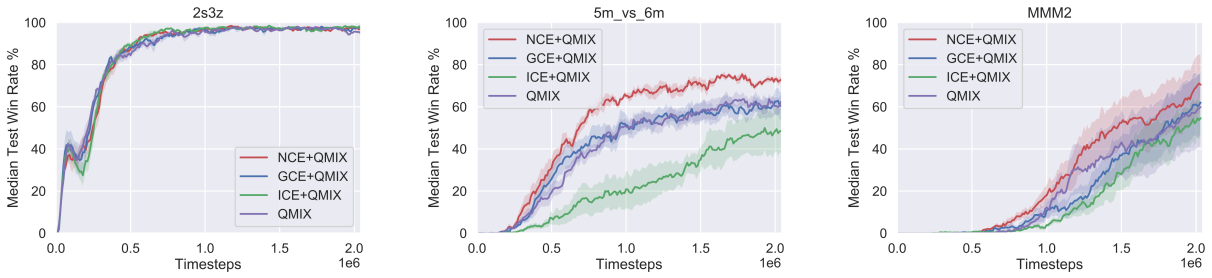
Fig. 5. Ablation study on temporal relation regularization. In the 2s3z map, the cooperation relationship between agents changes little during training, so the ablation of the temporal relation regularization is not obvious. The more complex the maps, the more effective the temporal relation regularization module is, mainly because the short-term stable cooperative relationship facilitates the completion of long-term cooperative tasks.

which may be caused by occasional gradient collapse, it still shows a clear lead over other methods in all three metrics.

To get a more intuitive idea of whether the NCE+QMIX algorithm can enable agents to learn complex cooperative strategies, we analyze a video of a battle on the MMM2 map. Figure 4 shows the battle process; the left side is our agent controlled by the NCE+QMIX algorithm, the right side is the enemy agent controlled by AI, and (a) to (f) are intercepted in chronological order; we only analyze the strategy adopted by our agent. Figure 4(a) shows the beginning of the battle, with all agents moving towards the enemy. At the moment in Figure 4(b), the medic makes the decision to back off to avoid being killed by the enemy so that he can continue to treat wounded Allies. At the moment, in Figure 4(c), the agent with the lowest health is hiding behind the other healthy units to avoid sacrifice and get medical treatment. At the moment in Figure 4(d), our agent is preferentially targeting the enemy's low-health units. At the moment in Figure 4(e), all our agents with low health are at a distance from the enemy, and the agents with higher health are taking damage in front of the agents with low health. Figure 4(f) shows the end of the battle, we have only sacrificed one agent, successfully annihilated all enemy units, and won the battle. It can be seen that the NCE+QMIX algorithm can make the agent learn complex cooperative strategies, so as to win the battle.

B. Ablation Study

We conduct ablation studies to verify the role of temporal relation regularization in our method. NCE-wo-T is used to represent the NCE method without temporal relation regularization. We compare the performances of NCE+QMIX, NCE-wo-T+QMIX, and QMIX in simple map 2s3z, hard map 5m_vs_6m, and super hard map MMM2. As shown in Figure 5, NCE+QMIX obtains the best performance in all these maps, and the performance of NCE-wo-T+QMIX in the MMM2 map is even worse than QMIX. It indicates that temporal relation



(a) Easy (b) Hard (c) Super Hard

Fig. 6. Effect of different kinds of curiosity in QMIX. Agents adding global curiosity and individual curiosity mechanisms show performance loss on both hard and super-hard maps. It is obvious that too large or too small curiosity neighborhood can not help the complex collaborative task. On the contrary, the attention mechanism is used to weight the neighborhood features in the graph convolutional network, which has a remarkable effect.

Algorithm 1 NCE+QMIX

initialize $[Q_i]$,Q_{tot} with random parameters θ
Initialize target parameters $\theta^- \leftarrow \theta$
initialize neighborhood curiosity module with random parameters η
Learning rate $\leftarrow \alpha$, $D \leftarrow \{\}$
for episode = 1 to *MAX_EPISODES* **do**
  Get state s^t, adjacency matrix M^t and observation o^t=$[O(s^t, i)]_{i=1}^N$ for each agent i
  for t = 1 to T **do**
    With probability ϵ select a random action a_i^t
    Otherwise $a_i^t = argmax_{u_i^t} Q_i(\tau_i^t, u_i^t)$ for each agent i
    Take action $\boldsymbol{a}^t$, and retrieve next state, next observation, next adjacency matrix and reward $(s^{t+1}, o^{t+1}, M^{t+1}, r^t)$
    Store transition $(s^t, o^t, M^t, \boldsymbol{a}^t, r^t, s^{t+1}, o^{t+1}, M^{t+1})$ in D
    if $|D| \geq$ batch_size **then**
      Sample from D
      Set $r^{ext} = r$, and compute r^{int} by Formula (6)
      Update neighborhood curiosity module by minimizing Formula (7)
      Update Q_{tot} and $[Q_i]_{i=1}^N$by minimizing Formula (9)
    end if
    if at target update interval **then**
      Update target parameters $\theta^- \leftarrow \theta$
    end if
  end for
end for

regularization is vital for improving the performance of NCE, as it prevents agents from frequently changing cooperative relationships in the short term.

We conduct additional ablation experiments to compare the effects of global curiosity, individual curiosity, and our proposed neighborhood curiosity in MARL. When calculating global curiosity and individual curiosity, both the prediction network and the target network of RND are composed of multi-layer MLP instead of the graph convolution module. Moreover, the input of RND is the local observation of an agent (individual curiosity) or the joint observation of all agents (global curiosity). Exploration based on global curiosity and individual curiosity are denoted by GCE and ICE, respectively. We compare the performance of NCE+QMIX, GCE+QMIX, ICE+QMIX, and QMIX in 2s3z, 5m_vs_6m, and MMM2. As shown in Figure 6, our method and ICE+QMIX converge the fastest in 2s3z, indicating that agents merely require simple exploration strategies to obtain good performance in easy maps. However, in hard and super-hard maps that demand more efficient and collaborative exploration, individual curiosity hurts the performance of QMIX because it motivates agents to explore alone. Global curiosity is inefficient in scenarios with large joint observation space, such as SMAC, because it is difficult to fully explore all joint observations, resulting in the behavior of agents still being dominated by external rewards. Therefore, neither global curiosity nor individual curiosity can enhance the performance of QMIX, while our proposed neighborhood curiosity obtains the best performance in these maps.

VI. Conclusion

This paper proposes NCE, an efficient multi-agent collaborative exploration method that can be applied to almost all the existing multi-agent value decomposition methods. By considering the neighboring agents' relationships when exploring with curiosity, NCE can promote agents to explore novel states and new cooperative relationships. Our approach shows significant performance improvements on the StarCraft II micromanagement benchmark. In the future, we consider combining representation learning with NCE to improve its performance, i.e., using a representation learning module instead of a simple MLP to extract features from each agent before aggregating neighborhood features and calculating intrinsic rewards.

References

[1] J. Li, H. Shi, and K.-S. Hwang, "Using fuzzy logic to learn abstract policies in large-scale multi-agent reinforcement learning," *IEEE Transactions on Fuzzy Systems*, 2022.
[2] H. Shi, J. Li, J. Mao, and K.-S. Hwang, "Lateral transfer learning for multiagent reinforcement learning," *IEEE Transactions on Cybernetics*, 2021.
[3] F. Baghbani, M. R. Akbarzadeh-T, and M. Sistani, "Cooperative adaptive emotional neuro-control for a class of higher-ordered heterogeneous uncertain nonlinear multi-agent systems," *Neurocomputing*, 2021.

[4] Z. He, J. Li, F. Wu, H. Shi, and K.-S. Hwang, "Derl: Coupling decomposition in action space for reinforcement learning task," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2023.

[5] M. Hüttenrauch, A. Šošić, and G. Neumann, "Guided deep reinforcement learning for swarm systems," *arXiv preprint arXiv:1709.06011*, 2017.

[6] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Transactions on Industrial informatics*, vol. 9, no. 1, pp. 427–438, 2012.

[7] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein, "The complexity of decentralized control of markov decision processes," *Mathematics of operations research*, vol. 27, no. 4, pp. 819–840, 2002.

[8] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls *et al.*, "Value-decomposition networks for cooperative multi-agent learning," *arXiv preprint arXiv:1706.05296*, 2017.

[9] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, "Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *International conference on machine learning*. PMLR, 2018, pp. 4295–4304.

[10] K. Son, D. Kim, W. J. Kang, D. E. Hostallero, and Y. Yi, "Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning," in *International conference on machine learning*. PMLR, 2019, pp. 5887–5896.

[11] J. Wang, Z. Ren, T. Liu, Y. Yu, and C. Zhang, "Qplex: Duplex dueling multi-agent q-learning," *arXiv preprint arXiv:2008.01062*, 2020.

[12] A. Mahajan, T. Rashid, M. Samvelyan, and S. Whiteson, "Maven: Multi-agent variational exploration," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[13] G. Ostrovski, M. G. Bellemare, A. Oord, and R. Munos, "Count-based exploration with neural density models," in *International conference on machine learning*. PMLR, 2017, pp. 2721–2730.

[14] H. Tang, R. Houthooft, D. Foote, A. Stooke, O. Xi Chen, Y. Duan, J. Schulman, F. DeTurck, and P. Abbeel, "# exploration: A study of count-based exploration for deep reinforcement learning," *Advances in neural information processing systems*, vol. 30, 2017.

[15] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *International conference on machine learning*. PMLR, 2017, pp. 2778–2787.

[16] Y. Burda, H. Edwards, A. Storkey, and O. Klimov, "Exploration by random network distillation," *arXiv preprint arXiv:1810.12894*, 2018.

[17] J. Li, X. Shi, J. Li, X. Zhang, and J. Wang, "Random curiosity-driven exploration in deep reinforcement learning," *Neurocomputing*, 2020.

[18] R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel, "Vime: Variational information maximizing exploration," *Advances in neural information processing systems*, vol. 29, 2016.

[19] T. Wang, J. Wang, Y. Wu, and C. Zhang, "Influence-based multi-agent exploration," *arXiv preprint arXiv:1910.05512*, 2019.

[20] I.-J. Liu, U. Jain, R. A. Yeh, and A. Schwing, "Cooperative exploration for multi-agent deep reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2021, pp. 6826–6836.

[21] N. Savinov, A. Raichuk, R. Marinier, D. Vincent, M. Pollefeys, T. Lillicrap, and S. Gelly, "Episodic curiosity through reachability," *arXiv preprint arXiv:1810.02274*, 2018.

[22] M. B. Hafez and S. Wermter, "Behavior self-organization supports task inference for continual robot learning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 6739–6746.

[23] M. B. Hafez, C. Weber, M. Kerzel, and S. Wermter, "Curious meta-controller: Adaptive alternation between model-based and model-free control in deep reinforcement learning," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[24] R. Sekar, O. Rybkin, K. Daniilidis, P. Abbeel, D. Hafner, and D. Pathak, "Planning to explore via self-supervised world models," in *International Conference on Machine Learning*. PMLR, 2020, pp. 8583–8592.

[25] H. Mao, W. Liu, J. Hao, J. Luo, D. Li, Z. Zhang, J. Wang, and Z. Xiao, "Neighborhood cognition consistent multi-agent reinforcement learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 05, 2020, pp. 7219–7226.

[26] J. Jiang, C. Dun, T. Huang, and Z. Lu, "Graph convolutional reinforcement learning," *arXiv preprint arXiv:1810.09202*, 2018.

[27] J. Jiang and Z. Lu, "Learning attentional communication for multi-agent cooperation," *Advances in neural information processing systems*, vol. 31, 2018.

[28] Y. Yang, R. Luo, M. Li, M. Zhou, W. Zhang, and J. Wang, "Mean field multi-agent reinforcement learning," in *International conference on machine learning*. PMLR, 2018, pp. 5571–5580.

[29] M. Samvelyan, T. Rashid, C. S. De Witt, G. Farquhar, N. Nardelli, T. G. Rudner, C.-M. Hung, P. H. Torr, J. Foerster, and S. Whiteson, "The starcraft multi-agent challenge," *arXiv preprint arXiv:1902.04043*, 2019.

[30] F. A. Oliehoek and C. Amato, *A concise introduction to decentralized POMDPs*. Springer, 2016.

[31] M. C. Machado, M. G. Bellemare, and M. Bowling, "Count-based exploration with the successor representation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 5125–5133.

[32] M. Henaff, R. Raileanu, M. Jiang, and T. Rocktäschel, "Exploration via elliptical episodic bonuses," *Advances in Neural Information Processing Systems*, vol. 35, pp. 37 631–37 646, 2022.

[33] Z. Gao, Y. Li, K. Xu, Y. Zhai, B. Ding, D. Feng, X. Mao, and H. Wang, "Dynamic memory-based curiosity: A bootstrap approach for exploration in reinforcement learning," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2023.

[34] H. Sun, L. Han, R. Yang, X. Ma, J. Guo, and B. Zhou, "Exploit reward shifting in value-based deep-rl: Optimistic curiosity-based exploration and conservative exploitation via linear reward shaping," *Advances in Neural Information Processing Systems*, vol. 35, pp. 37 719–37 734, 2022.

[35] S. Xu, Y. Liang, Y. Li, S. S. Du, and Y. Wu, "Beyond information gain: An empirical benchmark for low-switching-cost reinforcement learning," *Transactions on Machine Learning Research*, 2022.

[36] B. You, J. Xie, Y. Chen, J. Peters, and O. Arenz, "Self-supervised sequential information bottleneck for robust exploration in deep reinforcement learning," *arXiv preprint arXiv:2209.05333*, 2022.

[37] A. Graves, "Practical variational inference for neural networks," *Advances in neural information processing systems*, vol. 24, 2011.

[38] H. Kim, B. K. Lee, H. Lee, D. Hwang, S. Park, K. Min, and J. Choo, "Learning to discover skills through guidance," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[39] X. Liu, Y. Chen, and D. Zhao, "Comsd: Balancing behavioral quality and diversity in unsupervised skill discovery," *arXiv preprint arXiv:2309.17203*, 2023.

[40] M. I. Sener, Y. Nagai, E. Oztop, and E. Ugur, "Exploration with intrinsic motivation using object–action–outcome latent space," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 15, no. 2, pp. 325–336, 2021.

[41] C. Bai, R. Yang, Q. Zhang, K. Xu, Y. Chen, T. Xiao, and X. Li, "Constrained ensemble exploration for unsupervised skill discovery," *arXiv preprint arXiv:2405.16030*, 2024.

[42] C. Ying, Z. Hao, X. Zhou, X. Xu, H. Su, X. Zhang, and J. Zhu, "Peac: Unsupervised pre-training for cross-embodiment reinforcement learning," *arXiv preprint arXiv:2405.14073*, 2024.

[43] M. Xu, S. S. Ge, D. Zhao, and Q. Zhao, "Stable exploration via imitating highly-scored episode-decayed exploration episodes in procedurally-generated environments," *IEEE Transactions on Cognitive and Developmental Systems*, 2023.

[44] R. Yang, C. Bai, H. Guo, S. Li, B. Zhao, Z. Wang, P. Liu, and X. Li, "Behavior contrastive learning for unsupervised skill discovery," in *International Conference on Machine Learning*. PMLR, 2023, pp. 39 183–39 204.

[45] H. Liu and P. Abbeel, "Aps: Active pretraining with successor features," in *International Conference on Machine Learning*. PMLR, 2021, pp. 6736–6747.

[46] B. Eysenbach, J. Ibarz, A. Gupta, and S. Levine, "Diversity is all you need: Learning skills without a reward function," in *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[47] H. Liu and P. Abbeel, "Behavior from the void: Unsupervised active pre-training," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18 459–18 473, 2021.

[48] J. Kim, S. Park, and G. Kim, "Unsupervised skill discovery with bottleneck option learning," in *International Conference on Machine Learning*. PMLR, 2021, pp. 5572–5582.

[49] V. Campos, A. Trott, C. Xiong, R. Socher, X. Giró-i Nieto, and J. Torres, "Explore, discover and learn: Unsupervised discovery of state-covering skills," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1317–1327.

[50] A. Sharma, S. Gu, S. Levine, V. Kumar, and K. Hausman, "Dynamics-aware unsupervised discovery of skills," in *International Conference on Learning Representations*, 2019.

[51] N. Jaques, A. Lazaridou, E. Hughes, C. Gulcehre, P. Ortega, D. Strouse, J. Z. Leibo, and N. De Freitas, "Social influence as intrinsic motivation for multi-agent deep reinforcement learning," in *International conference on machine learning*. PMLR, 2019, pp. 3040–3049.

[52] H. Gao, X. Xu, C. Yan, Y. Lan, and K. Yao, "Gcen: Multi-agent deep reinforcement learning with grouped cognitive feature representation," *IEEE Transactions on Cognitive and Developmental Systems*, 2023.
[53] Y. Chen, K. Wang, G. Song, and X. Jiang, "Entropy enhanced multi-agent coordination based on hierarchical graph learning for continuous action space," *IEEE Transactions on Cognitive and Developmental Systems*, 2023.
[54] X. Li, J. Li, H. Shi, and K.-S. Hwang, "A decentralized communication framework based on dual-level recurrence for multiagent reinforcement learning," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 16, no. 2, pp. 640–649, 2023.
[55] L. Zheng, J. Chen, J. Wang, J. He, Y. Hu, Y. Chen, C. Fan, Y. Gao, and C. Zhang, "Episodic multi-agent reinforcement learning with curiosity-driven exploration," *Advances in Neural Information Processing Systems*, vol. 34, pp. 3757–3769, 2021.
[56] R. Gorsane, O. Mahjoub, R. J. de Kock, R. Dubb, S. Singh, and A. Pretorius, "Towards a standardised performance evaluation protocol for cooperative marl," *Advances in Neural Information Processing Systems*, vol. 35, pp. 5510–5521, 2022.
[57] L. Zhang, J. Li, H. Shi, K.-S. Hwang *et al.*, "Multi-agent reinforcement learning by the actor-critic model with an attention interface," *Neurocomputing*, vol. 471, pp. 275–284, 2022.