Neighborhood-oriented Decentralized Learning Communication in Multi-Agent System

Hao Dai[1], Jiashu Wu[1], André Brinkmann[2], and Yang Wang[1(✉)]

[1] Shenzhen Institutes of Advanced Technology, University of Chinese Academy of Science
[2] Zentrum für Datenverarbeitung, Johannes Gutenberg-Universität Mainz
yang.wang1@siat.ac.cn

Abstract. Since the partial observation issue is one of the crucial obstacles in multi-agent systems (MAS), a so-called "Centralized Training Decentralized Execution (CTDE)" paradigm has been widely studied by virtue of its integration of the global observations in the course of training process. Traditional CTDE paradigm suffers from observed locally during the execution phase, so numerous efforts have been made to study the communication efficiency among agents to promote the cognitive consistency and better cooperation. However, the vast majority of approaches still take effect in a centralized manner, which facilitates the agents to communicate with each other in a broadcast way. As a consequence, this centralized broadcast-based training process is infeasible when we adopt it to a more complex scenario. To address this issue, we propose a neighborhood-based learning communication approach in this paper to enable the agents to perform the training and execution in decentralized fashion based on the messages of its neighbor nodes. In particular, we design a novel encoder network whereby a two-step decision model is proposed to improve the performance of this decentralized training. To evaluate the method, we further implement a prototype and carry out a number of simulation-based experiments to demonstrate the effectiveness of our proposed method in multi-agent cooperation, when compared with the selected existing multi-agent methods to achieve the best rewards and drastically reduce the training data transmission.

Keywords: Multi-Agent System, Deep Reinforcement Learning, Learning Communication

1 Introduction

Multi-agent reinforcement learning (MARL) has emerged as a cutting-edge artificial intelligence technology, which has achieved significant success in massive challenging tasks [10, 17, 13]. However, although MARL shows excellent prospects in solving optimization problems, it encounters many additional obstacles when adapted to real-world tasks [20, 19, 8]. One is the well-known non-stationary problem. The simultaneous action of multiple agents brings not only the dimensional explosion of the observation space, but also the difficulty of

reaching consistency between actions and environment. Another is the matter of partial observation. A single agent has a limited perspective and can only observe the situation in its own neighborhood, which leads to a lack of overall consideration in its decision-making. To overcome these technical hurdles, a straightforward approach that came into researchers' minds is to aggregate all observations for centralized model training. Meanwhile, in the execution phase, the trained model relies on local observations to make decisions independently. Through optimization algorithms, such as value decomposition and credit assignment, this paradigm, called centralized training and decentralized execution (CTDE) [17, 13, 15], alleviates parts of the difficulty in the convergence of model training. However, it still suffers from chaotic decisions caused by partial observation in the execution stage.

Another intuitive idea comes from bionics, where animals use communication to negotiate and cooperate. Introducing communication into multi-agents means that agents can share their perception of the environment and their intentions to act, thus achieving unanimity. Agents usually encode their own observations and send them to other agents to assist decision-making. The open problems in learning communication include how to encode and decode the message, select communication objects, and design communication mechanisms [7, 14, 21, 19, 2]. Tremendous researchers work towards these aspects and have made marvelous achievements, showing the advantages of communication in MARL, such as CommNet [16], IC3Net [14], TarMAC [1], I2C [2], and so on. Although promising, these methods somehow imply centralization requirements, for instance, communicating in a broadcast manner [16, 6, 1], concentrating all observations to train the encoder or communication networks [7, 2].

To address these issues, we get inspiration from social psychology, which finds that cognitive consistency within a neighborhood matters, and people tend to cooperate with neighbors [11]. To this end, we design a brand-new neighbor-oriented learning communication approach, in which agents make decisions and train models by utilizing the neighboring messages. We made the following contributions in this paper: (1) We design an encoder network according to the idea of neighborhood cognitive consistency. By calculating the KL divergence of different neighbor messages, the encoder network can represent the consensus information of the neighborhood. (2) We propose a *pseudo-pre-acting* mechanism. This mechanism can send decision information along with messages to assist neighbors' decision-making and reduce the non-stationary of MAS. (3) We develop a brand-new learning communication method based on the AC algorithm, *Neighborhood-Oriented Actor-Critic (NOAC)*, and construct abundance experiments to validate our findings.

The organization of the paper is as follows: we discuss related works regarding MARL in Section 2 and introduce some background knowledge in Section 3. We illustrate the formulation and methodology in Section 4. Afterward, we present the simulation studies to validate our findings in Section 5, followed by the conclusion of the paper in the last section.

2 Related Work

A critical problem of a MARL is how to characterize the interaction between agents, so numerous works propose that we can learn a joint value to guide agents' actions [10, 17, 13, 15, 5]. Based on this idea, researchers developed a widely used paradigm, centralized training decentralized execution (CTDE). In this paradigm, all agents share the information of joint value to mitigate non-stationary. Although This paradigm ha won a series of medals, the lack of extra information in the execution phase still plagues the robustness of the action policy. To address the agents' limited perspective issue, exchanging observations among agents to gain an understanding of the entire environment could be an intuitive and favorable idea. Plenty of works have demonstrated that learning communication is a promising MARL approach [3, 16, 14, 1, 6, 7, 2]. These methods mainly focus on how to combine communication into deep reinforcement learning networks.

As a pioneer of learning communication, DIAL [3] takes DQN to implement a learnable communication, thus introducing backpropagation into communication networks first time. DIAL's shortcoming is that it only handles discrete messages. Therefore, CommNet [16] employs a hidden layer to encode observations as continuous messages. In these methods, messages are shared through a fully connected network among agents, that is, communicating in a broadcast manner. This coarse message delivery mode brings redundant communication and massive interference information. For a refined measure of the message, Lowe *et al.* [9] analyzed the urgency of messages and proposed two indicators, positive signaling and positive listening, to measure the utility of messages. Based on the concern that messages are not always beneficial to decision-making, some works tried to investigate how to communicate efficiently. Sort of methods represented by ATOC [6] and IC3Net [14] introduced a gate mechanism to determine whether a message needs to be transmitted. In contrast, other alternative approaches, *e.g.*, SchedNet [7], TarMAC [1], and I2C [2], adopted some weighting mechanisms to reduce the communication between some agents.

All the aforementioned methods show high efficiency and advancement. Still, inevitably there is a common problem: global information (including all observations and actions) is needed to calculate TD-error during training. This feature puts them at a disadvantage when dealing with large-scale multi-agent systems. The dilemma seems to stem from the fact that non-stationary requires global information to counteract it. To this end, we designed a learning communication method that trains and executes depending on neighborhood information. One of the essential differences between previous works and ours is that we no longer take global information into account to calculate TD-error. This setting is more practical and can flexibly parallelize training.

3 Background

3.1 Dec-POMDP

Decentralized partially observable Markov decision process (*Dec-POMDP*), is commonly used to characterize multi-agent systems, which assume that each agent possesses only partial observations of the environment and follows a hidden Markov decision process for state transitions. A *Dec-POMDP* can be defined as a tuple:

$$\mathcal{D} = \langle \mathcal{N}, \mathcal{A}, \mathcal{R}, \mathcal{O} \rangle \tag{1}$$

here, $\mathcal{N}$ denotes the number of agents in total, $\mathcal{A} = a_1 \times ... \times a_{\mathcal{N}}$ is the set of joint action, $\mathcal{R} = \{r_1, ..., r_{\mathcal{N}}\}$ is the reward set, $\mathcal{O} = \{o_1, ..., o_{\mathcal{N}}\}$ is the set of observations, which satisfies $o_i \cup o_j \nsubseteq o_i$ or o_j.

Our goal is to guide the agent to achieve the maximum cumulative reward $\mathbb{E}[\sum_{t=0}^{+\infty} \gamma \vec{r}_t]$, here γ is the discount factor. To this end, we define a set of policies $\vec{\pi} = \{\pi_1(a_1|o_1), ..., \pi_{\mathcal{N}}(a_{\mathcal{N}}|o_{\mathcal{N}})\}$, and the final objective is to learn the optimal policy to maximize the cumulative reward.

$$\mathcal{J}(\theta) = \mathbb{E}_{a \sim \pi(\theta)}[\sum_{t=0}^{+\infty} \gamma \vec{r}_t] \tag{2}$$

Remarkably, considering that we mainly focus on the cooperative agents, there is commonly only one global reward r, which is one of the crucial reasons why MARL is hard to converge.

3.2 Actor-Critic

Actor-Critic is a typical reinforcement learning algorithm, characteristic of combining the advantages of value-based and policy gradient methods. It includes two networks: actor and critic. Critic network is used to estimate the current status value $V(o; \theta^c)$, which is updated by calculating the current TD-error:

$$\mathcal{L}(\theta^c) = \gamma * V(o'; \theta^c) + r - V(o; \theta^c) \tag{3}$$

Meanwhile, the actor network adopts the policy gradient method to perform actions for agents. The idea of policy gradient is to give the bigger action-value higher sampling probability. Therefore, combined with the advantage function of the critic network, its update method is as follows:

$$\nabla_{\theta^a} \mathcal{J}(\theta^a) = \nabla_{\theta^a} [\log_{\theta^a} \pi_{\theta^a}(a|o) \mathcal{L}(\theta^c)] \tag{4}$$

It is worth noting that, to avoid non-stationary problems in multi-agents when training, we usually default that the critic network is centralized, or each critic can obtain global information through communications or other means. That is, o and a in the above formula are the set of observations and actions of all agents, respectively. This kind of method, though, has shown good convergence guarantee, the cost of aggregating all the information is unbearable in large-scale multi-agent systems.

4 Methodology

To break through the centralized dilemma of multi-agent training, we get some inspiration from the literature [11]. In most cases, agents only interact with their neighbors, which is also in line with the run-on form of human society. To this end, we design a neighborhood-oriented method for multi-agent training in the subsequence section.

4.1 Neighborhood Cognitive

We denote the set of neighbors of agent i as $N(i)$, *i.e.*, agent $j \in N(i)$ is the neighbor of agent i. According to the assumption of [11], there is a so-called **true hidden cognitive variable** C in each neighborhood, and all partial observations are the interplay of these variables. This assumption is intuitive, we

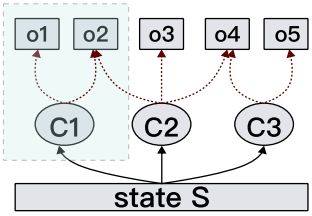


Fig. 1. The partial observations are generated from the interplay of hidden cognitive variables.

can imagine that multiple neighbors observe a global state S and attain multiple hidden variable C_k, and agent i observes C_k and get observation o_i, as illustrated in *Fig. 1* We can assume that $\{C_k\}$ has a strong representation of global state S, and the observation of agent i can be derived as follows:

$$p(o_i|S) = \sum_k p(o_i|C_k) \tag{5}$$

Therefore, we can regard the aggregation of C_k as an intermediate representation of S. Although this hidden state does not change the situation that the global reward cannot be decomposed to a single agent (or neighborhood), it did mitigate part of the uncertainties by such conversion. As a consequence, we design an encoder network, $m_i = \mathcal{M}(o_i)$, to encode the observation of agent i and send it to the neighbors. We can rewrite the AC algorithm with neighborhood communication as follows:

$$\begin{aligned} \mathcal{L}(\theta_i^c) &= \gamma * V_{j\in N(i)}(m_i', m_j'; \theta_i^c) + r - V_{j\in N(i)}(m_i, m_j; \theta_i^c) \\ \nabla_{\theta_i^a}\mathcal{J} &= \nabla_{\theta_i^a}[\log_{\theta_i^a} \pi_{\theta_i^a; j\in N(i)}(a_i|m_i, m_j)\mathcal{L}(\theta_i^c)] \end{aligned} \tag{6}$$

Note that the input to the AC algorithm becomes an encoded group of messages. Neighborhood consistency assumes that all observations are based on the same hidden variable C_k, which means that o_i and o_j are inherently correlated, so we can achieve consistency in cognition by minimizing message differences. Although the exact value of C_k is unknown, we can leverage KL-divergence to measure the differential of messages. Therefore, we derive the loss function of the encoder

network as follows:

$$\mathcal{L}^e(\theta_i^m) = \sum_{j \in N(i)} D_{\mathrm{KL}}(P(m_i; \theta_i^m) \| P(m_j)) \tag{7}$$

By the designed encoder network, the benefits of encoding are not only improved neighborhood cognize consistency but also a significant reduction in the amount of data transferred.

4.2 Pseudo Pre-Acting

Passing messages brings additional information to the actor network, and affects its decision-making by integrating neighborhood information. The neighborhood message changes the receiver's decision, and we use this differential to indicate how positive the agent is to listen to the message. By using the causal inference method, we can define indicators to show how decision-making affecting by its neighbors.

$$\mathbb{I}_i = D_{\mathrm{KL}, j \in N(i) \cup i}(\pi(a|o_i, m_j) \| \pi(a|o_i)) \tag{8}$$

Unfortunately, this kind of influence demonstrates the impact of neighborhood information on decision-making and does not indicate whether the changes increase or decrease the reward. Hence, we look at the role of messages from the perspective of game theory.

As we know, multi-agent systems can usually be formalized as a game, in which each agent takes their own action and gets a payoff. An important concept in game theory is the Nash equilibrium (NE), which means that the system is in a sort of steady state. Although NE is not always meant to maximize social welfare (reward), it is robust enough and usually better than the non-stationary. Moreover, the maximize reward must also be some element of the NE set. Therefore, if we let multi-agents reach Nash equilibrium cooperatively, we can change the problem of maximizing rewards to finding the optimal point among multiple NE states.

There are many ways to solve Nash equilibrium, and the most effective one is to calculate the best response. We let Ai and u be the action space and the utility function of agent i, respectively, then the best response can be calculated as follows:

$$a^* = \arg\max_{a_i \in A_i} u(a_i, \vec{a}_{-i}) \tag{9}$$

here, $\vec{a}_{-i}$ represents the action set of other agents except i. One of the definitions of Nash equilibrium is that all agents are in the best response state, which means that no agent can unilaterally change its action for greater rewards. That is, when agents know the actions of other agents, they can adopt strategies to get better rewards. We can share agents' actions through communication and compute actions through $\pi_i(a_i|o_i, a_j; j \in N(i))$.

However, it is not practical to send all neighborhood actions to agent i for decision-making, because agents act simultaneously rather than sequentially. When one agent receives the other agents and changes its action, this change will lead to the action of the other agents should also correspond to the change.

To avoid this recurring chain reaction, we propose a two-step decision-making method:

1) obtain $\hat{a}_i$ by $\hat{\pi}_i(\hat{a}_i|o_i)$, and send it to neighbors;

2) execute action $\pi_i(a_i|o_i, a_j; j \in N(i))$ after the actions of neighbors are received.

We refer to this approach as **pseudo-pre-acting (PPA)** mechanism, in which $\hat{a}$ and $\hat{\pi}$ are called pseudo-action and pseudo-policy, respectively. Note that π_i is the policy we actually learned by interacting with the environment, so we update the pseudo-policy network $\hat{\pi}_i$ by the aforementioned causal inference indicator:

$$\mathcal{L}^{\hat{\pi}}(\theta_i^m) = D_{\mathrm{KL}, j \in N(i)}(\hat{\pi}_i(\hat{a}_i|o_i; \theta_i^m) \| \pi_i(a_i|o_i, a_j, m_j)) \tag{10}$$

There are two perspectives to explain why we make the pseudo-action approx to the actual action: On the one hand, the consistency of the pseudo action and the actual action make the best response calculated by other agents effective; on the other hand, on the premise of receiving other acts, the approximate of the two types of actions indicates that the current state is in some kind of equilibrium.

4.3 Neighborhood-Oriented Actor-Critic

Combined with the above methods, we propose a neighborhood-oriented MARL approach based on actor-critic: **Neighborhood-Oriented Actor-Critic (NOAC)**. The overall architecture of NOAC is illustrated in *Fig. 2*, which consists of three parts: encoder network, actor network, and critic network.

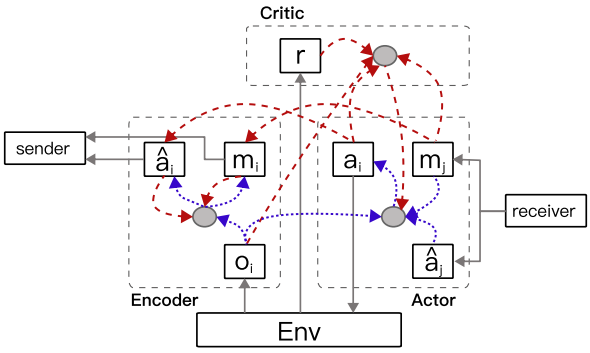


Fig. 2. The overall architecture of NOAC. The blue lines are the execution dataflow, while the red lines are training dataflow. Gray cricles are neural networks.

In the execution phase, the encoder network encodes the local observation, outputs the message m_i and pseudo action $\hat{a}_i$, and afterward sends it to other neighbors. After receiving messages and pseudo-actions from other neighbor nodes, the actor network conducts these messages into forward computing. It will select appropriate actions to interact with the environment and step into the next epoch.

In the training stage, the encoder network calculates $\mathcal{L}^e$ and $\mathcal{L}^{\hat{\pi}}$ according to the received messages and the actions performed by the actor network, respec-

tively, and updates parameters with the following loss function.

$$\mathcal{L}(\theta_i^m) = \mathcal{L}^e(\theta_i^m) + \mathcal{L}^{\hat{\pi}}(\theta_i^m) \tag{11}$$

Concerning the actor network, we adopt the policy gradient to update parameters as follows:

$$\nabla_{\theta_i^a}\mathcal{J} = \nabla_{\theta_i^a}[\log_{\theta_i^a}\pi_{\theta_i^a;j\in N(i)}(a_i|o_i, m_i, a_j, m_j)\mathcal{L}(\theta_i^c)] \tag{12}$$

As for the training of the critic network, the main concern is on how to calculate TD-loss. As mentioned above, the critic network is usually centralized since it needs to estimate Q_{total}, which is directly relevant to gloabl reward. Since the global reward cannot be directly assigned to individuals in multi-agent cooperation, the centralized network is needed to evaluate the value function. Likewise, although we propose that C_k can characterize part of the global state, we still cannot assign the global reward to a concrete C_k. However, considering that in some environments where agents move around, each agent transforms the C value as it moves, we have relaxed this constraint somewhat. As the size of the neighborhood directly affects the approximation of Q_C and Q_total, we define the TD-loss of the critic network as follows:

$$\mathcal{L}(\theta_i^c) = \gamma * V_{j\in N(i)\cup i}(m_i', m_j'; \theta_i^c) + \frac{|N(i)|}{\mathcal{N}} * r - V_{j\in N(i)\cup i}(m_i, m_j; \theta_i^c) \tag{13}$$

Thanks to the decentralized network design, all agents run in parallel during the execution and training process, which only needs to be synchronized during communication and environment step. Therefore, the networks of agents can be deployed in different servers and communicate through protocols, *e.g.*, GLOO, NCCL, TCP, etc. This characteristic is particularly efficient in large-scale multi-agent environments.

5 Experiments

To validate our findings, we conducted empirical studies to evaluate the performance of the proposed NOAC. We implemented a test platform based on multi-agent particle environment [12] and took a cooperative game as the environment simulator.

5.1 Setup

Environment. We took the cooperative navigation game [10] as the simulation environment. As shown in *Fig. 3*, there are $\mathcal{N}$ agents and $\mathcal{N}$ landmarks in the environment, and the agents need to cooperate with each other to occupy all the landmarks. The environment takes the sum of the minimum distance between all agents and landmarks as the global reward value, that is, there is no individual reward for each agent.

Each agent has its own observation o_i and takes the closer agents as its neighbors. And each agent can only stand on one landmark, and there will be a penalty (negative reward) for collisions. In the experiment, we set $\mathcal{N} = 7$, $N(i) = 3$ and $max\text{-}cycles = 40$. All other settings are default values in the *PettingZoo* open-source library [18]. It is worth noting that the agent's environment is open

rather than bounded in a particular range, which may differ from some other cooperative navigation settings.

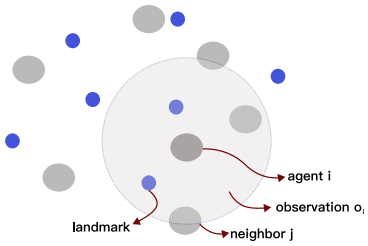


Fig. 3. Illustration of cooperative navigation environment.

Baselines. We compared the proposed method NOAC to other state-of-the-art MARL baselines :

- *TarMAC* [1]: An attention-based learning communication method, it weighs the importance of incoming messages.

- *IC3Net* [14]:A gate-based method to decide whether to communicate with others, in which messages are transmitted in broadcast mode.

- *MADDPG* [10]:A classical CTDE algorithm without communication.

- *DDQN* [4]:A typical single-agent algorithm. In our setting, it can sense the global state and simultaneously outputs all agents' actions.

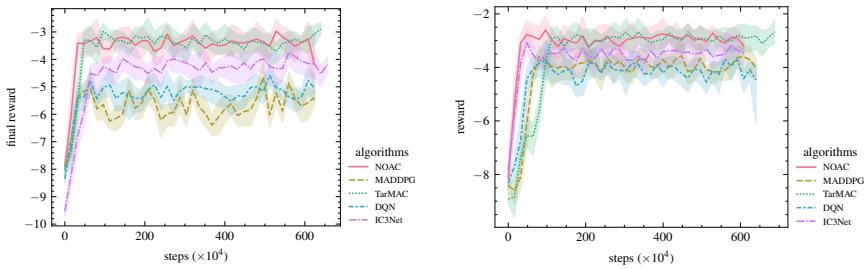
Hyperparameters. To decline the off-site factors in the comparison, we adopted the same network structure in most baselines. In addition, we set the learning rate $lr = 1 \times 10^{-3}, batch\_size = 64$ for all methods. We implemented the testbed based on *PyTorch* and *PettingZoo* and ran it on a $3 \times$ Tesla V-100 server.

5.2 Numerical Results

Global Reward. Firstly, we investigated the rewards of several baselines, which is the primary concern for MARL problems. Previous cooperative navigation experiments often only focused on the average reward, while we are more concerned with the state the agent is finally in when the round ends, the final reward. Therefore, we compared the two types of rewards of baselines: final reward and total reward.

The comparison of the final rewards is illustrated in *Fig. 4(a)*, in which the final rewards of *NOAC* are approx to *TarMAC*, more significant than several other baselines. This result indicates the effectiveness of our approach. It is worth noting that these methods, including *TarMAC*, require a central exchange of information, while *NOAC* only collects neighborhood information. Similarly, *NOAC* also performs superbly on total rewards, as shown in *Fig. 4(b)*. It is noted to point out that the gap of baselines on total reward is less different than that on the final reward, so we treat the final reward as the metric for the following experiments.

We executed 500 episodes with these trained models, and *Tab. 1* reports the mean and standard deviations of the total and final rewards. *NOAC* outperforms



(a) (b)

Fig. 4. Comparison of final reward (a) and total reward (b).

all the baselines with the highest average of final rewards. And these results suggest that our partial information and messaging training approach is comparable to these methods concentrating all the information.

Table 1. Summary of final reward and total reward of MARL baselines

Algorithms	NOAC	MADDPG	TarMAC	IC3Net	DQN
Final Reward Mean	**-3.292036**	-5.627667	-3.335745	-4.164599	-5.120261
Final Reward Std	0.595101	0.646373	0.766422	0.457646	0.626979
Total Reward Mean	-3.358014	-4.168850	-3.201513	-3.512844	-4.132432
Total Reward Std	1.430986	1.219422	1.408928	0.964834	0.860662

Neighborhood Impact. Although *NOAC* is designed for neighborhoods, it can also handle broadcast messages. To investigate whether *NOAC* does indeed adopt neighbors' messages, we contrast it with two exceptional cases: one where no communication occurs at all, which is a decentralized AC algorithm with independent control, denoted as "*NOAC-No-Comm*"; the other is a fully connected communication network with *NOAC*, which we call "*NOAC-FC*."

After training with the same settings, the comparison of the experimental results is illustrated in *Fig. 5(a)*. Not surprisingly, the decentralized algorithm *NOAC-No-Comm* without communication performs the worst, even worse than with all other communication or CTDE baselines, because this method only trains with local observations of the agent itself. This finding confirms that communication makes considerable progress to MARL.

Nevertheless, it is worth noting that *NOAC-FC* with fully connected communication does not bring a significant improvement in performance. At the beginning of training, the convergence of *NOAC-FC* is even slower than that of partial connection. This result illustrates, in a way, that more messages are not always better, and actually is consistent with a proposition of selective communication of related works. It also demonstrates that neighborhood cognitive consistency does exist, and the agent can achieve an approximate effect of the global observation through the messages of the neighbors.

Ablation. Finally, to further investigate the contribution of the proposed encoder network and pseudo-pre-acting to the *NOAC*, we performed ablation experiments. We conducted three different sets of experiments: 1) the *NOAC* without pseudo-pre-acting mechanism, "*NOAC-No-Pseudo*"; 2) The method of not encoding the neighbor observation, "*NOAC-No-Encoder*", it should be noted that the raw observation of the neighbor will still be transmitted; 3) The "*NOAC-*

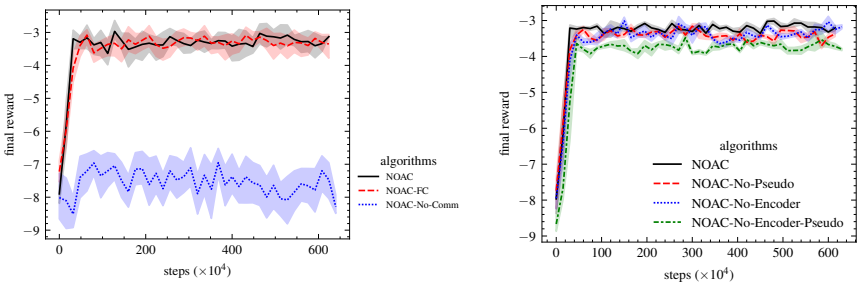


Fig. 5. Reward comparison of different neighborhood impacts (a) and total reward (b).

No-Encoder-Pseudo" algorithm without an encoder or pseudo-pre-acting mechanism, in which each agent aggregates the neighborhood observations for training.

Fig. 5(b) shows a comparison of the ablation experiments, and it can be seen that the removal of either mechanism leads to a slight decrease in performance. In particular, the *NOAC-No-Encoder-Pseudo* with the encoder and **PPA** removed shows a significant drop in reward. To indicate the gap more clearly, *Tab. 2* reports the difference in rewards for each setting. It should be noted that either method makes use of information from neighbors.

Table 2. Summary of the final reward of ablation experiments

	NOAC	**No-Encoder**	**No-PPA**	**No-Encoder-PPA**
Final Reward Mean	-3.292036	-3.458257	-3.514603	-3.828242
Final Reward Std	0.595101	0.411429	0.955895	0.592898

Despite the absence of an encoder, actor and critic networks can still extract critical information from raw observations. Thus, the gap is not such significant. However, the encoded message is around 3/4 dimensional less than the original observation ($42 \rightarrow 12$), which shows a considerable advantage in terms of latency in both network transmission and tensor operations.

6 Conclusions

In this paper, we proposed a neighborhood-oriented MARL training method, which leverages solely neighbors' messages rather than global information to learn policies. Experiments demonstrate that this decentralized training method is comparable to the mainstream CTDE methods. Although this is a crude attempt, it shows the potential of decentralized learning methods in solving MARL problems. Decentralized MARL learning is not only closer to real-world scenarios but also has excellent benefits in terms of computational efficiency. We anticipate this paradigm could be developed and applied to more practical problems.

References

1. Das, A., Gervet, T., Romoff, J., Batra, D., Parikh, D., Rabbat, M., Pineau, J.: Tarmac: Targeted multi-agent communication. In: ICML (2019)

2. Ding, Z., Huang, T., Lu, Z.: Learning individually inferred communication for multi-agent cooperation. NeurIPS (2020)
3. Foerster, J., Assael, I.A., de Freitas, N., Whiteson, S.: Learning to communicate with deep multi-agent reinforcement learning. In: NeurIPS (2016)
4. Hasselt, H.v., Guez, A., Silver, D.: Deep reinforcement learning with double q-learning. In: AAAI (2016)
5. Iqbal, S., Sha, F.: Actor-attention-critic for multi-agent reinforcement learning. In: International Conference on Machine Learning (ICML) (2019)
6. Jiang, J., Lu, Z.: Learning attentional communication for multi-agent cooperation. Advances in Neural Information Processing Systems (NeurIPS) (2018)
7. Kim, D., Moon, S., Hostallero, D., Kang, W.J., Lee, T., Son, K., Yi, Y.: Learning to schedule communication in multi-agent reinforcement learning. In: International Conference on Learning Representations (ICLR) (2019)
8. Leonardos, S., Overman, W., Panageas, I., Piliouras, G.: Global convergence of multi-agent policy gradient in markov potential games. In: ICLR (2022)
9. Lowe, R., Foerster, J., Boureau, Y.L., Pineau, J., Dauphin, Y.: On the pitfalls of measuring emergent communication. In: AAMAS (2019)
10. Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, O.P., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. In: NeurIPS (2017)
11. Mao, H., Liu, W., Hao, J., Luo, J., Li, D., Zhang, Z., Wang, J., Xiao, Z.: Neighborhood cognition consistent multi-agent reinforcement learning. AAAI (2020)
12. Mordatch, I., Abbeel, P.: Emergence of grounded compositional language in multi-agent populations. arXiv preprint arXiv:1703.04908 (2017)
13. Rashid, T., Samvelyan, M., de Witt, C.S., Farquhar, G., Foerster, J., Whiteson, S.: Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In: International Conference on Machine Learning (ICML) (2018)
14. Singh, A., Jain, T., Sukhbaatar, S.: Individualized controlled continuous communication model for multiagent cooperative and competitive tasks. In: International Conference on Learning Representations (ICLR) (2019)
15. Son, K., Kim, D., Kang, W.J., Hostallero, D.E., Yi, Y.: Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In: International Conference on Machine Learning (ICML) (2019)
16. Sukhbaatar, S., Fergus, R., et al.: Learning multiagent communication with backpropagation. In: NeurIPS (2016)
17. Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W.M., Zambaldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J.Z., Tuyls, K., Graepel, T.: Value-decomposition networks for cooperative multi-agent learning based on team reward. In: AAMAS (2018)
18. Terry, J.K., Black, B., Grammel, N., Jayakumar, M., Hari, A., Sulivan, R., Santos, L., Perez, R., Horsch, C., Dieffendahl, C., Williams, N.L., Lokesh, Y., Sullivan, R., Ravi, P.: Pettingzoo: Gym for multi-agent reinforcement learning. arXiv preprint arXiv:2009.14471 (2020)
19. Wang, T., Wang, J., Zheng, C., Zhang, C.: Learning nearly decomposable value functions via communication minimization. In: ICLR (2020)
20. Wen, Y., Yang, Y., Wang, J.: Modelling bounded rationality in multi-agent interactions by generalized recursive reasoning. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence. IJCAI'20 (2021)
21. Zhang, S.Q., Zhang, Q., Lin, J.: Efficient communication in multi-agent reinforcement learning via variance based control. In: Advances in Neural Information Processing Systems (NeurIPS) (2019)