Heterogeneous Multi-Agent Communication Learning via Graph Information Maximization

Wei Du
School of Computer Science and Technology
China University of Mining and Technology
Xuzhou, China
1394471165@qq.com

Shifei Ding
School of Computer Science and Technology
China University of Mining and Technology
Xuzhou, China
dingsf@cumt.edu.cn

***Abstract*—Communication learning is an effective way to solve complicated cooperative tasks in multi-agent reinforcement learning (MARL) domain. Graph neural network (GNN) has been widely adopt for learning the multi-agent communication and various GNN-based MARL methods have emerged. However, most of these methods are not specially designed for heterogeneous multi-agent scenarios, where agents have heterogeneous attributes or features based on different observation spaces or action sets. Without effective processing and transmission of heterogeneous feature information, communication learning will be useless and even reduce the performance of cooperation. To solve this problem, we propose a communication learning mechanism based on heterogeneous GNN and graph information maximization to learn effective communication for heterogeneous agents. Specifically, we use heterogeneous GNN for learning the efficient message representations, which aggregate the local feature information of neighboring agents. Furthermore, we maximize the mutual information (MI) between message representations and local values to make efficient use of information. Besides, we present a MARL framework that can flexibly integrate the proposed communication mechanism with existing value factorization methods. Experiments on various heterogeneous multi-agent scenarios demonstrate the effectiveness and superiority of the proposed method compared with baselines.**

***Index Terms*—Communication learning, Multi-agent reinforcement learning, Mutual information, Graph neural network.**

I. Introduction

In recent years, multi-agent reinforcement learning (MARL) has seen tremendous growth and attracted wide attention [1]. The paradigm of centralized training and decentralized execution (CTDE) is popular and widely used in MARL because it can address scalability issues and partial observability limitations of MARL [2]. Most of the CTDE based MARL methods can be roughly divided into value factorization methods [3], [4], [5] and communication learning methods [6], [7], which provide different solutions for further exploiting CTDE paradigm. The value factorization methods factorize the global value function into the set of the local individual value function of each agent to further tackle the scalability issue. The communication learning methods enable agents to share important information in the decentralized execution period, which can further promote action coordination.

This work was supported by the Fundamental Research Funds for the Central Universities No. 2022XSCX37.
DOI reference number: 10.18293/SEKE2023-099

Recently, graph neural network (GNN), as an efficient representation learning method [8], has been widely utilized to build communication learning mechanism of MARL, which generally regards agents as nodes in the graph, with the communication channels corresponding to edges. Many state-of-the-art MARL methods fall into this GNN-based communication paradigm [9], [10]. However, most GNN-based communication learning methods are not specially designed for heterogeneous scenarios, where agents have different observation spaces or action sets. Therefore, these methods can not effectively process and transmit heterogeneous feature information, which leads to inefficient communication learning and affects action coordination.

To solve these problems, we present a Communication Learning mechanism of multi-Agent Reinforcement learning (CLAR) for heterogeneous scenarios. The proposed mechanism utilizes heterogeneous GNNs to model the heterogeneous agents and fuse feature information of neighboring agents to obtain high-level message representations. Besides, the proposed mechanism leverages mutual information (MI) optimization to obtain high-quality message representations for action coordination. Furthermore, we present a MARL framework that integrates the value factorization and the proposed communication learning mechanism. This framework can maintain the advantages of the stability and scalability of the value factorization methods, and promote better action coordination between agents by effectively processing and utilizing heterogeneous feature information. The following are the primary contributions of the proposed method,

1) We present a MARL framework that integrates communication learning mechanism and value factorization methods for heterogeneous scenarios, which solved the communication learning challenge of heterogeneous scenarios and the action discoordination issue of value factorization methods.

2) We first introduce the MI between the local values and the message representations in MARL. We use the MI maximization to learn the most valuable and expressive information from different classes of agents for better action coordination.

3) We design the heterogeneous GNN to learn heterogeneous multi-agent communication, which efficiently models the heterogeneous scenarios and achieves the fusion and transmission of heterogeneous information.

II. Related work

A. GNN-based MARL

Recently, learning multi-agent communication via GNN in MARL has attracted popular attention. Jiang et al. [9] first extends GNN to MARL for multi-agent communication learning. Das et al. [10] leverages GNN with soft attention mechanism to learn whom to receive messages and what messages to pass. Sheng et al. [11] utilizes the hierarchical GNN to achieve effective communication learning by sharing information among agents and groups. Ryu et al. [12] presents a hierarchical attention mechanism based on GNNs, which models the relationships between agents effectively. Liu et al. [13] utilizes a two-stage attention mechanism to model the complete graph for communication learning. Niu et al. [14] proposes an attentional GNN to tackle the challenges of how to process information and when to communicate.

The existing GNN-based MARL methods have achieved efficient communication learning by modeling interactions or relationships between agents. However, most of these methods are not specially designed for heterogeneous scenarios, where agents have heterogeneous attributes or features based on different observation spaces or action sets. Although some works attempt to use heterogeneous GNNs to learn communication in heterogeneous scenarios [15], [16], these works do not further optimize the high-level message representation, resulting in communication learning less effective. Different from them, the proposed method utilizes MI optimization to obtain high-quality message representations for action selection.

B. Graph Convolutional Network

Graph Convolutional Network (GCN) as a popular GNN module generally utilizes the information passing between the graph nodes to learn the structural dependency between nodes [8]. Concretely, each node aggregates the feature of adjacent graph nodes to compute a new high-level feature vector, the feature aggregation procedure is shown in Eq.(1).

$$h_i' = \sigma\Big(\sum_{b\in N(i)} \omega h_b / d_{ib}\Big) \tag{1}$$

where h_i' represents the aggregated feature vector of node i , $\sigma(.)$ denotes the activation function and ω denotes the learnable weights. $b \in N(i)$ contains the immediate neighbor nodes of node i , where b represents the index of the neighbors. d_{ib} denotes the normalization term, which has several options and the common one is $\sqrt{|N(i)N(b)|}$. After feature aggregation through several layers, the high-level feature representation of node i can integrate the structural information of nodes reachable from graph node i.

In our work, we use an efficient enhancement of Eq. (1), which replace d_{ib} with the attention coefficients α_{ib} as follows: $\alpha_{ib} = \mathrm{softmax}_b\left(\sigma'\left(\bar{a}^T\left[\omega h_i \| \omega h_b\right]\right)\right)$ where σ' utilizes the LeakyReLU nonlinearity function, $\bar{a}$ represents the learnable weight, $||$ denotes concatenation operation. The softmax function is utilized to normalize the coefficients over all neighbor nodes b .

III. Methodology

In this section, we introduce the proposed MARL framework and the proposed communication learning mechanism based on the heterogeneous GCN and MI optimization.

A. Problem formulation

In our work, we formulate the heterogeneous multi-agent issue as the Heterogeneous Multi-Agent POMDP represented by $G = \langle C, I, \{S^c\}_{c\in C}, \{A^c\}_{c\in C}, \{O^c\}_{c\in C}, R\rangle$ [16]. C represents the set of all classes of agents in the heterogeneous scenarios and the index $c \in C$ represents the class that the agent belongs to, the total number of classes is denoted as n. $I = \sum_{c\in C} I^c$ represents the total number of collaborating agents in which I^c denotes the number of agents that belong to class c . $\{S^c\}_{c\in C}$ denotes the state space in which S^c represents the joint state of agents of class c. $S^c = [s_i^c]|_{i=1}^{I^c}$ and s_i^c represents the state of agent i of class c. $\{A^c\}_{c\in C}$ represents the action space in which A^c represents the joint action space of agent i of class c. $A^c = [a_i^c]|_{i=1}^{I^c}$ and a_i^c represents the action of agent i that belongs to class c. $\{O^c\}_{c\in C}$ represents the observation space where O^c represents the observation space of agents of class c. Each agent i of class c obtains a partial observation $o_i^c \in O^c$ and takes an action $a_i^c \in A^c$, which forms the joint action $a \in \{A^c\}_{c\in C}$. Then agent i can obtain an immediate shared reward $R(s,a)$, which encourages cooperated behavior among agents. The target of all the agents is to learn the optimal joint action-value function $Q_{tot}(\tau, a) = \mathbb{E}_{s,a}[\sum_{t=0}^{\infty} \gamma^t R(s,a)]$, where τ represents the joint action-observation history, and γ represents the temporal discount factor.

B. Overall Framework

The overall framework of the proposed method is shown in Fig. 1, which contains 3 modules: feature encoding module, communication learning module, and value decomposition module. For agent i, it receives the local observation o_i and then utilizes Multi-Layer Perceptron (MLP) and Gated Recurrent Unit (GRU) to process the local observation and produce the feature h_i . Then h_i is fed to encoder to generate type-specific features $\left(h_i^1, h_i^c, \ldots, h_i^n\right)$ for communication. The communication module is built by the heterogeneous GCN to pass the heterogeneous feature information among agents and learn specific communication policies based on agent types. By stacking multiple heterogeneous GCN layers, the high-level embedding m_i of agent i can be extracted through multiple rounds of communication.

In value decomposition module, the local individual action-value function $Q_i\left(\tau_i, a_i, m_i\right)$ is calculated based on local observation history τ_i and feature messages m_i received from the communication learning module. Then the local Q values obtained by all the agents are input into a mixing network to generate an estimation of the global value. Besides, we utilize mutual information optimization to further strengthen the correlation between the communication learning module and value decomposition module. The proposed communication mechanism can be fused with any value factorization

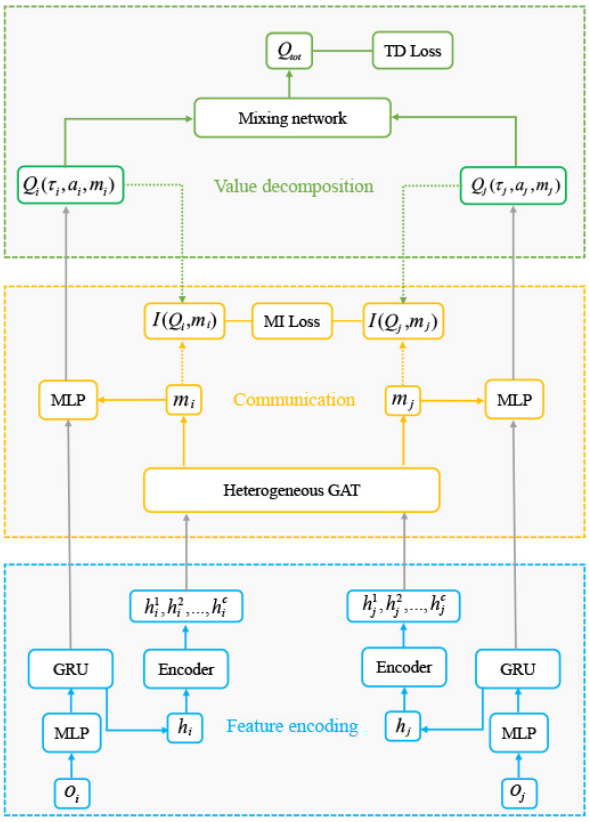


Fig. 1: Framework of the proposed method

methods under the paradigm of centralized training and decentralized execution. During the decentralized execution phase, the agents can communicate and take actions in a decentralized manner based on the communication learning module.

C. Communication Learning

In this section, we utilize the heterogeneous GCN to build the communication learning mechanism. The overall framework of the communication module is shown in Fig. 2, which contains the message sender procedure and receive procedure. For simplicity and universality, we consider a heterogeneous scenario with three types of agents, $C = \langle F, K, L \rangle$ and take the agent of class F as an example. For agent f of class F , its obtained feature h_f is processed by different weight matrixes and sent to other agents during the sender procedure. On the one side, h_f is processed utilizing a class-specific weight matrix $w_F \in \mathbb{R}^{d \times d}$, where d and d' represents the dimension of the input feature and the output feature, respectively. On the other side, h_f is processed by heterogeneous edgetypes utilizing the edgetype-specific weight matrix $w_{\text{Edgetype}} \in \mathbb{R}^{d'' \times d}$,where the d'' represents the output feature dimension of the agent that agent f sent messages to.

For example, $F \rightarrow K$ represents the edgetype from the agent of class F to the agent of class K. $h_f^K = w_{F \rightarrow K} h_f$ denotes the feature processed by edgetype-specific weight matrix $w_{F \rightarrow K}$ that from agent f of class F to the any agent of class K. Then the obtained feature h_f^K are sent to any agent of class K . During the message receive phase, for each edge

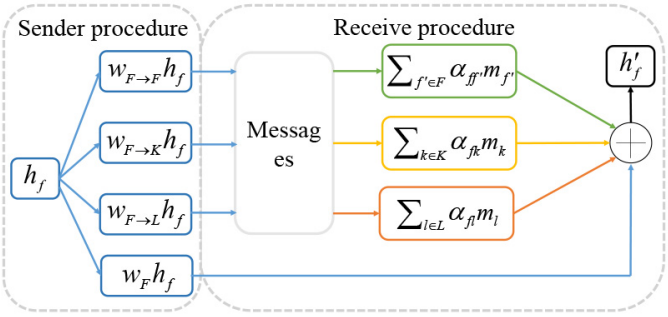


Fig. 2: Framework of the communication learning module.

type that the agent f of class F is connected to other agents, we utilize the heterogeneous GCN with attention mechanism to obtain the per-edge-type aggregation feature. It is obtained by weighted calculation of messages received by neighbor agents along the same edge type with α^{Edgetype} , which denotes normalized attention coefficients.

Then, the aggregation embeddings are integrated with transformed embedding $w_F h_f$ to compute the output message embedding. Therefore, for agent f of class F, the message aggregation equation can be represented as follows, where $N_f(F)$, $N_f(K)$ and $N_f(L)$ represents the neighbor agents that belongs to class F, K and L , respectively.

$$m_f = \sigma\left(w_F h_f + m_f^F + m_f^K + m_f^L\right) \tag{2}$$

where $m_f^F = \sum_{f' \in N_f(F)} \alpha_{ff'} h_{f'}^F$, $m_f^K = \sum_{k \in N_f(K)} \alpha_{fk} h_k^F$, and $m_f^L = \sum_{l \in N_f(L)} \alpha_{fl} h_l^F$. To consider heterogeneous communication, we utilize Eq.(3)-Eq.(5) to compute the attention coefficient $\alpha_{ff'}$, α_{fk}, α_{fl} in the message representations.

$$\alpha_{ff'} = \text{softmax}_{f'}\left(\sigma'\left(\bar{a}^T\left[\omega_F h_f \| \omega_{F \rightarrow F} h_{f'}\right]\right)\right) \tag{3}$$

$$\alpha_{fk} = \text{softmax}_{k}\left(\sigma'\left(\bar{a}^T\left[\omega_F h_f \| \omega_{K \rightarrow F} h_{k}\right]\right)\right) \tag{4}$$

$$\alpha_{fl} = \text{softmax}_{l}\left(\sigma'\left(\bar{a}^T\left[\omega_F h_f \| \omega_{L \rightarrow F} h_{l}\right]\right)\right) \tag{5}$$

The similar calculation procedure can be carried out for agent k of class K and agent l of class L , the corresponding equations are represented as follows:

$$m_k = \sigma\left(w_K h_k + m_k^K + m_k^F + m_k^L\right) \tag{6}$$

$$m_l = \sigma\left(w_L h_l + m_l^L + m_l^F + m_l^K\right) \tag{7}$$

Besides, the corresponding message representation are computed in a similar manner as abovementioned. At each time step, obtained messages embeddings are passed to other neighbor agents. In this way, one heterogeneous GCN layer can correspond to one round of message passing among neighbor agents and feature updates within each agent. We can extract the high-level message representation of each agent by stacking multiple heterogeneous attention network layers, which correspond to multiple rounds of communication. To stabilize the communication learning process, we extend the multi-head variant of the attentional mechanism to heterogeneous settings and utilizes M heads to obtain features in parallel.

D. Mutual information optimization

In this section, we introduce mutual information optimization to enable more efficient communication learning among agents. For agent i, it obtains message embeddings $\left(m_i^1, m_i^c, \ldots, m_i^n\right)$ from neighbor agents of class $(1, c, \ldots, n)$ and fuses them to generate the final message m_i. We define the agent's immediate neighbor agents as other agents within the field of view of this agent. We utilize the random walk with restart (RWR) [17] to sample a fixed number of samples from the defined neighbors of agent i.

Specifically, the neighbors sampling process are as follows: (1) random walk starting from the agent i , select agents using probability p , and put selected agents to set Z_i. Its total number of agents is fixed, and the number of different types of neighbors are limited to ensure that all types of agents in the initial immediate neighbors are included in the new set Z_i. (2) The agents in set Z_i are then grouped via types. For agents that belongs to class c, we choose top k_c agents from set Z_i according to frequency and collect them as the new set $N_i(c)$ of c-class neighbors of agent i.

For agent i, at each timestep, it fuses messages of all n class $\left(m_i^1, m_i^c, \ldots, m_i^n\right)$ to obtain the final message embeding m_i. Nevertheless, some messages of some class may be not useful at a certain time-step. To tackle this issue, we utilize mutual information to implicitly learn which class messages of agent is more valuable at certain time-step, so that the agent can learn the most expressive information from different types of information, so as to better coordinate actions. The mutual information can be calculated by learning a discriminator $\mathcal{D}$ inspired by the idea of [17]. The discriminator $\mathcal{D}$ is aim at telling a positive message-value pairs sample (m_i^c, Q_i) from a negative sample $(\tilde{m}_i^c, Q_i)$, therefore the corresponding loss function is represented as follows:

$$L_{MI} = \sum_{i \in I^+} \log \mathcal{D}_i\left(m_i, Q_i\right) + \sum_{i \in I^-}\left[1 - \log \mathcal{D}_i\left(\tilde{m}_i, Q_i\right)\right] \quad (8)$$

For agent i , we aim to maximize the MI between the messages m_i^c of c class and corresponding individual action value Q_i. The discriminator $\mathcal{D}_i$ is designed to score message-value pairs (m_i^c, Q_i), we utilize a bilinear layer to be the scoring function which is represented as follows:

$$\mathcal{D}_i\left(m_i^c, Q_i\right) = \sigma\left[\left(m_i^c\right)^T M_i^c Q_i\right] \quad (9)$$

where M_i^c represents a learnable scoring vector, σ utilizes the logistic sigmoid activation function. Therefore, given messages embeddings of c class and the discriminators, we can maximize the MI utilizing the message embedding-value loss function for N agents as follows:

$$L_{MI} = \sum_{i=1}^{I} \sum_{c=1}^{n} L_i^c \quad (10)$$

$$L_i^c = \sum_{\langle c,j\rangle \in N_i^+} \log \mathcal{D}_i\left(m_i^c, Q_i\right) + \sum_{\langle c,j\rangle \in N_i^-} \log\left[1 - \mathcal{D}_i\left(\tilde{m}_i^c, Q_i\right)\right] \quad (11)$$

where L_i^c represents the message-value loss of c class, the set N_i^+ represents a sub-set of set Z_i , in which the agents are sampled from the Z_i utilizing the RWR. Specifically, a sampling sub-set U_i is built by selecting agents from set Z_i . For agent j of the sub-set U_i, if the conditions $\operatorname{dist}(i, j) \leq \delta$ are met, the agent j and its corresponding class are together added into the sub-set N_i^+ until the number of sub-set N_i^+ reaches the batch-size. Where $\operatorname{dist}(\cdot)$ represents the distance of two agents, and δ denotes an adjustable parameter that can be set according to different scenarios in the experiment. N_i^- denotes the complement set of the N_i^+ . We utilize the communication learning mechanism designed above to generate the negative message embedding $\tilde{m}_i^c$ based on the set N_i^- . The designed mutual-information loss function in Eq. (11) can be utilized to maximize the MI.

Except for the proposed MI constraints on the message representations, all the parameters in other modules are generally updated by minimizing the global TD loss. In this paper, CLAR utilizes the mixing network of [4]. Therefore, TD loss function utilized in CLAR is presented as follows:

$$L_{ID} = \left[r + \gamma \max_{a'} Q_{tot}\left(\tau', a'; \theta^-\right) - Q_{tot}(\tau, a; \theta)\right]^2 \quad (12)$$

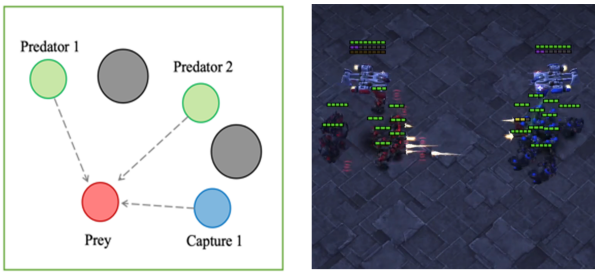
where θ^- represents the parameters of target network, θ denotes all parameters in CLAR. Then, the overall optimization of CLAR is presented as follows:

$$L = L_{TD} + \lambda L_{MI} \quad (13)$$

where λ represents the adjustable hyper-parameter to achieve a trade-off between the TD loss L_{TD} and the sum of MI loss of all agents L_{MI}. We set $\lambda = 0.1$ for it performs best compared to the other values of λ in the experiments.

IV. Experiments

In this section, we select Predator-Capture-Prey (PCP) [16] and StarCraft II Multi-Agent Challenge (SMAC) [18] as our benchmarks. We conduct various experiments on these benchmarks with GPU Nvidia RTX 2080 to answer: **Q1**: Whether CLAR can improve performance in diverse heterogeneous scenarios? **Q2**: Whether CLAR can be applied to large-scale multi-agent scenarios? **Q3**: Does the superiority of CLAR come from communication learning and MI optimization? **Q4**: How are the learned message embeddings distributed in the representation space and how do they affect team-work and action coordination? More details about experiment setting and algorithm are published to facilitate future research[1].



(a) PCP (b) SMAC

Fig. 3: Illustrations of PCP and SMAC

[1] https://github.com/Ayeliauk/Clar

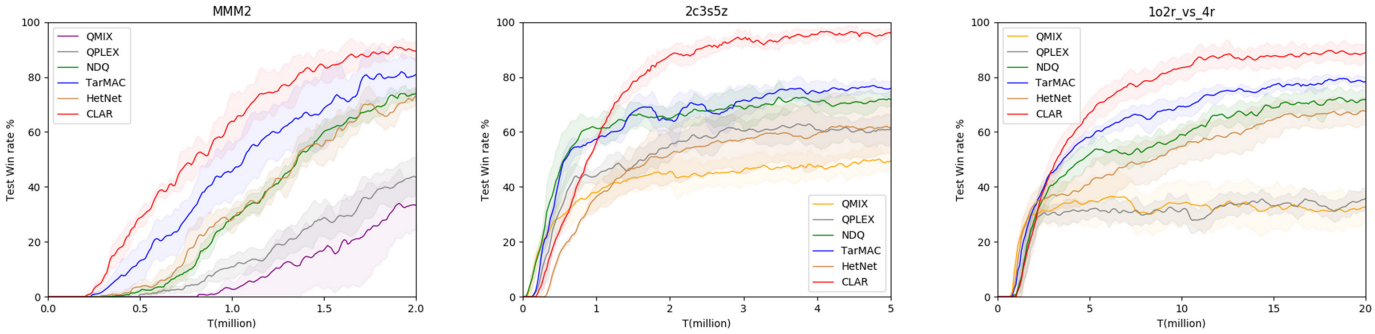


Fig. 4: Learning curves of the proposed method and baselines on heterogeneous scenarios of SMAC.

A. Environments and baselines

As shown in Fig. 3, we first select the heterogeneous environment PCP to conduct experiments, which contains three types of agents, predators, captures, and preys. Predators and captures are cooperative, while preys is adversary. Predators can observe environment while the captures cannot obtain any observation from the environment. Therefore, captures necessitate communication and coordination with predators. We build the communication learning module utilizing heterogeneous graph attention networks with $M = 4$ attention heads. Because the PCP scenario is relatively simple, we do not use MI optimization in the proposed method.

To further demonstrate the effectiveness and superiority of CLAR, we evaluate it on the more complicated benchmark SMAC and select the challenging heterogeneous scenarios of SMAC as shown in Fig. 3. Our experiments are conducted based on the PyMARL framework [18] and utilize its default structure and hyper-parameter settings of the value decomposition module. The hyper-parameters of the proposed communication learning module are set as follows: p is set to 0.6, δ is set to 5. Z_i, N_i^- and N_i^+ can be adjusted according to different scenarios. The rest part is set as same as in the PCP environment.

We select 2 value decomposition methods Q Mixing network (QMIX) [4], Q duplex network (QPLEX) [5] and 3 communication learning methods Nearly Decomposable Q network (NDQ) [3], Targeted Multi-Agent Communication (TarMAC) [9], and Heterogeneous Policy Network (HetNet) [16] as baselines, in which TarMAC and HetNet utilize GCN in the communication learning module.

B. Performance

Effectiveness (Q1). Table I shows the average reward of the baselines and CLAR of 3 different random-seed initialization on PCP, in which the bold numbers represents the highest performance results. Fig. 4 shows the average win rate of the baselines and CLAR of 5 different random-seed initialization on SMAC, while the shadow in represent a 95% confidence interval. As shown in Table I and Fig. 4, the proposed method outperforms other MARL baselines on diverse heterogeneous scenarios, which may be due to the effective representation and communication learning of heterogeneous agent features.

Scalability (Q2). To further verify that CLAR can be extended to large-scale heterogeneous scenarios, we compared the performance of CALR and baselines with different numbers of agents. The number of agents varies from 5 to 40, with the predators, captures, and preys ratio being 3:1:1. As shown in Table I, CLAR always performs optimally as the number of agents increases. The results demonstrate that the proposed method CLAR can be extended to large-scale scenarios.

TABLE I: PERFORMANCE OF DIFFERENT METHODS ON PCP.

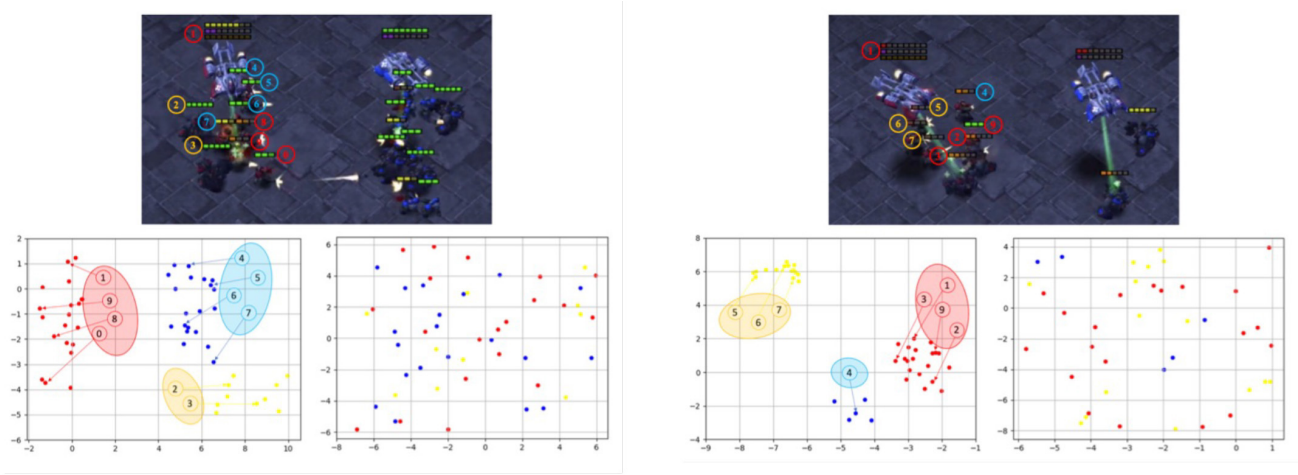
Methods	n=5	n=10	n=20	n=40
QMIX	-0.42±0.07	0.38±0.06	-0.33±0.06	-0.30±0.05
QPLEX	-0.39±0.05	0.35±0.04	-0.29±0.03	-0.26±0.03
NDQ	-0.34±0.05	0.28±0.04	-0.17±0.04	-0.12±0.03
TarMAC	-0.30±0.04	0.23±0.04	-0.05±0.02	+0.06±0.02
HetNet	-0.31±0.05	0.25±0.04	-0.19±0.02	-0.13±0.03
CLAR	**-0.26±0.03**	**-0.18±0.02**	**+0.07±0.01**	**+0.15±0.01**

Contributions (Q3). To evaluate the contributions of each component of CLAR, we design an ablation experiments, in which three variants of CLAR are selected as the baselines. As shown in Table II, CLAR-H is the CLAR without heterogeneous GCN. It directly uses the normal GCN for communication learning. CLAR-V is CLAR without the value decomposition component. CLAR-M is the CLAR without MI optimization. The performance of all three variants decreases compared to the CLAR, illustrating the effectiveness of each component. The heterogeneous GCN can achieve efficient communication for heterogeneous agents, the MI optimization can further enhance the quality of communication, and the value decomposition can promote the policy learning.

TABLE II: ABLATION EXPERIMENTS ON SMAC.

Methods	MMM2	1c3s5z	2c3s5z	1o2r vs 4r
CLAR-H	77.32±6.09	89.32±4.16	82.13±6.43	78.49±7.14
CLAR-M	83.16±5.52	94.17±3.59	87.16±4.60	83.05±5.17
CLAR-V	85.83±4.75	95.25±2.43	90.61±3.24	87.73±3.67
CLAR	**87.55±4.52**	**97.58±1.94**	**93.03±2.17**	**90.42±3.29**

Visualizations (Q4). As shown in Fig. 5, the message representations produced by CLAR-M are almost randomly distributed in the representation space. On the contrary, with the proposed MI optimization, the message representations produced by CLAR automatically cluster several clusters in the space. According to the locations of message representations, we divide the agents into groups. We color the message embeddings by the group to which each agent belongs. That is, for each group in the space, the message embeddings are colored



(a) 10-th timestep

(b) 20-th timestep

Fig. 5: Visualization of the video frames and t-SNE projection of the message embedding representation space, e.g., in (a), top subgraph (Visualization), left subgraph (t-SNE projection of CLAR), right subgraph (t-SNE projection of CLAR-M).

uniformly. In the video frame of the same time-step, we can see the correspondence between the agent groups formed in the game and those formed in the message representation space. Agents in the same group tend to receive similar message embedding and accomplish more cooperation.

V. Conclusions

This paper provides a novel GNN-based communication learning mechanism and MARL framework for heterogeneous scenarios. To our knowledge, our work is the first attempt to solve the heterogeneous multi-agent tasks by integrating heterogeneous GNN, MI optimization and value factorization, which simultaneously solves the issues of scalability, effective communication and action coordination in heterogeneous scenarios. We believe that the proposed method can provide a new way for other researchers to solve the MARL problem.

In the future, further implementation of sub-task partitioning of heterogeneous agents is a promising direction that can be explored to build efficient and scalable heterogeneous multi-agent systems. We intend to apply the proposed method in real-world heterogeneous multi agent scenarios, such as traffic signal control.

References

[1] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi. "Deep reinforcement learning for multiagent systems: a review of challenges, solutions, and applications," in *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3826-3839, 2020.

[2] W. Du, and S. Ding. "A survey on multi-agent deep reinforcement learning: from the perspective of challenges and applications," in *Artificial Intelligence Review*, vol. 54, no. 5, pp.3215-3238, 2021.

[3] T. Wang, J. Wang, C. Zheng, and C. Zhang. "Learning nearly decomposable value functions via communication," in *Proceedings of the 8th International Conference on Learning Representations*, pp.1-15, 2020.

[4] T. Rashid, M. Samvelyan, C. Schroeder, J. Foerster, and S. Whiteson. "Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *Proceedings of the 35th International Conference on Machine Learning*, Stockholm Sweden, pp. 4295-4304, 2018.

[5] J. Wang, Z. Ren, T. Liu, Y. Yu, and C. Zhang. "QPLEX: Duplex dueling multi-agent Q-learning," in *Proceedings of the 7th International Conference on Learning Representations*, pp. 1-27, 2019.

[6] S. Sukhbaatar, R. Fergus. "Learning multiagent communication with backpropagation," in *Proceedings of the 30th Advances in neural information processing systems*, Barcelona, Spain, pp. 2244-2252, 2016.

[7] J. Jiang, Z. Lu. "Learning attentional communication for multi-agent cooperation," in *Proceedings of the 32nd Advances in neural information processing systems*, Montreal, Canada, pp. 7254–7264, 2018.

[8] W. Fang, and L. Lu "Deep Graph Attention Neural Network for Click-Through Rate Prediction," in *Proceedings of The 33th International Conference on Software Engineering Knowledge Engineering*, pp. 483-488, 2020.

[9] J. Jiang, C. Dun, and Z. Lu. "Graph convolutional reinforcement learning for multi-agent cooperation," in *Proceedings of the 6th International Conference on Learning Representations*, pp. 1-10, 2018.

[10] A. Das, T. Gervet, J. Romoff, D. Batra, D. Parikh, M. Rabbat, and J. Pineau. "Tarmac: Targeted multi-agent communication," in *Proceedings of the 36st International Conference on Machine Learning*, Long Beach, California, USA, pp. 1538-1546, 2019.

[11] J. Sheng and X. Wang and B. Jin and J. Yan and W. Li and T.H. Chang and , and H. Zha. "Learning structured communication for multi-agent reinforcement learning," in *arXiv preprint arXiv:2002.04235*, 2020.

[12] H. Ryu, H. Shin, and J. Park. "Multi-agent actor-critic with hierarchical graph attention network," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, New York, USA, pp. 7236-7243, 2020.

[13] Y. Liu, W. Wang, Y. Hu, J. Hao, X. Chen, and Y. Gao. "Multi-agent game abstraction via graph attention neural network," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, New York, USA, pp. 7211-7218, 2020.

[14] Y. Niu and R. Paleja, and M. Gombolay. "Multi-agent graph-attention communication and teaming," in *International Conference on Autonomous Agents and Multi Agent Systems*, pp. 964-973, 2021.

[15] D. D. R. Meneghetti, and R. A. D. C.J. Bianchi "Towards heterogeneous multi-agent reinforcement learning with graph neural networks," in *arXiv preprint arXiv: 2009.13161*, 2020.

[16] E. Seraj, Z. Wang, R. Paleja, M. Sklar, A. Patel, and M. Gombolay "Heterogeneous graph attention networks for learning diverse communication," in *arXiv preprint arXiv: 2108.09568*, 2021.

[17] Tong, H., Faloutsos, C., Pan, J. Y.. "Fast random walk with restart and its applications," in *Proceedings of the 6th Conference and Workshop on Neural Information Processing Systems*, pp. 613-622, 2006.

[18] M. Samvelyan, T. Rashid, C. Schroeder de Witt, G. Farquhar, N. Nardelli, T. G. Rudner, and S. Whiteson. "The StarCraft Multi-Agent Challenge," in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, Montreal, Canada, pp. 2186-2188, 2019.