Multi-Agent Communication with Information Preserving Graph Contrastive Learning

Wei Du[1,2], **Shifei Ding**[3], **Wei Guo**[1,2], **Yuqing Sun**[1], **Guoxian Yu**[1,2,*] and **Lizhen Cui**[1,2,*]
[1]School of Software, Shandong University, China
[2]Joint SDU-NTU Centre for Artificial Intelligence Research (C-FAIR), Shandong University, China
[3]School of Computer Science and Technology, China University of Mining and Technology, China
duwei@sdu.edu.cn, dingsf@cumt.edu.cn, {guowei, sun_yuqing, gxyu, clz}@sdu.edu.cn

Abstract

Recent research in cooperative Multi-Agent Reinforcement Learning (MARL) has shown significant interest in utilizing Graph Neural Networks (GNNs) for communication learning due to their strong ability to process feature and topological information of agents into message representations for downstream action selection and coordination. However, GNNs generally assume network homogeneity that nodes of the same class tend to be interconnected. In real-world multi-agent systems, such assumptions are often unrealistic, as agents within the same class can be distant from each other. Furthermore, GNN-based MARL methods overlook the crucial role of feature similarity of agents in action coordination, which also restricts their performance. To overcome these limitations, we propose a Multi-Agent communication mechanism with Information preserving graph contrastive Learning (MAIL), which enhances message representation by preserving the comprehensive features of adjacent agents while integrating topological information. Specifically, MAIL considers three distinct graph views: original view, agent feature view, and global topological view. MAIL performs contrastive learning across three views to extract comprehensive information. MAIL effectively learns robust and expressive message representations for downstream tasks. Extensive experiments across various environments demonstrate that MAIL outperforms existing GNN-based MARL methods.

1 Introduction

Multi-agent reinforcement learning (MARL) has attracted considerable interest and achieved impressive results in various complex real-world applications, including autonomous driving [Xu *et al.*, 2024], traffic signal control [Zhang *et al.*, 2024], and auction market [Qiu *et al.*, 2021]. Recently, the centralized training with decentralized execution (CTDE) framework has gained widespread adoption as an effective solution to address the challenges of non-stationarity and scalability in MARL. This framework involves developing decentralized policies in a centralized manner, enabling the sharing of experiences and parameters throughout the training process. Building on the CTDE framework, several value decomposition methods have been proposed [Liu *et al.*, 2023; Rashid *et al.*, 2020; Du *et al.*, 2024]. These methods apply various constraints or restrictions to factorize the global value function to a combination of individual value functions. While the CTDE framework presents multiple benefits, the partial observability and stochasticity encountered during the decentralized execution period can heighten agents' uncertainty about the states and actions of other agents, potentially leading to miscoordination in their actions.

Recently, several MARL methods have utilized communication learning protocols to improve coordination among agents, enabling them to share information, such as feature embeddings, during the execution phase. By facilitating inter-agent communication, these methods significantly enhance coordination across a variety of tasks [Gilmer *et al.*, 2017]. Graph Neural networks (GNNs) effectively integrate both the feature and topological information in graph-structured data, facilitating robust feature representation learning for subsequent tasks. Due to the effective representation learning capability of GNN, communication learning through GNN has garnered significant research interest in MARL. In this case, agents can typically be represented as nodes, while communication channels between them are denoted as edges in a graph. Various MARL methods utilize this GNN-based communication framework, such as DGN [Jiang *et al.*, 2020], LSC [Sheng *et al.*, 2022], and MAGIC [Niu *et al.*, 2021].

While GNN-based MARL methods [Liu *et al.*, 2020; Das *et al.*, 2019; Sukhbaatar and Fergus, 2016]have achieved success, they overlook a key limitation of GNN it relies on: **homophily assumption**. This assumption suggests that nodes within the same class are more likely to be connected. However, in real-world multi-agent systems, this assumption often fails, as agents of the same type can be spatially distant from one another. Under this situation, GNN-based MARL methods struggle with poor performance because the propagation mechanism within graph neighborhoods of GCN becomes problematic, leading to the mixing of irrelevant information from agents of different classes. Moreover, GNN-based MARL methods tend to overlook the critical role that feature similarity between agents plays in action coordina-

*Corresponding author

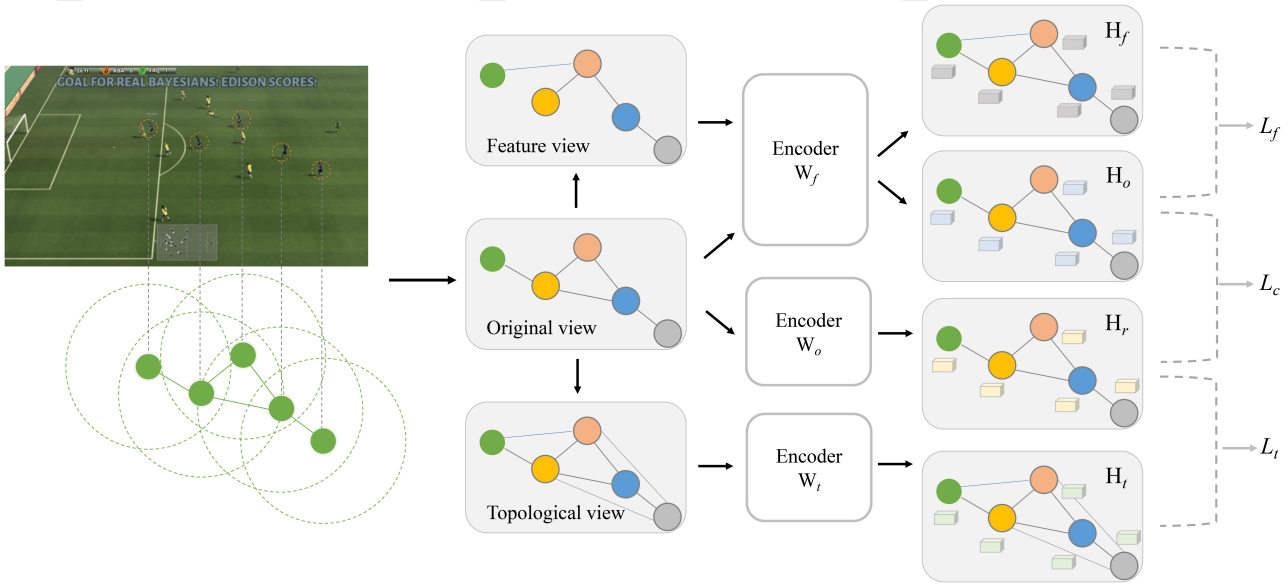


Figure 1: The overview of MAIL. We initially create three graph views from the built graph: original view, agent feature view, and global topological view. These three views are then processed by the encoders to produce their respective node representations. Rather than directly contrasting the graph views, we combine the node representations from the original view and agent feature view to form new node representations H_f for the subsequent contrastive learning process. By minimizing the feature contrastive loss L_f, cross-module loss L_c, and topological contrastive loss L_t, MAIL effectively learns expressive message representations.

tion, which limits their overall effectiveness.

In cooperative MARL, expressive message representations that incorporate comprehensive information are essential for efficient action coordination. Therefore, it is crucial to extract and preserve valuable information from neighboring agents to learn effective message representations. To address these issues, we present a Multi-Agent communication protocol with Information preserving graph contrastive Learning (MAIL), which comprehensively preserves agent feature information while exploiting topological information.

Consider a football game as illustrated in Figure 1, and assume that each agent chooses adjacent agents within its range to establish the original graph. In MAIL, we utilize the original graph as one of the contrastive views and consider the combination of the feature graph and the original graph as a second contrastive view. By maximizing the agreement between the two contrastive views, MAIL effectively preserves the agents' feature information. To maintain the topological information, we introduce an additional contrastive learning module specifically designed to preserve global topological details. We directly utilize the higher-order view of the graph as the global topological view and contrast it with the original graph view for preserving global topological information. MAIL acquires high-quality message representations by extracting crucial information at both the topological and feature levels. We select several baselines and conduct experiments across various environments, with the results confirming that MAIL outperforms the baselines. The key contributions of our work can be outlined as follows:

- To the best of our knowledge, our research is the first to utilize graph contrastive learning in the context of MARL, enabling effective communication learning.
- We present two contrastive learning modules that can individually extract the feature and topological information of the input graph while training them in a coordinated manner.
- The proposed method encourages the agent to learn significant information from both feature and topological aspects, enabling it to generate high-quality message representations for downstream action coordination.

2 Related Work

2.1 Graph Contrastive Learning

Graph Neural Networks (GNNs) have become a highly effective method for learning robust node representations. The term GNN refers to a broad spectrum of methodologies. In this study, we specifically define it as the Message Passing Graph Neural Network [Gilmer *et al.*, 2017], recognized as the most general architecture within the GNN framework. Prominent examples of this architecture include GCN [Duvenaud *et al.*, 2015], GAT [Veličković *et al.*, 2017], and GraphSAGE [Hamilton *et al.*, 2017]. While GNNs have shown impressive performance in many scenarios, their effectiveness may be impaired when the homophily assumption fails [Chen and Kou., 2023].

Graph contrastive learning (GCL) has emerged as a promising research direction, typically involving the design of different views and aiming to maximize the agreement between the representations of these views. DGI [Velickovic *et al.*, 2019] concentrates on maximizing the mutual information between global graph-level representations and local

node-level representations. Building upon DGI, GMI [Peng *et al.*, 2020] utilizes two discriminators to directly compute mutual information between the input and the embeddings of both edges and nodes. In contrast, MVGRL [Hassani and Khasahmadi, 2020] focuses on learning both node-level and graph-level embeddings by contrasting node representations and facilitating node diffusion using augmented graph summary representations.

Most previous GCL methods rely on GNNs as the backbone encoder, achieving impressive performance in homophilic scenarios. However, limited attention has been given to developing GCL methods for situations where the homophily assumption fails. Moreover, current contrastive learning methods primarily generate views by transforming existing graph data. However, these data transformation strategies are inadequate for learning comprehensive and robust node embeddings. In our research, we adopt the graph contrastive learning scheme introduced in ASP [Chen and Kou., 2023], which effectively retains both feature and topological information from the input built graph.

2.2 GNN-based MARL

In our work, we utilize the following GNN-based communication learning methods as baseline methods: TarMAC [Das *et al.*, 2019] employs GAT to learn communication through a fully constructed communication graph. MAGIC [Niu *et al.*, 2021] also utilizes GAT to facilitate multiple rounds of communication between agents. CommNet [Sukhbaatar and Fergus, 2016] introduces a communication channel that allows agents to dynamically enter and exit each other's communication range, similar to GNN methods that utilize mean aggregation. IC3Net [Singh *et al.*, 2019] incorporates a gating mechanism to regulate communication learning. DGN [Gilmer *et al.*, 2017] establishes communication protocols on graphs derived from the environment. LSC [Sheng *et al.*, 2022] presents a hierarchical GNN that enhances communication learning by enabling message exchanges within groups and between agents. G2ANet [Liu *et al.*, 2020] integrates both soft and hard attention mechanisms to dynamically adapt communication. DICG [Li *et al.*, 2021] features a module for inferring the topology of the coordination graph, using GNNs for implicit reasoning on joint actions. Although GNN-based MARL has achieved success, the network homogeneity assumption of GNN in MARL is often unrealistic, which limits the performance of GNN-based MARL methods. Furthermore, GNN-based MARL methods often neglect the crucial role of feature similarity between agents in action coordination, which hampers their overall effectiveness.

3 Method

3.1 Preliminary

Let $G = (V, E)$ represents a graph, where $V = \{v_i, ..., v_n\}$ denotes the node set and $E \subseteq V \times V$ denotes the edge set. The feature matrix is represented as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times f}$, where $\mathbf{x}_i \in \mathbb{R}^f$ denotes the feature of v_i. $A \in \{0, 1\}^{n \times n}$ represents the adjacency matrix. In our study, the multi-agent system is represented as a graph $G = (V, E, \mathbf{X})$, where the set of agents is denoted by $V = \{v_i, ..., v_n\}$, the edge set is represented as $E \subseteq V \times V$, and the agent/node features are indicated by $X \in \mathbb{R}^{n \times f}$. Adjacent agents are defined as those agents located within a specified range of each agent, which is used to construct the original graph. Given a graph $G = (V, E, \mathbf{X})$, K distinct transformations $\mathcal{F}_1, \ldots, \mathcal{F}_K$ can be applied to obtain different views $\mathbf{v}_1, \ldots, \mathbf{v}_K$ of the graph:

$$\mathbf{v}_i = \mathcal{F}_i(\mathbf{A}, \mathbf{X}), i = 1, \ldots, K \tag{1}$$

Graph transformation techniques encompass various methods such as node feature masking, edge perturbation, etc. A set of encoders $f_1, \ldots, f_K$ receives the corresponding views as inputs and produces the representations $\mathbf{h}_1, \ldots, \mathbf{h}_K$ of the graph from each view:

$$\mathbf{h}_i = f_i(\mathbf{v}_i), i = 1, \ldots, K \tag{2}$$

A cooperative MARL problem could be formally modeled as Dec-POMDP [Oliehoek, 2012], which can be represented by a tuple $< I, S, O, U, P, R >$. I denotes the agents' set, which is indexed from 1 to n. S denotes the state set of the environment. O represents the observations set, where $o_i \in O$ represents observation of agent i. U denotes the finite space of joint actions. For agent i, at each timestep, agent i takes its action a_i depending on its local observation o_i, and forms a joint action $a = (a_1, \ldots, a_n) \in U$. Based on the Markovian transition function $P : S \times U \to S$, the state changes and the agent obtains reward r_i^t based on the reward function $R : S \times U \to R$. The target of the agent i is to maximize its total discounted reward $R_i = \sum_{t=0}^{T} \gamma^t r_i^t$, in which $\gamma \in [0, 1]$ denotes a discount factor. In Dec-POMDPs, the target of all agents is to learn an optimal joint policy $\pi(\tau, a)$ to maximize the global value $Q_{tot}^{\pi}(\tau, a) = \mathbb{E}_{s,a}[\sum_{t=0}^{\infty} \gamma^t R(s, a)]$, where τ represents the joint observation history.

The framework of the MAIL is illustrated in Figure 2. For agent i, it receives local observation o_i and employs multilayer perceptron (MLP) with the gated recurrent unit (GRU) to produce the agent feature $\mathbf{x}_i$. Next, $\mathbf{x}_i$ is fed into the GCL module to generate the expressive message embedding $\mathbf{h}_i$.

Subsequently, the message representation $\mathbf{h}_i$ and local history τ_i are concatenated to generate the input for the individual action-value function to select an action and compute individual value $Q_i(\tau_i, a_i, \mathbf{h}_i)$. Finally, all individual action-values are fed to the mixing network to calculate the estimation of global value Q_{tot}. In this study, the mixing network of QMIX [Rashid *et al.*, 2020] is adopted, and it could be substituted with any mixing network from current value function factorization approaches.

3.2 Graph Contrastive Learning module

View Generation

As depicted in Figure 1, the original built graph data, without any data augmentation techniques, is referred to as the original view $\mathbf{v}_o$, serving as an anchor for the other views. To capture the diverse feature similarity relationships among agents, we employ the feature matrix $\mathbf{X}$ to create a k-Nearest-Neighbor (kNN) graph G_f, which we designate as the feature view $\mathbf{v}_f$. The kNN graph G_f can be constructed using various distance metrics, including Euclidean distance, Jaccard

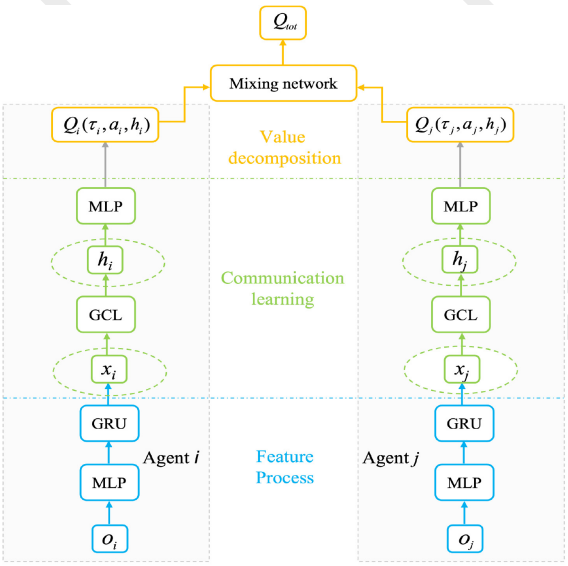


Figure 2: The framework of MAIL.

distance, or cosine distance. In our framework, we utilize cosine distance. Given a node pair (v_i, v_j), their cosine distance is defined as follows:

$$d_{\cos}(v_i, v_j) = 1 - \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{|\mathbf{x}_i|\,|\mathbf{x}_j|} \tag{3}$$

where $|\cdot|$ represents the magnitude of a vector. The adjacency matrix is denoted as $\mathbf{A}_F$, and the corresponding degree matrix of the kNN graph is denoted as $\mathbf{D}_F$.

To extract global topological information, we create a straightforward yet effective graph view. In contrast to edge perturbation strategies that arbitrarily alter the graph topology, edge diffusion methods [Gilmer *et al.*, 2017] effectively preserve the expressive global information contained in graphs. Current edge diffusion approaches primarily utilize the heat kernel [Kondor and Lafferty., 2002] to create a global view. However, this solution requires computing matrix exponentials or inverses, which can be computationally inefficient. While approximations can help reduce computational complexity, they often necessitate tedious and time-consuming hyperparameter tuning.

In contrast to existing approaches, we directly employ a higher-order view of the graph as the global topological view $\mathbf{v}_t$. In this framework, each node gathers information from neighbors that are l hops away. As l increases, more global topological information is incorporated. Both the kNN graph G_f and the original graph G can be utilized to generate these higher-order views. Our findings indicate that simply contrasting the higher-order views with the original graph views can yield comparable performance.

Feature Preserving Contrastive Learning

Current graph contrastive learning methods typically use GNN as the foundational encoder, effectively extracting the topological information inherent in graphs. Despite their effectiveness, these methods often neglect the similarities between node features derived from the feature matrices. In this study, we utilize a more robust mechanism that retains feature knowledge by leveraging both the feature graph view and the original graph view. While our framework permits the use of various GNNs, we have selected SGC [Wu *et al.*, 2019] as the base encoder because of its simplicity and competitive performance. SGC streamlines the topology of GCN by eliminating intermediate nonlinearities, resulting in all learnable parameters being consolidated into a single matrix:

$$\mathbf{H} = \mathbf{S}^P \mathbf{X} \boldsymbol{W} \tag{4}$$

where $\mathbf{S}$ represents the normalized adjacency matrix with added self-loops $\mathbf{S} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$ and $\boldsymbol{W}$ represents a trainable weight matrix. $\mathbf{S}^P$ represents the representations generated by propagating information from agents that are P-hop away.

Rather than directly comparing the original view with the feature view, we utilize the feature view to complement the original view. Specifically, we incorporate the node embeddings from the feature view into the original view, treating the sum of these embeddings as the final contrastive view. Formally, we express this as follows:

$$\begin{aligned} \mathbf{H}^o &= \mathbf{S}^p \mathbf{X} \boldsymbol{W}_f \\ \mathbf{H}^f &= \mathbf{S}^p \mathbf{X} \boldsymbol{W}_f + \mathbf{S}_F \mathbf{X} \boldsymbol{W}_f \end{aligned} \tag{5}$$

where $\mathbf{S}_F = \mathbf{D}_F^{-1/2}\mathbf{A}_F\mathbf{D}_F^{-1/2}$. The weight matrix $\boldsymbol{W}_f$ is shared between $\mathbf{H}^f$ and $\mathbf{H}^o$.

After acquiring the message representations $\mathbf{H}^o$ and $\mathbf{H}^f$, we employ InfoNCE [Gutmann and Hyvärinen, 2020] to estimate the lower bound of the MI between them. For node v_i, the learned node embeddings $\mathbf{h}_i^o$ and $\mathbf{h}_i^f$ serve as positive samples, while the embeddings of all other nodes are considered negative samples. With these definitions of negative and positive samples, the loss function for the feature-preserving contrastive learning module can be formulated as follows:

$$L_f(v_i) = -\log \frac{e^{\mathcal{D}\left(\mathbf{h}_i^o, \mathbf{h}_i^f\right)/\theta}}{\sum_{j=1}^{N} e^{\mathcal{D}\left(\mathbf{h}_i^o, \mathbf{h}_j^f\right)/\theta} + \sum_{v\in\{o,f\}} \sum_{j=1}^{N} \mathbb{I}_{[j\neq i]} e^{\mathcal{D}\left(\mathbf{h}_i^v \cdot \mathbf{h}_j^v\right)/\theta}} \tag{6}$$

where θ represents the temperature parameter, and $\mathcal{D}(\cdot)$ denotes the discriminator that calculates the agreement score between two vectors. In this context, we utilize cosine similarity for $\mathcal{D}(\cdot)$.

Topological Preserving Contrastive Learning

While feature information has been uncovered through feature preserving contrastive learning, global topological information remains unexplored. Therefore, to enhance the expressiveness and robustness of the proposed framework, we introduce a topological preserving contrastive learning module. We continue to use SGC as the base encoder in this module. To capture global topological information, we directly compare the original graph view with the global topological view:

$$\begin{aligned} \mathbf{H}^r &= \mathbf{S}^p \mathbf{X} \boldsymbol{W}_r \\ \mathbf{H}^t &= (\mathbf{S}_G)^l \mathbf{X} \boldsymbol{W}_t \end{aligned} \tag{7}$$

where l is a positive integer significantly larger than p, and $\mathbf{S}_G \in \{\mathbf{S}, \mathbf{S}_F\}$. It is important to note that we employ separate weight matrices for $\mathbf{H}^o$ and $\mathbf{H}^r$ to ensure that they do not interfere with one another.

Given the node embeddings $\mathbf{h}_i^r$ and $\mathbf{h}_i^t$ of node v_i, we compute the contrastive loss for topological preserving learning:

$$L_t(v_i) = -\log \frac{e^{\mathcal{D}\left(\mathbf{h}_i^o, \mathbf{h}_i^t\right)/\theta}}{\sum_{j=1}^{N} e^{\mathcal{D}\left(\mathbf{h}_i^o, \mathbf{h}_j^t\right)/\theta} + \sum_{v\in\{r,t\}} \sum_{j=1}^{N} \mathbb{I}_{[j\neq i]} e^{\mathcal{D}\left(\mathbf{h}_i^v \cdot \mathbf{h}_j^v\right)/\theta}} \tag{8}$$

Model Training

With the two main components of MAIL established, we introduce the cross-module loss, which we employ to align the representations of the original view across various modules. Following this, we will outline the overall objective loss for training of MAIL.

The purpose of the cross-module loss is to align the representations of $\mathbf{H}^o$ and $\mathbf{H}^r$. We have observed that contrasting representations from the same view but across different modules can effectively enhance the quality of learned message representations.

$$L_c(v_i) = -\log \frac{e^{\mathcal{D}(\mathbf{h}_i^o, \mathbf{h}_i^r)/\theta}}{\sum_{j=1}^{N} e^{\mathcal{D}\left(\mathbf{h}_i^o, \mathbf{h}_j^r\right)/\theta} + \sum_{v\in\{o,r\}} \sum_{j=1}^{N} \mathbb{I}_{[j\neq i]} e^{\mathcal{D}\left(\mathbf{h}_i^v \cdot \mathbf{h}_j^v\right)/\theta}} \tag{9}$$

The overall GCL objective loss of MAIL is defined as the sum of the feature preserving loss, the topological preserving loss, and the cross-module loss:

$$\begin{aligned} L_{GCL} &= L_f + \lambda_1 L_t + \lambda_2 L_c = \\ &\frac{1}{N} \sum_{i=1}^{N} \left[L_f(v_i) + \lambda_1 L_t(v_i) + \lambda_2 L_c(v_i)\right] \end{aligned} \tag{10}$$

where λ_1 and λ_2 denote the tuning parameters to weight the importance of L_t and L_c, respectively.

Through the above description, we can obtain high-quality message representations in MAIL. Except for the graph contrastive learning optimization constraints on learning message representations in the communication module, all the parameters across the remaining modules of the framework are updated by minimizing the TD loss L_{TD}. In the end, TD loss is explicitly expressed as follows:

$$L_{TD} = \left[r + \gamma \max_{a'} Q_{tot}\left(\tau', a'; \theta^-\right) - Q_{tot}(\tau, a; \theta)\right] \tag{11}$$

where θ denotes all the parameters in the remaining modules and θ^- denotes the parameters of the target network. Therefore, the overall optimization target of MAIL is expressed as follows,

$$L = L_{TD} + \beta L_{GCL} \tag{12}$$

where β represents a hyper-parameter that can be fine-tuned to balance between the graph contrastive learning optimization loss L_{GCL} and the TD loss L_{TD}. A detailed description of our framework is provided in Algorithm 1.

Algorithm 1 MAIL

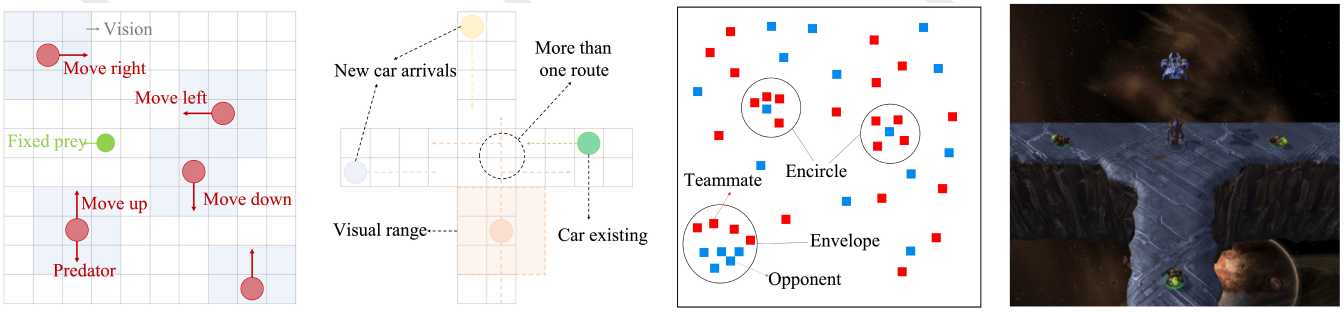
```
Initialize: the parameters of networks, the maximum size of the replay buffer, and the frequency of network updating.
for each timestep t ∈ T do
  for each agent i ∈ N do
    // During the decentralized execution period
    Generate agent feature x_i by GRU and MLP
    Construct graph G = (V, E, X) based on x_i
    Receive node representations H^o, H^f, H^r, and H^t
    Calculate feature loss L_f, topological loss L_t and, cross-module loss L_c with Eq.6, Eq.8, and Eq.9, respectively
    Update parameters according to the overall GCL objective loss L_GCL in Eq.10
    Obtain final message representation h_i^o
    Calculate action-value Q_i based on h_i and τ_i
    a_i^t ← π(Q_i) (ϵ− greed )
    Store τ_i and a_i^t to replay buffer
    // During centralized training period
    Fed Q_i to mixing network and obtain Q_tot
    Minimize loss function according to Eq.12
    Update weights of all networks
  end for
end for
```

4 Experiments

To verify the effectiveness of MAIL, we perform a range of experiments across 4 benchmarks: Predator-Prey [Sukhbaatar and Fergus, 2016], Traffic Junction [Sukhbaatar and Fergus, 2016], Battle [Zheng *et al.*, 2018], StarCraft Multi-Agent Challenge [Vinyals *et al.*, 2019]. Experiments are conducted with a GPU NVIDIA RTX 4090. The hyperparameters that we adjust are as follows: (i) $k \in \{3, 5, 10\}$, for k nearest neighbors, (ii) aggregation hops $l \in \{3, 5, 7\}$, (iii) $\lambda_1 = 0.2$, $\lambda_2 = 0.3$, and $\beta = 0.2$ depending on the experimental results. For each environment, 4 GNN-based MARL baselines (introduced in Related Work) have been chosen for ease of comparison without losing generality. The detailed hyperparameters and some experiments are given in the Appendix.

4.1 Predator-Prey

As shown in Figure 3(a), the Predator-Prey environment [Singh *et al.*, 2019] involves multiple predators with limited vision, aiming to capture prey. Predators can move in four directions: down, up, left, or right. An episode is considered successful if all predators find the prey within the given time limit. We define two difficulty levels: a 10 × 10 grid with 5 predators, and a 20 × 20 grid with 10 predators. A superior approach is characterized by its ability to minimize the mean number of steps necessary to accomplish an episode. As shown in Table 1, MAIL captures the prey faster than the baselines in both scenarios. Figure 4(a) illustrates the success rate of approaches in a 20 × 20 grid with 10 predators, MAIL performs significantly better than baselines. Compared to GNN-based MARL, MAIL avoids the mixing of irrelevant information from agents and, therefore learns high-quality message representations for action coordination.



a) Predator-Prey b) Traffic Junction c) Battle d) SMAC

Figure 3: Illustration of the four selected MARL benchmarks.

Method	10 × 10, 5 agents	20 × 20, 10 agents
CommNet	13.12 ± 0.06	75.24 ± 1.38
IC3Net	13.06 ± 0.04	50.26 ± 2.73
TarMAC	13.26 ± 0.10	36.22 ± 0.95
MAGIC	12.81 ± 0.05	33.12 ± 0.17
MAIL	**10.31 ± 0.03**	**27.42 ± 0.08**

Table 1: The mean number of the steps needed to accomplish an episode in Predator-Prey environment.

Method	Medium	Hard
CommNet	53.62 ± 13.81	51.56 ± 9.37
IC3Net	87.83 ± 3.06	73.26 ± 8.72
MAGIC	94.73 ± 2.46	93.51 ± 2.13
TarMAC	94.04 ± 1.42	85.32 ± 2.15
MAIL	**98.81 ± 0.52**	**98.03 ± 0.84**

Table 2: The mean win rate of MAIL and baselines in Traffic Junction environment.

4.2 Traffic Junction

Traffic Junction environment [Sukhbaatar and Fergus, 2016] consists of vehicles (agents with limited visibility) and intersecting routes. This environment is a valuable evaluation environment for assessing the efficiency of communication. In this environment, the main goal is to ensure effective communication between vehicles to prevent collisions. In this scenario, vehicles approach junctions from diverse entry points using a probability represented as p. There is a maximum limit, denoted as N_c, imposed on the number of vehicles permitted in the environment. At each time step, the vehicles can take one of two actions: "brake" or "gas". We evaluate the performance of MAIL and baseline methods in Traffic Junction with two difficulty levels.

As depicted in Figure 3(b), in the medium difficulty level, the traffic junction environment comprises 2 two-way roads sorted within a 14 × 14 grid. In this scenario, the supreme agents' number allowed is 10 (N_c= 10, p = 0.2). On the other hand, the scenario with a hard difficulty level entails 4 two-way roads within an 18 × 18 grid. In this scenario, the supreme agents' number is 20 (N_c= 20, p = 0.05). The objective in both scenarios with two difficulty levels is to maximize the mean success rate, which is specified as having no collisions occur in an episode.

Table 2 shows the mean success rate for each method upon reaching convergence. As shown in Table 2, in both scenarios, MAIL outperforms other baselines. Figure 4(b) illustrates the learning curves of methods in the scenario with a medium difficulty level. As shown in Figure 4(b), the mean number of steps needed to accomplish an episode of MAIL is significantly fewer than other baseline methods, which demonstrates the effectiveness of the graph contrastive learning module of the proposed MAIL.

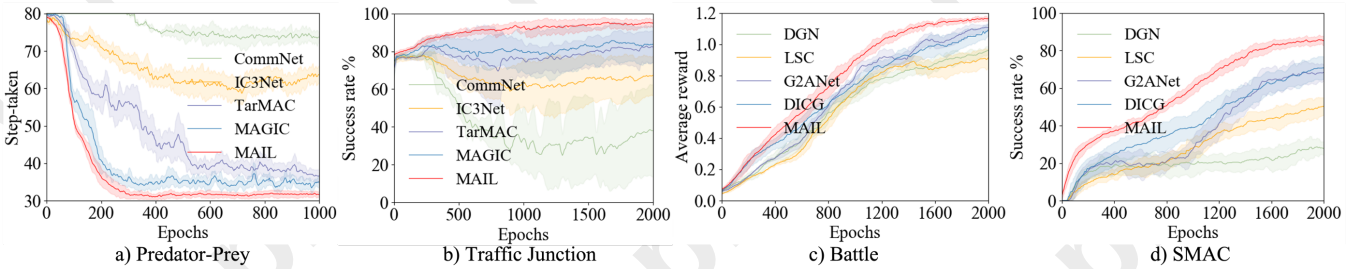
4.3 Battle

We chose the Battle environment [Zheng *et al.*, 2018] to further evaluate the effectiveness of MAIL. Battle scenario, which includes ally agents and enemy agents. Ally agent can select one action of two actions: attack or move. The target for the ally agents is to kill enemy agents. In Battle, MAIL and other baselines are trained with the identical configuration. The ally agent can obtain a positive reward of +5 if it successfully attacks an enemy. Conversely, a negative reward -2 is applied if an ally agent is killed by the enemy agent.

Figure 4(c) depicts the mean reward of different methods in the Battle environment. It can be observed that MAIL significantly outperforms baseline methods. MAIL trained agents can acquire various tactical skills, such as encircling and enveloping. Against a single enemy, MAIL trained agents can effectively learn to coordinate and surround it to kill it. When facing a group of enemies, the agents can adeptly learn to target and attack one flank. Other baselines trained agents initially adopt suboptimal strategies, such as clustering in the corner to evade attacks. Table 3 shows the performance of MAIL compared to baseline methods in the Battle environment. MAIL consistently surpasses baseline methods in terms of mean reward, kills, and kill-death (K/D) ratio.

Method	Kills	K/D ratio	Mean reward
DGN	216 ± 6	2.32 ± 0.16	0.92 ± 0.17
LSC	184 ± 7	1.73 ± 0.25	0.90 ± 0.17
G2ANet	247 ± 5	2.60 ± 0.21	1.12 ± 0.03
DICG	190 ± 6	2.01 ± 0.13	1.02 ± 0.07
MAIL	**261 ± 5**	**2.82 ± 0.26**	**1.21 ± 0.03**

Table 3: Performance of MAIL and baseline methods in Battle environment.



a) Predator-Prey b) Traffic Junction c) Battle d) SMAC

Figure 4: Learning curves of MAIL and baseline methods in four benchmarks.

4.4 StarCraft Multi-Agent Challenge

StarCraft Multi-Agent Challenge (SMAC) [Vinyals *et al.*, 2019] is an established environment developed within the prevalent game StarCraft II, as shown in Figure 3(d). In SMAC environment, MARL methods are employed to train ally agents, while enemy agents are managed by the built-in AI. The features collected by the agent encompass the following attributes of both allied and enemy units within its field of vision: unit type, location, distance, health, and shield. To heighten the coordination challenge for ally agents, the default experimental settings have been fine-tuned, resulting in a reduction of the visual field for agents from 9 to 2.

Figure 4(d) shows the mean win rates of different methods in scenario 1o10b vs 1r. We can observe that MAIL significantly outperforms other baselines. Table 4 shows the performance of MAIL and baselines in the different scenarios: MMM2, 1c3s5z, and 27m vs 30m. As shown in Table 4, MAIL significantly outperforms baselines in these scenarios.

Method	MMM2	1c3s5z	27m vs 30m
DGN	78.71 ± 6.92	85.23 ± 4.42	61.45 ± 9.10
LSC	76.24 ± 7.28	89.42 ± 2.91	73.47 ± 6.03
G2ANet	80.24 ± 4.37	91.25 ± 1.73	75.02 ± 5.84
DICG	83.45 ± 5.42	93.46 ± 1.26	71.53 ± 7.72
MAIL	**94.21 ± 3.75**	**98.91 ± 0.63**	**90.16 ± 3.02**

Table 4: The mean win rate of MAIL and baselines in several scenarios in SMAC environment.

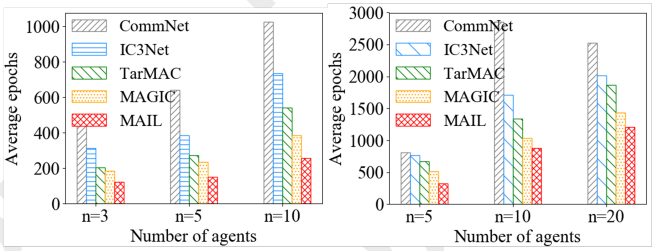
4.5 Ablation analysis

We further evaluate the contribution of each component in MAIL. Specifically, we design three ablations masking different components: (i) w/o FP is MAIL without the feature preserving contrastive learning component. (ii) w/o TP is MAIL without the topological preserving contrastive learning component. (iii) w/o CR is MAIL without cross-module loss. The results are summarized in Table 5. We can observe declines in performance when any of the components are removed, highlighting the effectiveness of each one. In particular, when the feature preserving contrastive learning module is removed, the performance in all three scenarios drops significantly, indicating that this module is essential for boosting performance. Meanwhile, the topological preserving contrastive learning module and the cross-module loss contribute to further improvements.

Ablation	MMM2	MMM3	1c3s5z
MAIL	94.21 ± 3.75	75.17 ± 5.02	98.91 ± 0.73
-w/o FP	74.18 ± 6.53	57.06 ± 7.26	84.31 ± 3.17
-w/o TP	79.10 ± 6.83	65.93 ± 7.30	91.04 ± 2.76
-w/o CR	89.64 ± 4.26	72.41 ± 5.13	95.63 ± 1.21

Table 5: Ablation study on MAIL.

Figure 5 illustrates the average number of epochs needed for each method to reach convergence. Compared to GNN-based MARL, MAIL prevents mixing irrelevant information from agents of different classes, enabling more efficient communication learning. MAIL consistently outperforms baselines as the number of agents increases, highlighting its scalability for tackling large-scale communication learning tasks.



a) Predator-Prey b) Traffic Junction

Figure 5: Performance of methods as the number of agents increases.

5 Conclusion

We have introduced the graph contrastive learning optimization concept for communication learning methods of MARL. MAIL efficiently preserves both feature and topological information and learns expressive message representations, therefore enhancing action coordination between agents. Experimental results on various environments validate that the proposed MAIL outperforms existing GNN-based communication approaches. The effectiveness of MAIL offers significant potential for advancing efficient multi-agent communication learning. Our research promotes further research into multi-agent communication methods and broader, robust forms of communication learning. In future research, we plan to explore more GCL mechanisms and apply them to real-world multi-agent communication learning scenarios.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (No. 2023YFF0725500), the National Natural Science Foundation of China under Grant (No.92367202, 62276265, and 62376138), and the Innovative Development Joint Fund Key Projects of Shandong NSF (ZR2022LZH007).

References

[Chen and Kou., 2023] Jialu Chen and Gang Kou. Attribute and structure preserving graph contrastive learning. In *AAAI*, pages 7024–7032, 2023.

[Das *et al.*, 2019] Abhishek Das, Théophile Gervet, and Joshua. Romoff. Tarmac: Targeted multi-agent communication. In *ICML*, pages 1538–1546, 2019.

[Du *et al.*, 2024] Wei Du, Shifei Ding, Lili Guo, and et al. Expressive Multi-Agent Communication via Identity-Aware Learning. In *AAAI*, pages 17354–17361, 2024.

[Duvenaud *et al.*, 2015] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, and et al. Convolutional networks on graphs for learning molecular fingerprints. In *NeurIPS*, 2015.

[Gilmer *et al.*, 2017] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, and et al. Neural message passing for quantum chemistry. In *ICML*, pages 1263–1272, 2017.

[Gutmann and Hyvärinen, 2020] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, pages 297–304, 2020.

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graph. In *NeurIPS*, 2017.

[Hassani and Khasahmadi, 2020] Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *ICML*, pages 4116–4126, 2020.

[Jiang *et al.*, 2020] Jiechuan Jiang, Chen Dun, and Tiejun Huang. Graph convolutional reinforcement learning. In *ICLR*, 2020.

[Kondor and Lafferty., 2002] Risi Imre Kondor and John Lafferty. Diffusion kernels on graphs and other discrete structures. In *ICML*, pages 315–322, 2002.

[Li *et al.*, 2021] Sheng Li, Jayesh K Gupta, Peter Morales, and et al. Deep Implicit Coordination Graphs for Multi-agent Reinforcement Learning. In *AAMAS*, 2021.

[Liu *et al.*, 2020] Yong Liu, Weixun Wang, and Yujing. Hu. Multi-agent game abstraction via graph attention neural network. In *AAAI*, pages 7211–7218, 2020.

[Liu *et al.*, 2023] Boyin Liu, Zhiqiang Pu, Yi Pan, and et al. Lazy agents: a new perspective on solving sparse reward problem in multi-agent reinforcement learning. In *ICML*, pages 21937–21950, 2023.

[Niu *et al.*, 2021] Yaru Niu, Rohan R Paleja, and Matthew C. Gombolay. Multi-agent graph-attention communication and teaming. In *AAMAS*, pages 964–973, 2021.

[Oliehoek, 2012] Frans A Oliehoek. Decentralized pomdps. *Reinforcement Learning*, pages 471–503, 2012.

[Peng *et al.*, 2020] Zhen Peng, Wenbing Huang, Minnan Luo, and et al. Graph representation learning via graphical mutual information maximization. In *WWW*, pages 259–270, 2020.

[Qiu *et al.*, 2021] Dawei Qiu, Jianhong Wang, Junkai Wang, and et al. Multi-agent reinforcement learning for automated peer-to-peer energy trading in double-side auction market. In *IJCAI*, pages 2913–2920, 2021.

[Rashid *et al.*, 2020] Tabish Rashid, Mikayel Samvelyan, De Witt, and et al. Monotonic value function factorisation for deep multi-agent reinforcement learning. *JMLR*, 21(178):1–51, 2020.

[Sheng *et al.*, 2022] Junjie Sheng, Xiangfeng Wang, Bo Jin, and et al. Learning structured communication for multi-agent reinforcement learning. In *AAMAS*, pages 436–438, 2022.

[Singh *et al.*, 2019] Amanpreet Singh, Tushar Jain, and Sainbayar. Sukhbaatar. Learning when to communicate at scale in multiagent cooperative and competitive tasks. In *ICLR*, 2019.

[Sukhbaatar and Fergus, 2016] Sainbayar Sukhbaatar and Rob Fergus. Learning multiagent communication with backpropagation. In *NeurIPS*, pages 2244–2252, 2016.

[Veličković *et al.*, 2017] Petar Veličković, Guillem Cucurull, Arantxa Casanova, and et al. Graph attention networks. In *ICLR*, 2017.

[Velickovic *et al.*, 2019] Petar Velickovic, William Fedus, William L Hamilton, and et al. Deep graph infomax. In *ICLR*, 2019.

[Vinyals *et al.*, 2019] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, and et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[Wu *et al.*, 2019] Felix Wu, Amauri Souza, Tianyi Zhang, and et al. Simplifying graph convolutional networks. In *ICML*, pages 6861–6871, 2019.

[Xu *et al.*, 2024] Yaqi Xu, Yan Shi, Xiaolu Tong, and et al. A multi-agent reinforcement learning based control method for connected and autonomous vehicles in a mixed platoon. *TVT*, 2024.

[Zhang *et al.*, 2024] Yutian Zhang, Guohong Zheng, Zhiyuan Liu, and et al. Marlens: understanding multi-agent reinforcement learning for traffic signal control via visual analytics. *TVCG*, 2024.

[Zheng *et al.*, 2018] Lianmin Zheng, Jiacheng Yang, Han Cai, and et al. Magent: A many-agent reinforcement learning platform for artificial collective intelligence. In *AAAI*, pages 8222–8223, 2018.