Expressive Multi-Agent Communication via Identity-Aware Learning

Wei Du[1], **Shifei Ding**[1,2,*], **Lili Guo**[1,2], **Jian Zhang**[1,2], **Ling Ding**[3]

[1] School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China
[2] Mine Digitization Engineering Research Center of Ministry of Education of the People's Republic of China, Xuzhou 221116, China
[3]College of Intelligence and Computing, Tianjin University, Tianjin, 300350, China
1394471165@qq.com, dingsf@cumt.edu.cn, liliguo@cumt.edu.cn, zhangjian10231209@cumt.edu.cn, dltjdx2022@tju.edu.cn

Abstract

Information sharing through communication is essential for tackling complex multi-agent reinforcement learning tasks. Many existing multi-agent communication protocols can be viewed as instances of message passing graph neural networks (GNNs). However, due to the significantly limited expressive ability of the standard GNN method, the agent feature representations remain similar and indistinguishable even though the agents have different neighborhood structures. This further results in the homogenization of agent behaviors and reduces the capability to solve tasks effectively. In this paper, we propose a multi-agent communication protocol via **ide**ntity-**a**ware **l**earning (IDEAL), which explicitly enhances the distinguishability of agent feature representations to break the diversity bottleneck. Specifically, IDEAL extends existing multi-agent communication protocols by inductively considering the agents' identities during the message passing process. To obtain expressive feature representations for a given agent, IDEAL first extracts the ego network centered around that agent and then performs multiple rounds of heterogeneous message passing, where different parameter sets are applied to the central agent and the other surrounding agents within the ego network. IDEAL fosters more expressive communication between agents and generates more distinguishable feature representations, which promotes action diversity and individuality emergence. Experimental results on various benchmarks demonstrate IDEAL can be flexibly integrated into various multi-agent communication methods and enhances the corresponding performance.

Introduction

Communication is crucial for multi-agent systems as each individual typically has limited visibility or capabilities of the environment. Through effective communication, humans are able to function as a team rather than a mere collection of individuals. To emulate the human team, multi-agent reinforcement learning (MARL) has investigated to develop agents that can autonomously learn to coordinate to solve complex tasks in real-world environments. Multiple agents must consider the behaviors of other agents, making it vital to allow agents to share information in MARL. Since the introduction of pioneering works such as DIAL (Foerster et al. 2016) and CommNet (Sukhbaatar, Fergus, and et al. 2016), how to achieve efficient multi-agent communication learning has been a popular research area.

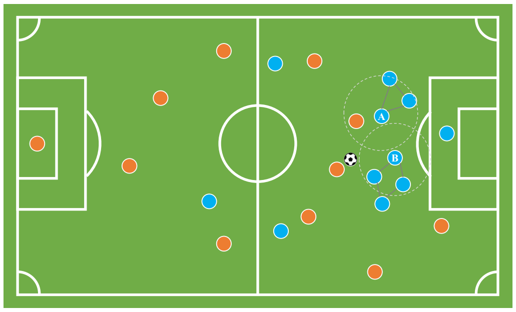


Figure 1: An example on the football game.

Recently, graph neural network (GNN) has been developed as an efficient representation learning method, which can process the topological information and attribute information of the graph-structured data to feature representation learning for the final tasks. GNN has been widely employed in constructing multi-agent communication. In this context, agents are typically represented as nodes within a graph, where the communication channels between them are depicted as edges. Many state-of-the-art MARL methods fall into this GNN-based communication paradigm, including CommNet (Sukhbaatar, Fergus, and et al. 2016), IC3Net (Singh, Jain, and Sukhbaatar 2019), TarMAC (Das, Gervet, and Romoff 2019), MAGIC (Niu, Paleja, and Gombolay 2021), DGN (Jiang, Dun, and Huang 2020), LSC (Sheng et al. 2022), DICG (Li et al. 2021), G2ANet (Liu, Wang, and Hu 2020). Other MARL methods such as DIAL (Foerster et al. 2016), and SchedNet (Kim et al. 2019) do not fall within the GNN-based communication paradigm as they utilize a fixed message-passing structure (Morris, Barrett, and Pretorius 2022).

However, recent studies have demonstrated that the expressive capabilities of these traditional GNNs are upper-bounded by the 1-WL test (Morris et al. 2019), further limiting the expressivity of multi-agent communication constructed using them. Specifically, in the context of multi-agent communication utilizing current GNNs, a significant limitation arises where agents with distinct neighborhood

[*] Corresponding author

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

structures can result in identical computational graphs, rendering them indistinguishable. As shown in Figure 1, consider a football game where different agents have similar observations. We assume that the agent selects the two nearest neighbor agents within its field of view to establish a communication edge and construct the graph. For convenience, Figure 1 only shows the neighborhood structure of Agents A and Agent B, and it can be seen that Agent A and Agent B have different neighborhood structures.

However, it can be observed that after using the previous GNN-based communication learning protocol, Agent A and Agent B still generally chose a similar behavior, that is, to move forward together to intercept the football. Since the neighborhood structures of Agent A and Agent B are different, we expect them to take different actions, such as defending different players or one of them backing up to defend and the other moving forward to intercept the football. The limited expressive ability of traditional GNN leads to the similar feature representation of Agent A and Agent B, which results in similar behavior. The homogenization of behavior tends to lead to local optimization of cooperative strategies, which seriously hinders effective exploration and reduces the final performance.

In our work, we investigate the expressive multi-agent communication from a perspective of identity information. We present a multi-agent communication protocol via **ide**ntity-**a**ware **l**earning (IDEAL) to break diversity bottleneck and achieve expressive communication by explicitly improving the distinguishability of agent feature representations. IDEAL introduces the identity information to GNN-based multi-agent communication protocol by applying inductive identity coloring and multiple rounds of heterogeneous message passing. Specifically, for a specific agent, IDEAL constructs the ego network centered around the agent. Subsequently, the message passing process is applied, with different sets of parameters utilized to compute the message representations from the center agent and the rest of the agents. Compared with existing GNN-based communication methods, IDEAL produces more distinguishable agent feature representations and fosters more expressive communication between agents, thus facilitating the emergence of action diversity and individuality. IDEAL can be flexibly combined with existing multi-agent communication protocols. We select 8 baselines and conduct experiments on 4 popular MARL benchmarks, and the results demonstrate improved performance over these baselines, which emphasize the effectiveness and versatility of IDEAL. Our key contributions include:

- We investigate the expressiveness of multi-agent communication, which is a key factor in effectively accomplishing multi-agent tasks, but has been largely ignored by the existing communication protocol.
- We propose a novel communication protocol with identity-aware learning (IDEAL) to promote expressive communication by explicitly encouraging feature representation distinguishability. IDEAL imposes no constraints on the GNN architecture, making it easily applicable to various communication protocols.

Related Work

Graph Neural Network The term graph neural network (GNN) can cover a large variety of different models. In our work, we define it as corresponding to the definition of Message Passing Graph Neural Network (Gilmer, Schoenholz, and Riley 2017), which is considered to be the most general GNN architecture. Prominent examples of this GNN architecture contain GCN (Duvenaud et al. 2015), GAT (Veličković et al. 2017), and GraphSAGE (Hamilton, Ying, and Leskovec 2017). Recently, several GNN models have been presented that possess expressive capability beyond the 1-WL test, as discussed in (Chen et al. 2019; Li et al. 2020). However, these works often introduce additional components that are typically specific to certain tasks or domains, extending beyond the standard GNN.

Some GNN models utilize node coloring techniques with augmented features to enhance the performance of existing GNNs (Veličković et al. 2020; Xu et al. 2019). However, these coloring techniques are problem and domain-specific, focusing on tasks such as link prediction, without general applicability to node-level tasks. On the contrary, ID-GNN (Jiaxuan et al. 2021) stands out as a versatile method that can be utilized for various node-level and edge-level tasks. It takes a more inclusive approach by adopting a heterogeneous message passing method. It is compatible with scenarios, where edges or nodes possess plentiful and diverse features. In our work, we manage to bring this advantage to the multi-agent communication setting.

GNN-based MARL Various MARL works have utilized GNNs to establish a communication protocol among agents. In our work, we use the following GNN-based MARL methods as the baselines. CommNet (Sukhbaatar, Fergus, and et al. 2016) defines a learnable communication channel that enables agents to enter and exit the other agents' communication range. It can be directly mapped to GNN methods where the mean is utilized for aggregation. IC3Net (Singh, Jain, and Sukhbaatar 2019) operates similarly to CommNet, with the distinction that its communication is controlled by a gating mechanism. TarMAC (Das, Gervet, and Romoff 2019) utilizes a soft attention mechanism to determine the extent to which a message is processed by an agent. This method implicitly generates a complete communication graph, which can be effectively modeled using GAT.

MAGIC (Niu, Paleja, and Gombolay 2021) constructs a communication graph and utilizes GAT for multiple rounds of communication. DGN (Jiang, Dun, and Huang 2020) operates on graphs derived deterministically from the environment. It is worth noting that DGN is the only value-based one among our selected baselines. LSC (Sheng et al. 2022) introduces a hierarchical GNN to facilitate efficient multi-agent communication learning through exchanging messages among agents and groups. G2ANet (Liu, Wang, and Hu 2020) proposes a game abstraction technique that combines both hard and soft-attention mechanisms, which enables dynamical learning of interactions between agents. DICG (Li et al. 2021) comprises a module dedicated to inferring the structure of a dynamic coordination graph, and then utilizes GNN to learn implicit reasoning about joint actions.

Preliminaries

A graph can be denoted as $G = (V, E)$, in $V = \{1, ..., n\}$ represents the node set and $E \subseteq V \times V$ denotes the edge set. Nodes have the feature $X = \{x_i \mid \forall i \in V\}$, while edges can be paired with the feature $F = \{f_{ij} \mid \forall e_{ij} \in E\}$. A GNN contains multiple message passing layers, where each layer updates the node features/embeddings/labels (these terms can be used interchangeably). The target of a GNN is to learn meaningful embeddings h_i for nodes through iterative aggregation of local neighborhoods network. For node i, the l-th iteration of the message passing, or the l-th layer of a GNN, can be represented as follows:

$$m_i^l = \mathrm{ME}^l\left(h_i^{l-1}\right), \tag{1}$$

$$h_i^l = \mathrm{AG}^l\left(\left\{m_i^l, i \in N(i)\right\}, h_i^{l-1}\right), \tag{2}$$

where h_i^l denotes the node embedding after l iterations, $h_i^0 = x_i$, m_i^l denotes the message representation and $N(i)$ denotes the local neighbor nodes of node i. Different GNN methods have a variety of definitions of $\mathrm{ME}^l(\cdot)$ and $\mathrm{AG}^l(\cdot)$. For instance, GraphSAGE utilizes the definition (W^l , U^l are trainable weights) as follows:

$$m_i^l = \mathrm{RELU}\left(W^l h_i^{l-1}\right), \tag{3}$$

$$h_i^l = U^l\,\mathrm{CON}\left(\mathrm{MAX}\left(\left\{m_j^l, j \in N(i)\right\}\right), h_i^{l-1}\right), \tag{4}$$

where CON denotes the concatenate operation. The node embeddings $h_i^l, \forall i \in V$ are then utilized for node, edge prediction tasks.

Methodology

Here we present expressive multi-agent communication via **id**entity-**a**ware **l**earning (IDEAL), which can make any message between agents more expressive in GNN-based MARL methods. It is worth noting that IDEAL is universal and any existing GNN-based MARL methods can be flexibly integrated with IDEAL.

Overview of IDEAL

In general, a fundamental limitation of existing communication in these methods is that two agents with different neighborhood structures may have the same computational graph, making the feature embeddings of agents appear indistinguishable. As shown in Figure 2, we assume that the agents have similar observations, so there are no distinguishing features initially. In multi-agent tasks, we need to distinguish the final feature embeddings of Agent A and Agent B with different neighbor structures. However, all GNN-based MARL methods, regardless of their communication rounds i.e., the number of GNN layers, always assign similar final feature embeddings to both Agent A and Agent B because their computational graphs are the same (as shown in the middle row). On the contrary, IDEAL provides a colored computational graph that can clearly distinguish the feature embeddings of Agent A and Agent B, because the colored computational graphs of two agents are no longer the same.

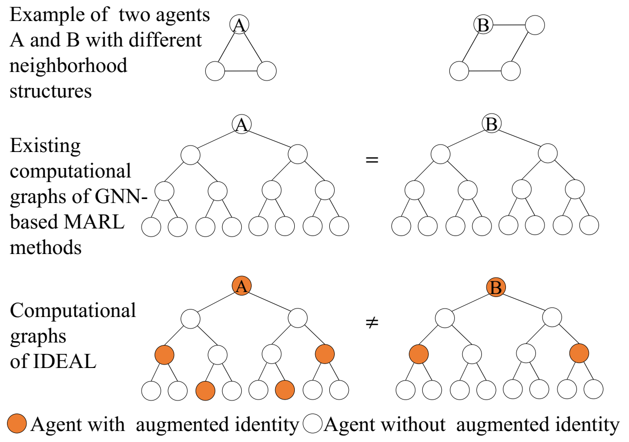

Example of two agents A and B with different neighborhood structures
A
B
Existing computational graphs of GNN-based MARL methods
A
=
B
Computational graphs of IDEAL
A
≠
B
● Agent with augmented identity ○ Agent without augmented identity

Figure 2: An overview of the proposed IDEAL.

Problem Formulation

We investigate the Decentralized Partially Observable Markov Decision Process (DEC-POMDP) (Oliehoek 2012) setting augmented with multi-agent communication. In this setting, at each timestep t, each agent i receives a local partial observation o_i^t, then selects and takes an action a_i^t, and obtains a reward r_i^t. Two paradigms of MARL are considered: value-based and actor-critic. To maintain conciseness, the term "actor network" is utilized to refer collectively to the Q-network in the value-based paradigm and the policy network in the actor-critic paradigms. In our work, we assume parameter sharing by default among the networks of different agents. Parameter sharing is a commonly employed technique in MARL to facilitate faster and more stable training. Most successful multi-agent communication protocols can be modeled within the framework as follows.

At each time step, we construct graph $G = (V, E)$ where nodes $V(G)$ represent all agents and edges $E(G)$ represent communication channels (i, j) between agents i and j. Besides, the node i in the graph is labeled with the local observation of agent i. Then we fed this graph into a GNN, which produces high-level feature embeddings for each agent. These embeddings are subsequently passed through the actor network of the corresponding agent. We assume that the actor networks employ shared weights, and use them to replace a final GNN layer L. In this case, $a_i = D(h_i^L)$, where h_i^L represents the feature representation after passing through layer L and D represents the shared actor network.

Based on the definition provided in (Morris, Barrett, and Pretorius 2022), communication protocols that fit within this framework are referred to as graph decision network (GDN). Assuming shared weights of the actor network, any GDN simplifies to an issue of labeling nodes in GNN. For a given node, the correct label is the output of the corresponding actor network that aims to maximize the reward. From here on, we focus solely on GDNs with a shared actor network. Although action selection is analogous to the node-labeling problem, the agents are trained not in a supervised learning way but in the typical MARL way (using reward signal).

Inductive Identity Coloring

IDEAL consists of two crucial components: 1) inductive identity coloring, which involves injecting identity information into each agent; 2) heterogeneous message passing, which utilizes identity information during the process of message passing. Specifically, we utilize the inductive identity coloring technique to differentiate the agent itself (the root agent in the computational graph as shown in Figure 2) from other neighboring agents within their respective computational graphs. To embed the feature of a specific agent $i \in G$ utilizing a $L-$ layer GNN with IDEAL, we first extract the $L-$ hop ego network G_i^L of agent i.

Next, we distribute a unique coloring to the central agent of the ego network G_i^L. In total, agents in G_i^L are classified into two classes during the embedding procedure: agents with coloring and agents without coloring. This coloring technique is considered inductive as it enables the center agent of the ego network to be distinguished from other neighboring agents, regardless of the permutation of their order. In MARL, the policy of an agent should not be based on the order that messages are obtained at a given time step, essentially being permutation invariant. Contrarily, methods like ATOC (Jiang and Lu 2018) utilize LSTMs for message aggregation, which are not permutation invariant and thus do not fall within the GNN-based communication paradigm. Since the order of agents can often be permuted in MARL, this coloring technique is more effective compared to labeling each agent feature with a one-hot encoding, which is transductive and cannot be generalized to unseen graphs.

Heterogeneous Message Passing

We subsequently conduct L rounds of message passing on all the extracted ego networks. To obtain the feature embedding of agent $j \in G_i^L$, we extend Eq.(1) and Eq.(2) to facilitate heterogeneous message passing as follows:

$$m_c^l = \mathrm{ME}_{\mathbb{I}[c=i]}^l \left(h_c^{l-1}\right), \tag{5}$$

$$h_j^l = \mathrm{AG}^l \left(\left\{m_c^l, c \in N(j)\right\}, h_j^{l-1}\right), \tag{6}$$

where only h_i^L is utilized as the feature embedding for agent i after applying L rounds of Eq.(6). Unlike Eq.(1), we utilize two sets of ME^l functions, where $\mathrm{ME}_1^l(\cdot)$ is utilized for agents with identity coloring, and $\mathrm{ME}_0^{(l)}(\cdot)$ is applied to agents without coloring. We utilize the indicator function $\mathbb{I}[c = i]$ to index the selection of these functions, where $\mathbb{I}[c = i] = 1$ if $c = i$ else 0. This enables the encoding of inductive identity coloring into the IDEAL computational graph. An advantage of this message passing technique is its applicability to any GNN-based MARL. For instance, consider the following message passing strategy, which expands upon the definition of GNNs in Eq.(5) and Eq.(6) by incorporating edge attributes f_{cj} during the process of message passing:

$$m_{cj}^l = \mathrm{ME}_{\mathbb{I}[c=i]}^l \left(h_c^{l-1}, f_{cj}\right), \tag{7}$$

$$h_j^l = \mathrm{AG}^l \left(\left\{m_{cj}^l, c \in N(j)\right\}, h_j^{l-1}\right). \tag{8}$$

Algorithm 1: IDEAL

Input: $G(V; E)$, agent observations $\{o_i, \forall i \in V\}$, number of layers L ; trainable functions $\mathrm{ME}_1^l(\cdot)$ for agents with identity coloring, $\mathrm{ME}_0^l(\cdot)$ for the rest of agents; $\mathrm{EGO}(i, l)$ extracts the L -hop ego network centered at agent i, indicator function$\mathbb{I}[c = i] = 1$ if $c = i$ else 0
Output: a_i for all $i \in V$

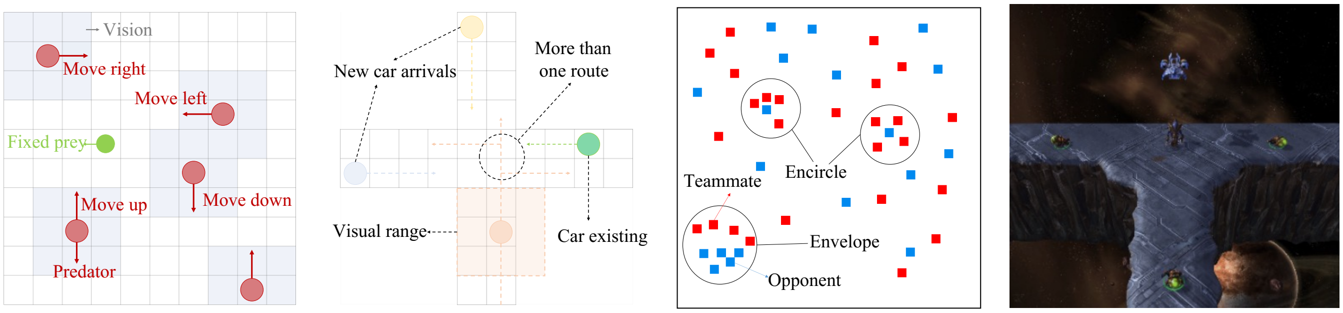
```
for $i \in V$ do
    $x_i \leftarrow \mathrm{EN}(o_i)$
    $G_i^L \leftarrow \mathrm{EGO}(i, L)$
    $h_j^0 \leftarrow x_j, \forall j \in G_i^L$
    for $l = 1, \ldots, L$ do
        for $j \in G_i^L$ do
            $h_j^l \leftarrow \mathrm{AG}^l \Big($
                $\Big\{\mathrm{ME}_{\mathbb{I}[c=i]}^l \left(h_c^{l-1}\right), l \in N(j)\Big\}, h_j^{l-1}\Big)$
    $h_i \leftarrow h_i^L$
    $a_i \leftarrow D(h_i^L)$
```

Algorithm of IDEAL

Algorithm 1 shows the instantiation of IDEAL. Given the agent observations $\{o_i, \forall i \in V\}$, we use the encoder to process it and generate the initial agent feature $\{x_i, \forall i \in V\}$. Then x_i is fed to IDEAL, which consists of two crucial components: inductive identity coloring and heterogeneous message passing. After L rounds of message passing, we obtain the expressive feature representation h_i^L. Assuming that the actor networks utilize shared weights, we can utilize them to replace the final GNN layer L. Next, we use the actor networks to obtain the action $a_i = D(h_i^L)$. IDEAL generates more distinguishable feature representations, which promotes diversity of actions.

Given the objective of analyzing expressivity, it is not essential to consider how the IDEAL agent is trained. The focus is solely on the capability of the method to generate the desired output, rather than the specifics of the training process leads to convergence. What matters is that there exist "optimal" outputs for each agent in the actor network, determined by some metric of optimality. We can then evaluate the model's ability to produce these outputs. Even in scenarios involving heterogeneous agents, this paradigm can still be applied by incorporating a portion of the observations to indicate the agent's class, often achieved through one-hot encoding. Methods incorporating recurrent networks can still fit into this framework, in which the cell or hidden states of the networks can be regarded as part of the observations.

In addition to introducing identity coloring and two classes of message passing, the computation of IDEAL closely resembles the widely used GNNs in other multi-agent communication methods. The simplicity of IDEAL holds promise for further exploration into the expressiveness of multi-agent communication. In the experiment, when the number of trainable parameters is matched, the training time of the MARL method using IDEAL and traditional GNN is almost identical.



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

Figure 3: Illustration of the four selected MARL benchmarks.

Experiments

To demonstrate the effectiveness of IDEAL, we conduct various experiments on four MARL benchmarks: Predator-Prey, Traffic Junction, Battle, and SMAC. For each benchmark, we chose two different GNN-based MARL methods as the baselines. Table 1 summarizes these baselines. All implementations have been extended to support multiple rounds of message-passing, and the capability for baseline communication to be masked by the environment has been enhanced. The detailed hyper-parameters are given in the Appendix. All baseline methods have been introduced in related work and are briefly summarized as follows.

- TarMAC utilizes GAT with a soft attention mechanism to determine the extent to which a message is processed.
- MAGIC constructs a communication graph and utilizes GAT for multiple rounds of communication.
- CommNet can be directly mapped to GNN methods, where the mean is utilized for aggregation.
- IC3Net has complete communication and uses a gating mechanism to control communication.
- DGN operates on graphs derived deterministically from the environment and is the only value-based method.
- LSC introduces a hierarchical GNN to facilitate efficient multi-agent communication learning.
- G2ANet utilizes GAT that combines both hard and soft-attention mechanisms to learn communication.
- DICG utilizes GNN to learn communication and implicit reasoning about joint actions.

Baseline	GNN structure	Benchmark
TarMAC	Implicit GAT	Predator-Prey
MAGIC	Explicit GAT	Predator-Prey
CommNet	Sum aggregation	Traffic Junction
IC3Net	Sum aggregation	Traffic Junction
DGN	Explicit GCN	Battle
LSC	Explicit GCN	Battle
G2ANet	Explicit GAT	SMAC
DICG	Explicit GCN	SMAC

Table 1: GNN structure and benchmark of baselines.

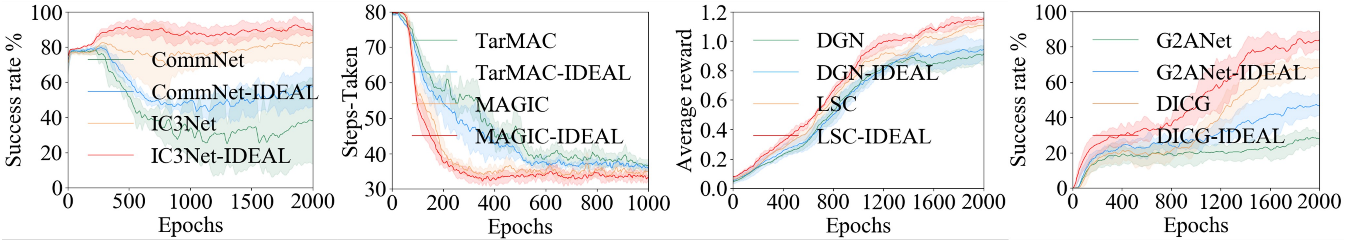
Predator-Prey

We employ the Predator-Prey benchmark introduced in (Singh, Jain, and Sukhbaatar 2019). As shown in Figure 3(a), there are multiple predators with limited sight, whose goal is to find stationary prey. The predators are able to select actions such as moving left, right, up, or down. We use the "mixed" mode of the Predator-Prey environment, where the predators receive a reward of -0.05 for each time step until they find the prey. The success of an episode is determined by whether all the predators find the prey before reaching a predefined maximum limited time. We have designed two difficulty levels in the Predator-Prey benchmark. The difficulty level increases with the number of predators and grid size, requiring more effective communication among the predators to achieve success. The two difficulty levels are defined as follows: 10×10 grid with 5 predators, and 20×20 grid with 10 predators. In this domain, a higher-performing method is defined as one that minimizes the average number of steps required to complete an episode.

Table 2 illustrates the average number of steps taken to complete an episode at convergences (predator capture the prey). In two scenarios, TarMAC-IDEAL and MAGIC-IDEAL capture prey faster than the corresponding baseline TarMAC and MAGIC, respectively. Figure 4(a) illustrates the learning curves of IDEAL and baseline methods in the high-difficulty benchmark with size 20×20. In this scenario, there is a significant improvement over the baselines, with an average gap of nearly 5 steps in terms of performance. Figure 5(a) depicts the average number of epochs required to converge with different numbers of agents. TarMAC-IDEAL and MAGIC-IDEAL maintain more quick convergence compared with the corresponding baseline as the number of agents increases.

Method	10×10	20×20
TarMAC	13.26 ± 0.10	36.22 ± 0.95
TarMAC-IDEAL	10.94 ± 0.07	31.57 ± 0.62
MAGIC	12.81 ± 0.05	33.12 ± 0.17
MAGIC-IDEAL	9.72 ± 0.03	28.45 ± 0.11

Table 2: The average number of steps required to complete an episode at convergence in Predator-Prey.



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

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

Traffic Junction

We utilize the Traffic Junction benchmark as introduced in (Sukhbaatar, Fergus, and et al. 2016). This environment serves as a useful benchmark for evaluating the effectiveness of communication. It consists of intersecting routes and cars acting as agents, each with limited visibility. The primary objective in this environment is to ensure effective communication among the cars to prevent collisions. In this environment, cars enter the junction from various entry points with a probability denoted as p. The maximum number of cars N_c allowed in the junction at any given time is constrained. At each time step, cars have two possible actions: “gas” or “brake”. The task is designed with three levels of difficulty, which differ based on the number of potential routes, entry points, and junctions present in the scenario.

We evaluate the performance of IDEAL and other baselines on two difficulty levels. In the medium difficulty level, as shown in Figure 3(b), the traffic junction benchmark contains two, two-way roads arranged on a 14 × 14 grid, and the maximum number of agents in the domain is ten (N_c= 10, p = 0.2). The hard difficulty level involves four, two-way roads on a 18 × 18 grid, and the maximum number of agents in the domain is twenty (N_c= 20, p = 0.05). The objective is to maximize the success rate, defined as the absence of collisions within an episode.

Table 3 presents the success rate achieved by each method at convergence in the two difficulty levels. Figure 4(b) illustrates the learning curve for each method in the medium difficulty level. In both scenarios, CommNet-IDEAL and IC3Net-IDEAL outperform their respective baseline methods, CommNet and IC3Net. Figure 5(b) displays the average number of epochs required for each method to converge. As the number of agents grows, IC3Net-IDEAL and CommNet-IDEAL exhibit faster convergence rates compared to their corresponding baseline methods.

Method	Medium	Hard
CommNet	53.62 ± 13.81	51.56 ± 2.37
CommNet-IDEAL	65.27 ± 11.07	67.04 ± 4.28
IC3Net	87.83 ± 3.06	73.26 ± 8.72
IC3Net-IDEAL	92.15 ± 2.63	82.45 ± 6.81

Table 3: The win rate of IDEAL and baseline methods in Traffic Junction.

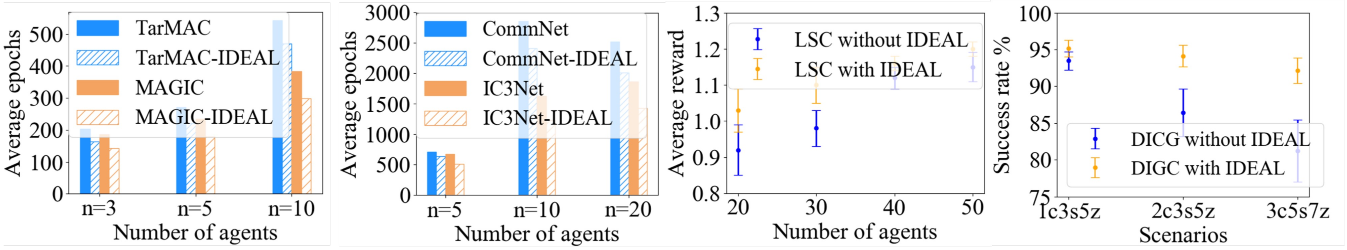
Battle

We select Battle scenario from MAgent (Zheng et al. 2018). Figure 3(c) displays the Battle scenario, consisting of Y ally agents and Z enemy agents. The objective for the ally agents is to learn to defeat all enemy agents. Each agent can choose between two actions: move or attack. While individual enemy agents possess more capabilities than individual ally agents, the ally agents must develop cooperative strategies, such as encircling, to effectively fight against the enemies. Since the Battle scenario tends to become unbalanced after the death of agents, we introduce stochastic additions of new ally or enemy agents to maintain balance. In our experiments, IDEAL and other baseline methods are trained under the same settings, with $Y = 40$ and $Z = 24$. an agent receives a positive reward of +5 when successfully attacking an enemy. A negative reward of -2 is incurred when an agent is killed by an enemy, and a negative reward of -0.01 is given when an agent hits a blank grid.

Figure 4(c) illustrates the mean reward for each method in Battle scenario. It is observed that LSC-IDEAL and DGN-IDEAL outperform their corresponding baselines, LSC and DGN, respectively. When dealing with an individual enemy, agents trained with IDEAL demonstrate the ability to coordinate their actions, surround the enemy, and achieve victory. Table 4 presents the performance of IDEAL and baseline methods in Battle scenario. IDEAL consistently outperforms the other baselines in terms of kills, kill-death ratio, and mean reward. To investigate the scalability of IDEAL in large-scale multi-agent scenarios, we compared it with other baselines in Battle scenario under different numbers of agents $Y \in \{20, 30, 40, 50\}$. As depicted in Figure 5(c), IDEAL consistently outperforms the baselines as the number of agents grows. This demonstrates the scalability ability of IDEAL, highlighting its effectiveness in handling large-scale multi-agent communication problems.

Method	Kills	K/D ratio	Mean reward
DGN	216 ± 6	2.32 ± 0.16	0.92 ± 0.17
DGN-IDEAL	225 ± 8	2.41 ± 0.19	0.96 ± 0.21
LSC	247 ± 5	2.60 ± 0.21	1.12 ± 0.03
LSC-IDEAL	252 ± 5	2.68 ± 0.23	1.16 ± 0.02

Table 4: Performance of IDEAL and baseline methods in Battle scenario.



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

Figure 5: Performance of IDEAL and baseline methods as the number of agents increase in four benchmarks.

SMAC

The StarCraft Multi-Agent Challenge (SMAC) (Vinyals et al. 2019) is a benchmark designed within the popular strategy game StarCraft II. In SMAC, all the ally agents are trained using MARL methods, while the enemy agents are controlled by the built-in AI. The action space for the agents in SMAC consists of four actions: move, attack, no-op (no operation), and stop. The attack action allows ally agents to fire at enemy agents within a range of 6. At each time step, ally agents can choose to attack to obtain a global reward. Additionally, they receive an additional reward for killing an enemy agent or winning the game. To provide a more challenging coordination task for the ally agents, we have fine-tuned the default experimental settings, which reduces the scope of vision for ally agents from 9 to 2.

Figure 4(d) displays the win rates achieved by different methods in scenario 1o10b vs 1r in SMAC. This scenario contains an Overseer, 10 Baneling, and an enemy Roach. The objective for the teammates of 1 Overseer and 10 Baneling is to eliminate this Roach to obtain the winning reward. In an effective communication strategy, the Banelings have the option to remain silent while the Overseer encodes its position features and communicates it to the Banelings. Table 5 exhibits the performance of different methods in the other two scenarios: MMM2 and 1c3s5z. In MMM2, symmetric teams comprising 7 Marines, 2 Marauders, and 1 Medivac spawn at fixed points, with the enemy team assigned the task of attacking the ally team. To emerge victorious, agents must learn to effectively communicate their health status to Medivac. 1c3s5z scenario consists of Colossus, Stalkers, and Zealots for both the ally agents and enemy agents. In this scenario, the ally agents must learn numerous tactics to win. The presence of efficient information interaction makes it easier for agents to learn these strategies and coordinate their actions effectively.

Method	MMM2	1c3s5z
G2ANet	80.24 ± 4.37	91.25 ± 1.73
G2ANet-IDEAL	85.31 ± 3.62	94.12 ± 0.94
DICG	83.45 ± 5.42	93.46 ± 1.26
DICG-IDEAL	89.23 ± 3.32	95.13 ± 1.13

Table 5: The win rate of IDEAL and baseline methods in some scenarios in SMAC.

As depicted in Figure 4(d) and Table 5, DICG-IDEAL and G2ANet-IDEAL methods exhibit superior performance compared to their respective baselines, DICG and G2ANet. Generally, as the number and classes of agents grow, the interactions between diverse classes of agents become more intricate, therefore learning policies become progressively challenging. To validate the scalability of the proposed method, we conduct experiments in scenarios 1c3s5z, 2c3s5z, and 3c5s7z, where the types of agents are the same but the number is increasing. Figure 5(d) illustrates that as the number of agents increases, the difficulty of communication learning intensifies, leading to a drop in the win rate of the baselines. However, even in the face of these challenges, IDEAL consistently maintains a remarkably high win rate while the number of agents grows.

Conclusions

We have introduced IDEAL as a versatile and powerful extension to existing GNN-based communication protocols. IDEAL enhances multi-agent communication protocols by incorporating agents' identity information in the message passing process. By facilitating more expressive communication among agents and generating distinct feature representations, IDEAL promotes the emergence of action diversity and individuality. Experimental results on various benchmarks validate that IDEAL can be flexibly integrated into diverse multi-agent communication methods and improve the performance of these methods.

The simplicity of IDEAL holds great promise for further exploration of expressive multi-agent communication. Our work encourages additional research into GNN-based multi-agent communication approaches as well as more generic yet powerful forms of communication learning. We anticipate that the enhanced expressiveness and proven practical applicability of the proposed IDEAL will enable exciting new applications and advancements in multi-agent communication. In future work, we intend to further investigate the relationship between communication expression ability and the performance of downstream real-world tasks.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant no.62276265, no.61976216.

References

Chen, Z.; Villar, S.; Chen, L.; and Bruna, J. 2019. On the equivalence between graph isomorphism testing and function approximation with gnns. In *Advances in neural information processing systems (NeurIPS)*.

Das, A.; Gervet, T.; and Romoff, J. 2019. Tarmac: Targeted multi-agent communication. In *International Conference on Machine Learning (ICML)*.

Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; and et al. 2015. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems (NeurIPS)*.

Foerster, J.; Assael, I. A.; De Freitas, N.; and Whiteson, S. 2016. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Gilmer, J.; Schoenholz, S. S.; and Riley, P. F. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning (ICML)*.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graph. In *Advances in neural information processing systems (NeurIPS)*.

Jiang, J.; Dun, C.; and Huang, T. 2020. Graph convolutional reinforcement learning. In *International Conference on Learning Representations (ICLR)*.

Jiang, J.; and Lu, Z. 2018. Learning attentional communication for multi-agent coopera- tion. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Jiaxuan, Y.; Jonathan, G. S.; Rex, Y.; and Jure, L. 2021. Identity-aware graph neural networks. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Kim, D.; Moon, S.; Hostallero, D.; and et al. 2019. Learning to schedule communication in multi-agent reinforcement learning. In *International Conference on Learning Representations (ICLR)*.

Li, P.; Wang, Y.; Wang, H.; and Leskovec, J. 2020. Distance encoding: Design provably more powerful neural networks for graph representation learning. In *Advances in neural information processing systems (NeurIPS)*.

Li, S.; Gupta, J. K.; Morales, P.; Allen, R.; and Kochenderfer, M. J. 2021. Deep Implicit Coordination Graphs for Multi-agent Reinforcement Learning. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*.

Liu, Y.; Wang, W.; and Hu, Y. 2020. Multi-agent game abstraction via graph attention neural network. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Morris, C.; Ritzert, M.; Fey, M.; Hamilton, W. L.; Lenssen, J. E.; Rattan, G.; and Grohe, M. 2019. Weisfeiler and leman go neural: Higher-order graph neural networks. In *AAAI conference on artificial intelligence (AAAI)*.

Morris, M.; Barrett, T. D.; and Pretorius, A. 2022. Universally expressive communication in multi-agent reinforcement learning. In *Advances in neural information processing systems (NeurIPS)*.

Niu, Y.; Paleja, R. R.; and Gombolay, M. C. 2021. Multi-agent graph-attention communication and teaming. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*.

Oliehoek, F. A. 2012. Decentralized pomdps. *Reinforcement Learning*, 471–503.

Sheng, J.; Wang, X.; Jin, B.; Yan, J.; Li, W.; Chang, T.-H.; Wang, J.; and Zha, H. 2022. Learning structured communication for multi-agent reinforcement learning. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*.

Singh, A.; Jain, T.; and Sukhbaatar, S. 2019. Learning when to communicate at scale in multiagent cooperative and competitive tasks. In *International Conference on Learning Representations (ICLR)*.

Sukhbaatar, S.; Fergus, R.; and et al. 2016. Learning multiagent communication with backpropagation. In *Advances in neural information processing systems (NeurIPS)*.

Veličković, P.; Cucurull, G.; Casanova, A.; and et al. 2017. Graph attention networks. In *International Conference on Learning Representations (ICLR)*.

Veličković, P.; Ying, R.; Padovano, M.; Hadsell, R.; and Blundell, C. 2020. Neural execution of graph algorithms. In *International Conference on Learning Representations (ICLR)*.

Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; and et al. 2019. Grandmaster level in StarCraft II using multi- agent reinforcement learning. *Nature*, 575(7782): 350–354.

Xu, K.; Li, J.; Zhang, M.; Du, S. S.; Kawarabayashi, K.-i.; and Jegelka, S. 2019. What can neural networks reason about? In *International Conference on Learning Representations (ICLR)*.

Zheng, L.; Yang, J.; Cai, H.; Zhou, M.; Zhang, W.; Wang, J.; and Yu, Y. 2018. Magent: A many-agent reinforcement learning platform for artificial collective intelligence. In *AAAI Conference on Artificial Intelligence (AAAI)*.