Hierarchical Cognitive Graph Autoencoder for Multi-Agent Reinforcement Learning

Peng He
Beijing University of Posts and Telecommunication
hepeng123@bupt.edu.cn

Wei Wang
Beijing University of Posts and Telecommunication
weiwang@bupt.edu.cn

Chuxiong Sun
Institute of Software Chinese Academy of Sciences
chuxiong2016@iscas.ac.cn

Yi Wang
Beijing University of Posts and Telecommunication
yiwang@bupt.edu.cn

Abstract

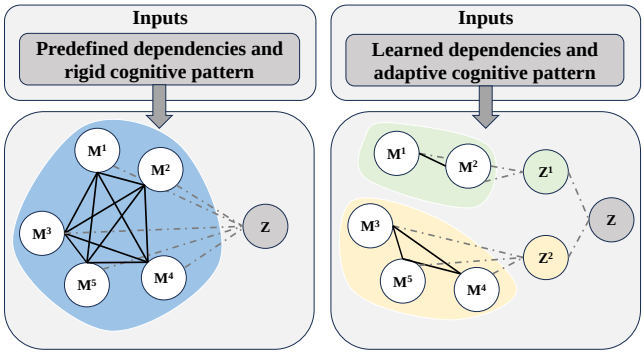
Communication is essential for enhancing the cognition and cooperation of agents in multi-agent reinforcement learning (MARL). However, existing methods often rely on predefined and rigid cognitive patterns, which cannot adapt to dynamic environmental changes and complex inter-agent interactions. In this work, we introduce the Hierarchical Cognitive Graph Autoencoder (HCGA), an adaptive framework that addresses these limitations. HCGA represents inter-agent messages as nodes in a graph with learnable edges, employs a grouping mechanism to integrate related local information into compact latent representations, and then applies hierarchical aggregation to construct a comprehensive global cognition. This approach effectively distills essential information and adaptively uncovers cognitive patterns from dynamic environments, thereby enhancing the overall robustness and efficiency of cognitive processing in MARL tasks. Experimental results demonstrate that HCGA significantly outperforms state-of-the-art methods across various MARL tasks, highlighting its robustness, adaptability, and efficiency.

Keywords: multi-agent reinforcement learning; graph convolution network; cooperative cognition

Introduction

Reinforcement Learning (RL) has achieved great success across a diversity of intricate real-world domains, ranging from Game AI (Osband, Blundell, Pritzel, & Van Roy, 2016; Silver et al., 2017, 2018; Vinyals et al., 2019) and robotics (Andrychowicz et al., 2020) to autonomous driving (Leurent, 2018). However, when applied to complex multi-agent systems, distinct challenges emerge. A primary challenge is partial observability—agents are confined to their local perspectives, lacking a comprehensive view of the environment. This limitation poses a fundamental cognitive challenge in multi-agent reinforcement learning (MARL), as agents must interpret and integrate fragmented observations to form a coherent understanding of dynamic surroundings for effective cooperative decision-making.

Multi-agent communication offers a promising solution by enabling agents to share their observed information, thereby enriching their collective perception and cognition of the environment. Despite substantial advancements, current approaches predominantly concentrate on the sending aspects of communication—for instance, generating meaningful messages (Zhang, Zhang, & Lin, 2019, 2020; Yuan et al., 2022; Sun et al., 2021; Sun, He, Wang, & Zheng, 2025; Sun, He, et al., 2024), selecting communication timing (Singh, Jain, & Sukhbaatar, 2018; Kim et al., 2019) and communication target (Sun, Zang, et al., 2024)to conserve bandwidth and improve efficiency. However, these methods often overlook the critical receiving-side challenge: effectively assimilating and leveraging the communicated data to boost the agents' overall situational cognition and build a unified, actionable representation of the environment. Concretely, existing methods typically rely on predetermined and rigid cognitive patterns, where all available information is integrated at every step, regardless of how the environment changes. Such a one-size-fits-all approach fails to capture the dynamic evolution of the environment and the nuanced interactions among agents.



(a) Predefined cognitive aggregation (b) Learned cognitive aggregation

Figure 1: Pre-defined vs. Adaptive cognitive patterns.

In human cognitive decision-making, individuals influence others' beliefs and behaviors through communication. This mechanism also applies to agent-based cognition(Sumers, Hawkins, Ho, & Griffiths, 2021). Inspired by how humans dynamically adjust their cognitive processes and structures in response to external changes, we propose the Hierarchical Cognitive Graph Autoencoder (HCGA)—a novel adaptive framework designed to support complex multi-agent reinforcement learning tasks. To capture the adaptive dependencies inherent in the received messages, we represent messages from different agents as nodes in a graph and model their relationships as learnable edges. Our framework leverages both grouping and hierarchical strategies to enhance cognitive processing. Specifically, it first employs a grouping stage, wherein local cognitive patterns are learned to integrate related messages together into a compact latent graph node. This grouping mechanism effectively distills critical

In D. Barner, N.R. Bramley, A. Ruggeri and C.M. Walker (Eds.), *Proceedings of the 47th Annual Conference of the Cognitive Science Society*

local information, filters out noise, and reduces redundancy. Building upon these grouped representations, a subsequent hierarchical layer integrates the latent graph into a comprehensive global representation. This two-tiered approach allows HCGA to dynamically adapt its information aggregation strategies in response to environmental changes. As illustrated in Fig.1, our method enables end-to-end learning of cognitive patterns and information integration without relying on predefined assumptions, thereby offering significant advantages in robustness, adaptability, and efficiency. By enhancing cognitive processing, HCGA outperforms state-of-the-art baselines by a large margin across a variety of MARL tasks.

Related Works and Preliminaries

Decentralized Partially Observable Markov Decision Process (Dec-POMDP)

A Dec-POMDP is a framework for modeling cooperative tasks involving multiple agents under partial observability and decentralized decision-making. It is defined as: $\langle I, \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^n, P, \{O_i\}_{i=1}^n, \{\pi_i\}_{i=1}^n, R, \gamma \rangle$ where I is the set of agents ($|I| = n$), $\mathcal{S}$ represents the state space of the environment and $s \in \mathcal{S}$ is the true state of the environment. At each time step t, agent i receives partial information through its observation $o_i^t \in O_i$, determined by the observation function $O(o_i^t \mid s_t)$. Each agent selects an action $a_i^t \in \mathcal{A}_i$ based on its local policy $\pi_i(a_i^t \mid \tau_i^t)$, where $\tau_i^t = (o_i^0, a_i^0, \ldots, o_i^{t-1}, a_i^{t-1}, o_i^t)$ represents its observation history. The joint action $\mathbf{a}_t = (a_1^t, \ldots, a_n^t)$ determines the next state $s_t + 1$ via the transition function $P(s_{t+1} \mid s_t, \mathbf{a}_t)$, while the reward function $R(s_t, \mathbf{a}_t)$ provides a shared global reward. The goal is to maximize the global cumulative discounted return: $Q_{\text{tot}}(s, \mathbf{a}) = \mathbf{E}\left[\sum_{t=0}^{T} \gamma^t R(s_t, \mathbf{a}_t) \mid s_0 = s, \mathbf{a}_0 = \mathbf{a}\right]$, where $\gamma \in [0, 1]$ is the discount factor. Agents collaborate by learning joint policies $\{\pi_i\}_{i=1}^n$ to optimize this objective.

The potential communication and cooperation relationships of agents can generally be represented by a meaningful dynamic topology $G = (\mathcal{H}, \mathcal{E})$,where $\mathcal{H}$ is the embedding set of agents and $\mathcal{E}$ is the set of edges encoding their interactions. Agents may exchange information through message passing, where each agent i receives messages $c_i^t = \sum_{j \neq i} m_j^t$ from other agents j. These messages, combined with local observations, allow agents to make better decisions.

Graph-based MARL

One of the primary challenges in MARL is managing the exponential growth of the joint action space as the number of agents increases (Oroojlooy & Hajinezhad, 2023). In response, the Centralized Training with Decentralized Execution (CTDE) framework (Lowe et al., 2017; Foerster, Farquhar, Afouras, Nardelli, & Whiteson, 2018) was introduced to strike a balance between computational efficiency and the need for effective multi-agent collaboration. However, CTDE often falls short in capturing the intricate dependencies among agents, which are crucial for optimal coordination. To overcome this limitation, Graph Neural Networks (GNNs) have emerged as a powerful tool for modeling relational dependencies (Wu et al., 2021; Liu et al., 2020), positioning graph-based approaches as a promising direction in MARL. These methods can be broadly categorized into two main types.

Graph for utility and payoff function. The first type uses graphs as coordination graphs to facilitate policy training, exemplified by techniques such as DCG (Boehmer, Kurin, & Whiteson, 2020), SOP-CG (Yang et al., 2022), and CASEC (Wang et al., 2022). In this framework, the joint action-value function is decomposed and defined as follows:

$$Q_{tot}(s_t, a) = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} q^i\left(a^i \mid s_t\right) + \frac{1}{|\mathcal{E}|} \sum_{\{i,j\} \in \mathcal{E}} q^{ij}\left(a^i, a^j \mid s_t\right) \quad (1)$$

In this formulation, the first term focuses on computing the Q-value for individual actions, which is commonly referred to as the utility function, while the second term evaluates the pairwise interactions between agents' actions, known as the payoff function. This structure explicitly assesses both the individual contributions of actions and the collaborative quality of joint actions among agents.

Graph for message generation. Another type leverages graphs to enable efficient information sharing and communication among agents, as seen in methods like DICG (Wang, Wang, Zheng, & Zhang, 2020) and G2ANet (Duan, Xuan, Qiao, & Lu, 2022). This approach is formally expressed as:

$$m_i = \text{AGGREGATE}_{j \in \mathcal{N}_i}(f(o_j, a_j)), \quad Q_{tot} = \sum_{i=1}^{n} Q_i(o_i, a_i, m_i) \quad (2)$$

In this formulation, $\mathcal{N}_i$ represents agent i's neighbor set. The function $f(\cdot)$ transforms raw observations and actions into embeddings, while AGGREGATE$(\cdot)$ aggregates these embeddings based on the graph structure to generate message m_i. This message enhances decision-making and implicitly captures agent coordination. While these methods do not explicitly compute the payoff-utility function using a coordination graph, they rely on the same principle of inferring joint actions through agent interactions.

Information aggregation

Information aggregation is a key focus in Graph Neural Networks (GNNs), where methods like GraphSAGE (Hamilton, Ying, & Leskovec, 2018) and GAT (Veličković et al., 2018) have advanced the field by efficiently integrating local and global features. Inspired by these, our work builds on DiffPool (Ying et al., 2019), which uses hierarchical graph pooling to dynamically cluster nodes, preserving structural information while reducing computational costs. This approach is particularly suited for multi-agent systems, where interactions often exhibit layered dependencies.

Among nongraph-based multiagent communication methods, MAISA is closely related to our approach, particularly in the information extraction phase, where we adopt a similar strategy. However, MAISA does not explicitly model the

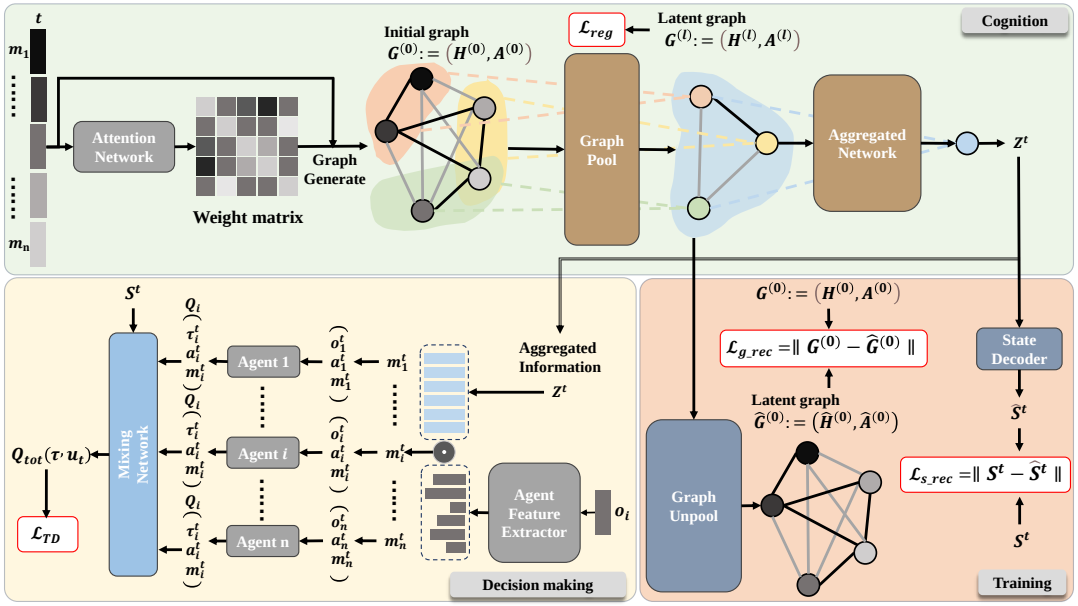


Figure 2: The HCGA framework consists of three modules: cognition, training, and decision making. The cognition module constructs a weighted communication graph via attention and pools it into a latent graph for global representation. The training module refines information through graph unpooling and state reconstruction. In the decision-making module, agents extract personalized features from the aggregated representation to guide their actions.

generation process of information aggregation, whereas our method significantly enhances interpretability through a hierarchical aggregation mechanism, clearly illustrating the process of information aggregation from local to global levels.

Method

Figure 2 illustrates the framework of HCGA, designed to learn adaptive information aggregation patterns and enhance cognition in MARL. Leveraging the representational power of GNNs and a grouping-and-hierarchical aggregation mechanism, HCGA uncovers hidden patterns from inter-agent communication graphs to boost agents' cognition. Our method is structured into three key parts.

Grouping and hierarchical cognition

We first define a communication graph based on the received messages. In this graph, each node represents a message from a sending agent, while the edges capture the relationships among messages from different sources, the initial graph is denoted as: $G^{(0)} := (H^{(0)}, A^{(0)})$. $A^{(0)} \in [0,1]^{N\times N}$ is the adjacency matrix between N agents and $H^{(0)} \in \mathbb{R}^{N\times d}$ is the matrix of node feature. Concretely, the relationships among messages are captured by an attention network $f_{att}(\cdot)$:

$$A_{ij}^{(0)} = f_{att}(m_i^t, m_j^t) \tag{3}$$

$A^{(0)}$ is regarded as the weight matrix of agent-pair, whose dimension is $N \times N$. This weight matrix not only quantifies the strength of the relationships between agent pairs but also serves as a crucial input for subsequent graph-based learning algorithms, facilitating more effective information propagation and collaborative decision-making among agents.

After building the communication graph, our next step is to aggregate the information between the different nodes in the initial graph. Our approach is based on graph neural networks and therefore follows GNN's "message-passing" architecture as follows:

$$H^{(k)} = M(A, H^{(k-1)}; \theta^{(k)}) \tag{4}$$

where $H^{(k)}$ are the node embeddings calculated after k steps of the GNN. The message propagation function is represented by M. This function relies on three key elements: the adjacency matrix A, the trainable parameters $\theta^{(k)}$, and the node embeddings $H^{(k-1)}$ that were generated during the previous message-passing step.The graph obtained after message passing is commonly referred to the latent graph which can be formally represented as: $G^{(l)} := (H^{(l)}, A^{(l)})$.

GNN-based MARL methods typically aggregate neighborhood information by controlling message-passing steps (k). However, we propose considering higher-level global relationships. Since agents with similar messages provide redundant information, our framework hierarchically aggregates nodes with significant information differences. Additionally, it adaptively selects optimal aggregation strategies based on message content, reducing redundancy and better capturing complex multi-agent dependencies.

Graph Pool. In order to achieve hierarchical aggregation, we propose a graph pool operation to get the latent graph by adapting the DIFFPOOL (Ying et al., 2019):

$$(A^{(l)}, H^{(l)}) = \text{POOL}(A^{(l-1)}, H^{(l-1)}) \tag{5}$$

We denote the input adjacency matrix at this layer as $A^{(l)} \in [0,1]^{K' \times K'}$ and and denote the input node embedding matrix at this layer as $H^{(l)} \in \mathbb{R}^{K' \times r}$, where l represents the l-th layer of our model, r represents the latent embedding dimension, K' represent the nodes number after pooling. In our method, the ratio K'/K and the output dimension r are hyperparameters.

The essence of the POOL$(\cdot)$ method is to learn the cluster assignment matrix over the nodes by using the output of the GNN model.This matrix is expressed as $S^{(l)} \in \mathbb{R}^{K \times K'}$. At the l layer of graph pool, we can compute the assignment matrix as follows:

$$S^{(l)} = \text{softmax}\left(\text{GNN}_{l,\text{pool}}(A^{(l)}, H^{(l)})\right) \tag{6}$$

In this process, we define the embedding $X^{(l)}$ after the message passing of GNN, as calculated below:

$$X^{(l)} = \text{GNN}_{l,\text{embed}}(A^{(l)}, H^{(l)}) \tag{7}$$

After obtaining the embedding and assignment matrix by Eq. 6 and 7, we can define the POOL$(\cdot)$ by the following two equation:

$$A^{(l+1)} = \left(S^{(l)}\right)^T A^{(l)} S^{(l)} \in \mathbb{R}^{K' \times K'} \tag{8}$$

$$H^{(l+1)} = \left(S^{(l)}\right)^T X^{(l)} \in \mathbb{R}^{K' \times r} \tag{9}$$

Through Eq.8 and 9,we get the latent graph $G^{(l+1)} := (H^{(l+1)}, A^{(l+1)})$ from layer $l+1$. Each node in layer $l+1$ corresponds to multiple nodes in layer l. The latent graph's node count decreases with deeper layers. Finally, we obtain the output aggregated representation z^t by the aggregated Network. During the training phase, we augment the value function by incorporating both individual observations and the aggregate representation z^t contains the information required to infer the true state, this reduces environmental uncertainty and enhances Q-value estimation in value-based policy learning.

Cognition-enhanced decision making

After building the global cognitive representation, and considering that different agents have distinct roles and tasks, we extract the cognitive components most critical to each agent's local decision-making from the global cognition. Concretely, each agent employs a feature extraction network to derive relevant features, which are then element-wise multiplied with the globally aggregated information, as shown in the following equation:

$$z_i^t = z^t \otimes f_{extra}(o_i^t) \tag{10}$$

where $\otimes$ is an element-wise Hadamard product function. $f_{extra}(\cdot)$ is implemented by a Multi-Layer Perception (MLP).This yields a unique representation for each agent derived from the global information. This approach enables each agent to process each dimension of the aggregated information with different weights, thereby effectively filtering out redundant information. To some extent, this reflects the cognitive differences among agents regarding the same environment. Subsequently, the local information extracted by each agent from the global features is fed into the subsequent network for Q-value computation and decision-making.The process of Q-value computation is mathematically represented as:

$$Q_{tot} = \sum_{i=1}^{n} Q_i(o_i, a_i, z_i) \tag{11}$$

Training

Regularization loss. Training the pooling function (Eq. 8) with only unsupervised loss gradients often proves ineffective in practice. A key issue is that the function may converge to trivial solutions, where information is evenly distributed across latent nodes, failing to capture the desired diversity in local information aggregation. To counteract this, we incorporate an orthogonality regularization term, which encourages the model to learn more meaningful and discriminative latent representations.

$$\mathcal{L}_{reg} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{C} \sum_{k=2}^{K'} \sum_{j=1}^{k-1} \left\| \rho\left(h_i^k, h_i^j\right) \right\|_1 \tag{12}$$

where $C = \frac{K' \cdot (K'-1)}{2}$ represents the total number of pairwise correlations, and $\rho(\cdot)$ is computed using cosine similarity.

Reconstruction loss. The graph unpool reconstructs the original multi-agents' input from the decoder latent graph representation.

$$(\hat{A}^{(0)}, \hat{H}^{(0)}) = \text{UNPOOL}(A^{(l)}, H^{(l)}) \tag{13}$$

Unpool is mathematically equivalent to the inverse process of the pool step(Eq. 6 to 9) At the same time, for the generation process of the aggregate representation z^t, we employ an additional decoder designed to reconstruct the global state from the aggregate representation to allow self-monitoring of the global encoder.Thus, our reconstruction loss comprises the following two components:

$$\mathcal{L}_{rec} = \frac{1}{NK} \sum_{i=1}^{N} \sum_{k=1}^{K} \|x_i^k - \hat{x}_i^k\|_2^2 + \mathbb{E}_{z_i^t, s_t} \|s_t' - s_t\|_2^2 \tag{14}$$

where the first term is a loss on reconstructed node embeddings, and the second term is a loss on the recovered the global state.

Overall training objective. Our algorithm extends QMIX (Rashid et al., 2018), integrating all individual Q values for overall reward maximization. The training involves minimizing a loss function, composed of a temporal-difference (TD) loss and the group distance loss, as follows:

$$\mathcal{L}(\theta) = \mathcal{L}_{TD}(\theta^-) + \lambda \mathcal{L}_{reg}(\theta_{reg}) + \mu \mathcal{L}_{rec}(\theta_{rec}) \tag{15}$$

where θ includes all parameters in the model, λ and μ are hyperparameters, representing the weights of regularization loss

and reconstruction loss, respectively. The TD loss is defined as:

$$\mathcal{L}_{TD}(\theta^{-}) = \left[r + \gamma \max_{a'} Q_{tot}\left(s', a'; \theta'\right) - Q_{tot}(s, a; \theta^{-}) \right]^2 \tag{16}$$

where θ' represents the parameters of a target network that is updated at regular intervals.

Experiment

In this section, we aim to explore the following questions experimentally.

- **Q1.** How does HCGA perform compared to other coordination graph methods and communication methods?
- **Q2.** Does the hierarchical GNN framework enhance the agent's perception of the environment?
- **Q3.** Does the regularization and reconstruction methods play a role in optimizing HCGA?

Setup

Benchmark To demonstrate the effectiveness and generality of our method, experiments in this study were conducted across the following three multi-agent scenarios: SMAC(StarCraft Multi-Agent Challenge)(Samvelyan et al., 2019), SMAC Communication, Hallway. Within SMAC, the maps encompass a range of difficulties, and we selected two tasks of hard difficulty, namely *MMM2* and *10m_vs_11m*. For SMAC Communication, we chose three tasks *1o_10b_vs_1r, 1o_2r_vs_4r, 5z_vs_1ul*. In the Hallway scenario, we employed the hallway group task, where agents are divided into different groups that must reach distinct destinations. The lengths of the Markov chains for these two groups were set to (3,5,7) and (4,6,8,10), respectively.

Baseline We employ several state-of-the-art Graph-based algorithms and communication method as baselines. DCG(Boehmer et al., 2020) establishes direct connections among all agents to form an unweighted fully connected graph, which is utilized to compute action pair functions. DICG(Li, Gupta, Morales, Allen, & Kochenderfer, 2021) leverages attention mechanisms to construct weighted fully connected graphs for information exchange between agents. SOP-CG(Yang et al., 2022) selects sparse graphs from a precomputed candidate set. LSTCG(Duan, Lu, & Xuan, 2024) generates and samples sparse graphs based on agents' historical observations, facilitating knowledge sharing. TarMAC(Das et al., 2019) uses attention mechanisms at the sender to determine which information to transmit to receivers. MAIC(Yuan et al., 2022) implements an implicit communication mechanism, enabling agents to infer intent by observing each other's behaviors, thereby enhancing their understanding of the environment and teammates.Since our method is proposed based on value decomposition, we also choose QMIX(Rashid et al., 2018) as a baseline for comparison.

Performance(Q1)

Fig.3 compares our method with baselines across six tasks, demonstrating HCGA's consistent superiority. On the 1o_10b_vs_1r and *1o_2r_vs_4r* maps, which evaluate communication-driven MARL, our method achieves substantial performance gains, confirming the effectiveness of our hierarchical framework in promoting agent collaboration through communication.

In the *5z_vs_1ul* scenario, HCGA outperforms all baselines except QMIX, likely due to the reduced partial observability where agents focus on a single target. In the more complex *MMM2* task, coordination graph methods perform poorly with win rates below 80%. Although QMIX and MAIC achieve similar final results, HCGA exhibits faster convergence and superior early-stage performance.

In the *10m_vs_11m* and *hallway_group* scenarios, we achieved near-perfect win rates (100%), with HCGA showing faster convergence than baselines in early training (below 1 million steps for *10m_vs_11m*, 0.4 million steps for *hallway_group*). Notably in *hallway_group*, baseline performance struggled significantly: DICG, MAIC, QMIX, and SOP-CG failed to coordinate agents' simultaneous arrival at endpoints across decision steps. While TarMAC and LTSCG eventually succeeded in late training phases, they exhibited high variance. These results demonstrate that our hierarchical aggregation framework enables agents to rapidly develop comprehensive environmental awareness while efficiently learning optimal cooperative strategies, ultimately achieving superior performance metrics.

Ablation experiment(Q2,Q3)

In order to further verify the effectiveness of HCGA method, we proposed the following three Settings to conduct ablation experiments:

- *HCGA w/o Pool*: Aggregated initial graph representation into one node through the aggregated network, removing the Graph pool and unpool, and keeping the state decoder
- *HCGA w/o Reg*: The loss of latent graph regularization was excluded from the HCGA
- *HCGA w/o Rec*: Excluding the loss for initial graph reconstruction and state reconstruction from HCGA

We conducted ablation experiments under the *1o_10b_vs_1r* and *MMM2* scenarios. As depicted in Fig.4, the performance of *HCGA w/o Pool* in both scenarios is inferior to *HCGA*, substantiating the superior efficacy of our hierarchical GNN framework in multi-agent environments. Compared to single-layer aggregation, our method better guides agents toward optimal decisions. The pronounced performance drop in *MMM2* versus *1o_10b_vs_1r* when removing Pool indicates that hierarchical aggregation benefits scenarios with more units.

In Fig.4, comparisons among *HCGA w/o Reg*, *HCGA w/o Rec*, and *HCGA* reveal performance degradation when

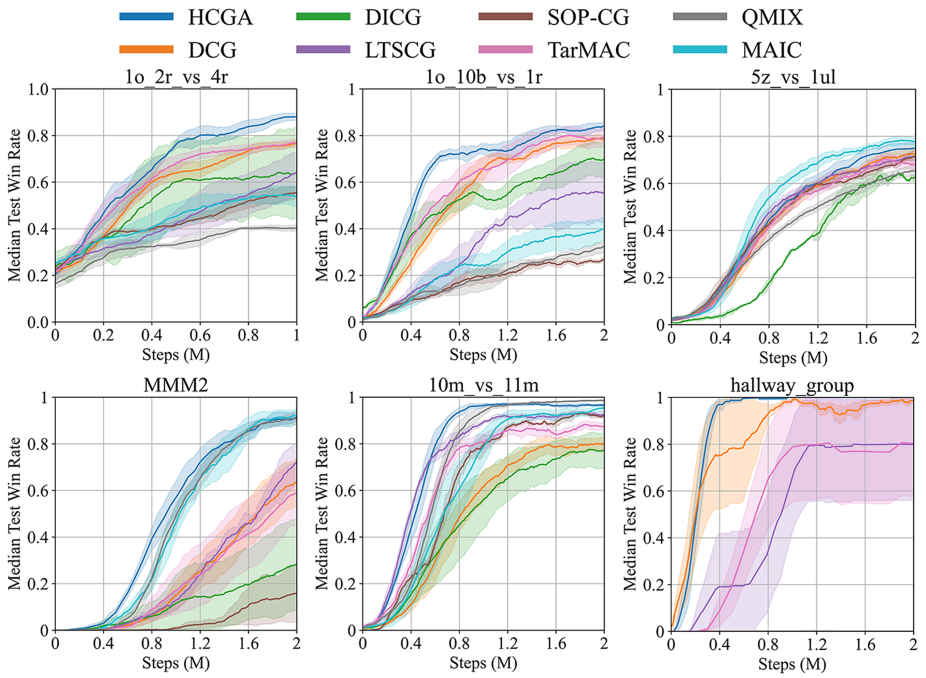


Figure 3: Performance on multiple benchmarks. All results are reported as the median performance with a 95% confidence interval over five random seeds.

regularization or reconstruction methods are removed from HCGA's representation generation. Specifically, *HCGA w/o Reg* shows milder decline, as the absence of regularization may lead to a scenario where multiple groups contain the same node during the pool process, resulting in redundancy of node information. However, this redundancy has a minimal impact on subsequent decision-making. Conversely, *HCGA w/o Rec* suffers significant win-rate reduction due to aggregated representations lacking environmental understanding without reconstruction loss.

Table.1 explores pooling layer impacts under fixed K'/K ratios, where d (HCGA's GNN layers) includes $d-1$ pooling layers and one aggregated network. Deeper pooling improves performance with more agents. When $d=1$, equivalent to direct aggregation from the initial graph.

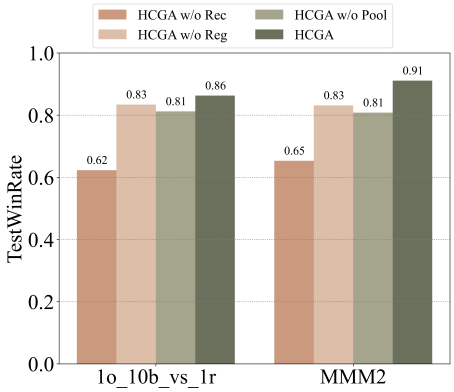


Figure 4: Ablation studies of HCGA modules.

Table 1: Performance under different hyperparameters.

Parameter	**1o_10b_vs_1r**	**MMM2**
$d=1$	81.2 (1.8)	80.8 (1.9)
$d=2, K'/K=0.5$	**86.3 (1.4)**	88.3 (2.4)
$d=3, K'/K=0.5$	84.6 (1.9)	**91.1 (2.3)**
$d=4, K'/K=0.5$	85.4 (1.6)	90.3 (2.5)

Conclusion

In this work, we delved into the challenge of cognition in complex MARL tasks. To achieve effective cognition in dynamic environments, we introduced HCGA, a novel framework that dynamically uncovers dependencies in received messages and enhances the efficiency of information integration. HCGA employs an innovative grouping-and-hierarchical cognitive pattern, which distills local information into a coherent global representation and enhances agents' adaptability in dynamic, complex environments. Experimental results demonstrate that HCGA significantly improves cooperative cognition among agents, thereby boosting cooperative performance across various MARL tasks.

Just as humans adjust their mental models and integrate information from social interactions to make informed decisions, HCGA enables agents to build flexible, context-sensitive cognitive representations through communication. This human-inspired perspective underscores the value of adaptive, socially grounded cognition in artificial multi-agent systems.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China under grants 62076232 and 62172049. Corresponding author: Wei Wang.

References

Andrychowicz, O. M., Baker, B., Chociej, M., Jozefowicz, R., McGrew, B., Pachocki, J., ... others (2020). Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, *39*(1), 3–20.

Boehmer, W., Kurin, V., & Whiteson, S. (2020). Deep coordination graphs. In *the 37th international conference on machine learning (ICML 2020), virtual event* (Vol. 119, pp. 980–991).

Chalnick, A., & Billman, D. (1988). Unsupervised learning of correlational structure. In *Proceedings of the tenth annual conference of the cognitive science society* (pp. 510–516). Hillsdale, NJ: Lawrence Erlbaum Associates.

Das, A., Gervet, T., Romoff, J., Batra, D., Parikh, D., Rabbat, M., & Pineau, J. (2019). Tarmac: Targeted multi-agent communication. In *International conference on machine learning* (pp. 1538–1546).

Duan, W., Lu, J., & Xuan, J. (2024). *Inferring latent temporal sparse coordination graph for multi-agent reinforcement learning.* Retrieved from https://arxiv.org/abs/2403.19253

Duan, W., Xuan, J., Qiao, M., & Lu, J. (2022). Learning from the dark: Boosting graph convolutional neural networks with diverse negative samples. In *the 36th AAAI conference on artificial intelligence (AAAI 2022), virtual event* (pp. 6550–6558). AAAI Press.

Feigenbaum, E. A. (1963). The simulation of verbal learning behavior. In E. A. Feigenbaum & J. Feldman (Eds.), *Computers and thought.* New York: McGraw-Hill.

Foerster, J. N., Farquhar, G., Afouras, T., Nardelli, N., & Whiteson, S. (2018). Counterfactual multi-agent policy gradients. In *the 32nd AAAI conference on artificial intelligence (AAAI 2018), new orleans, louisiana, usa* (pp. 2974–2982).

Hamilton, W. L., Ying, R., & Leskovec, J. (2018). *Inductive representation learning on large graphs.* Retrieved from https://arxiv.org/abs/1706.02216

Hill, J. A. C. (1983). A computational model of language acquisition in the two-year old. *Cognition and Brain Theory*, *6*, 287–317.

Kim, D., Moon, S., Hostallero, D., Kang, W. J., Lee, T., Son, K., & Yi, Y. (2019). Learning to schedule communication in multi-agent reinforcement learning. *arXiv preprint arXiv:1902.01554*.

Leurent, E. (2018). A survey of state-action representations for autonomous driving.

Li, S., Gupta, J. K., Morales, P., Allen, R. E., & Kochenderfer, M. J. (2021). Deep implicit coordination graphs for multi-agent reinforcement learning. In *the 20th international conference on autonomous agents and multiagent systems (AAMAS 2021), virtual event, united kingdom* (pp. 764–772).

Liu, Y., Wang, W., Hu, Y., Hao, J., Chen, X., & Gao, Y. (2020). Multi-agent game abstraction via graph attention neural network. In *the 34th AAAI conference on artificial intelligence (AAAI 2020), new york, ny, usa,* (pp. 7211–7218).

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., & Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. In *the 30th annual conference on neural information processing systems (NIPS 2017), long beach, ca, USA* (pp. 6379–6390).

Matlock, T. (2001). *How real is fictive motion?* Doctoral dissertation, Psychology Department, University of California, Santa Cruz.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.

Ohlsson, S., & Langley, P. (1985). *Identifying solution paths in cognitive diagnosis* (Tech. Rep. No. CMU-RI-TR-85-2). Pittsburgh, PA: Carnegie Mellon University, The Robotics Institute.

Oroojlooy, A., & Hajinezhad, D. (2023). A review of cooperative multi-agent deep reinforcement learning. *Appl. Intell.*, *53*(11), 13677–13722.

Osband, I., Blundell, C., Pritzel, A., & Van Roy, B. (2016). Deep exploration via bootstrapped dqn. In *Advances in neural information processing systems* (pp. 4026–4034).

Rashid, T., Samvelyan, M., De Witt, C. S., Farquhar, G., Foerster, J., & Whiteson, S. (2018). Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. *arXiv preprint arXiv:1803.11485*.

Samvelyan, M., Rashid, T., de Witt, C. S., Farquhar, G., Nardelli, N., Rudner, T. G., ... Whiteson, S. (2019). The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*.

Shrager, J., & Langley, P. (Eds.). (1990). *Computational models of scientific discovery and theory formation*. San Mateo, CA: Morgan Kaufmann.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., ... others (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, *362*(6419), 1140–1144.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., ... others (2017). Mastering the game of go without human knowledge. *nature*, *550*(7676), 354–359.

Singh, A., Jain, T., & Sukhbaatar, S. (2018). Learning when to communicate at scale in multiagent cooperative and competitive tasks. *arXiv preprint arXiv:1812.09755*.

Sumers, T. R., Hawkins, R. D., Ho, M. K., & Griffiths, T. L. (2021, may). Extending rational models of communication from beliefs to actions. *arXiv preprint arXiv:2105.11950*. Retrieved from https://arxiv.org/abs/2105.11950 (Proceedings of the 43rd Annual Meeting of the Cognitive Science Society) doi: 10.48550/arXiv.2105.11950

Sun, C., He, P., Ji, Q., Zang, Z., Li, J., Wang, R., & Wang, W. (2024, dec). M2i2: Learning efficient multi-agent communication via masked state modeling and intention inference. *arXiv preprint arXiv:2501.00312*. Retrieved from https://arxiv.org/abs/2501.00312

Sun, C., He, P., Wang, R., & Zheng, C. (2025, jan). Revisiting communication efficiency in multi-agent reinforcement learning from the dimensional analysis perspective. *arXiv preprint arXiv:2501.02888*. Retrieved from https://arxiv.org/abs/2501.02888 doi: 10.48550/arXiv.2501.02888

Sun, C., Wu, B., Wang, R., Hu, X., Yang, X., & Cong, C. (2021). Intrinsic motivated multi-agent communication. In *Proceedings of the 20th international conference on autonomous agents and multiagent systems* (p. 1668–1670). Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Sun, C., Zang, Z., Li, J., Li, J., Xu, X., Wang, R., & Zheng, C. (2024). T2mac: Targeted and trusted multi-agent communication through selective engagement and evidence-driven integration. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 38, pp. 15154–15163).

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). *Graph attention networks.* Retrieved from https://arxiv.org/abs/1710.10903

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., ... others (2019). Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, *575*(7782), 350–354.

Wang, T., Wang, J., Zheng, C., & Zhang, C. (2020). Learning nearly decomposable value functions via communication minimization. In *the 8th international conference on learning representations (ICLR 2020), addis ababa, ethiopia.*

Wang, T., Zeng, L., Dong, W., Yang, Q., Yu, Y., & Zhang, C. (2022). Context-aware sparse deep coordination graphs. In *the 10th international conference on learning representations (ICLR 2022), virtual event.*

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2021). A comprehensive survey on graph neural networks. *IEEE Trans. Neural Networks Learn. Syst.*, *32*(1), 4–24.

Yang, Q., Dong, W., Ren, Z., Wang, J., Wang, T., & Zhang, C. (2022). Self-organized polynomial-time coordination graphs. In *International conference on machine learning (ICML 2022), baltimore, maryland, USA* (Vol. 162, pp. 24963–24979).

Ying, R., You, J., Morris, C., Ren, X., Hamilton, W. L., & Leskovec, J. (2019). *Hierarchical graph representation learning with differentiable pooling.* Retrieved from https://arxiv.org/abs/1806.08804

Yuan, L., Wang, J., Zhang, F., Wang, C., Zhang, Z., Yu, Y., & Zhang, C. (2022). Multi-agent incentive communication via decentralized teammate modeling. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 36, pp. 9466–9474).

Zhang, S. Q., Zhang, Q., & Lin, J. (2019). Efficient communication in multi-agent reinforcement learning via variance based control. In *Advances in neural information processing systems* (pp. 3235–3244).

Zhang, S. Q., Zhang, Q., & Lin, J. (2020). Succinct and robust multi-agent communication with temporal message control. *Advances in Neural Information Processing Systems*, *33*, 17271–17282.