Group-Aware Coordination Graph for Multi-Agent Reinforcement Learning

Wei Duan , Jie Lu , Junyu Xuan
Australian Artificial Intelligence Institute (AAII), University of Technology Sydney
wei.duan@student.uts.edu.au, {jie.lu, junyu.xuan}@uts.edu.au

Abstract

Cooperative Multi-Agent Reinforcement Learning (MARL) necessitates seamless collaboration among agents, often represented by an underlying relation graph. Existing methods for learning this graph primarily focus on agent-pair relations, neglecting higher-order relationships. While several approaches attempt to extend cooperation modelling to encompass behaviour similarities within groups, they commonly fall short in concurrently learning the latent graph, thereby constraining the information exchange among partially observed agents. To overcome these limitations, we present a novel approach to infer the Group-Aware Coordination Graph (GACG), which is designed to capture both the cooperation between agent pairs based on current observations and group-level dependencies from behaviour patterns observed across trajectories. This graph is further used in graph convolution for information exchange between agents during decision-making. To further ensure behavioural consistency among agents within the same group, we introduce a group distance loss, which promotes group cohesion and encourages specialization between groups. Our evaluations, conducted on StarCraft II micromanagement tasks, demonstrate GACG's superior performance. An ablation study further provides experimental evidence of the effectiveness of each component of our method.

1 Introduction

Cooperative Multi-Agent Reinforcement Learning (MARL) requires agents to coordinate with each other to achieve collective goals [Cui *et al.*, 2020; Wang *et al.*, 2022a]. To address the challenges of the expansive action space posed by multi-agents [Orr and Dutta, 2023], a straightforward approach is breaking down the global training objective into manageable parts for each agent. Methods like VDN [Sunehag *et al.*, 2018], QMIX [Rashid *et al.*, 2018] empower individual agents to select actions that maximize their own value function, contributing to the overall reward maximization. Nevertheless, it is important to recognize that many real-world tasks necessitate complex agent interactions and are not readily decomposable into simpler, individual tasks. This realization underscores the need to depict agent relationships, often assumed to involve latent graph structures in MARL. As the graph is not explicitly given, inferring dynamic graph topology remains a significant and persistent challenge.

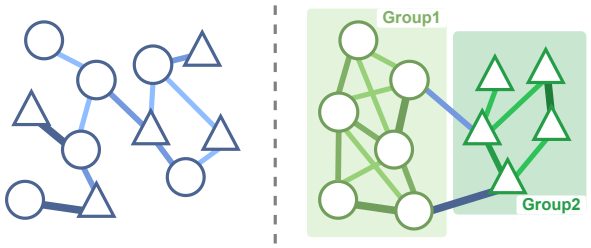

Figure 1: In a multi-agent environment, agents may exhibit diverse behaviours represented by triangles and circles. Existing methods for modelling agent interactions primarily focus on agent-pair relations. Concurrently recognizing the importance of higher-order group relationships among agents in coordination graphs is critical.

Current methods for learning the underlying graph in MARL can be categorized into three types: creating complete graphs by directly linking all nodes/agents [Liu *et al.*, 2020a; Boehmer *et al.*, 2020], employing attention mechanisms to calculate fully connected weighted graphs [Li *et al.*, 2021; Liu *et al.*, 2020b], and designing drop-edge criteria to generate sparse graphs [Yang *et al.*, 2022; Wang *et al.*, 2022b]. However, these methods focus exclusively on agent-pair relations when modelling interactions. In multi-agent scenarios, such as orchestrating multi-robot formations [Rizk *et al.*, 2019] or controlling a group of allies in strategic multi-agent combats [Samvelyan *et al.*, 2019b], relying solely on pairwise relations is inadequate for comprehensively understanding collaboration. A critical aspect often overlooked is the importance of higher-order relationships, including group relationships/dependencies. Advancements have recently emerged that utilize group division, aiming to explore value factorization for sub-teams [Phan *et al.*, 2021] or to specialize the character of the different group [Iqbal *et al.*, 2021]. Despite their efficacy in group partitioning, these methods do not concurrently learn the underlying graph structure. This limitation significantly affects the efficiency of information exchange among partially observed agents, which is vital for precise

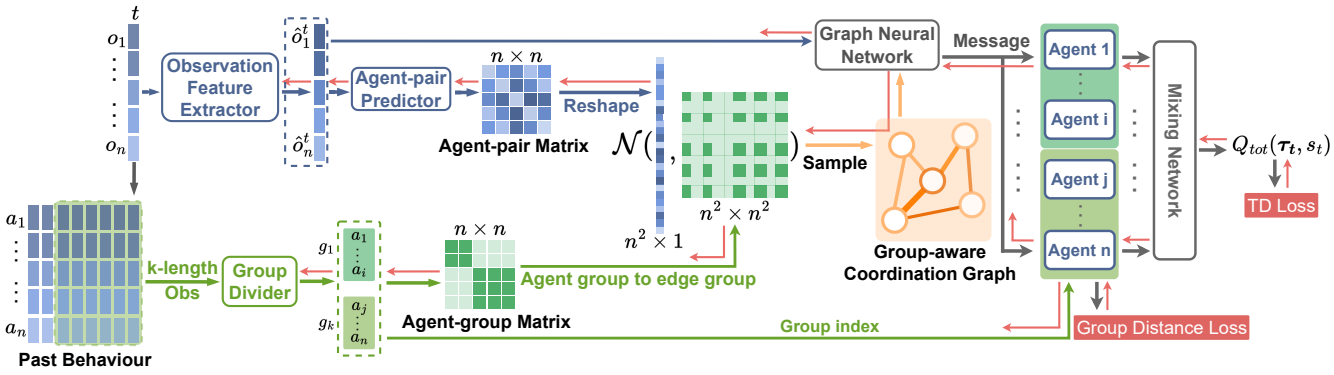

Figure 2: The framework of our method. GACG is designed to calculate cooperation needs between agent pairs based on current observations and to capture group-level dependencies from behaviour patterns observed across trajectories. All edges in the coordination graph are represented by a Gaussian distribution. This graph helps agents exchange knowledge when making decisions. During agent training, the group distance loss regularizes behaviour among agents with similar observation trajectories.

coordination and informed decision-making.

In light of these limitations, this paper proposes a novel approach to infer the Group-Aware Coordination Graph (GACG). GACG is designed to calculate cooperation needs between agent pairs based on current observations and to capture group dependencies from behaviour patterns observed across trajectories. A key aspect of our methodology is representing all edges in the coordination graph as a Gaussian distribution. This approach not only integrates agent-level interactions and group-level dependencies within a latent space but also transforms discrete agent connections into a continuous expression. Such a transformation is instrumental in modelling the uncertainty in various relationship levels, leading to a more informative and comprehensive representation of cooperation. Following this approach, the GACG is sampled from the distribution and used in graph convolution for information exchange between agents through a graph neural network during decision-making. To further ensure behavioural consistency among agents within the same group, we introduce a group distance loss. This loss function is designed to increase the differences in behaviour between groups while minimizing them within groups. By doing so, it promotes group cohesion and encourages specialization between groups.

Experimental evaluations on StarCraft micromanagement tasks demonstrate GACG's superior performance. Our ablation study provides experimental evidence for the effectiveness of each component of our method. The contributions of this paper are summarized as follows:

- We propose a novel MARL approach named the Group-Aware Coordination Graph (GACG). To the best of our knowledge, this is the first method to simultaneously calculate cooperation needs between agent pairs and capture group-level dependencies within a coordination graph.
- The edges of GACG are expressed as a Gaussian distribution, which models the uncertainty in various relationship levels, leading to a more informative and comprehensive representation of cooperation.
- We introduce the group distance loss to regularize behaviour among agents with similar observation trajectories, which enhances group cohesion and fosters distinct roles between different groups.

2 Related Work

The relationships among agents can be assumed to have latent graph structures [Tacchetti *et al.*, 2019; Li *et al.*, 2021]. Graph Convolutional Networks (GNN) have demonstrated remarkable capability in modelling relational dependencies [Wu *et al.*, 2021; Duan *et al.*, 2022; Duan *et al.*, 2024a], making graphs a compelling tool for facilitating information exchange among agents [Wang *et al.*, 2020b; Liu *et al.*, 2020b] and serving as coordination graphs during policy training [Boehmer *et al.*, 2020; Wang *et al.*, 2022b]. Although the current methods use different graph structures, such as a complete graph [Jiang *et al.*, 2020; Liu *et al.*, 2020a], weighted graph [Wang *et al.*, 2020a] and sparse graph [Yang *et al.*, 2022; Duan *et al.*, 2024b], they primarily focus on agent-pair relations for inferring graph topology, neglecting higher-order relationships among agents.

Moving beyond the individual agent level, attention to group development becomes pivotal for maintaining diversified policies and fostering efficient collaborations. ROMA [Wang *et al.*, 2020a] learns dynamic roles that depend on the context each agent observes. VAST [Phan *et al.*, 2021] approximates sub-group factorization for global reward to adapt to different situations. REFIL [Iqbal *et al.*, 2021] randomly group agents into related and unrelated groups, allowing exploration of specific entities. SOG [Shao *et al.*, 2022] selects conductors to construct groups temporally, featuring conductor-follower consensus with constrained communication between followers and their respective conductors. GoMARL [Zang *et al.*, 2023] uses a "select-and-kick-out" scheme to learn automatic grouping without domain knowledge for efficient cooperation. Despite their efficacy in group partitioning during training, these methods do not concurrently learn the underlying graph structure, thereby hindering the transfer of information within or between groups.

3 Background

We focus on cooperative multi-agent tasks modelled as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) [Oliehoek and Amato, 2016] consisting of a tuple $\langle \mathcal{A}, \mathcal{S}, \{\mathcal{U}_i\}_{i=1}^n, P, \{\mathcal{O}_i\}_{i=1}^n, \{\pi_i\}_{i=1}^n, R, \gamma \rangle$, where $\mathcal{A}$ is the finite set of n agents, $s \in \mathcal{S}$ is the true state of the environment. At each time step, each agent a_i observes the state partially by drawing observation $o_i^t \in \mathcal{O}^i$ and selects an action $u_i^t \in \mathcal{U}^i$ according to its own policy π_i. Individual actions form a joint action $\boldsymbol{u} = (u_1, ..., u_n)$, which leads to the next state s' according to the transition function $P(s'|s, \boldsymbol{u})$ and a reward $R(s, \boldsymbol{u})$ shared by all agents. This paper considers episodic tasks yielding episodes $(s^0, \{o_i^0\}_{i=1}^n, \boldsymbol{u}^0, r^0, ..., s^T, \{o_i^T\}_{i=1}^n)$ of varying finite length T. Agents learn to collectively maximize the global return $Q_{tot}(s, \boldsymbol{u}) = \mathbb{E}_{s_{0:T}, u_{0:T}} \left[\sum_{t=0}^{T} \gamma^t R\left(s^t, \boldsymbol{u}^t\right) \mid s^0 = s, \boldsymbol{u}^0 = \boldsymbol{u} \right]$, where $\gamma \in [0, 1)$ is the discount factor.

4 Method

The framework of our method is depicted in Fig. 2, which is designed to calculate cooperation needs between agent pairs based on current observations and to capture group-level dependencies from behaviour patterns observed across trajectories. This graph helps agents exchange knowledge when making decisions. During agent training, the group distance loss regularizes behaviour among agents with similar observation trajectories, which enhances group cohesion and encourages specialization between groups.

4.1 Group-Aware Coordination Graph Inference

Definition 1. *(Coordination graph (CG)). Given a cooperative task with n agents, the coordination graph is defined as $\mathcal{C} = \{\mathcal{A}, \mathcal{E}\}$, where $\mathcal{A} = \{a_1, \ldots, a_n\}$ are agents/node and $\mathcal{E} = \{e_{11}, \ldots, e_{nn}\}$ edges/relations between agent. $|\mathcal{E}| = n^2$ indicates the number of possible edges. CG can be written in an adjacent matrix form as C.*

To effectively capture the evolving importance of interactions between agents, our approach leverages two components: the observation feature extractor $f_{oe}(\cdot)$ and agent-pair predictor $f_{ap}(\cdot)$. These components are designed to extract hidden features from the current observations of all agents at time t and transform them into a meaningful structure we term the **agent-pair matrix**:

$$\hat{o}_i^t = f_{oe}(o_i^t), \quad \boldsymbol{\mu}_{ij}^t = f_{ap}(\hat{o}_i^t, \hat{o}_j^t), \tag{1}$$

where $f_{oe}(\cdot)$ is realized as a multi-layer perceptron (MLP), $f_{ap}(\cdot)$ is an attention network. The dimension of agent-pair matrix $\boldsymbol{\mu}^t$ is $n \times n$, and it represents the edge weights for each agent pair, indicating the importance of their interaction at time t.

In a multi-agent environment, understanding the dynamics solely through pairwise relations is insufficient. To address this, we introduce the group concept, allowing us to extract higher-level information for more informed cooperative strategies among agents. We define a group as:

Definition 2. *(Individual and Group). Given n agents $\mathcal{A}$, we have a set of groups $\mathcal{G} = \{g_1, \ldots, g_m\}, 1 \leq m \leq n$. Each group g_i contains n_i $(1 \leq n_i \leq n)$ different agents $g_i = \{a_1^i, \ldots, a_{n_i}^i\} \subseteq \mathcal{A}$, where $g_i \cap g_j = \emptyset, i \neq j, \cup_i g_i = \mathcal{A}$, and $i, j \in [1, m]$.*

Given the inherent partial observability in MARL, we assume that agents with similar observations over specific time periods are likely to encounter similar situations, leading to similar behaviours. Following this premise, we introduce the group divider model represented by $f_g : \mathcal{A} \mapsto \mathcal{G}$, which aims to capture similarities among agents based on behaviour patterns observed across trajectories:

$$\mathcal{G} = f_g(\mathcal{O}^{t-k:t}), \tag{2}$$

where $\mathcal{O}^{t-k:t}$ denotes the observations of all agents from time $t-k$ to t. The parameter k provides flexibility in determining the duration of the time steps, which allows us to choose the length of the trajectory. The reasons for using $\mathcal{O}^{t-k:t}$ to indicate trajectory behaviour are twofold: firstly, group dependencies are often observed over a time period, as in scenarios like coordinating a group of allies in an attack, where observations (such as facing the enemy) and behaviours (like attacking) tend to similar until the objective is achieved. Secondly, historical trajectory data has been shown to represent agents' behaviours more accurately than one-step observations [Pacchiano *et al.*, 2020], making it a more realistic and reliable source for our group divider.

Once groups are determined, we calculate the agent-group matrix $\mathbf{M}$ to indicate whether two agents belong to the same group at time t. This matrix is crucial for understanding group relations, which is defined as:

$$\mathbf{M}_{ij}^t = \begin{cases} 1 & \text{if } a_i, a_j \in g_m \text{ at time } t \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

To effectively calculate cooperation needs between agent pairs based on current observations and capture group dependencies from behaviour patterns observed across trajectories, we model all edges as a Gaussian distribution. Specifically, the mean values of this distribution indicate the importance of interactions between agent pairs, where a larger mean value indicates a stronger interaction. The covariance matrix of this distribution encapsulates the dependencies between edges. This feature becomes particularly crucial when agents are part of the same group, as it underscores a heightened level of dependence among them.

Building on this idea, we first convert the agent-group matrix into the edge-group matrix, enabling direct incorporation of group information into edge relationships. This is described by the following operation:

$$\hat{\mathbf{M}}^t = \text{vec}(\mathbf{M}^t) \times \text{vec}(\mathbf{M}^t)^\top, \quad \boldsymbol{\Sigma} = \hat{\mathbf{M}}, \tag{4}$$

where the shape of $\text{vec}(\mathbf{M}^t)$ is $n^2 \times 1$ (the number of possible edges) and $\hat{\mathbf{M}}^t$ is $n^2 \times n^2$. An element in $\hat{\mathbf{M}}^t$ being 1 means that the corresponding edges belong to the same group.

Definition 3. *(Edegs in the same group). Given two edge $e_{ij}, e_{lk} \in \mathcal{E}$, if $a_i, a_j, a_l, a_k \in g_m$, we assert that e_{ij} and e_{lk} in same group.*

Utilizing the **agent-pair matrix** $\boldsymbol{\mu}^t$ and **edge-group matrix** $\hat{\mathbf{M}}^t$, the Gaussian distribution is formally represented as:

$$\mathcal{E} \sim \mathcal{N}(\boldsymbol{\mu}^t, \hat{\mathbf{M}}^t), \tag{5}$$

where the shapes of $\boldsymbol{\mu}^t$ and $\hat{\mathbf{M}}^t$ are $n^2 \times 1$ and $n^2 \times n^2$, respectively. This approach not only provides a practical tool for capturing edge dependencies within the graph but also modelling the uncertainty in various agent relationship levels. For instance, considering two edges $e_i, e_j \in \mathcal{E}$, they follow the distribution (one property of multivariate Gaussian distribution):

$$(e_i, e_j) \sim \mathcal{N}\left((\mu_i^t, \mu_j^t), \begin{bmatrix} \hat{\mathbf{M}}_{ii}^t & \hat{\mathbf{M}}_{ij}^t \\ \hat{\mathbf{M}}_{ji}^t & \hat{\mathbf{M}}_{jj}^t \end{bmatrix}\right). \tag{6}$$

According to the above definitions, if e_i and e_j are in the same group (as indicated by $\hat{\mathbf{M}}_{ij}^t = 1$), they will exhibit high dependence, implying a closely aligned probability of their simultaneous occurrence or absence. Conversely, if the edges belong to different groups ($\hat{\mathbf{M}}_{ij}^t = 0$), their existences are modelled as independent. This approach aligns with the expectation that edges within the same group should exhibit stronger contextual dependencies, enhancing the model's capability to accurately represent and adapt to complex interactions among agents.

4.2 Group-Aware Cooperative MARL

Utilizing the Gaussian distribution defined in Eq.(5), we sample the edges of the Group-Aware Coordination Graph at each timestep, reshaping them into matrix form C^t. This sampled graph structure facilitates information exchange between agents through a graph neural network (GNN) [Duan *et al.*, 2023]. The GNN's message-passing mechanism is essential for agents to efficiently share and integrate information, adapting their strategies to the dynamic MARL environment. The GNN is defined as follows:

$$H_l^t = ReLU\left(\hat{C}^t H_{(l-1)}^t W_{(l-1)}\right), \tag{7}$$

where l is the index of GNN layers, $\hat{C}^t = \tilde{D}^{-\frac{1}{2}} C^t \tilde{D}^{-\frac{1}{2}}$, $\tilde{D}_{ii} = \sum_j C^t[i,j]$. The initial input of the GNN, H_t^0, is set to the extracted observation features $\{\hat{o}_1^t, ..., \hat{o}_n^t\}$ from Eq.(1). The output of GNN $m_i^t = H_l^t$ treated as exchanged knowledge between agents, which is then utilized in the local action-value function defined as $Q_i(\tau_i, \mu_j, m_i^t)$.

Building upon our earlier assumption that agents with similar trajectories are likely to exhibit similar behaviours, it becomes straightforward to regularize the behavioural consistency of agents during the policy training phase. This behaviour is reflected in the output of $\pi_i(\mu_i|\tau_i)$ (or Q_i), representing the probability distribution of actions for the current state. To assess the similarity of behaviour among agents, we compare their policy outputs and introduce the group distance loss, defined as:

$$\mathcal{L}_g = \frac{\frac{1}{(m-1)^2}\sum_{i\neq j}\left(\frac{1}{|g_i||g_j|}\sum_{a_l\in g_i}\sum_{a_k\in g_j}\|\pi_l - \pi_k\|_2\right)}{\frac{1}{m}\sum_i\left(\frac{1}{|g_i|^2}\sum_{a_l,a_v\in g_i}\|\pi_l - \pi_v\|_2\right)}. \tag{8}$$

Method	Graph type	Edge	Group
QMIX	×	×	×
DCG	Complete	Unweighted	×
DICG	Complete	Weighted	×
CASEC	Sparse	Weighted	×
VAST	×	×	√
GACG	**Sparse**	**Weighted**	√

Table 1: Comparison of different experiment methods in terms of graph type, edge representation, and group utilization.

This equation calculates the average pairwise behavioural distances between agents of different groups (numerator) and within the same group (denominator). By minimizing intra-group distances, this loss function promotes uniform behaviour within groups while maximizing inter-group distances to encourage diversity and specialization.

Our algorithm is built on top of 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:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{TD}(\boldsymbol{\theta}^-) + \lambda \mathcal{L}_g(\boldsymbol{\theta}_g), \tag{9}$$

where $\boldsymbol{\theta}$ includes all parameters in the model, λ is the weight of group distance loss. The TD loss $\mathcal{L}_{TD}(\boldsymbol{\theta}^-)$ is defined as

$$\mathcal{L}_{TD}(\boldsymbol{\theta}^-) = \left[r + \gamma \max_{\boldsymbol{a}'} Q_{tot}\left(s', \boldsymbol{\mu}'; \boldsymbol{\theta}'\right) - Q_{tot}(s, \boldsymbol{\mu}; \boldsymbol{\theta}^-)\right]^2, \tag{10}$$

where $\boldsymbol{\theta}'$ denotes the parameters of a periodically updated target network, as commonly employed in DQN.

5 Experiments

In this section, we design experiments to answer the following questions: (1) How well does GACG perform on complex cooperative multi-agent tasks compared with other state-of-the-art CG-based methods? (2) Is the choice and calculation method for the Gaussian distribution promising for sampling edges in CG? (3) Does the inclusion of the group loss improve GACG's performance? (4) What is the influence of group number on the GACG performance? (5) How does the selected length of trajectory affect the final result?

The experiments in this study are conducted using the StarCraft II benchmark [Samvelyan *et al.*, 2019a], which offers complex micromanagement tasks with varying maps and different numbers of agents. The benchmark includes scenarios with a minimum of eight agents, encompassing both homogeneous and heterogeneous agent setups. The environments are configured with a difficulty level of 7. The experiments are systematically carried out with 5 random seeds to ensure robustness and reliability in the assessment of the proposed methods. The code is available at: *https://github.com/Wei9711/GACG*

5.1 Compared with Other CG-Based Methods

We compare our methods with the following baselines, and each method's graph type, edge representation, and group utilization are summarised in Tab.1.

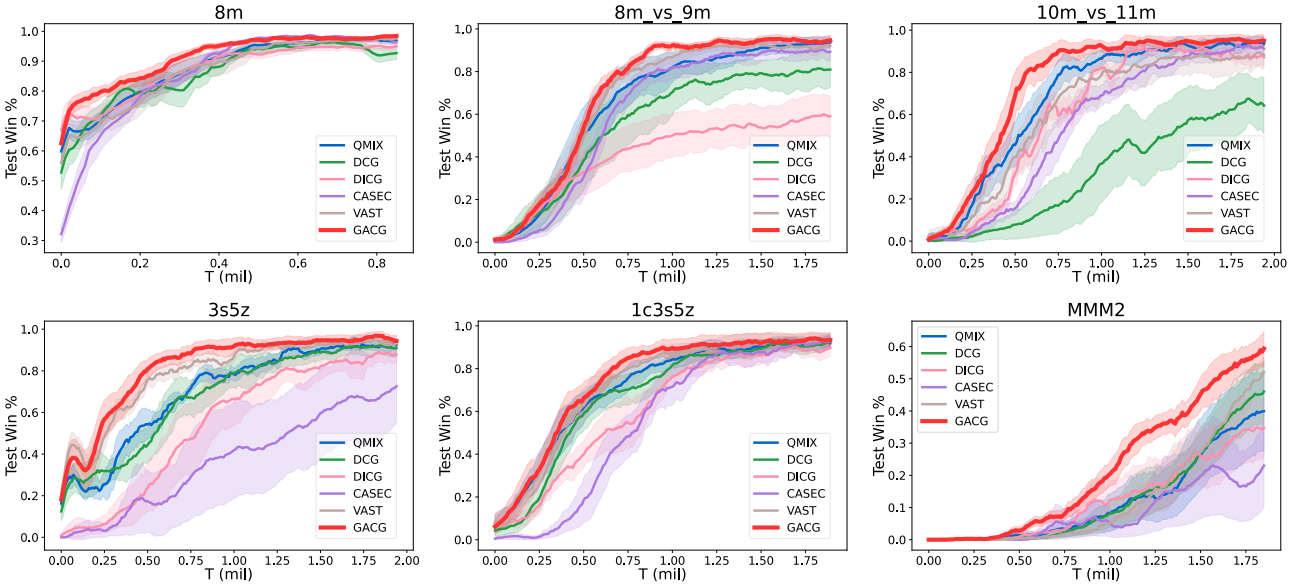

Figure 3: Performance of GACG and baselines on six maps of the SMAC. The x-axis represents the time steps (in millions), while the y-axis quantifies the test win rate in the games..

- **QMIX**[1] [Rashid *et al.*, 2018] is effective but without cooperation between agents, also without group division.
- **DCG** [2][Boehmer *et al.*, 2020] directly links all the edges to get an unweighted fully connected graph. The graph is used to calculate the action-pair values function.
- **DICG**[3] [Li *et al.*, 2021] uses attention mechanisms to calculate weighted fully connected graph. The graph is used for information passing between agents.
- **CASEC**[4] [Wang *et al.*, 2022b] drop edges on the weighted fully connected graph using the variance payoff function.
- **VAST** [5][Phan *et al.*, 2021] explores value factorization for sub-teams based on a predetermined group number. The sub-team values are linearly decomposed for all sub-team members.

Results

In Figure 3, we present the comprehensive results of our experiments conducted across six diverse maps, highlighting the superior performance of our Group-Aware Coordination Graph (GACG) method. GACG consistently achieves high win rates with rapid convergence and reliability, outperforming competing methods in complex multi-agent scenarios. On the *8m* and *1c3s5z* maps, all methods demonstrate similar convergence patterns. They reach their highest test win rates at the end of 2 million training steps. However, performance disparities become more evident on other maps. For instance, DICG exhibits the weakest performance on the *8m_vs_9m* map, struggling to match the effectiveness of other methods. Similarly, DCG falls behind on the *10m_vs_11m* map, where the competing approaches outperform it. On the challenging 3s5z map, CASEC shows a lower win rate, suggesting that its strategy is less suited to the intricacies of this particular scenario.

The limitations of comparative methods are apparent: QMIX lacks graph structures, DCG treats all interactions uniformly without weight considerations, DICG misses group dynamics, and CASEC, despite addressing message redundancy, overlooks the importance of group-level behaviour. VAST, while exploring sub-team dynamics, does not utilize dynamic graph structures. GACG's nuanced approach—leveraging agent pair cooperation and group-level dependencies—affords a deeper understanding of agent interactions. The group distance loss embedded in training sharpens within-group behaviour and enhances inter-group strategic diversity, contributing to GACG's effectiveness.

In summary, the comparative analysis validates that a sophisticated approach to graph learning, attentive to pairwise and group-level information, is instrumental in achieving superior performance in MARL. With its innovative graph sampling technique and the seamless incorporation of group dynamics, the Group-Aware Coordination Graph method offers more intelligent and adaptable cooperative learning algorithms.

Computational Complexity Analysis

The computational complexity of our model is primarily influenced by the necessity to discern group relationships among agents. Given that interactions are represented as edges in a fully connected graph, an environment with n agents results in n^2 pairwise edges. To capture group dy-

[1]https://github.com/oxwhirl/pymarl
[2]https://github.com/wendelinboehmer/dcg
[3]https://github.com/sisl/DICG
[4]https://github.com/TonghanWang/CASEC-MACO-benchmark
[5]https://github.com/thomyphan/scalable-marl

	$1k$ steps time (s)	$1m$ steps time (h)
QMIX	20.13 ± 3.59	6.79 ± 0.37
DCG	33.57 ± 4.65	11.63 ± 0.64
CASEC	30.50 ± 2.03	10.12 ± 0.51
VAST	21.28 ± 3.59	6.21 ± 0.56
GACG	22.33 ± 4.22	7.84 ± 0.49

Table 2: Time computational consumption on map *10m_vs_11m*.

namics, an edge-group matrix $\hat{\mathbf{M}}^t$ is constructed to represent these relationships, with dimensions $n^2 \times n^2$. Consequently, the time complexity for computing the group matrix is $O(n^4)$.

While the theoretical computational complexity for computing the group matrix in our model seems high, this computation is mainly attributed to the multiplication operation $\hat{\mathbf{M}}^t = \text{vec}(\mathbf{M}^t) \times \text{vec}(\mathbf{M}^t)^\top$ in Eq.(4). Importantly, this operation is highly amenable to parallelization, a task at which GPUs excel, substantially accelerating the actual training process. Tab.2 presents empirical running times for our model on the *10m_vs_11m* map, demonstrating that despite the theoretically high complexity, our GACG method is competitive in practice. In fact, the running times for GACG are faster or on par with other graph-based methods.

5.2 Ablation Study

Different Edge Distributions

In this part, we experiment on two maps aiming to provide insights into the unique aspects of the Gaussian distribution in GACG and its effectiveness in capturing agent-level and group-level information for cooperative multi-agent tasks. We substitute this part with other distributions. The detailed settings are shown below:

- **Attention (without distribution):** This approach uses no specific distribution for the coordination graph but learns the edges directly from the agent-pair matrix, where edges are obtained as $e_{ij}^t = f_{ap}(\hat{o}_i^t, \hat{o}_j^t)$.
- **Bernoulli:** The distribution is changed to a Bernoulli distribution, where the agent-pair matrix serves as the probability of this distribution, expressed as $P_B(e_{ij} = 1) = f_{ap}(\hat{o}_i^t, \hat{o}_j^t)$.
- **Inde-Gaussian:** Each edge is considered independent and follows a Gaussian distribution. Each element in the agent-pair matrix serves as mean values for corresponding edges, formulated as $e_{ij} \sim \mathcal{N}(\boldsymbol{\mu}_{ij}, \sigma^2)$, where $0 \leq \sigma^2 \leq 1$. This setting investigates the impact of group-level dependence when designing the multivariate Gaussian distribution.
- **GACG (w/o $\mathcal{L}_g$):** The final loss function is without the inclusion of the group distance loss $\mathcal{L}_g$. This ensures that the only difference between the compared methods is distribution.

The result is shown in Fig.4. Across various maps, our GACG consistently outperforms other settings, confirming the efficacy of our method. In the *3s5z* map, regardless of the distribution used, treating the learning of the graph as the learning of the edge distribution is more effective than utilizing attention alone (without distribution). However, in the *10m_vs_11m* map, the use of the Bernoulli distribution performs worse than attention, indicating that the choice of distribution is not arbitrary. This observation underscores the importance of carefully selecting the distribution method in constructing the coordination graph.

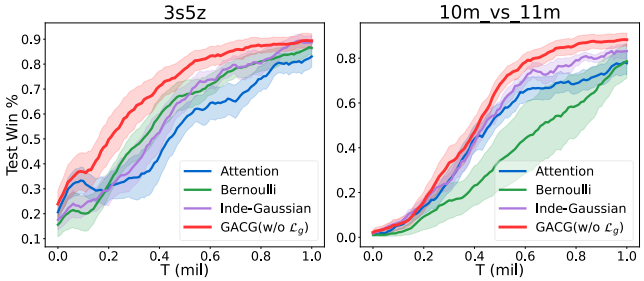

Figure 4: Experiment of choosing different edge distributions when learning the CG.

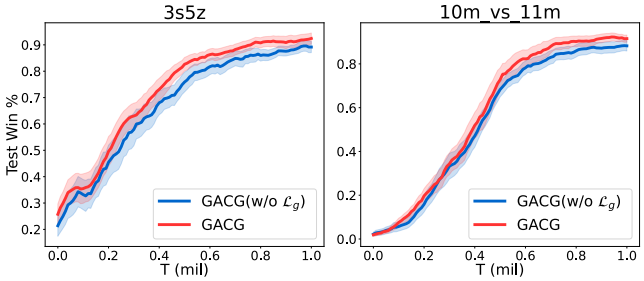

Figure 5: Experiment of training GACG with/without $\mathcal{L}_g$.

When compared with the setting where each edge is considered independent and follows a Gaussian distribution, our method yields better results. This finding emphasizes the importance and effectiveness of capturing group dependency when learning the edge distribution. The ability to model dependencies among agents at the group level contributes significantly to the improved performance of our approach.

Effectiveness of Group Distance Loss

In this part, we test the effectiveness of group distance loss by training the GACG with and without $\mathcal{L}_g$.

The results are shown in Fig.5, revealing several key findings: (1) GACG trained with $\mathcal{L}_g$ exhibits a faster convergence speed and achieves a higher average final performance compared to its counterpart trained without $\mathcal{L}_g$. This observation strongly suggests that $\mathcal{L}_g$ effectively guides the model towards more efficient and cooperative learning. (2) This component contributes to enlarging inter-group distances while concurrently decreasing intra-group distances, fostering effective agent cooperation. (3) This result highlights the essential role that group-level information plays in enhancing the overall effectiveness and cooperation capabilities of GACG, demonstrating the significance of considering both agent-level and group-level dynamics for optimal performance.

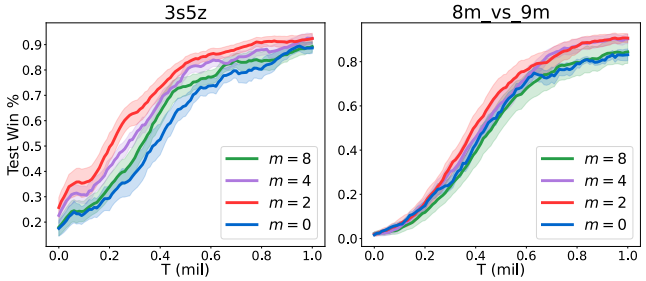

Figure 6: Experiment of dividing n agents into different numbers of groups(m) on *3s5z* and *8m_vs_9m*. The former has two types of agents, while the latter consists of a single type.

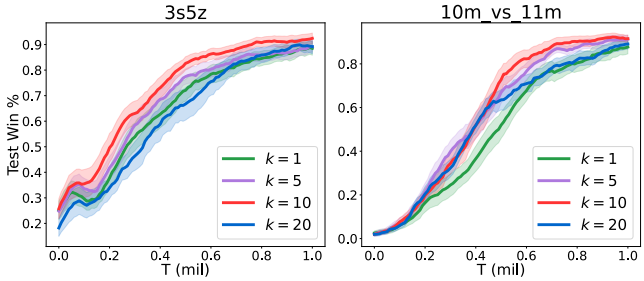

Figure 7: Experiment of varying observation window lengths (k) on the group divider model f_g.

Number of Groups

We further investigate the impact of varying the number of groups (m) on two distinct maps: *3s5z* and *8m_vs_9m*, each featuring 8 agents. The former has two types of agents, while the latter consists of a single type. The experiment is conducted with m set to $\{0, 2, 4, 8\}$.

When $m = 0$, it implies the absence of group divisions between agents, indicating that no group-level information is utilized during training (including graph reasoning and group loss calculation). Conversely, when $m = 8$, each agent is treated as an individual group. In this setting, while there are no group divisions, the presence of the group distance loss encourages each agent to exhibit diverse behaviours and dissimilarity.

The results are illustrated in Fig. 6. Optimal performance is observed when $m = 2$, underscoring the beneficial impact of a moderate level of group division on cooperation. Across both maps, the introduction of group divisions ($m \in 2, 4$) consistently outperforms scenarios where there are either no group divisions or a high number of them ($m \in 0, 8$). This emphasizes the crucial role of incorporating group-level information in achieving superior MARL outcomes. Notably, in map *3s5z*, where two agent types are present, the convergence speed of $m = 8$ is faster than that of $m = 0$. This acceleration is likely due to the importance of policy diversity in a multi-agent setting with distinct agent types, and the group distance loss facilitates each agent in achieving diverse behaviours.

Length of Trajectory for Group Division

In this analysis, our objective is to investigate the influence of varying observation trajectory lengths, parameterized by k, on the effectiveness of the group divider model f_g. We explore different values for k, specifically $\{1, 5, 10, 20\}$. When $k = 1$, a single timestep is considered, whereas $k \in \{5, 10, 20\}$ enables the assessment of longer temporal relationships.

The results are depicted in Fig. 7. Across both maps, the performance data indicates that a moderate observation window length of $k = 10$ yields the highest test win percentage. This setting strikes a balance, providing agents with sufficient historical data to discern meaningful patterns and group relationships. Conversely, the $k = 1$ setting, representing the shortest observation window, fails to deliver adequate historical data for effective group differentiation and decision-making, resulting in a lower test win percentage. As we increase the window length from $k = 1$ to $k = 5$, more information becomes available for group division, yet it does not surpass the performance achieved with $k = 10$.

An interesting observation arises when $k = 20$, the overall performance decreases and approaches that of $k = 1$. This phenomenon suggests a point of diminishing returns, where additional historical information may not contribute to better decision-making. This could be attributed to challenges such as overfitting or an inability to adapt quickly to new information. Therefore, selecting an optimal observation window, such as $k = 10$, allows agents to integrate just enough temporal information. This enables adaptation to dynamic environments without the drawbacks associated with processing excessive or potentially noisy data.

6 Conclusion

In this paper, we have presented the Group-Aware Coordination Graph (GACG), a novel MARL framework that addresses the limitations of existing approaches. Unlike previous methods that handle agent-pair relations and dynamic group division separately, GACG seamlessly unifies the integration of pairwise interactions and group-level dependencies. It adeptly computes cooperation needs from one-step observations while capturing group behaviours across trajectories. Employing the graph's structure for information exchange during agent decision-making significantly enhances collaborative strategies. Incorporating a group distance loss during training enhances behavioural similarity within groups and encourages specialization across groups. Our extensive experimental evaluations reveal that our method consistently outperforms current leading methods. An ablation study confirms the efficacy of each individual component, highlighting the importance of incorporating both pairwise and group-level insights into the learning model. The outcomes of this research emphasize the importance of multi-level agent information integration, establishing our framework as a substantial contribution to advancing MARL.

Acknowledgements

This work is supported by the Australian Research Council under Australian Laureate Fellowships FL190100149 and Discovery Early Career Researcher Award DE200100245.

References

[Boehmer *et al.*, 2020] Wendelin Boehmer, Vitaly Kurin, and Shimon Whiteson. Deep coordination graphs. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020), Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 980–991. PMLR, 2020.

[Cui *et al.*, 2020] Jingjing Cui, Yuanwei Liu, and Arumugam Nallanathan. Multi-agent reinforcement learning-based resource allocation for UAV networks. *IEEE Trans. Wirel. Commun.*, 19(2):729–743, 2020.

[Duan *et al.*, 2022] Wei Duan, Junyu Xuan, Maoying Qiao, and Jie Lu. Learning from the dark: Boosting graph convolutional neural networks with diverse negative samples. In *Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI 2022), Virtual Event*, pages 6550–6558. AAAI Press, 2022.

[Duan *et al.*, 2023] Wei Duan, Junyu Xuan, Maoying Qiao, and Jie Lu. Graph convolutional neural networks with diverse negative samples via decomposed determinant point processes. *IEEE Trans. Neural Networks Learn. Syst.*, 2023.

[Duan *et al.*, 2024a] Wei Duan, Jie Lu, Yu Guang Wang, and Junyu Xuan. Layer-diverse negative sampling for graph neural networks. *Transactions on Machine Learning Research*, 2024.

[Duan *et al.*, 2024b] Wei Duan, Jie Lu, and Junyu Xuan. Inferring latent temporal sparse coordination graph for multi-agent reinforcement learning. *CoRR*, abs/2403.19253, 2024.

[Iqbal *et al.*, 2021] Shariq Iqbal, Christian A. Schröder de Witt, Bei Peng, Wendelin Boehmer, Shimon Whiteson, and Fei Sha. Randomized entity-wise factorization for multi-agent reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning (ICML 2021), 18-24 July, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 4596–4606. PMLR, 2021.

[Jiang *et al.*, 2020] Jiechuan Jiang, Chen Dun, Tiejun Huang, and Zongqing Lu. Graph convolutional reinforcement learning. In *8th International Conference on Learning Representations (ICLR 2020), Addis Ababa, Ethiopia*, 2020.

[Li *et al.*, 2021] Sheng Li, Jayesh K. Gupta, Peter Morales, Ross E. Allen, and Mykel J. Kochenderfer. Deep implicit coordination graphs for multi-agent reinforcement learning. In *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Virtual Event, United Kingdom*, pages 764–772. ACM, 2021.

[Liu *et al.*, 2020a] Iou-Jen Liu, Raymond A Yeh, and Alexander G Schwing. Pic: permutation invariant critic for multi-agent deep reinforcement learning. In *Proceedings of the 3rd Conference on Robot Learning (CoRL 2019), Osaka, Japan*, pages 590–602, 2020.

[Liu *et al.*, 2020b] Yong Liu, Weixun Wang, Yujing Hu, Jianye Hao, Xingguo Chen, and Yang Gao. Multi-agent game abstraction via graph attention neural network. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2020), New York, NY, USA,*, pages 7211–7218. AAAI Press, 2020.

[Oliehoek and Amato, 2016] Frans A. Oliehoek and Christopher Amato. *A Concise Introduction to Decentralized POMDPs*. Springer Briefs in Intelligent Systems. Springer, 2016.

[Orr and Dutta, 2023] James Orr and Ayan Dutta. Multi-agent deep reinforcement learning for multi-robot applications: A survey. *Sensors*, 23(7):3625, 2023.

[Pacchiano *et al.*, 2020] Aldo Pacchiano, Jack Parker-Holder, Yunhao Tang, Krzysztof Choromanski, Anna Choromanska, and Michael Jordan. Learning to score behaviors for guided policy optimization. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning, (ICML 2020)*, volume 119 of *Proceedings of Machine Learning Research*, pages 7445–7454, 13–18 Jul 2020.

[Phan *et al.*, 2021] Thomy Phan, Fabian Ritz, Lenz Belzner, Philipp Altmann, Thomas Gabor, and Claudia Linnhoff-Popien. VAST: value function factorization with variable agent sub-teams. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems (NIPS 2021), December 6-14, virtual*, pages 24018–24032, 2021.

[Rashid *et al.*, 2018] Tabish Rashid, Mikayel Samvelyan, Christian Schröder de Witt, Gregory Farquhar, Jakob N. Foerster, and Shimon Whiteson. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML 2018), Stockholmsmässan, Stockholm, Sweden*, volume 80, pages 4292–4301, 2018.

[Rizk *et al.*, 2019] Yara Rizk, Mariette Awad, and Edward W. Tunstel. Cooperative heterogeneous multi-robot systems: A survey. *ACM Comput. Surv.*, 52(2):29:1–29:31, 2019.

[Samvelyan *et al.*, 2019a] Mikayel Samvelyan, Tabish Rashid, Christian Schröder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob N. Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2019), Montreal, QC, Canada,*, pages 2186–2188. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

[Samvelyan *et al.*, 2019b] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philiph H. S. Torr, Jakob Foerster, and Shimon Whiteson. The StarCraft Multi-Agent Challenge. *CoRR*, abs/1902.04043, 2019.

[Shao *et al.*, 2022] Jianzhun Shao, Zhiqiang Lou, Hongchang Zhang, Yuhang Jiang, Shuncheng He, and Xiangyang Ji. Self-organized group for cooperative multi-agent reinforcement learning. In *NeurIPS*, 2022.

[Sunehag *et al.*, 2018] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinícius Flores Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2018), Stockholm, Sweden*, pages 2085–2087, 2018.

[Tacchetti *et al.*, 2019] Andrea Tacchetti, H. Francis Song, Pedro A. M. Mediano, Vinícius Flores Zambaldi, János Kramár, Neil C. Rabinowitz, Thore Graepel, Matthew M. Botvinick, and Peter W. Battaglia. Relational forward models for multi-agent learning. In *7th International Conference on Learning Representations (ICLR 2019), New Orleans, LA, USA*, 2019.

[Wang *et al.*, 2020a] Tonghan Wang, Heng Dong, Victor R. Lesser, and Chongjie Zhang. ROMA: multi-agent reinforcement learning with emergent roles. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020), Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9876–9886, 2020.

[Wang *et al.*, 2020b] Tonghan Wang, Jianhao Wang, Chongyi Zheng, and Chongjie Zhang. Learning nearly decomposable value functions via communication minimization. In *8th International Conference on Learning Representations (ICLR 2020), Addis Ababa, Ethiopia*, 2020.

[Wang *et al.*, 2022a] Min Wang, Libing Wu, Jianxin Li, and Liu He. Traffic signal control with reinforcement learning based on region-aware cooperative strategy. *IEEE Trans. Intell. Transp. Syst.*, 23(7):6774–6785, 2022.

[Wang *et al.*, 2022b] Tonghan Wang, Liang Zeng, Weijun Dong, Qianlan Yang, Yang Yu, and Chongjie Zhang. Context-aware sparse deep coordination graphs. In *The Tenth International Conference on Learning Representations (ICLR 2022), Virtual Event*. OpenReview.net, 2022.

[Wu *et al.*, 2021] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Networks Learn. Syst.*, 32(1):4–24, 2021.

[Yang *et al.*, 2022] Qianlan Yang, Weijun Dong, Zhizhou Ren, Jianhao Wang, Tonghan Wang, and Chongjie Zhang. Self-organized polynomial-time coordination graphs. In *International Conference on Machine Learning (ICML 2022), Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 24963–24979. PMLR, 2022.

[Zang *et al.*, 2023] Yifan Zang, Jinmin He, Kai Li, Haobo Fu, QIANG FU, Junliang Xing, and Jian Cheng. Automatic grouping for efficient cooperative multi-agent reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems, (NIPS 2023)*, 2023.