From Local to Global: Progressive Consensus via Hierarchical Communication in Multi-Agent Reinforcement Learning

Jiangjin Yin[1], Zhonglin Lv[1], Hangyu Mao[2], Rongbo Zhu[1],
Yiwei Shi[3], Kun Hu[4], Xianfang Tang[5], Zhiwei Xu[6]

[1]College of Informatics, Huazhong Agricultural University, Key Laboratory of Smart Farming for Agricultural Animals, Hubei Engineering Technology Research Center of Agricultural Big Data, and Engineering Research Center of Intelligent Technology for Agriculture
[2]AI Center, Institute of Microelectronics, Chinese Academy of Sciences
[3]University of Bristol
[4]College of Computer Science and Technology, National University of Defense Technology
[5]School of Computer Science and Artificial Intelligence, Wuhan Textile University
[6] School of Artificial Intelligence, Shandong University

Abstract

Effective team collaboration hinges on consensus formation through information exchange, a principle equally critical in multi-agent reinforcement learning (MARL). However, existing communication-based and consensus-learning methods often struggle to form a coherent global understanding from agents' limited local views. We propose STAGE, a local-to-global progressive consensus framework based on hierarchical communication. By first grouping agents according to their perceptual focuses, STAGE enables intra-group communication to form local consensuses, then selects group leaders to exchange these consensuses across groups, progressively expanding them into a unified global consensus with reduced communication redundancy. To further improve consensus quality, we introduce KL-divergence constraints for consensus alignment and a variational autoencoder (VAE) objective for preserving task-relevant global information. Extensive experiments on challenging MARL benchmarks show that STAGE consistently outperforms state-of-the-art baselines, especially on more difficult tasks and larger-scale multi-agent systems. https://github.com/hzau-yjj/STAGE.

1 Introduction

Social psychology highlights the critical role of consensus formation in fostering team cooperation, as individuals leverage communication to align their understanding and strategies [1]. Empirical observations further suggest that such communication is more effective when organized progressively. For example, in a town hall setting, residents first deliberate within local neighborhoods to form partial agreements, after which representatives negotiate to reach a broader consensus. Compared to simultaneous global discussion, this process reduces communication overload and improves agreement efficiency. Inspired by this insight, we propose a progressive consensus paradigm based on hierarchical communication to enhance the cooperation in Multi-Agent Reinforcement Learning (MARL) [6, 21, 38].

Although both explicit multi-agent communication and implicit consensus methods have been studied in MARL, they reveal limitations in achieving a global consensus. Communication-based approaches [26, 33, 39] typically focus on acquiring additional information beneficial for decision-making to mitigate partial observability issues, overlooking whether a consistent consensus is formed among agents. In contrast, implicit consensus methods [37, 41] generally aim to derive a consistent consensus

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

through agents' local observations. However, deriving a complex consensus solely through local observations is highly challenging, particularly when the observation information is severely limited. Taking Figure 1 as an example, communication-based methods either reconstruct only a partial structure or form an incorrect one (Figure 1(b)). Similarly, implicit consensus methods attempt to reconstruct the entire structure from a single agent's perspective, which is evidently highly challenging (Figure 1(c)).

This paper presents a two-stage communication framework that progressively transforms local consensus into global consensus. First, agents communicate within dynamically formed groups to establish local consensus. And then, group leaders exchange these local consensuses across groups, refining them into a unified global consensus. This local-to-global consensus formation enables agents to integrate information from multiple levels—individual experiences, group-level local consensus, and system-level global consensus—into the decision-making process, empowering them to make comprehensive decisions. Refer to Figure 1(d), our framework effectively guides agents in completing the 3D reconstruction task.

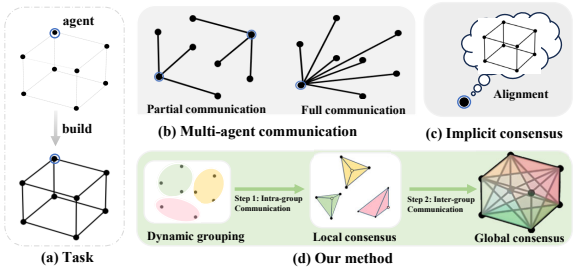


Figure 1: (a) 3D Reconstruction task, where each agent only observes its own position and is required to build the cube structure. (b)–(d) Sketches of different methods.

However, several key challenges must be addressed to ensure the effectiveness of the proposed framework. First, how to effectively group agents and enable efficient inter-group communication. Drawing from the human tendency for like-minded individuals to group together, we introduce a grouping strategy that unites agents with similar focus on local observations. Since members within a group share similar local consensus, selecting a leader for inter-group communication is often sufficient. Our strategy efficiently identifies group leaders, allowing only inter-leader exchanges to form global consensus, which significantly reduces communication overhead. Second, how to allow the leader to convey the global consensus, formed through inter-group communication, to team members. This is achieved by applying a KL divergence constraint that aligns each member's consensus distribution with the leader's. Third, how to preserve the validity of the global consensus. To this end, we employ a Variational Autoencoder (VAE) [4] to reconstruct the environment state from the global consensus representation. By minimizing the reconstruction loss, the model encourages the global consensus to preserve task-relevant information about the environment state.

We evaluate the proposed method through extensive experiments on the Cooperative Navigation, StarCraft Multi-Agent Challenge (SMAC and SMACv2), and Battle scenarios, which are widely used benchmarks for MARL. Results show our approach significantly outperforms existing methods, especially on harder tasks and larger-scale multi-agent coordination.

2 Related Work

MARL. MARL involves multiple agents interacting with each other to complete tasks. In these settings, the environment is inherently non-stationary from the viewpoint of an individual agent, due to the continual learning and adaptation of other agents. To mitigate this challenge, one popular paradigm is Centralized Training with Decentralized Execution (CTDE) [22, 27, 34], which allows agents to leverage global information during training while maintaining the ability to act independently based on local observations during execution. Although CTDE has driven significant progress, its reliance on limited local views during execution often impairs coordination and leads to suboptimal performance.

Multi-agent Communication. Multi-agent communication enables agents to share information during execution to support better cooperation [7, 12, 19, 36]. As a pioneering approach, CommNet [32] adopts a fully connected communication structure, where each agent aggregates the mean information of all other agents. TarMAC [5] improves communication efficiency by introducing a soft attention mechanism for message aggregation. ATOC [15] and Mean Field [40] adopt partial communication frameworks based on the assumption that agents primarily interact with nearby peers. DGN [16]

models the multi-agent system as a dynamic graph and utilizes multi-head attention as a convolution kernel to capture neighbor interactions and enhance cooperative behaviors. Similarly, LSC [31] adopts a hierarchical graph neural network (GNN) to generate the communication messages, and TGCNet [44] learns the topological structure of a dynamic directed graph to represent the communication policy. GACG [9] learns a group-aware coordination graph that captures pairwise agent relations and uses the graph for GNN-based information exchange. LTS-CG [10] further infers latent temporal sparse coordination graphs from historical observations, enabling dynamic modeling of agent dependencies and reducing redundant communication. CommFormer [14] represents communication as a learnable graph within a bi-level optimization framework, where the upper level refines the graph structure and the lower level updates model parameters, with both processes alternating iteratively to improve performance. Gated-ACML [24] and CACOM [17] incorporate gating mechanisms to address bandwidth limitations. MAIC [42] and MASIA [13] focus on generating personalized information for individual agents. Yet, these methods primarily focus on information sharing rather than consensus building, which may lead to inconsistent actions among agents and degrade team performance.

Implicit Consensus. Implicit consensus methods aim to infer a shared consensus from the local observations of individual agents [30, 37], where the consensus represents agents' unified insight on the environmental state. NCC [23] introduces the concept of neighborhood consistency into MARL, enabling agents in the same neighborhood to form a shared understanding of the environment. COLA [37] uses a consensus builder to extract shared understanding from local observations, employing a student-teacher architecture where the student learns from teacher-generated pseudo-labels via knowledge distillation. CoS [28] utilizes vectorized variational autoencoders to group agents and extracts group consensus. COCOM [18] leverages global state information to constrain the consensus derived solely from individual agent observations, thereby enabling the emergence of more meaningful and coherent consensus among agents. However, directly building the consensus from agents' local observations may become challenging particularly when agents have significantly disparate observations. Unlike treating communication and consensus separately, this paper combines the advantages of both and designs a hierarchical communication framework that progressively lifts agent local consensuses to a unified global one that agents can efficiently comprehend and maintain.

3 Preliminaries

We consider a fully cooperative multi-agent setting modeled as a decentralized partially observable Markov decision process (DEC-POMDP) [3, 20], defined by the tuple $\langle N, S, \boldsymbol{A}, T, R, \boldsymbol{O}, Z, \gamma \rangle$. Here, N is the set of agents; S is the global state space; $\boldsymbol{A} = A_1 \times \cdots \times A_N$ is the joint action space, where A_i is the action space of agent i; $T(s' \mid s, \boldsymbol{a}) : S \times \boldsymbol{A} \times S \to [0, 1]$ is the state transition function; $\boldsymbol{O} = O_1 \times \cdots \times O_N$ is the joint observation space, where each agent i receives a local observation $o_i \in O_i$ according to the observation function $Z(\boldsymbol{o} \mid s, \boldsymbol{a}) : \boldsymbol{O} \times S \times \boldsymbol{A} \to [0, 1]$; $R(s, \boldsymbol{a}) : S \times \boldsymbol{A} \to \mathbb{R}$ is the shared reward function; and $\gamma \in [0, 1)$ is the discount factor. At each timestep t, each agent selects an action a_i^t based on its local observation, forming a joint action $\boldsymbol{a}^t = (a_1^t, \ldots, a_N^t)$, which induces a transition to s^{t+1} and yields a shared reward r^t. The objective is to learn a joint decentralized policy $\pi = \{\pi_i\}_{i=1}^N$, maximizing the expected discounted return $J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^H \gamma^t r^t\right]$, where H is the time horizon.

Partial observability in MARL restricts each agent to decentralized decision-making, complicating credit assignment. To address this, value decomposition methods, such as VDN [34] and QMIX [27], decompose the global action-value function Q_{total} into individual utilities Q_i and combine them via a mixing function f: $Q_{\text{total}} = f(Q_1, \ldots, Q_N)$. These methods follow the Individual-Global Max (IGM) principle to align individual and global optimality: $\forall i \in N, \arg\max_{a_i} Q_i(o_i, a_i) \in \arg\max_{\boldsymbol{a}} Q_{\text{total}}(\boldsymbol{o}, \boldsymbol{a})$. The model is typically trained by minimizing the temporal difference (TD) error:

$$\mathcal{L}_{\text{td}} = \left(r + \gamma \max_{\boldsymbol{a}'} Q_{\text{total}}^-(\boldsymbol{o}', \boldsymbol{a}') - Q_{\text{total}}(\boldsymbol{o}, \boldsymbol{a})\right)^2. \tag{1}$$

Here, Q_{total}^- is a periodically updated target network of Q_{total}, r is the reward, and $\boldsymbol{o}'$ and $\boldsymbol{a}'$ denote the next-step joint observations and candidate joint actions, respectively.

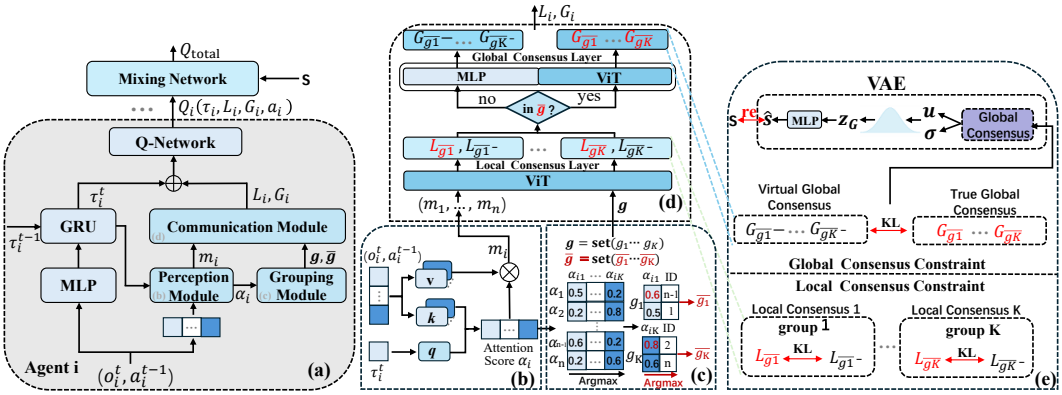


Figure 2: a) The structure of STAGE, b) Perception module, c) Grouping module, d) Communication module, e) Consensus Constraint.

4 Method

We propose STAGE, a hierarchical communication framework to progressively promote global agent consensus in MARL. As shown in Figure 2, STAGE consists of four modules: an experience-driven perception module, a perceptual-focus grouping module, a hierarchical communication module, and a consensus constraint module. At each timestep, agent i maintains an experience τ_i^t through a GRU, which serves as a compact memory of its accumulated interaction history $(o_i,\ a_i)$ and provides contextual guidance for subsequent perception and communication. Given the experience τ_i^t and current observation o_i^t, the experience-driven perception module leverages an attention mechanism to produce the communication message m_i^t and the attention scores α_i. Then, the perceptual-focus grouping module takes α_i to dynamically partition agents into communication groups $\boldsymbol{g} = \{g_1, g_2, \ldots\}$ and selects a representative leader set $\overline{\boldsymbol{g}}$. After that, the hierarchical communication module forms local consensus L_i via intra-group message exchange, then integrates them into global consensus G_i via inter-group leader communication. Finally, each agent combines its experience τ_i^t, local consensus L_i, and global consensus G_i to estimate its individual utility Q_i. To regularize the learned consensus, we introduce a consensus constraint module with a consistency term and a validity term. The former aligns local and global consensus distributions while the latter encourages the global consensus to preserve task-relevant state information.

4.1 Experience-Driven Perception Module

Each agent maintains an experience τ_i^t, updated by a GRU from its current observation o_i^t, previous action a_i^{t-1}, and previous experience τ_i^{t-1}. This experience summarizes local interaction history and guides task-relevant information extraction. Given o_i^t and τ_i^t, this module generates the communication message m_i^t via an attention mechanism [35]. [1] Specifically, we divide the observation o_i^t into K semantic components ($K \leq 4$): the agent's own feature o_i^{own}, partner feature o_i^{par}, adversary feature o_i^{adv}, and environmental feature o_i^{env}. [2] Each component is mapped to a corresponding key vector k_j and value vector v_j, and the experience τ_i^t is mapped to the query vector q_i. The attention coefficients $\alpha_i = \{\alpha_{i1}, \ldots, \alpha_{iK}\}$ are computed as:

$$\alpha_{ij} = \frac{\exp(q_i^\top k_j)}{\sum_{l=1}^{K} \exp(q_i^\top k_l)}. \tag{2}$$

The communication message is then obtained by a weighted sum over the value vectors as $m_i^t = \sum_{j=1}^{K} \alpha_{ij} v_j$. Here, m_i^t represents the agent's compact task-relevant perception, α_i reflects its focus over different observation components.

[1] Using multiple attention heads allows agents to perceive observation components from different perspectives and can dynamically expand the maximum number of communication groups.

[2] If the observation does not contain all four components, the missing ones are padded with zeros.

4.2 Perceptual-Focus Grouping Module

This module forms communication groups $\boldsymbol{g} = \{g_1, g_2, \ldots\}$ based on agents' perceptual focuses and selects the corresponding group leaders $\overline{\boldsymbol{g}} = \{\overline{g_1}, \overline{g_2}, \ldots\}$, where $\overline{g_k}$ denotes the leader of group g_k.

The attention scores α_i characterize the perceptual focus of agent i over different semantic components of its observation o_i^t. Agents focusing on the same semantic components are likely to possess complementary information about related aspects of the environment, making them suitable for forming local consensuses through intra-group communication. Although the similarity between attention vectors can be measured by cosine similarity or Euclidean distance, we adopt a simpler yet effective strategy: each agent is assigned to the group corresponding to the dimension with its highest attention weight. That is, agents i and j belong to the same group if $\arg\max(\alpha_i) = \arg\max(\alpha_j)$. Therefore, we obtain $\boldsymbol{g} = \{g_1, g_2, \ldots, g_K\}$ as follows,

$$g_k = \{i \mid \arg\max(\alpha_i) = k,\ i \in \{1, 2, \ldots, n\}\}. \tag{3}$$

Within each group, the agent with the highest attention weight on the corresponding semantic component is selected as the leader, as it is expected to provide the most informative representation for inter-group consensus formation. Thus, the leader of group g_k is defined as

$$\overline{g_k} = \arg\max_{i \in g_k} \alpha_i[k], k \in \{1, \ldots, K\}. \tag{4}$$

4.3 Hierarchical Communication Module

The hierarchical communication module progressively transforms local information into team-level consensus through two stages: the *Local Consensus Layer* and the *Global Consensus Layer*.

Local Consensus Layer: Agents within each group aggregate messages through a Vision Transformer (ViT) model [2, 8] to obtain local consensus representations:

$$[L_i]_{i \in g_k} = \mathrm{ViT}_{\mathrm{local}}\left([m_i]_{i \in g_k}\right). \tag{5}$$

Collecting local consensuses from all agents gives $\boldsymbol{L} = [L_1, \ldots, L_n]$. For each group g_k, the selected leader forwards its local consensus $L_{\overline{g_k}}$ for subsequent inter-group communication.

Global Consensus Layer: After establishing local consensuses, group leaders communicate with each other through another ViT:

$$[G_{\overline{g_1}}, \ldots, G_{\overline{g_K}}] = \mathrm{ViT}_{\mathrm{global}}\left([L_{\overline{g_1}}, \ldots, L_{\overline{g_K}}]\right). \tag{6}$$

Through inter-leader communication, each leader obtains a global consensus representation that incorporates information from all groups.

Non-leader agents do not directly participate in inter-group communication. To provide them with an approximation of the global consensus, each non-leader independently applies an MLP to its own local consensus representation. We refer to this locally predicted representation as a *virtual global consensus*:

$$G_i^{\mathrm{virtual}} = \mathrm{MLP}(L_i), \qquad i \notin \overline{\boldsymbol{g}}. \tag{7}$$

During training, the global consistency constraint encourages alignment between these predictions and the global-consensus representations obtained by the leaders through communication. We denote each agent's communicated or predicted representation by G_i, and their collection by $\boldsymbol{G} = [G_1, \ldots, G_n]$.

Each agent combines its experience τ_i^t, local consensus L_i, and global consensus G_i to compute its individual utility $Q_i(\tau_i^t, L_i, G_i, a_i; \theta_Q)$. During centralized training, the mixing network aggregates individual utilities with the global state to obtain $Q_{\mathrm{total}}(s, \boldsymbol{\tau}^t, \boldsymbol{L}, \boldsymbol{G}, \boldsymbol{a})$.

4.4 Consensus Constraint

The consensus constraint includes two parts: a consistency term and a validity term. The consistency term aligns agents' local and global consensuses by penalizing the KL divergence among local consensuses and among global ones separately. The validity constraint enriches agents' global consensus with task-relevant environment information by training a VAE model to reconstruct the global state s from it.

Consistency Constraint: We group the agents based on the similarity of attention weights, which facilitates the formation of consistent local and global consensus. However, this mechanism alone cannot guarantee their consistent formation.

To further ensure consistency of the local consensuses within the same group, we minimize the following KL divergence for each agent i:

$$\min\{\frac{1}{|g_k|}\sum_{j\in g_k} KL\left(p(L_i|o_i)||p(L_j|o_j)\right)\}, \tag{8}$$

where $p(L_i|o_i)$ and $p(L_j|o_j)$ represent the distributions obtained by transforming L_i and L_j using the softmax function, respectively. Here, $|g_k|$ denotes the number of agents in group k, and $i, j \in g_k$. Similarly, to ensure that the global consensus is shared by all agents, we minimize the KL divergence between each agent and all other agents:

$$\min\{\frac{1}{N-1}\sum\nolimits_{j=1}^{N} KL\left(p(G_i|o_i)||p(G_j|o_j)\right)\}, \tag{9}$$

where N is the total number of agents, and $j \in \{1,\ldots,N\} \setminus \{i\}$.

Validity Constraint: The consistency constraint promotes alignment among agents' global consensus representations, while the validity constraint encourages these representations to preserve information about the environment state. To this end, we adopt a VAE to reconstruct the global state s from a latent representation derived from the global consensus G_i. The reconstruction objective encourages G_i to retain state-relevant information, complementing the cross-agent alignment promoted by the consistency constraint.

The VAE model includes two parts: VAE encoder and VAE decoder. In our approach, the encoder generates the global consensus embedding z_G from the input state s, while the decoder reconstructs the state $\hat{s}$ based on z_G using a deep neural network. Here, z_G is sampled from the variational posterior distribution $q_\theta(z_G|s)$, whose mean μ and variance σ are produced by G_i and can be viewed as the low-dimensional representation of G_i. Note that in our method, the forward propagation process that input $\boldsymbol{o} = \{o_1, \cdots, o_n\}$ to obtain the output z_G is treated as the VAE encoder. The joint observation $\boldsymbol{o}$ is considered equivalent to the state s [22], resulting in the approximation $q_\theta(z_G|\boldsymbol{o}) \approx q_\theta(z_G|s)$, where θ represents the network parameters of the agents.

Specifically, given a state s, there exists a true posterior distribution $p(z_G|s)$ over z_G. According to Bayes' theorem, it can be calculated as

$$p(z_G|s) = \frac{p(s|z_G)p(z_G)}{p(s)}. \tag{10}$$

Considering that calculating the marginal probability distribution $p(s)$ is extremely complicated, we adopt a tactic of learning a variational posterior distribution $q_\theta(z_G|s)$ to approximate the true posterior distribution $p(z_G|s)$. By minimizing the following KL

$$\min\left(KL\left(q_\theta(z_G|s)||p(z_G|s)\right)\right), \tag{11}$$

we ensure the $q_\theta(z_G|s)$ is close to $p(z_G|s)$, which enables the VAE encoder to capture the core features of the global state s accurately. By deriving the Eq. (11), we get

$$\begin{aligned}\log p(s) &= KL(q_\theta(z_G \mid s) \,\|\, p(z_G \mid s)) + \sum_{z_G} q_\theta(z_G \mid s) \log\left(\frac{p(s, z_G)}{q_\theta(z_G \mid s)}\right) \\ &= KL(q_\theta(z_G \mid s) \,\|\, p(z_G \mid s)) + \text{ELBO}.\end{aligned} \tag{12}$$

Since $\log p(s)$, the log-likelihood of the state s, is a constant and the KL divergence is non-negative, maximizing the evidence lower bound (ELBO) is equivalent to minimizing the KL divergence in Eq. (11), which is

$$\begin{aligned}\text{ELBO} &= \mathbb{E}_{q_\theta(z_G|s)}\left[\log p(s|z_G)\right] - KL\left(q_\theta(z_G|s)||p(z_G)\right) \\ &= -\mathcal{L}^{re}\left(s, \widehat{s}; \theta\right) - KL\left(q_\theta(z_G|s)||p(z_G)\right).\end{aligned} \tag{13}$$

Here, the first term in the ELBO characterizes the reconstruction likelihood of the VAE decoder, whereas the second term regularizes the variational posterior distribution $q_\theta(z_G|s)$ to align with the true prior distribution $p(z_G)$. Maximizing the ELBO therefore encourages the global consensus G_i to retain information relevant to reconstructing the state s.

Table 1: Comparison of different algorithms across various tasks.

Environment	Task	Difficulty	**STAGE**	LTSCG	TGCNet	CommFormer	COLA	LSC	DGN	QMIX	TarMAC	GACG	NCC	Steps
Cooperative Navigation	CN_7agent	Easy	**-38.2**$_{(1.8)}$	-39.4$_{(4.5)}$	-50.5$_{(3.9)}$	-133.8$_{(8.0)}$	-77.4$_{(2.0)}$	-72.1$_{(2.8)}$	-120.1$_{(3.6)}$	-169.2$_{(6.2)}$	-39.4$_{(1.3)}$	-57.5$_{(2.6)}$	-48.5$_{(12.2)}$	1×10^6
	CN_15agent	Hard	**-73.3**$_{(3.4)}$	-154.4$_{(5.1)}$	-131.6$_{(1.9)}$	-207.5$_{(6.7)}$	-140.4$_{(5.9)}$	-126.1$_{(4.9)}$	-224.4$_{(6.5)}$	-254.0$_{(10.5)}$	-148.6$_{(10.7)}$	-115.5$_{(3.0)}$	-221.3$_{(20.6)}$	1×10^6
SMAC	MMM	Easy	**97.4**$_{(2.5)}$	96.0$_{(2.8)}$	96.8$_{(2.2)}$	21.6$_{(7.1)}$	91.0$_{(5.2)}$	90.0$_{(4.3)}$	48.2$_{(7.1)}$	42.4$_{(5.4)}$	91.9$_{(4.6)}$	93.4$_{(3.1)}$	0.0$_{(0.0)}$	1×10^6
	3s5z	Easy	**96.3**$_{(1.9)}$	95.5$_{(1.3)}$	80.4$_{(5.5)}$	26.9$_{(10.2)}$	94.6$_{(3.0)}$	33.5$_{(4.7)}$	0.1$_{(0.4)}$	50.9$_{(4.9)}$	39.3$_{(11.0)}$	95.0$_{(1.7)}$	0.0$_{(0.0)}$	2×10^6
	5m_vs_6m	Hard	**71.0**$_{(4.9)}$	70.8$_{(12.4)}$	47.6$_{(10.2)}$	1.7$_{(1.5)}$	61.2$_{(6.0)}$	53.7$_{(6.6)}$	0.4$_{(0.8)}$	58.8$_{(6.3)}$	14.3$_{(5.7)}$	67.9$_{(4.2)}$	0.5$_{(2.1)}$	2×10^6
	10m_vs_11m	Hard	**95.8**$_{(3.3)}$	95.8$_{(4.2)}$	93.0$_{(4.3)}$	10.4$_{(3.7)}$	95.4$_{(2.6)}$	91.3$_{(3.6)}$	0.8$_{(0.9)}$	66.6$_{(11.1)}$	11.9$_{(7.4)}$	93.5$_{(3.1)}$	0.0$_{(0.0)}$	2×10^6
	3s_vs_5z	Hard	88.6$_{(3.5)}$	85.0$_{(12.0)}$	**89.9**$_{(9.6)}$	0.0$_{(0.0)}$	45.5$_{(3.0)}$	69.4$_{(5.0)}$	0.4$_{(0.8)}$	78.6$_{(6.6)}$	31.2$_{(5.5)}$	79.8$_{(6.4)}$	0.0$_{(0.0)}$	2×10^6
	MMM2	Hard	**93.0**$_{(4.9)}$	90.5$_{(6.2)}$	88.0$_{(0.2)}$	1.8$_{(2.7)}$	89.0$_{(5.1)}$	41.1$_{(10.7)}$	10.5$_{(4.5)}$	68.1$_{(10.6)}$	37.1$_{(10.6)}$	83.4$_{(3.9)}$	0.0$_{(0.0)}$	3×10^6
	3s5z_vs_3s6z	Hard+	**68.2**$_{(6.4)}$	0.3$_{(1.3)}$	63.8$_{(6.7)}$	0.1$_{(0.4)}$	2.6$_{(2.1)}$	0.0$_{(0.0)}$	1.0$_{(1.1)}$	0.0$_{(0.0)}$	0.0$_{(0.0)}$	5.5$_{(2.5)}$	0.0$_{(0.0)}$	3×10^6
	6h_vs_8z	Hard+	**65.8**$_{(8.9)}$	1.8$_{(3.6)}$	0.0$_{(0.0)}$	0.0$_{(0.0)}$	0.0$_{(0.0)}$	0.0$_{(0.0)}$	0.0$_{(0.0)}$	0.2$_{(0.4)}$	0.6$_{(0.6)}$	0.6$_{(0.6)}$	0.0$_{(0.0)}$	3×10^6
SMACv2	Protoss5v5	Hard	**70.2**$_{(3.9)}$	62.8$_{(5.1)}$	38.3$_{(4.8)}$	25.8$_{(6.8)}$	61.1$_{(5.3)}$	48.0$_{(6.8)}$	19.0$_{(0.2)}$	57.0$_{(6.0)}$	51.0$_{(6.0)}$	61.0$_{(0.8)}$	18.5$_{(9.6)}$	5×10^6
	Terran5v5	Hard	**64.8**$_{(5.0)}$	62.8$_{(3.4)}$	60.3$_{(7.3)}$	28.3$_{(7.8)}$	63.5$_{(5.2)}$	55.5$_{(10.8)}$	27.3$_{(0.5)}$	61.0$_{(6.0)}$	55.7$_{(8.9)}$	64.2$_{(13.8)}$	10.4$_{(13.2)}$	5×10^6
	Protoss10v10	Hard	**61.5**$_{(6.3)}$	58.3$_{(3.3)}$	33.0$_{(4.3)}$	3.7$_{(3.0)}$	53.9$_{(6.2)}$	39.0$_{(8.9)}$	13.0$_{(0.3)}$	41.7$_{(4.9)}$	35.8$_{(7.1)}$	19.5$_{(12.3)}$	0.0$_{(0.0)}$	5×10^6
	Terran10v10	Hard	62.8$_{(5.0)}$	62.5$_{(3.8)}$	59.0$_{(6.4)}$	5.8$_{(3.5)}$	57.6$_{(5.2)}$	47.5$_{(10.8)}$	23.5$_{(0.1)}$	53.8$_{(6.0)}$	54.8$_{(10.2)}$	**67.5**$_{(8.1)}$	2.0$_{(4.0)}$	5×10^6
	Protoss5v6	Hard+	**12.8**$_{(7.5)}$	9.7$_{(5.2)}$	4.0$_{(2.6)}$	2.2$_{(1.7)}$	11.8$_{(1.9)}$	9.5$_{(3.5)}$	0.5$_{(0.0)}$	10.0$_{(3.4)}$	8.2$_{(2.8)}$	9.6$_{(5.7)}$	0.0$_{(0.0)}$	5×10^6
	Terran5v6	Hard+	**26.8**$_{(7.6)}$	18.2$_{(8.6)}$	9.8$_{(2.0)}$	3.3$_{(1.1)}$	23.2$_{(5.2)}$	12.5$_{(7.8)}$	1.7$_{(0.7)}$	21.2$_{(4.8)}$	16.3$_{(6.0)}$	20.3$_{(9.9)}$	1.5$_{(4.8)}$	5×10^6
	Protoss10v11	Hard+	**25.2**$_{(5.9)}$	20.5$_{(5.1)}$	10.3$_{(4.1)}$	0.0$_{(0.0)}$	19.9$_{(5.6)}$	14.0$_{(6.6)}$	1.7$_{(0.0)}$	23.7$_{(4.8)}$	14.3$_{(2.6)}$	20.8$_{(7.6)}$	0.0$_{(0.0)}$	5×10^6
	Terran10v11	Hard+	38.5$_{(8.3)}$	**38.8**$_{(6.7)}$	27.0$_{(4.4)}$	0.5$_{(1.0)}$	38.0$_{(3.7)}$	23.0$_{(9.5)}$	5.3$_{(0.1)}$	35.2$_{(7.0)}$	28.3$_{(10.0)}$	35.5$_{(8.2)}$	0.0$_{(0.0)}$	5×10^6

4.5 Training

The training of STAGE involves minimizing three loss functions: TD-loss, KL loss, and VAE loss. For simplicity, all parameters associated with agents are denoted by θ. The TD-loss, shared among all agents, is defined as

$$\mathcal{L}^{\text{td}}(\theta) = \mathbb{E}_{(s,\boldsymbol{o},\boldsymbol{a},r,\boldsymbol{o}')\sim D}\left[(y_{\text{total}} - Q_{\text{total}}(s,\boldsymbol{o},\boldsymbol{L},\boldsymbol{G},\boldsymbol{a};\theta))^2\right], \tag{14}$$

where $y_{\text{total}} = r + \gamma \max_{\boldsymbol{a}'} Q_{\text{total}}(s',\boldsymbol{o}',\boldsymbol{L}',\boldsymbol{G}',\boldsymbol{a}';\theta^-)$, θ^- is the parameter of a target network as in DQN [25], and D is the replay buffer that stores the recent experience tuples $(s,\boldsymbol{o},\boldsymbol{a},r,\boldsymbol{o}',s')$. According to Equations (8) and (9), the KL loss for each agent i is calculated as

$$\mathcal{L}^{\text{kl}}(\theta) = \mathbb{E}_{\boldsymbol{o}\sim D}\left[\frac{1}{|g_k|}\sum_{j\in g_k}\text{KL}(p(L_i \mid o_i;\theta)\,\|\,p(L_j \mid o_j)) + \frac{1}{N-1}\sum_{\substack{j=1\\ j\neq i}}^{N}\text{KL}(p(G_i \mid o_i)\,\|\,p(G_j \mid o_j))\right]. \tag{15}$$

The VAE is optimized by maximizing the ELBO; thus, the VAE loss for each agent i is

$$\mathcal{L}^{\text{vae}}(\theta) = \mathbb{E}_{\boldsymbol{o}\sim D}[-\text{ELBO}] = \mathbb{E}_{\boldsymbol{o}\sim D}\left[KL\left(q_\theta(z_G|s)\|p(z_G)\right) + \mathcal{L}^{re}(s,\widehat{s};\theta)\right]. \tag{16}$$

Finally, we combine the Equations (14), (15), (16) to obtain the total loss:

$$\mathcal{L}^{\text{total}} = \mathcal{L}^{\text{td}} + \alpha\sum_{i=1}^{N}\mathcal{L}^{\text{kl}} + \beta\sum_{i=1}^{N}\mathcal{L}^{\text{vae}}. \tag{17}$$

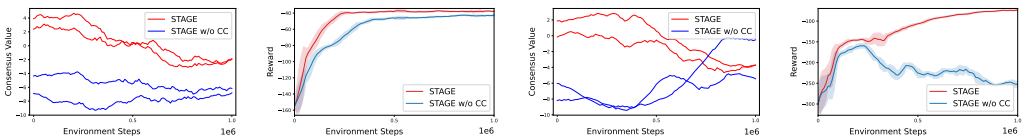
5 Evaluation

We evaluate our STAGE method in Cooperative Navigation (CN) [23], SMAC [29], SMACv2 [11], and Battle [45] scenarios, with various numbers of agents and different task difficulties. We compare STAGE with 10 baselines, all of which use their official implementations and original hyperparameters to ensure fairness. To ensure the reliability of the results, we report the average performance over five runs with different random seeds. Detailed settings and additional results are provided in the supplementary material.

5.1 Performance on various scenarios

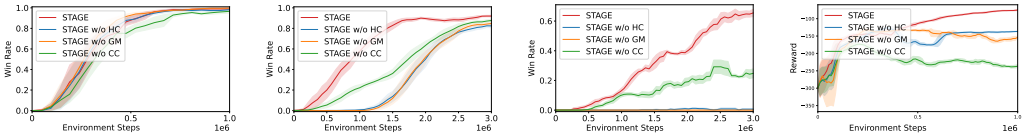
Table 1 shows the performance of all methods in the CN, SMAC, and SMACv2 scenarios.

CN scenario: In the CN benchmark, equal numbers of agents and landmarks are deployed. Agents aim to cover as many landmarks as possible without collisions. Evaluations are conducted on 7-agent (simple) and 15-agent (hard) settings. The results show that QMIX, as the most basic method without communication or implicit consensus alignment, performs the worst. DGN, LSC, TGCNet, GACG, and LTS-CG introduce the graph-based communication, improving over QMIX, but their limited communication range restricts performance. COLA and NCC are able to build some degree of



(a) *CN_7agent* (b) *CN_7agent* (c) *CN_15agent* (d) *CN_15agent*

Figure 3: Results of STAGE and *STAGE w/o CC* on the CN task. The consensus value is computed from the agents' high-dimensional consensus representations. (a) Consensus value and (b) mean reward in *CN_7agent*; (c) consensus value and (d) mean reward in the *CN_15agent*.



(a) *MMM* (b) *MMM2* (c) *6h_vs_8z* (d) *CN_15agent*

Figure 4: Ablation study results on tasks with different levels of difficulty. (a) *MMM* task (simple), (b) *MMM2* task (hard), and (c) *6h_vs_8z* task (super-hard). (d) *CN_15agent* represents the hard task in the CN scenario.

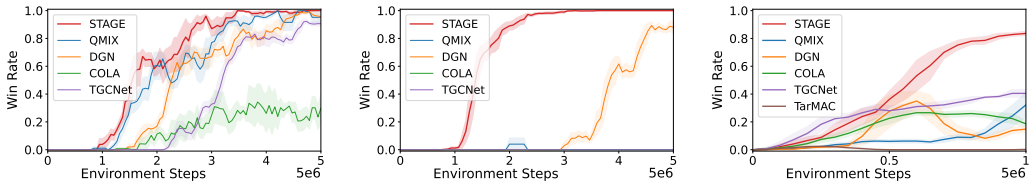
consensus from agent local observations and outperform both QMIX and DGN. However, their lack of a global perspective prevents them from achieving optimal coordination, especially in more complex tasks. TarMAC could obtain global information with its global communication mechanism and perform well in the 7-agent task. However, when faced with the 15-agent setting, which demands more complex global consensus and coordination, its performance significantly degrades. In contrast, STAGE builds global consensus from local ones via hierarchical communication, enabling a more coherent environmental understanding and consistently outperforming all baselines.

SMAC scenario: SMAC is a StarCraft II micromanagement benchmark where partially observable agents control allies to defeat rule-based enemies. In Table 1, STAGE outperforms most baselines across all tasks. As the task difficulty increases, STAGE demonstrates even greater performance gains compared to existing baselines. Notably, in the super-hard *6h_vs_8z* and *3s5z_vs_3s6z* tasks, where almost all baselines fail to perform, STAGE still achieves a win rate of over 65%. This superior performance stems from STAGE's ability to derive task-relevant global consensus through hierarchical local-to-global communication and consensus alignment.

SMACv2 scenario: Compared to SMAC, SMACv2 introduces greater diversity and stochasticity, demanding stronger generalization and adaptability from MARL algorithms. Consequently, nearly all methods show lower win rates on SMACv2. Nevertheless, STAGE achieves the highest average win rate across all tasks, demonstrating its robust and adaptive coordination in highly dynamic scenarios.

5.2 Consensus Visualization

We investigate whether STAGE can establish a consistent global consensus and whether such a consensus can enhance cooperative performance. Figure 3 presents the evolution of global consensus



(a) *Battle-20* (b) *Battle-64* (c) *M50v50*

Figure 5: Performance comparison in large-scale scenarios. (a) *Battle-20*, (b) *Battle-64*, and (c) *M50v50*. Note that CommFormer and GACG are excluded due to out-of-memory (OOM) issues on the GPU.

values and mean rewards over training steps in both *CN_7agent* and *CN_15agent* tasks, where *STAGE w/o CC* refers to the variant of STAGE without the consensus constraint module. As shown in Figure 3(a) and (c), agents trained with STAGE progressively align their consensus values, eventually achieving full convergence. This alignment correlates with a continuous increase in cooperative reward, as depicted in Figure 3(b) and (d). In contrast, *STAGE w/o CC* fails to reach a unified consensus, particularly in the more complex 15-agent setting where the consensus values remain highly divergent. Notably, in the simpler 7-agent task, even without consensus constraints, *STAGE w/o CC* still achieves relatively high rewards. This is because, in less complex environments, agents can more easily coordinate through basic communication, leading to naturally closer consensus values. However, as the number of agents increases, reaching consensus purely through communication becomes substantially harder. In the *CN_15agent* task, the loss of consensus constraints leads to severe divergence in agent consensus and a corresponding drop in performance. These results highlight that STAGE not only enables agents to reach a global consensus but also leverages this consensus to facilitate effective agent cooperation.

5.3 Ablation Study

As shown in Figure 4, we conduct ablation studies on four representative tasks with different difficulty levels to evaluate each STAGE component's contribution. STAGE w/o GM eliminates the grouping module, allowing each agent to communicate with all peers within its perceptual range. STAGE w/o HC discards the hierarchical mechanism for one-step communication, where a global consensus is constructed directly among all agents. It can be observed that, in the simple *MMM* task, STAGE and its variants exhibit only marginal performance differences. In the hard *MMM2* task, STAGE slightly outperforms all ablated variants. In the super-hard *6h_vs_8z* task, STAGE significantly outperforms all ablated variants, with *STAGE w/o GM* and *STAGE w/o HC* even exhibiting near-zero win rates. Similar performance gaps between STAGE and its ablated variants can also be observed in the *CN_15agent* task. The underlying reason is that, in simple tasks, agents do not require sophisticated coordination skills, and effective cooperation can be achieved through relatively simple communication mechanisms. As task difficulty increases, agents must learn more complex cooperative strategies to succeed. Consequently, the performance gap between STAGE and the ablated variants gradually widens.

5.4 Scalability in Large-Scale Scenarios

To evaluate the scalability of STAGE, we conduct experiments on large-scale Battle tasks with 20 and 64 agents, and a custom SMACv2 task with 50 Marines vs. 50 Marines, denoted *M50v50*. As shown in Figure 5, while several baselines achieve competitive results on *Battle-20*, STAGE still performs well. As agent numbers increase to *Battle-64* and *M50v50*, however, STAGE clearly outperforms all baselines. This validates the superior scalability of STAGE in handling large-scale tasks.

5.5 Communication Overhead

We further evaluate the communication overhead of STAGE, TarMAC, DGN, TGCNet across different tasks. We adopt a similar approach to that in [43] to compute the communication overhead C, defined as $C = \frac{\sum_{i=1}^{N} E_i}{N(N-1)}$, where N is the total number of agents, and E_i represents the number of communication edges associated with agent i. The denominator $N(N-1)$ is the total communication edges of all agents in a full communication network. As shown in Figure 6, STAGE demonstrates lower communication overhead than the existing TarMAC, DGN, and TGCNet methods across all tasks. This illustrates that our STAGE architecture strengthens team collaboration while mitigating communication overhead.

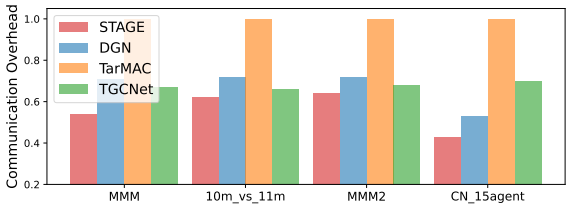
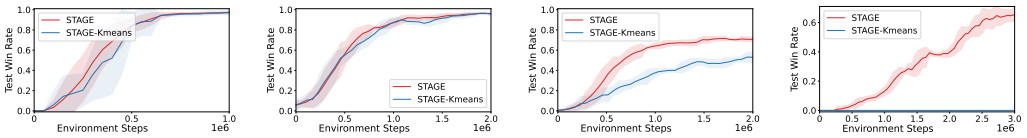


Figure 6: Comparison of communication overhead.



(a) *MMM* (b) *3s5z* (c) *5m_vs_6m* (d) *6h_vs_8z*

Figure 7: Comparison between STAGE and STAGE-Kmeans on tasks with different levels of difficulty.

5.6 Perceptual-Focus Grouping v.s. K-means Grouping

To investigate whether our proposed grouping strategy can be replaced by conventional clustering, we design a variant, STAGE-Kmeans. In this variant, agents are grouped by applying K-means to their current observations, and the agent closest to each cluster centroid is selected as the leader. All other components remain unchanged. Figure 7 shows that STAGE and STAGE-Kmeans perform similarly on the simple tasks *3s5z* and *MMM*, indicating that conventional clustering is viable in easier scenarios. This similarity diminishes as task difficulty increases. On the hard *5m_vs_6m* task, STAGE clearly outperforms STAGE-Kmeans, while on the super-hard *6h_vs_8z* task, STAGE-Kmeans nearly fails and STAGE remains effective. The underlying reason is that perceptual-focus grouping encourages agents to concentrate on the most critical observation semantics rather than dispersing attention across different dimensions, as K-means grouping does.

6 Conclusion

In this paper, we propose STAGE, a progressive local-to-global consensus framework for MARL. Agents first reach local consensus in dynamically formed groups, then group leaders communicate to form a unified global consensus. ViT aggregates messages, KL divergence aligns consensus distributions, and VAE preserves task-relevant information. Extensive experiments show STAGE outperforms baselines, particularly in scenarios requiring complex coordination.

7 Limitations

STAGE has been evaluated in fully cooperative settings with structured observations; its applicability to mixed cooperative–competitive tasks remains unexplored. The method assumes that the communication links required by the formed groups are available, but does not account for link availability during group formation. Applying STAGE to a new environment also requires defining the semantic observation components and their number. In addition, its robustness under bandwidth limitations, packet loss, and communication delays remains to be evaluated. Future work will explore connectivity-aware grouping and more adaptable semantic representations.

Acknowledgments and Disclosure of Funding

The authors thank the anonymous area chair and the reviewers for their insightful comments. This work was supported in part by the National Natural Science Foundation of China (Grant No. 62302189 and No. 62506210), the CAS Pioneer Hundred Talents Program (Grant No. 2026000063/E6YB01T and No. 2026000065/E6SR01T), and the Key Research and Development Project of Hubei Province, China (Grant No. 2024BAB070). Hangyu Mao is the corresponding author of this paper.

References

[1] Abdollah Amirkhani and Amir Hossein Barshooi. Consensus in multi-agent systems: a review. *Artificial Intelligence Review*, 55(5):3897–3935, 2022.

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021.

[3] Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4):819–840, 2002.

[4] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

[5] Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and Joelle Pineau. TarMAC: Targeted Multi-Agent Communication. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97, pages 1538–1546. PMLR, 2019.

[6] Shifei Ding, Wei Du, Ling Ding, Jian Zhang, Lili Guo, and Bo An. Multiagent reinforcement learning with graphical mutual information maximization. *IEEE Transactions on neural networks and learning systems*, 2023.

[7] Ling Ding, Tianbai Lyu, Zhiliang Bi, Hao Wang, Shanshan Feng, and Wei Yu. Communication-efficient multi-agent reinforcement learning with spatiotemporal information hub. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 20799–20807, 2026.

[8] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[9] Wei Duan, Jie Lu, and Junyu Xuan. Group-aware coordination graph for multi-agent reinforcement learning. *arXiv preprint arXiv:2404.10976*, 2024.

[10] Wei Duan, Jie Lu, and Junyu Xuan. Inferring latent temporal sparse coordination graph for multi-agent reinforcement learning. *arXiv preprint arXiv:2403.19253*, 2024.

[11] Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob Foerster, and Shimon Whiteson. Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 36:37567–37593, 2023.

[12] Zihong Gao, Hongjian Liang, Yuanhui Hao, Lei Hao, and Liangjun Ke. Grdc: A unified graph-driven framework for role discovery and communication in multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 29441–29449, 2026.

[13] Cong Guan, Feng Chen, Lei Yuan, Chenghe Wang, Hao Yin, Zongzhang Zhang, and Yang Yu. Efficient multi-agent communication via self-supervised information aggregation. *Advances in Neural Information Processing Systems*, 35:1020–1033, 2022.

[14] Shengchao Hu, Li Shen, Ya Zhang, and Dacheng Tao. Learning multi-agent communication from graph modeling perspective. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024.

[15] Jiechuan Jiang and Zongqing Lu. Learning Attentional Communication for Multi-Agent Cooperation. In *Proceeding of the 31th Advances in Neural Information Processing Systems (NeurIPS)*, pages 7254–7264, 2018.

[16] Jiechuan Jiang, Chen Dun, and Zongqing Lu. Graph Convolutional Reinforcement Learning for Multi-Agent Cooperation. In *International Conference on Learning Representations*, 2019.

[17] Xinran Li and Jun Zhang. Context-aware communication for multi-agent reinforcement learning. *arXiv preprint arXiv:2312.15600*, 2023.

[18] Dapeng Li, Na Lou, Zhiwei Xu, Bin Zhang, and Guoliang Fan. Efficient communication in multi-agent reinforcement learning with implicit consensus generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23240–23248, 2025.

[19] Pei Li, Yongkang Zhang, Zhonglin Lv, Jinmin Zhu, and Jiangjin Yin. CAIC: Congestion-aware intent communication for multi-agent reinforcement learning. In Emilija Perković and Daniel Malinsky, editors, *Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence*, volume 337 of *Proceedings of Machine Learning Research*, pages 3705–3718. PMLR, 17–21 Aug 2026.

[20] Michael L Littman. Markov Games as a Framework for Multi-Agent Reinforcement Learning. In *Proceedings of the 11th International Conference on Machine Learning*, pages 157–163, 1994.

[21] Chiqiang Liu and Dazi Li. Hygma: Hypergraph coordination networks with dynamic grouping for multi-agent reinforcement learning. *arXiv preprint arXiv:2505.07207*, 2025.

[22] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *Advances in Neural Information Processing Systems*, volume 30, pages 6379–6390. Curran Associates, Inc., 2017.

[23] Hangyu Mao, Wulong Liu, Jianye Hao, Jun Luo, Dong Li, Zhengchao Zhang, Jun Wang, and Zhen Xiao. Neighborhood Cognition Consistent Multi-Agent Reinforcement Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 7219–7226, 2020.

[24] Hangyu Mao, Zhengchao Zhang, Zhen Xiao, Zhibo Gong, and Yan Ni. Learning Agent Communication under Limited Bandwidth by Message Pruning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 5142–5149, 2020.

[25] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-Level Control Through Deep Reinforcement Learning. *Nature*, 518(7540):529, 2015.

[26] Yaru Niu, Rohan R Paleja, and Matthew C Gombolay. Multi-agent graph-attention communication and teaming. In *AAMAS*, volume 21, page 20th, 2021.

[27] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:1803.11485*, 2018.

[28] Jingqing Ruan, Xiaotian Hao, Dong Li, and Hangyu Mao. Learning to collaborate by grouping: A consensus-oriented strategy for multi-agent reinforcement learning. In *ECAI 2023*, pages 2010–2017. 2023.

[29] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019.

[30] Christian Schroeder de Witt, Jakob Foerster, Gregory Farquhar, Philip Torr, Wendelin Boehmer, and Shimon Whiteson. Multi-agent common knowledge reinforcement learning. *Advances in neural information processing systems*, 32, 2019.

[31] Junjie Sheng, Xiangfeng Wang, Bo Jin, Junchi Yan, Wenhao Li, Tsung-Hui Chang, Jun Wang, and Hongyuan Zha. Learning structured communication for multi-agent reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 36(2):50, 2022.

[32] Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning Multiagent Communication with Backpropagation. In *proceeding of the 29th Advances in Neural Information Processing Systems (NeurIPS)*, pages 2244–2252, 2016.

[33] Chuxiong Sun, Peng He, Rui Wang, and Changwen Zheng. Revisiting communication efficiency in multi-agent reinforcement learning from the dimensional analysis perspective. *arXiv preprint arXiv:2501.02888*, 2025.

[34] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-Decomposition Networks for Cooperative Multi-Agent Learning. *arXiv preprint arXiv:1706.05296*, 2017.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All You Need. In *Proceeding of the 30th Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017.

[36] Rundong Wang, Xu He, Runsheng Yu, Wei Qiu, Bo An, and Zinovi Rabinovich. Learning efficient multi-agent communication: An information bottleneck approach. In *International conference on machine learning*, pages 9908–9918. PMLR, 2020.

[37] Zhiwei Xu, Bin Zhang, Dapeng Li, Zeren Zhang, Guangchong Zhou, Hao Chen, and Guoliang Fan. Consensus learning for cooperative multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11726–11734, 2023.

[38] Zhiwei Xu, Kun Hu, Xin Xin, Weiliang Meng, Yiwei Shi, Hangyu Mao, Bin Zhang, Dapeng Li, and Jiangjin Yin. Reidentify: Context-aware identity generation for contextual multi-agent reinforcement learning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 69262–69278. PMLR, 13–19 Jul 2025.

[39] Di Xue, Lei Yuan, Zongzhang Zhang, and Yang Yu. Efficient multi-agent communication via shapley message value. In *IJCAI*, pages 578–584, 2022.

[40] Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. Mean Field Multi-Agent Reinforcement Learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80, pages 5567–5576. PMLR, 2018.

[41] Junyan Yu and Long Wang. Group consensus of multi-agent systems with undirected communication graphs. In *2009 7th Asian control conference*, pages 105–110. IEEE, 2009.

[42] Lei Yuan, Jianhao Wang, Fuxiang Zhang, Chenghe Wang, Zongzhang Zhang, Yang Yu, and Chongjie Zhang. Multi-agent incentive communication via decentralized teammate modeling. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 9466–9474, 2022.

[43] Sai Qian Zhang, Qi Zhang, and Jieyu Lin. Efficient communication in multi-agent reinforcement learning via variance based control. *Advances in neural information processing systems*, 32, 2019.

[44] Zhuohui Zhang, Bin He, Bin Cheng, and Gang Li. Bridging training and execution via dynamic directed graph-based communication in cooperative multi-agent systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23395–23403, 2025.

[45] Lianmin Zheng, Jiacheng Yang, Han Cai, Weinan Zhang, Jun Wang, and Yong Yu. Magent: A many-agent reinforcement learning platform for artificial collective intelligence, 2017.

A Implementation Details

A.1 Algorithmic Description

Algorithm 1 presents the pseudo-code of STAGE, and the source code can be found in the supplementary material.

Algorithm 1 The training procedure of STAGE.

Initialize network parameters θ, target network parameters θ^-, and replay buffer D.
for episode $= 1$ **to** $max_episodes$ **do**
  Reset the environment, obtain s^1 and $\boldsymbol{o}^1$, and initialize each agent's recurrent state and previous action.
  for $t = 1$ **to** $max_episode_length$ **do**
    Update each agent's recurrent state τ_i^t using GRU with o_i^t, a_i^{t-1}, and τ_i^{t-1}.
    Generate communication messages m_i^t and attention weights α_i from o_i^t and τ_i^t.
    Determine communication groups $\boldsymbol{g}$ and leader set $\overline{\boldsymbol{g}}$ according to α_i.
    for each nonempty group g_k **do**
      Aggregate intra-group messages and compute local consensus representations: $[L_i]_{i \in g_k} = \mathrm{ViT}_{\mathrm{local}}([m_i^t]_{i \in g_k})$.
    end for
    Aggregate local consensuses among leaders and compute global consensus representations: $[G_i]_{i \in \overline{\boldsymbol{g}}} = \mathrm{ViT}_{\mathrm{global}}([L_i]_{i \in \overline{\boldsymbol{g}}})$.
    for each non-leader agent $i \notin \overline{\boldsymbol{g}}$ **do**
      Generate virtual global consensus: $G_i^{\mathrm{virtual}} = \mathrm{MLP}(L_i)$.
    end for
    Select actions a_i^t according to $Q_i(\tau_i^t, L_i, G_i, a; \theta)$ and the exploration policy.
    Execute $\boldsymbol{a}^t$, receive reward r^t, and obtain s^{t+1} and $\boldsymbol{o}^{t+1}$.
    Store transition $(s^t, \boldsymbol{o}^t, \boldsymbol{a}^t, r^t, \boldsymbol{o}^{t+1}, s^{t+1})$ in replay buffer D.
    Sample a minibatch from D.
    Compute TD loss, global consistency loss, and VAE loss, and update θ using Equation (17).
  end for
  if episode $\mathrm{mod} M = 0$ **then**
    Update target network parameters $\theta^- \leftarrow \theta$.
  end if
end for

A.2 Implementation Details and Hyperparameters

In practical implementation, when reconstructing the global state s based on global consensus, we employ a single-layer MLP to compress the information of the global state, making the reconstruction process more manageable.

All experiments in this paper are conducted on two machines with the following configurations: (1) NVIDIA GeForce RTX 4090 D and AMD EPYC 9654 96-Core Processor. (2) NVIDIA GeForce RTX 3090 GPU and Intel(R) Xeon(R) Gold 6254 CPU, Table 2 shows the hyperparameters used for our algorithm STAGE.

The open-source implementations for all baseline methods are provided below.

- QMIX: https://github.com/starry-sky6688/MARL-Algorithms
- DGN: https://github.com/jiechuanjiang/pytorch_DGN
- TarMAC: https://github.com/TonghanWang/NDQ/blob/master
- CommFormer: https://github.com/charleshsc/CommFormer
- NCC: https://github.com/maohangyu/marl_demo
- COLA: https://github.com/deligentfool/COLA
- LSC: https://github.com/bellmanequation/LSC
- TGCNet: https://github.com/ZhuohuiZhang/TGCNet

Table 2: The hyperparameters used in different scenarios.

Hyperparameter	**SMAC**	**SMACv2**	**CN**	**Battle**
Optimizer	Adam	Adam	Adam	Adam
Learning rate	5×10^{-4}	5×10^{-4}	5×10^{-4}	1×10^{-3}
Hidden dimension	32 or 64	32 or 64	32 or 64	32 or 64
Activation function	ReLU	ReLU	ReLU	ReLU
Buffer size	5000	5000	5000	5000
Layer of ViT	2	2	1	2
Attention heads in ViT	2	2	2	2
Attention heads in perception	1	1	2	2
Discount factor	0.99	0.99	0.99	0.99
α	0.1	0.1	0.1	0.1
β	0.1	0.1	0.1	0.1

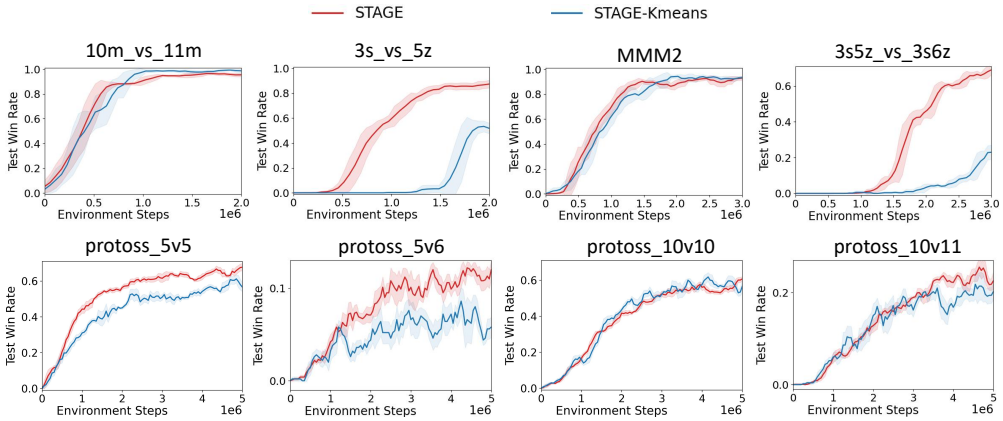


Figure 8: Comparison between STAGE and STAGE-Kmeans on eight SMAC maps.

B Additional Results

B.1 Additional Comparison with K-means Grouping

To further analyze the effectiveness of the proposed perceptual-focus grouping strategy, we compare STAGE with a K-means-based variant, denoted as STAGE-Kmeans. In STAGE-Kmeans, agents are grouped by applying K-means to their current observations, while the remaining components are kept unchanged. For fairness, we set the number of clusters to four, matching the maximum number of groups in STAGE under the default single-head setting.

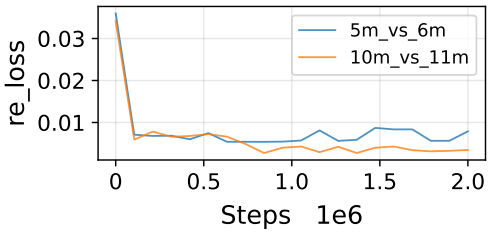
As shown in Figure 8, STAGE and STAGE-Kmeans perform similarly on the relatively simple *10m_vs_11m* task, while STAGE achieves better performance on the more challenging *3s_vs_5z* and *3s5z_vs_3s6z* tasks, as well as on SMACv2 scenarios. This suggests that grouping agents solely by raw observation similarity is not always reliable, especially in difficult scenarios requiring task-relevant coordination. A key limitation of K-means is that its cluster number must be predefined and is difficult to choose consistently across tasks. In contrast, STAGE forms groups from agents' experience-driven perceptual focuses without an additional clustering procedure, enabling task-adaptive communication.

B.2 Analysis of VAE State Reconstruction

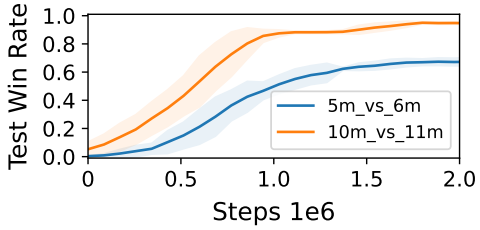
We further analyze the state reconstruction ability of the VAE module on two hard SMAC scenarios, *5m_vs_6m* and *10m_vs_11m*, and two super-hard scenarios, *MMM2* and *6h_vs_8z*. The VAE is introduced to encourage the learned global consensus to preserve task-relevant state information. Therefore, a lower reconstruction loss indicates that the global consensus captures more accurate information about the underlying state.

As shown in Figure 9, the reconstruction loss consistently decreases during training across all scenarios, indicating that the learned global consensus progressively preserves more state-relevant information. Moreover, the reconstruction quality is closely related to the final cooperative perfor-

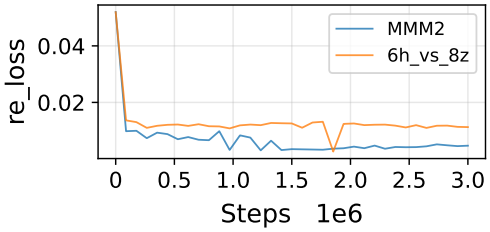
mance. Among the hard tasks, *10m_vs_11m* achieves a lower converged reconstruction loss than *5m_vs_6m*, and it also obtains a higher test win rate. Similarly, among the super-hard tasks, *MMM2* converges to a lower reconstruction loss than *6h_vs_8z* and achieves better performance. These results suggest that more accurate reconstruction of task-relevant state information can improve the quality of the learned global consensus, thereby benefiting cooperative decision-making.



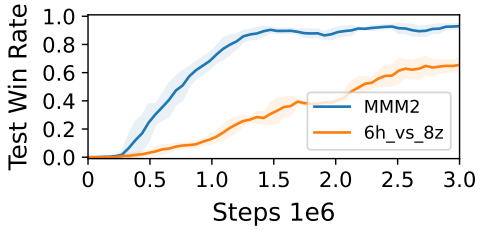
(a) Reconstruction loss on hard tasks



(b) Test win rate on hard tasks



(c) Reconstruction loss on super-hard tasks



(d) Test win rate on super-hard tasks

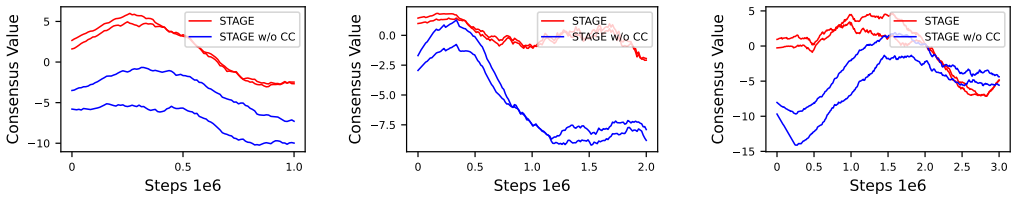
Figure 9: Analysis of VAE state reconstruction on hard and super-hard SMAC tasks.

B.3 Additional Visualization of Agent Consensus

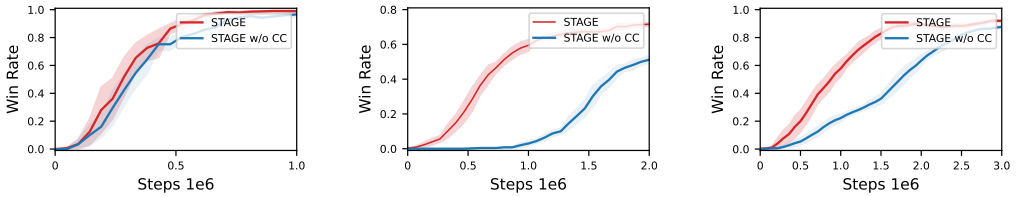
We first visualize the consensus values on representative SMAC and SMACv2 tasks to further examine whether STAGE can learn stable and coherent global consensus during training. As shown in Figure 10, STAGE consistently drives the consensus values toward a stable range across different tasks, while *STAGE w/o CC* exhibits more unstable or divergent consensus trajectories. This phenomenon is especially clear on more challenging tasks, where communication alone is insufficient to guarantee consistent consensus formation. These results indicate that the consensus constraint plays an important role in aligning agents' global consensus representations and stabilizing cooperative learning.

Furthermore, we provide a full visualization of the consensus values for all agents to examine whether STAGE truly achieves collective alignment. Each curve represents the evolution of one agent's consensus value during training. Compared with representative-agent visualization, plotting all agents provides a more complete view of whether the learned consensus is consistently shared across the whole team. As shown in Figure 11, agents trained with STAGE exhibit highly aligned consensus trajectories. Their consensus values gradually converge to a similar range, indicating that the proposed consensus constraint effectively encourages agents to form a shared global understanding. In contrast, without the consensus constraint, *STAGE w/o CC* shows significantly more divergent trajectories. The consensus values of different agents remain scattered during training, suggesting that communication alone is insufficient to guarantee collective agreement.

To complement the qualitative visualization, Table 3 reports the mean pairwise Euclidean distances and cosine similarities among agents' global consensus representations in the original high-dimensional space. In both CN scenarios, STAGE achieves lower Euclidean distances and higher cosine similarities than *STAGE w/o CC*, providing quantitative evidence that the consensus constraint promotes alignment among agents' global consensus representations.

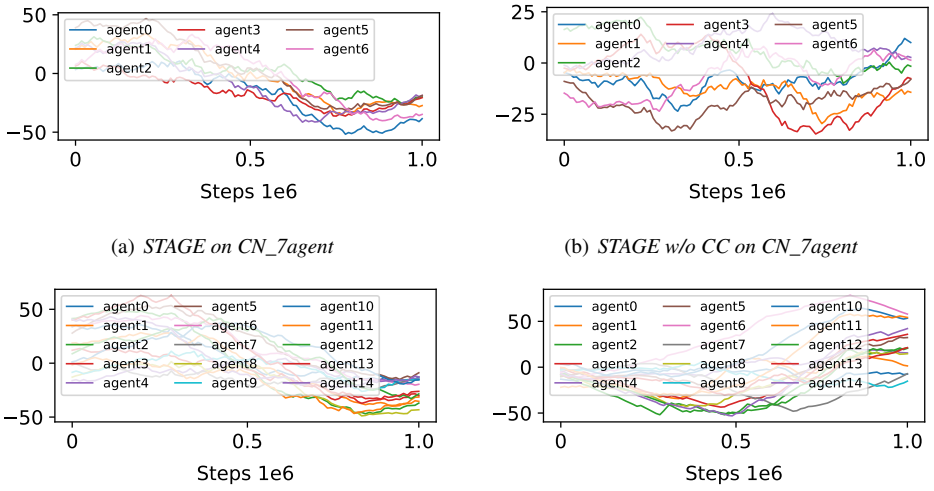


(a) *MMM*: consensus (b) *5m_vs_6m*: consensus (c) *MMM2*: consensus



(d) *MMM*: win rate (e) *5m_vs_6m*: win rate (f) *MMM2*: win rate

Figure 10: Consensus and performance visualization on representative SMAC tasks. The first row shows consensus values, and the second row shows the corresponding test win rates. STAGE achieves more stable consensus and better performance than *STAGE w/o CC*, indicating the effectiveness of the consensus constraint.



(a) *STAGE on CN_7agent* (b) *STAGE w/o CC on CN_7agent*

(c) *STAGE on CN_15agent* (d) *STAGE w/o CC on CN_15agent*

Figure 11: Consensus visualization of all agents on CN tasks. (a) and (b) show the consensus trajectories of all agents in *CN_7agent*; (c) and (d) show the consensus trajectories of all agents in *CN_15agent*.

B.4 Effectiveness of the Dynamic Leader Mechanism

In our design, a dynamic leader mechanism is introduced, where different agents take on the leader role at different time steps. To validate the effectiveness of this mechanism, we investigate two configurations: updating the communication topology at every step and updating it every five steps. The experimental results are shown in Figure 12.

As observed, on the relatively simple *MMM* and *3s5z* tasks, the two variants achieve nearly identical performance. In the hard *5m_vs_6m* and *10m_vs_11m* tasks, updating the communication topology at every step results in faster convergence and slightly higher final win rates. Notably, on the super-hard tasks, namely *MMM2* and *3s5z_vs_3s6z*, the per-step update strategy significantly outperforms the variant that updates the topology every five steps. These results clearly demonstrate that, in extremely challenging scenarios characterized by highly complex and rapidly changing environments, the dynamic leader mechanism that enables all agents to assume leadership roles or possess leadership capabilities is essential for achieving superior performance.

Table 3: Global-consensus consistency on CN tasks in the original representation space. ED denotes Euclidean distance (lower is better); CS denotes cosine similarity (higher is better).

Scenario	Metric	STAGE	STAGE w/o CC
CN_7agent	ED ↓	0.1722 ± 0.099	2.7096 ± 1.176
	CS ↑	0.772 ± 0.242	0.278 ± 0.292
CN_15agent	ED ↓	0.1129 ± 0.024	27.0786 ± 15.434
	CS ↑	0.714 ± 0.163	0.329 ± 0.227

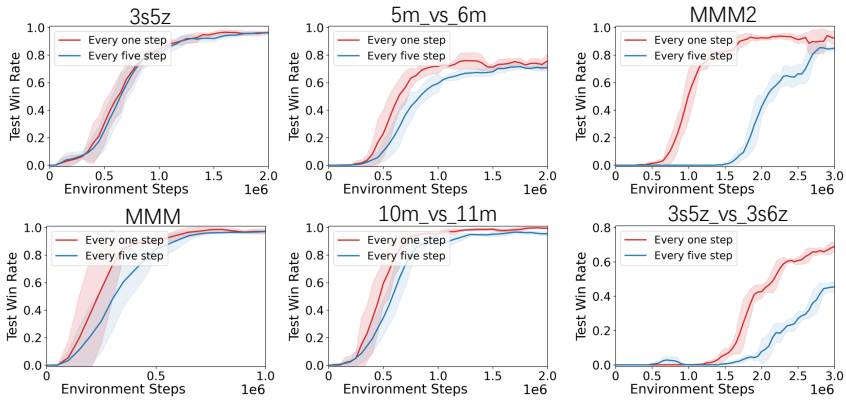


Figure 12: Performance comparison of per-step and five-step communication topology updates.

B.5 Training Efficiency

To evaluate computational efficiency, Table 4 presents the training times across all SMACv2 tasks. Notably, STAGE significantly reduces the training time compared to LSC and TGCNet, which incur substantial computational overheads with average training times of 129.4 and 62.3 hours, respectively. While CommFormer averages 30.9 hours, the computational cost of STAGE (averaging 44.6 hours) remains well within an acceptable and practical range. Furthermore, it is observed that while CommFormer is faster in smaller scenarios, its training time grows markedly as the number of agents increases, narrowing the efficiency gap with our method in complex tasks. This demonstrates that STAGE possesses superior scalability and provides a more stable solution for large-scale multi-agent coordination.

Table 4: Time cost (h) of different algorithms, where P and T are the abbreviations for Protoss and Terran.

Algorithm	P5v5	P5v6	P10v10	P10v11	T5v5	T5v6	T10v10	T10v11	**Avg.**
COLA	11.3	13.0	17.2	17.8	14.2	28.9	17.7	18.7	**17.4**
STAGE	35.0	45.6	43.8	39.7	49.8	46.6	47.5	48.4	**44.6**
CommFormer	22.6	22.9	34.7	34.4	20.2	30.3	34.2	48.1	**30.9**
LSC	87.0	90.0	144.9	140.1	113.8	114.2	169.1	176.3	**129.4**
TGCNet	55.2	60.3	54.2	62.4	72.3	58.6	76.5	58.5	**62.3**

B.6 Analysis of Attention Heads in Grouping

Our perceptual-focus grouping mechanism is scalable, where the maximum number of candidate groups K can be directly expanded by increasing the number of grouping attention heads h. Table 5 summarizes the results of STAGE for different (h, K). It shows that STAGE with $(h = 1, K = 4)$ is better than STAGE with $(h = 2, K = 8)$ on all three tasks, where the absolute performance degradation when increasing K from 4 to 8 is: 3s5z: (-3.9%), MMM: (-1.1%), and M50v50: (-7.1%). The underlying reason is that when increasing from $(h = 1, K = 4)$ to $(h = 2, K = 8)$, multiple attention heads may repeatedly attend to semantically similar information and result in semantic redundancy.

Nevertheless, the two-head variant $(h = 2, K = 8)$ still maintains highly competitive win rates across all three tasks. It also notably outperforms the Mean baseline (i.e., the average performance of all compared methods). Moreover, it matches the Best baseline on 3s5z and MMM and surpasses it on the M50v50 task. Overall, these results demonstrate that STAGE can flexibly scale up the number

Table 5: Performance under two grouping configurations. Values in parentheses for STAGE are standard deviations.

Configuration	3s5z	MMM	M50v50
STAGE ($h = 1$, $K = 4$)	**96.3 (1.9)**	**97.4 (2.5)**	**84.5 (5.6)**
STAGE ($h = 2$, $K = 8$)	92.4 (5.3)	96.5 (5.6)	77.4 (9.5)
Best baseline	95.5	96.8	40.9
Mean baseline	51.6	67.1	22.3

of groups by adding attention heads while preserving strong win rates, making it adaptable to tasks with varying complexity and agent scales.

C Environments

C.1 Cooperative Navigation

The cooperative navigation scenario is a classic multi-agent coordination benchmark in which an equal number of agents and landmarks are deployed. As depicted in Figure 13(a), the environment is a 10×10 grid world in which both agents and landmarks are randomly initialized. Each agent can observe the positions of other agents and landmarks and has a discrete action space: [up, down, left, right, stop]. At each time step, agents receive a negative reward proportional to their distance from the landmarks. As a result, each agent must learn which landmark to navigate toward while simultaneously avoiding collisions with other agents.

C.2 StarCraft Multi-Agent Challenge (SMAC)

The StarCraft Multi-Agent Challenge (SMAC) is a benchmark environment specifically designed to evaluate and advance Multi-Agent Reinforcement Learning (MARL) algorithms within the complex context of the real-time strategy game StarCraft II. It emphasizes micromanagement tasks, where independent learning agents control individual units and must coordinate their actions based solely on local observations. As depicted in Figure 13(c), agents are able to observe various attributes such as distance, position, health, shield, and type within their sight range, with the ultimate goal of defeating enemy units. The environment enables centralized training while supporting decentralized execution, offering additional state information during training to facilitate algorithm development. The details of the experimental tasks used in our study are provided in Table 6.

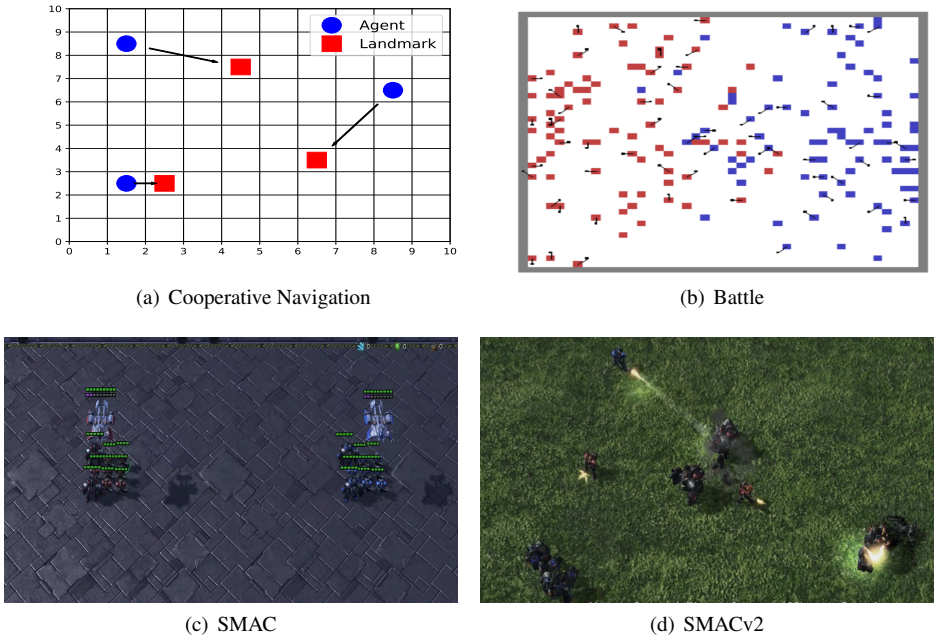
C.3 SMACv2

The SMACv2 is a benchmark environment that extends SMAC to evaluate Multi-Agent Reinforcement Learning algorithms under heightened stochasticity and diversity within the StarCraft II micromanagement domain. As shown in Figure 13(d), in each episode, procedurally generated team compositions, randomized allied and enemy starting positions—either mirrored or encircling—and unit-specific attack and sight ranges sampled from the original game mechanics jointly create a previously unknown battlefield. As a result, agents must coordinate while coping with uncertainty in their own unit types, initial locations, and relative engagement distances, all inferred solely from local observations of distance, position, health, shield, and unit type within their sight range. SMACv2 retains the centralized-training-with-decentralized-execution paradigm, providing extra state information during training to foster development of robust, adaptive strategies that generalize beyond fixed unit line-ups and deterministic spawn patterns. The details of the experimental tasks used in our study are provided in Table 7.

C.4 Battle

The Battle environment in MAgent2 is a large-scale team-combat scenario designed to evaluate coordination in multi-agent systems. Agents are divided into two opposing teams and must coordinate attacks to eliminate opponents. Unlike environments with explicit team-level rewards, agents receive rewards based only on individual performance, such as damage dealt or enemies eliminated, making emergent coordination particularly challenging. Each agent starts with 10 HP, loses 2 HP for each

attack received, and regenerates 0.1 HP per time step. This slow health regeneration encourages agents to eliminate opponents quickly before they recover, requiring efficient and timely cooperation. For both *Battle-20* and *Battle-64*, the step reward, death penalty, attack penalty, and attack-opponent reward are set to -0.005, -0.1, -0.05, and 0.2, respectively.



(a) Cooperative Navigation (b) Battle

(c) SMAC (d) SMACv2

Figure 13: Illustration of four benchmark scenarios: (a) Cooperative Navigation, (b) Battle, (c) SMAC, and (d) SMACv2.

Table 6: Different tasks in SMAC.

Task	**Ally Units**	**Enemy Units**	**Type**
MMM	1 Medivac 2 Marauders 7 Marines	1 Medivac 2 Marauders 7 Marines	heterogeneous & symmetric
3s5z	3 Stalkers 5 Zealots	3 Stalkers 5 Zealots	heterogeneous & symmetric
3s_vs_5z	3 Stalkers	5 Zealots	micro-trick: kiting
5m_vs_6m	5 Marines	6 Marines	homogeneous & asymmetric
10m_vs_11m	10 Marines	11 Marines	homogeneous & asymmetric
MMM2	1 Medivac 2 Marauders 7 Marines	1 Medivac 3 Marauders 8 Marines	heterogeneous & asymmetric
3s5z_vs_3s6z	3 Stalkers 5 Zealots	3 Stalkers 6 Zealots	heterogeneous & asymmetric
6h_vs_8z	6 Hydralisks	8 Zealots	micro-trick: focus fire

Table 7: Different map tasks in SMACv2 scenarios.

Task	Ally	Enemy	Race	Unit types	Generation probability
protoss5_vs_5	5	5	Protoss	Stalker, Zealot, Colossus	0.45, 0.45, 0.1
protoss5_vs_6	5	6			
protoss10_vs_10	10	10			
protoss10_vs_11	10	11			
terran5_vs_5	5	5	Terran	Marine, Marauder, Medivac	0.45, 0.45, 0.1
terran5_vs_6	5	6			
terran10_vs_10	10	10			
terran10_vs_11	10	11			
terran50_vs_50	50	50	Terran	Marine, Marauder, Medivac	1.0, 0.0, 0.0

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction describe the proposed local-to-global progressive consensus framework, including hierarchical communication, perceptual-focus grouping, and consensus regularization. These claims are supported by the method design, ablation studies, communication-overhead analysis, and experiments on CN, SMAC, SMACv2, and Battle scenarios.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations related to the reliance on semantically decomposed observations, the need for global state information during training for the VAE-based validity constraint, and the evaluation scope being limited to simulated MARL benchmarks.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present formal theoretical results, theorems, or proof-based claims. The contributions are primarily algorithmic and empirical.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides the implementation code of the proposed algorithm and the hyperparameter settings needed to reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility.

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The experiments are conducted on public MARL benchmarks, and the implementation of the proposed method is provided in the supplementary material with the necessary instructions and hyperparameter settings for reproduction. The baseline implementations are also credited with their corresponding public repositories.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper reports the evaluated environments, task configurations, training steps, baselines, optimizer, learning rates, hidden dimensions, attention heads, discount factor, and loss coefficients. Additional environment descriptions and hyperparameter settings are provided in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: The reported results are averaged over multiple random seeds, and the tables report mean performance with standard deviation.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper reports the hardware configurations used for experiments, including GPU and CPU types, and provides training-time comparisons across SMACv2 tasks. These details help estimate the computational resources required to reproduce the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses simulated multi-agent benchmarks and does not involve human subjects, private data, sensitive personal information, or deceptive data collection. The experiments are conducted in standard research environments and conform to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses that the proposed method may improve coordination efficiency in cooperative multi-agent systems, with potential applications such as robotics, traffic control, and distributed decision-making. It also acknowledges possible risks if such coordination methods are applied to safety-critical or adversarial domains without proper validation and oversight.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk models, scraped datasets, pre-trained language models, image generators, or datasets with sensitive content. The released materials, if any, are limited to code for reproducing experiments on standard simulated MARL benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper cites the original works for the benchmark environments and baseline methods, including SMAC, SMACv2, MAgent2/Battle, and the compared MARL algorithms. Public baseline implementations are credited in the appendix, and their usage follows the corresponding repository terms and licenses.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper introduces the implementation of STAGE as a new research artifact. The supplementary material documents the algorithm, model architecture, hyperparameters, and training procedure needed to use and reproduce the proposed method.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing, user studies, or research with human subjects. All experiments are conducted in simulated multi-agent reinforcement learning environments.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects, crowdsourcing, or collection of personal data. Therefore, IRB approval or equivalent review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method does not use LLMs as a model component, training signal, data generator, evaluator, or experimental baseline.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.