HHC: Hierarchical Hypergraph Communication for Multi-Agent Systems

Qifan Liang[1] Chenlong Li[1] Feiyu Wang[1] Jing Fu[1] Yixiang Shan[1] Lu Guo[1] Wei Liu[1] Lichang Song[1] Ting Long[*1]

[1]School of Artificial Intelligence, Jilin University, Changchun, Jilin, China, *Corresponding author.

Abstract

Cooperative multi-agent reinforcement learning faces significant coordination challenges due to partial observability. Although communication can mitigate these issues, traditional methods often overlook the fact that agents can simultaneously belong to multiple collaborative groups, playing diverse roles. This limitation hinders the effective integration of tactical and strategic information. To address this, we propose Hierarchical Hypergraph Communication. In our method, agents are modeled via a tactical-level hypergraph that supports overlapping multi-group memberships, facilitating the extraction of fine-grained tactical features. Simultaneously, these tactical hyperedges serve as virtual nodes to integrate and communicate strategic intents within a strategic-level hypergraph. This dual-layered architecture empowers agents to align individual actions with global strategies through a hierarchical communication process, thereby enhancing decision-making capabilities. Extensive experiments demonstrate the effectiveness of our method in complex coordination tasks.

1 INTRODUCTION

Cooperative multi-agent reinforcement learning (MARL) aims to enable agents to acquire collaborative behaviors to accomplish complex tasks that are infeasible for a single agent [McKee et al., 2020, Du et al., 2023]. This paradigm has demonstrated remarkable capabilities across various domains, including robotics [Orr and Dutta, 2023], intelligent traffic systems [Mushtaq et al., 2023], and network resource allocation [Allahham et al., 2022]. In MARL, each agent interacts with both other agents and the environment to learn optimal decision-making strategies [Neto, 2005, Bukharin et al., 2023]. However, agents often face significant challenges due to partial observability, which complicates coordination and learning [Tuyls and Nowé, 2005, Panait and Luke, 2005].

The Centralized Training with Decentralized Execution (CTDE) framework leverages a centralized critic with global states to address this [Sunehag et al., 2017, Rashid et al., 2020]. Nevertheless, agents are still limited by local observations during execution, which hinders effective coordination.

Communication has emerged as an effective solution to this issue; however, it is often plagued by uncertainties regarding whom to communicate with (communication targets) and what to communicate (communication content) [Zhang et al., 2019, 2020]. Traditional communication-based methods, whether relying on broadcasting [Sukhbaatar et al., 2016] or graph-based topologies (Figure 1(a)) [Fan et al., 2025], treat individual agents as the primary information nodes and communication targets, propagating private local observations across the network [Sheng et al., 2022]. While these approaches improve performance, they predominantly emphasize agent-level message exchange and underutilize emergent group structures [Jiang et al., 2021, Yu, 2023], thereby overlooking the overlapping nature of cooperative groups in complex coordination [Wang et al., 2023, Li and Zhang, 2023].

To address this limitation, existing hypergraph-based [Feng et al., 2019] methods (Figure 1(b)) cluster agents into disjoint groups to form cooperative groups [Zhu et al., 2024, Shi et al., 2025, Liu and Li, 2025], facilitating local intra-group communication. However, these methods still face two critical challenges. On one hand, they primarily focus on intra-group tactical communication while neglecting high-level strategic communication. On the other hand, it is difficult for them to model the overlapping multi-group membership of agents. In complex scenarios, an agent might simultaneously belong to different functional groups, such as executing an attack within an offensive group while providing flank cover for a defensive group. This inability to capture hierarchical and overlapping group structures ren-

Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence (UAI 2026), PMLR 337:3802–3818.

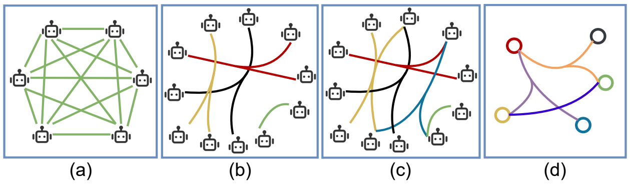


Figure 1: Comparison of communication topologies. (a) Standard graph-based communication. (b) Existing flat hypergraph-based communication. (c) Tactical level of HHC, where agents can join multiple tactical hyperedges to form overlapping groups. (d) Strategic level of HHC, where tactical hyperedges are elevated as virtual nodes to capture high-level coordination.

ders existing methods less effective in highly complex environments.

We observe that hyperedges, by aggregating group-level information, naturally serve as ideal virtual nodes for high-level strategic coordination. Building upon this insight, we propose Hierarchical Hypergraph Communication (HHC), a framework that models multi-level coordination via a dual-layered hypergraph architecture. HHC comprises two core modules: Adaptive Hierarchical Hypergraph Construction (AHHC) and Top-Down Hierarchical Hypergraph Communication (THHC). Specifically, AHHC evaluates agents' multi-group membership strength to adaptively construct the tactical-level hypergraph (Figure 1(c)) and extract hyperedge features. Subsequently, these hyperedges are treated as virtual nodes to construct the strategic-level hypergraph (Figure 1(d)). Following this structure, THHC utilizes a Structure-Constrained Hypergraph Attention mechanism to facilitate sparse and personalized communication. This ensures that strategic intents are filtered through the hierarchy to deliver tailored tactical guidance for each agent.

Our main contributions can be summarized as follows:

- We formulate MARL communication as hierarchical hypergraph modeling, where agents can join overlapping tactical groups and group-level structures are further abstracted for strategic coordination.
- We propose HHC, which combines adaptive hierarchy construction with top-down structure-constrained communication, enabling strategic intents to guide tactical coordination and agent decisions.
- Extensive experiments across diverse environments show that HHC improves coordination performance, especially in tasks with dense interactions and dynamic group structures.

2 RELATED WORK

Cooperative MARL is commonly studied under the centralized training with decentralized execution (CTDE) paradigm. Value-decomposition methods, such as VDN [Sunehag et al., 2017], QMIX [Rashid et al., 2020], QTRAN [Son et al., 2019], and QPLEX [Wang et al., 2021], decompose the joint value function into agent-wise utilities to address credit assignment. Policy-gradient methods such as MAPPO [Yu et al., 2022] optimize decentralized policies with centralized training signals. Communication is another important direction for addressing partial observability in MARL. Early methods such as CommNet [Sukhbaatar et al., 2016] rely on global broadcasting, while later methods improve communication efficiency through variance control, temporal message selection, or context-aware attention [Zhang et al., 2019, 2020, Li and Zhang, 2023]. Several works further introduce structured or hierarchical grouping mechanisms based on agent types, learned relations, or clustering [Jiang et al., 2021, Sheng et al., 2022, Yu, 2023, Wang et al., 2023]. However, these methods typically operate on agent-level communication or disjoint group assignments, making it difficult to model agents that participate in multiple functional groups. Hypergraphs provide a natural way to model higher-order relations beyond pairwise interactions [Feng et al., 2019]. Recent works have explored hypergraph structures for cooperative MARL and coordination, including HGCN-MIX [Bai et al., 2022], HGAC/ATT-HGAC [Zhang et al., 2022], HyperComm [Zhu et al., 2024], HYGMA [Liu and Li, 2025], and SDHN [Zhao et al., 2025]. These methods show the usefulness of hypergraphs for high-order coordination. In contrast, HHC combines overlapping tactical memberships, strategic-level hypergraph abstraction, and top-down strategic-to-tactical communication within a unified hierarchical framework.

3 PRELIMINARY

Dec-POMDP: A fully cooperative multi-agent reinforcement learning task can be modeled as a *Decentralized Partially Observable Markov Decision Process* (Dec-POMDP) [Oliehoek et al., 2016], defined by $\langle \mathcal{N}, \mathcal{A}, \mathcal{S}, \mathcal{P}, \mathcal{O}, \Omega, r, \gamma \rangle$. Here, $\mathcal{N} = \{1, \dots, N\}$ denotes the set of agents, and $\mathcal{A} = \prod_{i \in \mathcal{N}} \mathcal{A}^i$ is the joint action space with $\boldsymbol{a}_t = \{a_t^i\}_{i \in \mathcal{N}}$. The global state $s_t \in \mathcal{S}$ evolves according to $\mathcal{P}(s_{t+1} \mid s_t, \boldsymbol{a}_t)$. Each agent receives a local observation $o_t^i \in \mathcal{O}$, and we denote the joint observation as $\boldsymbol{o}_t = (o_t^1, \dots, o_t^N)$ generated by an observation function $\Omega(\boldsymbol{o}_t \mid s_t)$. All agents share the reward function $r(s_t, \boldsymbol{a}_t)$ and discount factor $\gamma \in [0, 1)$. Each agent maintains an action–observation history $\tau_t^i = (o_1^i, a_1^i, \dots, o_t^i)$ to condition its decentralized policy $\pi^i(a_t^i \mid \tau_t^i)$. The objective is to learn a joint policy $\boldsymbol{\pi}$ that maximizes the expected discounted return.

Hypergraph Structure: A hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a vertex set $\mathcal{V}$ (representing the agents) and a hyperedge set $\mathcal{E}$ containing M hyperedges, where each hyperedge $e_j \in \mathcal{E}$ connects an arbitrary subset of vertices. The topology can be encoded by an incidence matrix $\mathbf{H} \in \{0,1\}^{N\times M}$, where $H_{ij} = 1$ if vertex v_i participates in hyperedge e_j, and 0 otherwise. To enable *overlapping multi-group memberships*, we further define a *membership strength* matrix $\mathbf{P} \in [0,1]^{N\times M}$, where P_{ij} quantifies the degree of affiliation between agent v_i and hyperedge (group) e_j. Here, $\mathbf{H}$ specifies the connectivity pattern, while $\mathbf{P}$ provides soft membership strengths that support overlapping assignments, enabling the representation of higher-order coordination patterns in MARL tasks [Feng et al., 2019].

4 METHOD

We propose Hierarchical Hypergraph Communication (HHC), which consists of two modules, as shown in Figure 2. Adaptive Hierarchical Hypergraph Construction (**AHHC**) first infers sparse agent–group and group–supergroup topologies from learned membership strengths. Top-Down Hierarchical Hypergraph Communication (**THHC**) then propagates information from supergroups to groups and from groups to agents through structure-constrained attention. The resulting agent-level messages are used for decentralized action selection.

4.1 ADAPTIVE HIERARCHICAL HYPERGRAPH CONSTRUCTION (AHHC)

4.1.1 Agent Topology Construction

We first estimate each agent's membership strengths to latent tactical groups from its trajectory features, and then infer a sparse agent-group incidence matrix for communication.

Agent Membership Strength. To capture time-varying coordination patterns under partial observability, we encode each agent's local observation-action history with a recurrent encoder. In implementation, the encoder input includes the current local observation o_t^i and the previous action a_{t-1}^i:

$$h_t^i = \mathrm{RNN}([o_t^i, a_{t-1}^i], h_{t-1}^i), \tag{1}$$

where $\mathrm{RNN}(\cdot)$ denotes the recurrent neural network [Graves, 2012, Zaremba, 2014], and we implemented it with a Gated Recurrent Unit (GRU) [Chung et al., 2014]. The $[\cdot,\cdot]$ denotes concatenation. h_t^i, h_{t-1}^i denote the latent trajectory state of the current step and previous step, and h_0^i is initialized with a zero vector, that is $h_0^i = \mathbf{0}$. The recurrent encoder is used only to summarize each agent's local observation-action history under partial observability [Li et al., 2025]. Based on h_t^i, we compute the membership strength:

$$\mathbf{P}_{t,i}^{(1)} = \sigma\Big(f^{(1)}(h_t^i)\Big) \in (0,1)^M, \tag{2}$$

where $\mathbf{P}_{t,i}^{(1)} = [P_{t,i1}^{(1)}, \ldots, P_{t,iM}^{(1)}]$ denotes the membership strength vector of agent i over M latent tactical groups, function $f^{(1)}$ is implemented using a multiple-layer perceptron (MLP) and $\sigma(\cdot)$ denotes the Sigmoid activation function. Unlike Softmax normalization, the Sigmoid formulation treats group memberships independently and thus supports non-exclusive assignments.

Sparse Topology Inference. To obtain a sparse and robust communication topology, we convert the continuous membership strengths into a discrete incidence matrix $\mathbf{H}_t^{(1)} \in \{0,1\}^{N\times M}$. We adopt a dual-filtering protocol with Top-k selection and a confidence threshold δ. Formally, the entry $H_{t,ij}^{(1)}$ is determined as follows:

$$H_{t,ij}^{(1)} = \begin{cases} 1, & \text{if } j \in \text{Top-}k(\mathbf{P}_{t,i}^{(1)}) \text{ and } P_{t,ij}^{(1)} > \delta, \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where Top-$k(\cdot)$ selects the k largest entries of $\mathbf{P}_{t,i}^{(1)}$, and δ filters out low-confidence associations. This mechanism ensures that only the most significant and reliable coordination relationships are preserved.

Connectivity Guarantee. To prevent agents from being isolated during the sparsification process, we implement a fallback mechanism to ensure the basic connectivity of the communication network. In scenarios where all membership strengths of an agent fail to exceed the threshold (i.e., $\max_j P_{t,ij}^{(1)} \le \delta$), the aforementioned sparsification protocol would result in an empty set of connections. To mitigate this, we forcibly preserve the link between the agent and its most preferred group:

$$H_{t,ij}^{(1)} = 1, \quad \text{if } j = \arg\max_j P_{t,ij}^{(1)}. \tag{4}$$

This fallback rule guarantees that each agent is connected to at least one tactical hyperedge, preventing empty communication sets after sparsification and preserving effective gradient pathways for hierarchical communication learning.

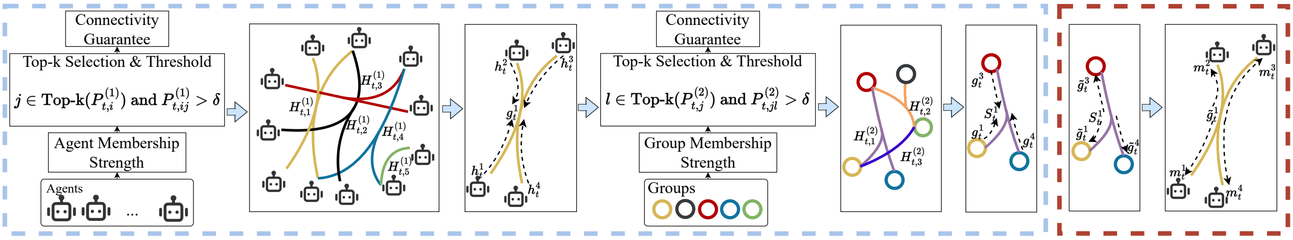
4.1.2 Tactical Group Abstraction

Given $P_t^{(1)}$ and $H_t^{(1)}$, we compute tactical group representations and use them to infer the group–supergroup topology.

Structural Feature Aggregation. Each tactical group is treated as a hyperedge over its affiliated agents. For group j, we compute a weighted summary from agent trajectory features $\{h_t^i\}_{i=1}^N$ according to $\mathbf{H}_t^{(1)}$ and $\mathbf{P}_t^{(1)}$:

$$\bar{g}_t^j = \frac{\sum_{i=1}^N H_{t,ij}^{(1)} P_{t,ij}^{(1)} h_t^i}{\sum_{i=1}^N H_{t,ij}^{(1)} P_{t,ij}^{(1)} + \epsilon}, \tag{5}$$

where ϵ is a small constant (e.g., 1e-8) to avoid division by zero. This structured aggregation effectively preserves the contributions of core members while filtering out irrelevant noise from non-member agents. Moreover, the normalization also maintains consistent feature scaling, thereby helping maintain numerical stability during training.



(a) AHHC (b) THHC

Figure 2: Overview of the HHC framework. (a) Agents generate membership strengths $P_{t,i}^{(1)}$, which are filtered through a dual-gate mechanism (Top-k selection and threshold δ) together with a fallback connectivity mechanism to form overlapping tactical hyperedges $H_{t,ij}^{(1)}$. Agent features h_t^i are then aggregated into tactical group representations $\{g_t^j\}$. A similar adaptive process further abstracts these groups into higher-level supergroups $H_{t,jl}^{(2)}$ and their corresponding strategic intents $\{S_t^l\}$. (b) The strategic intents are propagated to groups via structure-constrained hypergraph attention to produce strategically informed representations $\{\tilde{g}_t^j\}$. These representations are subsequently refined into agent-level tactical guidance m_t^i through the tactical hypergraph, facilitating coordinated decision-making.

Group Density Encoding. To enable the model to capture group size and membership strength [Xu et al., 2019], we further incorporate hyperedge cardinality (i.e., the number of nodes connected by a hyperedge). We define a dual-count representation $\mathbf{c}_t^j$ as:

$$c_t^j = f_{\text{enc}}\Big(\big[\log\big(1+\textstyle\sum_i H_{t,ij}^{(1)} P_{t,ij}^{(1)}\big), \quad \log\big(1+\textstyle\sum_i H_{t,ij}^{(1)}\big)\big]\Big), \tag{6}$$

where the two terms represent the soft membership count, capturing cumulative membership strength, and the hard cardinality, indicating the discrete group size, respectively. The logarithmic transform compresses scale variation and improves numerical stability. The encoder $f_{\text{enc}}(\cdot)$ maps these density features into the latent space.

Finally, the group representation g_t^j is obtained by fusing the aggregated summary $\bar{g}_t^j$ with the density embedding $\mathbf{c}_t^j$:

$$g_t^j = f_{\text{fuse}}([\bar{g}_t^j, \mathbf{c}_t^j]), \tag{7}$$

where f_{fuse} is implemented as a linear layer followed by a nonlinear activation. This design enables g_t^j to capture both aggregated member features and group scale information, supporting subsequent topology construction and cross-level message passing.

Strategic Topology Inference. Following a similar structural inference protocol as in Section 4.1.1, we extend the topology to a higher level by inferring supergroup memberships from group features. Analogous to agent-level assignment, each group is treated as a higher-level node.

First, we compute the supergroup membership strengths $\mathbf{P}_{t,j}^{(2)}$ for each group j:

$$\mathbf{P}_{t,j}^{(2)} = \sigma\Big(f^{(2)}(g_t^j)\Big) \in (0,1)^L, \tag{8}$$

where L denotes the number of latent supergroups and $f^{(2)}$ is an MLP-based encoder.

The group-level incidence matrix $\mathbf{H}_t^{(2)} \in \{0,1\}^{M\times L}$ is then obtained using the same dual-filtering and fallback strategy applied to $\mathbf{P}_{t,j}^{(2)}$:

$$H_{t,jl}^{(2)} = \begin{cases} 1, & \text{if } l \in \text{Top-}k(\mathbf{P}_{t,j}^{(2)}) \text{ and } P_{t,jl}^{(2)} > \delta, \\ 1, & \text{if } l = \arg\max_l P_{t,jl}^{(2)}, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

4.1.3 Supergroup Strategic Aggregation

To capture higher-level coordination context, we apply the same structural aggregation procedure to the group-level features. Analogous to Section 4.1.2, for each supergroup $l \in \{1,\ldots,L\}$, we aggregate its affiliated group features $\{g_t^j\}$ into a strategic supergroup representation S_t^l, which serves as a higher-level hyperedge representation:

$$S_t^l = f_{\text{fuse}}^{(2)}\left([\bar{S}_t^l, c_t^l]\right), \tag{10}$$

$$\bar{S}_t^l = \frac{\sum_{j=1}^M H_{t,jl}^{(2)} \cdot P_{t,jl}^{(2)} \cdot g_t^j}{\sum_{j=1}^M H_{t,jl}^{(2)} \cdot P_{t,jl}^{(2)} + \epsilon}, \tag{11}$$

$$c_t^l = f_{\text{enc}}^{(2)}\Big(\Big[\log\Big(1+\textstyle\sum_j H_{t,jl}^{(2)} P_{t,jl}^{(2)}\Big), \quad \log\Big(1+\textstyle\sum_j H_{t,jl}^{(2)}\Big)\Big]\Big). \tag{12}$$

Here, $\bar{S}_t^l$ denotes the weighted aggregation of group features, while c_t^l encodes both cumulative membership intensity and group cardinality within each supergroup. The functions $f_{\text{fuse}}^{(2)}$ and $f_{\text{enc}}^{(2)}$ are MLP-based fusion and encoding layers at the strategic level.

Through this hierarchical aggregation, we obtain a dual-layer hypergraph structure comprising the agent–group incidence matrix $\mathbf{H}_t^{(1)}$ and the group–supergroup incidence matrix $\mathbf{H}_t^{(2)}$, together with multi-scale representations including group features $\{g_t^j\}$ and supergroup features $\{S_t^l\}$.

4.2 TOP-DOWN HIERARCHICAL HYPERGRAPH COMMUNICATION (THHC)

4.2.1 Strategic Communication: Supergroup to Group

Strategic Guidance Issuance. Top-Down Hierarchical Hypergraph Communication (THHC) employs a Structure-Constrained Hypergraph Attention mechanism, which ensures that each group receives coordination signals conditioned on its structural connectivity and current representation, rather than a uniform global broadcast. Specifically, the attention coefficients and the strategic information aggregated by each group from its connected supergroups are computed as follows:

$$\alpha_{jl}^{(2)} = \frac{H_{t,jl}^{(2)} \exp\left(\frac{(\mathbf{q}_j^{(2)})^\top \mathbf{k}_l^{(2)}}{\sqrt{d}}\right)}{\sum_{l'} H_{t,jl'}^{(2)} \exp\left(\frac{(\mathbf{q}_j^{(2)})^\top \mathbf{k}_{l'}^{(2)}}{\sqrt{d}}\right)}, \tag{13}$$

$$\hat{\mathbf{s}}_t^j = \sum_{l=1}^{L} \alpha_{jl}^{(2)} \cdot \mathbf{v}_l^{(2)}, \tag{14}$$

where $\mathbf{q}_j^{(2)} = f_q^{(2)}(g_t^j)$ denotes the query projection derived from the group feature g_t^j, and $\mathbf{k}_l^{(2)} = f_k^{(2)}(S_t^l)$ and $\mathbf{v}_l^{(2)} = f_v^{(2)}(S_t^l)$ denote the key and value projections from the supergroup feature S_t^l, respectively. All projections are learnable linear mappings. The scalar d is the feature dimension used for scaling to maintain numerical stability. The resulting vector $\hat{\mathbf{s}}_t^j$ represents the aggregated strategic context, constrained by the inferred group–supergroup topology $H_{t,jl}^{(2)}$.

Strategic Guidance Internalization. After receiving the strategic context $\hat{\mathbf{s}}_t^j$, each group updates its representation to incorporate high-level coordination cues. To preserve tactical characteristics while integrating strategic context, we adopt a concatenation-based fusion mechanism:

$$\tilde{g}_t^j = \mathrm{LN}\left(f_{int}^{(2)}\left([\hat{\mathbf{s}}_t^j, g_t^j]\right)\right), \tag{15}$$

where $f_{int}^{(2)}$ is a learnable MLP, and $\mathrm{LN}(\cdot)$ denotes Layer Normalization.

4.2.2 Tactical Communication: Group to Agent

Tactical Guidance Issuance: Next, the updated group representations $\{\tilde{g}_t^j\}$ are propagated to agents through the tactical topology $H_t^{(1)}$. For each agent i, the attention weights $\alpha_{ij}^{(1)}$ are computed as

$$\alpha_{ij}^{(1)} = \frac{H_{t,ij}^{(1)} \exp\left(\frac{(\mathbf{q}_i^{(1)})^\top \mathbf{k}_j^{(1)}}{\sqrt{d}}\right)}{\sum_{j'} H_{t,ij'}^{(1)} \exp\left(\frac{(\mathbf{q}_i^{(1)})^\top \mathbf{k}_{j'}^{(1)}}{\sqrt{d}}\right)}, \tag{16}$$

and the resulting coordination message is defined as

$$m_t^i = f_{out}\left(\sum_{j=1}^{M} \alpha_{ij}^{(1)} \cdot \mathbf{v}_j^{(1)}\right), \tag{17}$$

where $\mathbf{q}_i^{(1)} = f_q^{(1)}(h_t^i)$ is the query projection derived from the agent's local latent state h_t^i, while $\mathbf{k}_j^{(1)} = f_k^{(1)}(\tilde{g}_t^j)$ and $\mathbf{v}_j^{(1)} = f_v^{(1)}(\tilde{g}_t^j)$ denote the key and value projections obtained from the updated group representations, respectively. The function f_{out} is a learnable linear transformation that maps the aggregated coordination context into the message space.

4.3 DECISION AND LEARNING TARGET

4.3.1 Decision Through Communication

For decentralized action selection, agent i concatenates its local observation with the received message:

$$\mathbf{z}_t^i = [o_t^i, m_t^i].$$

Unlike the recurrent encoder used for membership inference, the agent-level recurrent unit focuses on trajectory-based policy learning. The coordinated trajectory state is updated as

$$x_t^i = \mathrm{RNN}(\mathbf{z}_t^i, x_{t-1}^i), \tag{18}$$

where $\mathrm{RNN}(\cdot)$ denotes an agent-specific recurrent unit (implemented as a GRU in our work).

Based on the updated state x_t^i, each agent selects an action using an ϵ-greedy policy:

$$a_t^i = \epsilon\text{-Greedy}\Big(\arg\max_a \ \mathrm{MLP}(x_t^i, a)\Big), \tag{19}$$

where ϵ controls the exploration–exploitation trade-off.

The individual action-value function is then computed as

$$Q_i(x_t^i, a_t^i) = \mathrm{MLP}(x_t^i, a_t^i), \tag{20}$$

where the MLP denotes the individual Q-network that maps the coordinated trajectory state to action utilities. The executed action a_t^i is applied to the environment, and the corresponding Q-value contributes to the joint value decomposition during training.

4.3.2 Learning Target

We optimize the HHC framework using a joint objective that combines topology regularization and reinforcement learning:

$$\mathcal{L} = \lambda \mathcal{L}_p + w \mathcal{L}_s + \mathcal{L}_{td}. \tag{21}$$

The TD loss is the main task objective, while $\mathcal{L}_p$ and $\mathcal{L}_s$ are auxiliary structural regularizers. We optimize them jointly because memberships, topology, messages, and value estimation are coupled end-to-end; staged critic-first training would make topology learning depend on stale value estimates. Similar end-to-end optimization is adopted in hypergraph-based MARL communication methods [Zhu et al., 2024, Liu and Li, 2025].

Polarization Loss. To encourage decisive group assignments, we penalize ambiguous membership strengths:

$$\mathcal{L}_p = \mathbb{E}_t\Bigg[\sum_{i,j} P^{(1)}_{t,ij}\left(1 - P^{(1)}_{t,ij}\right) + \sum_{j,l} P^{(2)}_{t,jl}\left(1 - P^{(2)}_{t,jl}\right)\Bigg]. \tag{22}$$

The coefficient λ is annealed from a small value in early training to a larger value later. This schedule encourages structural exploration initially and gradually promotes sparse and decisive topology formation.

Temporal Stability Loss. To reduce abrupt topology changes over time, we introduce a stability regularizer:

$$\mathcal{L}_s = \mathbb{E}_t\Bigg(\sum_{i,j} \max\Big(0, \left|P^{(1)}_{t,ij} - P^{(1)}_{t-1,ij}\right| - \xi\Big) + \sum_{j,l} \max\Big(0, \left|P^{(2)}_{t,jl} - P^{(2)}_{t-1,jl}\right| - \xi\Big)\Bigg), \tag{23}$$

where w controls the penalty strength and ξ is a tolerance threshold that allows minor fluctuations. This term improves temporal consistency of the learned topology while preserving necessary structural adaptation.

Temporal Difference Loss. The primary reinforcement learning objective is the TD loss:

$$\mathcal{L}_{td} = \mathbb{E}_{\mathcal{D}}\left[\left(Q_{\text{tot}}(\boldsymbol{\tau}_t, \boldsymbol{a}_t, s_t) - y_{\text{tot}}\right)^2\right], \tag{24}$$

where the joint action-value is computed by the mixing network:

$$Q_{\text{tot}}(\boldsymbol{\tau}_t, \boldsymbol{a}_t, s_t) = f_{\text{mix}}\Big(Q_1(x^1_t, a^1_t), \ldots, Q_N(x^N_t, a^N_t), s_t\Big). \tag{25}$$

Here, each message m^i_t is concatenated with the local observation and encoded into the agent representation x^i_t through Eq. 18. The individual utility $Q_i(x^i_t, a^i_t)$ is then computed from this message-conditioned representation, and Q_{tot} is obtained by mixing the individual utilities with the global state s_t. The TD target is defined as

$$y_{\text{tot}} = r_t + \gamma \max_{\boldsymbol{a}'} Q^{\text{target}}_{\text{tot}}(\boldsymbol{\tau}_{t+1}, \boldsymbol{a}', s_{t+1}), \tag{26}$$

where the target network parameters are periodically updated to stabilize training, and samples are drawn from the replay buffer $\mathcal{D}$. The full training procedure is summarized in Appendix B.

5 EXPERIMENTS

We conduct extensive experiments to evaluate the performance of HHC, focusing on the following research questions: i) Does HHC achieve superior performance compared with existing methods (**RQ1**)? ii) Can HHC generalize effectively to more challenging environments (**RQ2**)? iii) What factors contribute to HHC's performance gains (**RQ3**)? iv) Can HHC learn flexible coordination structures via its hierarchical hypergraph representation (**RQ4**)? v) How does the number of groups (M and L) affect HHC (**RQ5**)?

5.1 EXPERIMENT SETTINGS

To evaluate the effectiveness of HHC, we conduct experiments in three MARL environments: Stag Hunt [Böhmer et al., 2020], SMAC [Samvelyan et al., 2019], and SMACv2 [Ellis et al., 2023]. We compare against six representative baselines (QMIX [Rashid et al., 2020], VBC [Zhang et al., 2019], TMC [Zhang et al., 2020], CACOM [Li and Zhang, 2023], HAMS [Wang et al., 2023], and HYGMA [Liu and Li, 2025]), using official implementations and recommended training settings when available. Each experiment is repeated over five independent runs with different random seeds, and results are reported as mean ± standard deviation. Implementation details and hyperparameters are provided in Appendix A. Additional sensitivity analyses of key hyperparameters, including k, δ, λ, and w, are reported in Appendix C. Computational overhead is reported in Appendix F.

5.2 BENCHMARK RESULTS (RQ1)

To evaluate the effectiveness of HHC, we compare it with a diverse set of baseline methods, as shown in Figure 3. We observe that:

(1) Compared with the value-decomposition baseline QMIX, HHC achieves superior performance, highlighting the benefit of communication for improving cooperative decision-making under partial observability. (2) HHC consistently outperforms communication methods without explicit grouping (VBC, TMC, and CACOM), demonstrating that modeling cooperative groups improves coordination effectiveness and communication efficiency. (3) Compared with prior group-based (HAMS) or Hypergraph-based (HYGMA) communication approaches, HHC achieves better performance, suggesting that overlapping memberships enable more flexible coordination patterns and facilitate more effective information sharing.

5.3 MORE CHALLENGING RESULTS (RQ2)

To further evaluate scalability, we test HHC in larger-scale and more crowded settings. Specifically, beyond the largest

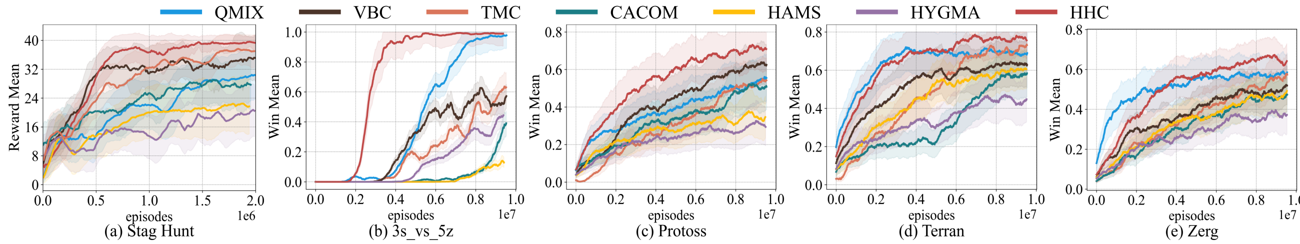


Figure 3: Performance comparison between HHC and baseline methods across benchmark tasks.

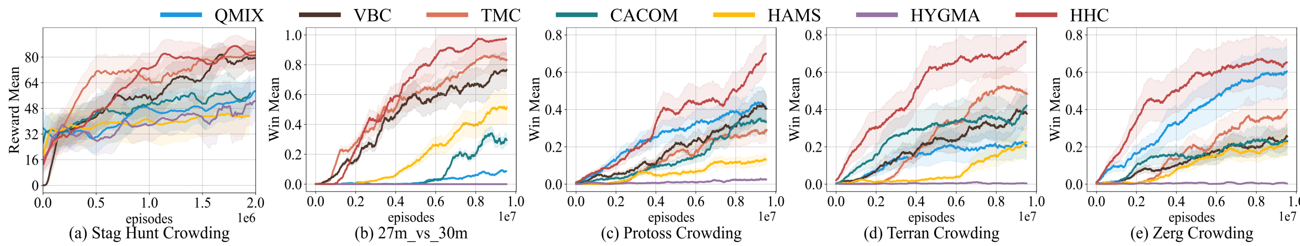


Figure 4: Performance comparison in large-scale and high-density scenarios, including the SMAC map 27m_vs_30m and crowding settings with up to 20 agents in Stag Hunt and SMACv2.

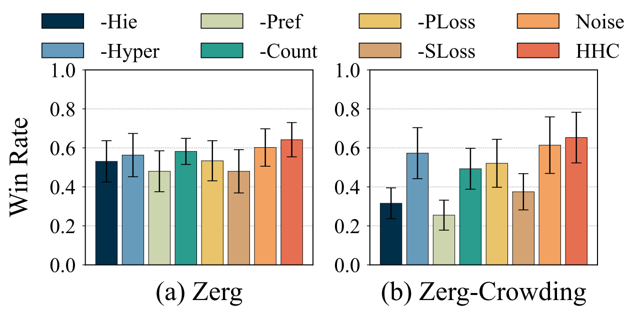


Figure 5: Ablation and robustness study evaluating the effects of hierarchical structure, hypergraph modeling, multi-group membership, group density encoding, structural regularization losses, and noisy trajectory representations.

SMAC map (27m_vs_30m), we increase the number of agents to the maximum supported scale (20 agents) in both Stag Hunt and SMACv2. A larger population introduces greater uncertainty, including more potential communication targets, richer interaction patterns, and increasingly complex grouping structures.

As shown in Figure 4, HHC significantly outperforms all baselines in these challenging scenarios. These results indicate that HHC scales effectively with population size and maintains robust coordination under increased interaction complexity, where existing methods tend to degrade.

5.4 ABLATION STUDY (RQ3)

To better understand the contributions of key components in HHC, we conduct an ablation study in high-uncertainty and high-density scenarios, namely Zerg and Zerg-Crowding. We compare the full HHC model with the following variants:

- **-Hie**: Removes the strategic-level hypergraph, reducing the framework to a single-layer hypergraph model.
- **-Hyper**: Replaces hypergraphs with simple graphs, reducing the model to a hierarchical graph-based architecture.
- **-Pref**: Disables the multi-group membership mechanism, restricting each agent to a single group assignment.
- **-Count**: Removes group density encoding at both the tactical and strategic levels.
- **-PLoss**: Removes the polarization constraint $\mathcal{L}_p$ (Eq. 22) to evaluate its role in producing decisive group assignments.
- **-SLoss**: Removes the stability constraint $\mathcal{L}_s$ (Eq. 23) to assess its impact on temporal consistency of the hypertopology.
- **Noise**: Injects Gaussian noise with standard deviation 0.1 into the trajectory representation h_t before membership inference.

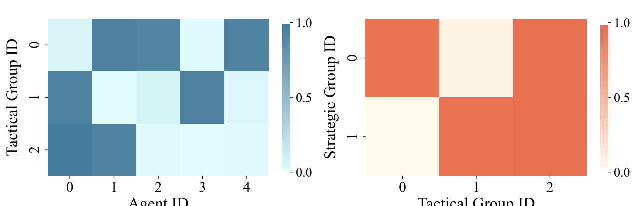
The results are illustrated in Figure 5, from which we observe the following:

(1) Removing the strategic-level hierarchy (-Hie) leads to a clear performance drop, confirming the importance of modeling coordination at multiple abstraction levels. (2) Replacing hypergraphs with simple graphs (-Hyper) also reduces performance, but the degradation is relatively moderate compared with other variants. This suggests that hyperedge-based high-order aggregation provides an additional benefit,

while pairwise graph communication can still capture useful interactions when the hierarchical communication structure is preserved. (3) The significant degradation of -Pref, particularly in the high-load Zerg-Crowding scenario, suggests that modeling multi-group memberships is important for flexible role adaptation in complex environments. (4) Removing group density encoding (-Count) also degrades performance, especially in Zerg-Crowding. This indicates that group-scale information is important for dense coordination scenarios where different group sizes may imply different strategic roles. (5) The inferior performance of -PLoss and -SLoss indicates that the polarization constraint ($\mathcal{L}_p$) and temporal stability constraint ($\mathcal{L}_s$) help produce decisive and temporally consistent agent-to-group assignments, supporting a stable communication topology. (6) Injecting noise into h_t causes a moderate performance drop but does not collapse HHC's performance. This suggests that although noisy trajectory representations can affect membership inference, HHC retains reasonable robustness due to Top-k selection, confidence-threshold filtering, the fallback rule, temporal stability regularization, and end-to-end task feedback.

Overall, these results suggest that HHC's gains mainly come from hierarchical coordination and overlapping multi-group membership, while group density encoding, hypergraph-based high-order aggregation, topology regularization, and robust membership inference provide complementary improvements for learning stable and flexible coordination structures. To further verify that the gain does not mainly come from the fallback rule, we provide group-formation statistics and Welch's t-tests in Appendix D.

5.5 VISUALIZATION OF MULTI-MEMBERSHIP STRENGTH (RQ4)



(a) Tactical Membership $P^{(1)}$ (b) Strategic Membership $P^{(2)}$

Figure 6: Visualization of the average membership strengths $\bar{\mathbf{P}}^{(1)}$ and $\bar{\mathbf{P}}^{(2)}$ over the battle phase of evaluation episodes in the Zerg scenario.

To further examine the coordination structures learned by HHC, we visualize the membership strength matrices $\bar{\mathbf{P}}^{(1)}$ and $\bar{\mathbf{P}}^{(2)}$ in the Zerg scenario, as illustrated in Figure 6. Here, $\bar{\mathbf{P}}^{(1)}$ represents the associations between agents and tactical groups, while $\bar{\mathbf{P}}^{(2)}$ captures how tactical groups are organized into higher-level supergroups. The visualization reveals several notable structural patterns:

(1) Multiple agents exhibit strong memberships to more than one tactical group, and certain tactical groups are associated with multiple supergroups. This demonstrates that HHC supports overlapping memberships, enabling agents and groups to participate in multiple coordination structures rather than being restricted to disjoint partitions. (2) The learned memberships tend to polarize toward values close to 0 or 1, producing sparse and decisive associations. This behavior is consistent with the effect of the polarization constraint $\mathcal{L}_p$, which encourages clear structural assignments and avoids diffuse all-to-all connectivity. (3) Considering both matrices jointly reveals a hierarchical organization in which tactical groups are further aggregated into higher-level coordination structures. This multi-level topology enables information to be propagated across different abstraction levels and provides structural context for coordinated decision-making.

Overall, these visualizations indicate that HHC learns sparse, overlapping, and hierarchical coordination structures that adapt to the interaction patterns in complex multi-agent environments. Additional membership visualizations for crowding scenarios are provided in Appendix E.

5.6 SENSITIVITY OF HIERARCHICAL CAPACITY (RQ5)

In HHC, the number of tactical groups M and the number of strategic groups L define the representational capacity of the hierarchical hypergraph. These hyperparameters do not assume that the true number of functional groups is known in advance. Instead, HHC learns timestep-dependent membership strengths $P_t^{(1)}$ and $P_t^{(2)}$, and the active sparse topology is induced by Top-k selection and the confidence threshold δ.

We select M and L according to task scale and interaction density. Specifically, M is used as a compact tactical-group capacity relative to the number of agents, while L provides a more compact strategic abstraction. For standard Zerg, we use $(M = 3, L = 2)$, while for Zerg-Crowding, we use a larger setting $(M = 5, L = 3)$ due to its higher interaction density. To evaluate the sensitivity of HHC to these hierarchy-capacity hyperparameters, we conduct additional experiments on Zerg and Zerg-Crowding. We fix one of M and L and vary the other. The results are reported in Tables 1 and 2. The results show the following:

(1) HHC is relatively robust to nearby choices of M and L. In Zerg-Crowding, the default setting $(M = 5, L = 3)$ achieves the best mean performance, while nearby settings such as $M = 3, 7$ and $L = 2, 4$ remain competitive. Similar stability is observed in Zerg, where $M = 2$ and $L = 1$ perform close to the default setting $(M = 3, L = 2)$. (2) Too small a hierarchy capacity may limit coordination modeling. When M or L is insufficient, different functional roles may be compressed into the same group, leading to interference among coordination patterns. This is reflected

Table 1: Sensitivity Analysis of the Tactical-Group Capacity M with Fixed L. Results are reported as win rate (%).

Task	Fixed L	$M=2$	$M=3$	$M=4$	$M=5$	$M=7$	$M=9$
Zerg	2	63.8±9.6	**64.2±8.8**	60.4±9.2	57.1±9.1	–	–
Zerg-Crowding	3	–	61.8±11.2	–	**65.3±13.0**	64.2±13.8	51.7±14.0

Table 2: Sensitivity Analysis of the Strategic-Group Capacity L with Fixed M. Results are reported as win rate (%).

Task	Fixed M	$L=1$	$L=2$	$L=3$	$L=4$
Zerg	3	63.1±8.6	**64.2±8.8**	58.1±10.5	–
Zerg-Crowding	5	58.2±9.6	62.2±12.3	**65.3±13.0**	63.1±12.8

by the performance drop of $L = 1$ in Zerg-Crowding. (3) Too large a hierarchy capacity can also hurt performance. Excessive groups introduce redundant structural candidates, enlarge the topology search space, and make membership learning more difficult. This is reflected by the degradation of $M = 9$ in Zerg-Crowding and by the lower performance of larger settings such as $M = 5$ or $L = 3$ in standard Zerg. Overall, M and L should be viewed as structural capacity hyperparameters rather than estimates of the true number of functional groups.

6 CONCLUSION

In this paper, we propose Hierarchical Hypergraph Communication (HHC) for cooperative MARL. HHC models overlapping tactical groups and lifts tactical hyperedges as higher-level units for strategic coordination. With adaptive hierarchical hypergraph construction and top-down structure-constrained communication, HHC enables strategic context to guide tactical communication and agent-level decision-making. Experiments across diverse tasks demonstrate the effectiveness of HHC, particularly in tasks requiring high-order coordination and overlapping group structures. One limitation is that the hierarchy capacity is still predefined. Future work will study adaptive group-number selection, flexible hierarchy-depth design, and more efficient communication.

References

Mhd Saria Allahham, Alaa Awad Abdellatif, Naram Mhaisen, Amr Mohamed, Aiman Erbad, and Mohsen Guizani. Multi-agent reinforcement learning for network selection and resource allocation in heterogeneous multi-rat networks. *IEEE Transactions on Cognitive Communications and Networking*, 8(2):1287–1300, 2022.

Yunpeng Bai, Chen Gong, Bin Zhang, Guoliang Fan, Xinwen Hou, and Yu Lu. Cooperative multi-agent reinforcement learning with hypergraph convolution. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022.

Wendelin Böhmer, Vitaly Kurin, and Shimon Whiteson. Deep coordination graphs. In *International Conference on Machine Learning*, 2020. URL https://arxiv.org/abs/1910.00091.

Alexander Bukharin, Yan Li, Yue Yu, Qingru Zhang, Zhehui Chen, Simiao Zuo, Chao Zhang, Songan Zhang, and Tuo Zhao. Robust multi-agent reinforcement learning via adversarial regularization: Theoretical foundation and stable algorithms. *Advances in Neural Information Processing Systems*, 36:68121–68133, 2023.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Yali Du, Joel Z Leibo, Usman Islam, Richard Willis, and Peter Sunehag. A review of cooperation in multi-agent learning. *arXiv preprint arXiv:2312.05162*, 2023.

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.

Wenzhe Fan, Zishun Yu, Chengdong Ma, Changye Li, Yaodong Yang, and Xinhua Zhang. Towards efficient collaboration via graph modeling in reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 16505–16513, 2025.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3558–3565, 2019.

Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45, 2012.

Shengchao Hu, Ziqing Fan, Li Shen, Ya Zhang, and Dacheng Tao. Communication learning in multi-agent systems from graph modeling perspective. *IEEE Transactions on Knowledge and Data Engineering*, 2026.

Hao Jiang, Dianxi Shi, Chao Xue, Yajie Wang, Gongju Wang, and Yongjun Zhang. Multi-agent deep reinforcement learning with type-based hierarchical group communication. *Applied Intelligence*, 51(8):5793–5808, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Mengxian Li, Qi Wang, and Yongjun Xu. GTDE: Grouped training with decentralized execution for multi-agent actor-critic. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 18368–18376, 2025.

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

Chiqiang Liu and Dazi Li. HYGMA: Hypergraph coordination networks with dynamic grouping for 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 38767–38788. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/liu25af.html.

Kevin R McKee, Ian Gemp, Brian McWilliams, Edgar A Duéñez-Guzmán, Edward Hughes, and Joel Z Leibo. Social diversity and social preferences in mixed-motive reinforcement learning. *arXiv preprint arXiv:2002.02325*, 2020.

Anum Mushtaq, Irfan Ul Haq, Muhammad Azeem Sarwar, Asifullah Khan, Wajeeha Khalil, and Muhammad Abid Mughal. Multi-agent reinforcement learning for traffic flow management of autonomous vehicles. *Sensors*, 23 (5):2373, 2023.

Gonçalo Neto. From single-agent to multi-agent reinforcement learning: Foundational concepts and methods. *Learning theory course*, 2, 2005.

Frans A Oliehoek, Christopher Amato, et al. *A concise introduction to decentralized POMDPs*, volume 1. Springer, 2016.

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

Liviu Panait and Sean Luke. Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11:387–434, 2005.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020.

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

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.

Yuchen Shi, Shihong Duan, Cheng Xu, Ran Wang, Fangwen Ye, and Chau Yuen. Dynamic deep factor graph for multi-agent reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 5887–5896. PMLR, 2019.

Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropagation. *Advances in Neural Information Processing Systems*, 29, 2016.

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.

Karl Tuyls and Ann Nowé. Evolutionary game theory and multi-agent reinforcement learning. *The Knowledge Engineering Review*, 20(1):63–90, 2005.

Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. QPLEX: Duplex dueling multi-agent q-learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Rcmk0xxIQV.

Jiao Wang, Mingrui Yuan, Yun Li, and Zihui Zhao. Hierarchical attention master–slave for heterogeneous multi-agent reinforcement learning. *Neural Networks*, 162: 359–368, 2023.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=ryGs6iA5Km`.

Chao Yu. Hierarchical mean-field deep reinforcement learning for large-scale multiagent systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11744–11752, 2023.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of PPO in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35: 24611–24624, 2022.

Wojciech Zaremba. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

Bin Zhang, Yunpeng Bai, Zhiwei Xu, Dapeng Li, and Guoliang Fan. Efficient policy generation in multi-agent systems via hypergraph neural network. In *International Conference on Neural Information Processing*, pages 219–230. Springer, 2022.

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.

Sai Qian Zhang, Qi Zhang, and Jieyu Lin. Succinct and robust multi-agent communication with temporal message control. *Advances in Neural Information Processing Systems*, 33:17271–17282, 2020.

Delin Zhao, Yanbo Shan, Chang Liu, Shenghang Lin, Yingxin Shou, and Bin Xu. SDHN: Skewness-driven hypergraph networks for enhanced localized multi-robot coordination. *arXiv preprint arXiv:2504.06684*, 2025.

Tianyu Zhu, Xinli Shi, Xiangping Xu, Jie Gui, and Jinde Cao. HyperComm: Hypergraph-based communication in multi-agent reinforcement learning. *Neural Networks*, 178:106432, 2024.

HHC: Hierarchical Hypergraph Communication for Multi-Agent Systems (Supplementary Material)

Qifan Liang[1] **Chenlong Li**[1] **Feiyu Wang**[1] **Jing Fu**[1] **Yixiang Shan**[1] **Lu Guo**[1] **Wei Liu**[1] **Lichang Song**[1] **Ting Long**[*1]

[1]School of Artificial Intelligence, Jilin University, Changchun, Jilin, China, [*]Corresponding author.

A IMPLEMENTATION DETAILS AND HYPERPARAMETERS

Our model was trained on a setup with 4 NVIDIA A40 GPUs, an Intel Gold 5220 CPU, and 504GB of memory, optimized using the Adam optimizer [Kingma and Ba, 2014]. We trained each method for 10^8 environment steps. For reproducibility, we report the hyperparameters used for HHC in Table 3.

Table 3: Hyperparameter Configurations for the HHC Framework.

Category	**Hyperparameter**	**Value**
AHHC Module (Construction)	Num. of groups (M)	3 / 5 for crowding
	Num. of supergroups (L)	2 / 3 for crowding
	Top-k Selection (k)	2
	Confidence threshold (δ)	0.8
	Numerical stability constant (ϵ)	10^{-8}
Architecture (Networks)	RNN/GRU hidden dimension	64
	MLP hidden units	64
	Attention dimension (d)	64
	Message space dimension	64
	Nonlinear activation	ReLU
	Topology activation	Sigmoid
MARL Training (General)	Learning rate (α)	10^{-3}
	Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.999$)
	Replay buffer size	5000
	Batch size	128
	Discount factor (γ)	0.99
	Target update interval	200 episodes
	ϵ-greedy range	$1.0 \rightarrow 0.05$
	ϵ-greedy annealing steps	100,000
Loss Objectives (Weights)	Polarization weight (λ)	$0.001 \rightarrow 0.05$
	Polarization annealing steps	2,000,000
	Stability weight (w)	0.1
	Stability threshold (ξ)	0.05

B HHC ALGORITHM

During training, each episode is processed as follows. At each timestep, AHHC first updates the recurrent trajectory states from local observations and infers the sparse agent–group and group–supergroup topologies. THHC then performs top-down communication, producing the agent-specific message m_t^i for each agent. Each agent concatenates m_t^i with its local observation, updates its decision-state representation, and selects an action using an ϵ-greedy policy. The resulting episode is stored in the replay buffer. During optimization, mini-batches of episodes are sampled from the buffer, and the model is updated end-to-end by minimizing the TD loss together with the polarization and temporal stability regularizers. For the readers' convenience, pseudocode of HHC has been summarized as Algorithm 1.

Algorithm 1 Training of Hierarchical Hypergraph Communication (HHC)

Initialize parameters for Mixing network, Communication module, and Agent networks.
Initialize target networks and replay buffer $\mathcal{D}$.
for each episode **do**
  Reset environment and initialize trajectory states $\{h_0^i, x_0^i\}_{i=1}^N$ to $\mathbf{0}$.
  for each timestep $t = 0$ to $T-1$ **do**
    {**Adaptive Hierarchical Hypergraph Construction (AHHC)**}
    for each agent $i \in \{1, \ldots, N\}$ **do**
      Observe o_t^i, take previous action a_{t-1}^i, and update trajectory latent state h_t^i via Eq. (1).
      Compute membership preference $\mathbf{P}_{t,i}^{(1)}$ and sparse incidence $H_{t,ij}^{(1)}$ via Eqs. (2)–(4).
    end for
    for each group $j \in \{1, \ldots, M\}$ **do**
      Aggregate agent features into group representation g_t^j via Eqs. (5)–(7).
      Infer supergroup membership $\mathbf{P}_{t,j}^{(2)}$ and sparse topology $H_{t,jl}^{(2)}$ via Eqs. (8)–(9).
    end for
    for each supergroup $l \in \{1, \ldots, L\}$ **do**
      Distill strategic features S_t^l via Eqs. (10)–(12).
    end for
    {**Top-Down Hierarchical Hypergraph Communication (THHC)**}
    Update strategically-informed group features $\{\tilde{g}_t^j\}_{j=1}^M$ via Eqs. (13)–(15).
    for each agent $i \in \{1, \ldots, N\}$ **do**
      Generate tactical message m_t^i and individual Q_i via Eqs. (16)–(20).
      Select and execute action a_t^i using ϵ-greedy policy.
    end for
  end for
  Store the full episode in $\mathcal{D}$.
  Sample a mini-batch of episodes from $\mathcal{D}$.
  Compute and minimize joint loss $\mathcal{L}$ according to Eqs. (21)–(26).
  Periodically update target network parameters.
end for

C SENSITIVITY ANALYSIS

C.1 SENSITIVITY OF CONFIDENCE THRESHOLD

As the confidence threshold δ controls the sparsity of hierarchical membership assignments, we conduct a sensitivity study by evaluating $\delta \in \{0.2, 0.4, 0.6, 0.8\}$ in both Zerg and Zerg-Crowding scenarios. As illustrated in Figure 7, we observe the following:

(1) HHC maintains stable and competitive performance across a wide range of δ values. This robustness can be attributed in part to the fallback mechanism, which ensures that each node (agent or group) remains connected to at least one hyperedge, thereby preventing degenerate empty assignments during sparsification. (2) Higher thresholds lead to improved performance, with $\delta = 0.8$ achieving the best results among the tested values. A stricter threshold filters low-confidence associations and promotes a more decisive and structured communication topology. (3) The improved performance at larger δ values is

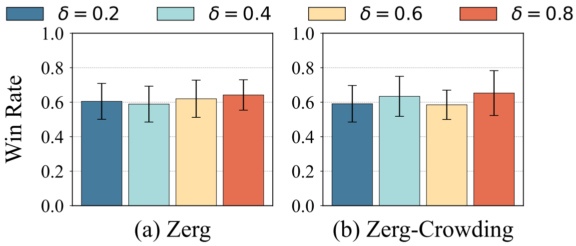


Figure 7: Sensitivity of the confidence threshold δ.

Table 4: Sensitivity Analysis of the Top-k Value. Results are reported as win rate (%).

Task	$k = 1$	$k = 2$	$k = 3$
Zerg	56.3±11.0	**64.2±8.8**	–
Zerg-Crowding	57.3±11.0	**65.3±13.0**	59.2±13.9

Table 5: Sensitivity Analysis of the Loss Weights. Results are reported as win rate (%). For λ, the tested values denote annealing upper bounds.

Weight	Task	0.01	0.05	0.1	0.5
λ	Zerg	61.6±9.6	**64.2±8.8**	62.3±7.9	55.1±9.1
	Zerg-Crowding	55.9±13.8	**65.3±13.0**	61.4±15.7	52.5±14.6
w	Zerg	58.3±9.2	61.9±10.6	**64.2±8.8**	58.2±10.3
	Zerg-Crowding	52.4±15.9	62.0±13.4	**65.3±13.0**	56.4±12.8

consistent with the effect of the polarization constraint $\mathcal{L}_p$, which encourages clear structural assignments. Together, these mechanisms promote sparse and stable coordination structures that support effective multi-agent cooperation.

C.2 SENSITIVITY OF TOP-k

The Top-k value controls how many high-confidence memberships are retained during sparse topology inference. As shown in Table 4, we observe the following:

(1) $k = 2$ achieves the best mean performance in both Zerg and Zerg-Crowding, suggesting that allowing each node to connect to more than one group is beneficial for modeling overlapping memberships. (2) When $k = 1$, the topology becomes closer to a single-group assignment, which weakens cross-group participation and reduces coordination flexibility. (3) Increasing k further does not necessarily improve performance. In Zerg-Crowding, $k = 3$ performs worse than $k = 2$, indicating that overly dense communication may introduce redundant or noisy group associations.

Overall, we use $k = 2$ to balance overlapping group participation and sparse communication.

C.3 SENSITIVITY OF LOSS WEIGHTS

We further analyze the sensitivity of the loss weights λ and w in Table 5. Here, λ controls the polarization loss $\mathcal{L}_p$, and w controls the temporal stability loss $\mathcal{L}_s$. For λ, the default setting anneals it from 0.001 to 0.05 within 2M environment steps, and we vary the annealing upper bound. For w, we test different fixed values.

The results show the following: (1) Both loss weights show a middle-range preference. The default setting, i.e., annealing λ to 0.05 and setting $w = 0.1$, achieves the best mean performance in both Zerg and Zerg-Crowding. (2) HHC is not overly brittle to nearby values. For example, $\lambda = 0.1$ and $w = 0.05$ only lead to moderate degradation compared with the default setting. (3) Extreme settings hurt performance more clearly. A very large polarization weight, such as $\lambda = 0.5$, may over-regularize membership assignments, while too small or too large w may either fail to stabilize the topology or overly suppress necessary structural adaptation.

Overall, we use the default setting $\lambda : 0.001 \rightarrow 0.05$ and $w = 0.1$ to balance decisive membership assignment and temporal topology stability.

Table 6: Group-Formation Statistics and Welch Tests on Zerg-Crowding. TF/SF: tactical/strategic fallback ratio; TM/SM: memberships per agent/tactical group; TG/SG: active tactical/strategic groups. Win rates are reported in %.

Variant	TF ↓	TM ↑	TG ↑	SF ↓	SM ↑	SG ↑	Win ↑	Gap	p
HHC	$0.03{\pm}0.02$	$1.75{\pm}0.18$	$4.89{\pm}0.10$	$0.05{\pm}0.03$	$1.32{\pm}0.14$	$2.62{\pm}0.25$	$65.3{\pm}13.0$	–	–
-Pref	N/A	1.00	$3.53{\pm}0.41$	N/A	1.00	$1.84{\pm}0.45$	$25.5{\pm}7.7$	39.8	0.0008
-SLoss	$0.09{\pm}0.06$	$1.48{\pm}0.25$	$4.10{\pm}0.55$	$0.16{\pm}0.08$	$1.23{\pm}0.16$	$2.15{\pm}0.40$	$37.5{\pm}9.3$	27.8	0.0055
-PLoss	$0.14{\pm}0.06$	$1.35{\pm}0.23$	$4.41{\pm}0.40$	$0.15{\pm}0.06$	$1.19{\pm}0.12$	$2.30{\pm}0.35$	$52.1{\pm}12.3$	13.2	0.1380
-Hie	$0.05{\pm}0.03$	$1.68{\pm}0.20$	$4.72{\pm}0.24$	N/A	N/A	N/A	$31.6{\pm}7.9$	33.7	0.0019
-Hyper	N/A	N/A	N/A	N/A	N/A	N/A	$57.3{\pm}13.0$	8.0	0.3599

D GROUP FORMATION ANALYSIS

To examine whether the performance gain of HHC mainly comes from the fallback mechanism rather than learned overlapping group formation, we conduct a group-formation analysis on the representative challenging task Zerg-Crowding in Table 6. We report six statistics: the fallback ratio, the average number of memberships per node, and the number of active groups at both the tactical and strategic levels. The fallback ratio measures how often the argmax fallback is triggered because all membership strengths are below the confidence threshold. The membership count measures whether the learned topology forms overlapping memberships rather than degenerating to single-group assignment. The number of active groups measures whether group usage collapses to only a few groups.

We also perform Welch's two-sided t-test on seed-wise win rates to compare HHC with each ablated variant. Here, "Tac." denotes the tactical level and "Strat." denotes the strategic level. N/A means that the statistic is not defined for the corresponding variant: *-Pref* disables overlapping membership by construction, *-Hie* removes the strategic level, and *-Hyper* replaces hyperedges with pairwise graph edges.

The results show the following:

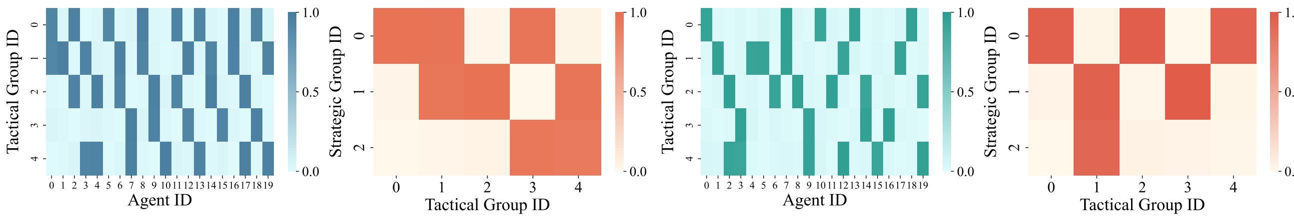
(1) HHC rarely relies on the fallback mechanism. The fallback ratios are 0.03 at the tactical level and 0.05 at the strategic level, indicating that most connections are induced by learned membership strengths rather than by the argmax fallback rule. (2) HHC learns overlapping and non-collapsed group structures. The average tactical memberships per agent is 1.75, and the average strategic memberships per tactical group is 1.32, both larger than one. The numbers of active tactical and strategic groups are 4.89 and 2.62, close to the configured capacities $M = 5$ and $L = 3$. This suggests that HHC does not degenerate to single-group assignment or collapse to only a few active groups. (3) The ablations clarify the role of different components. *-Pref*, which removes overlapping membership, causes a large and statistically significant drop. *-SLoss* increases fallback ratios, reduces active group usage, and also significantly degrades performance. *-Hie* preserves tactical grouping but removes the strategic level, leading to another significant drop. These results support the importance of overlapping membership, temporal stability, and strategic-level abstraction. (4) The effects of *-PLoss* and *-Hyper* are more moderate. Their Welch p values are above 0.05, so we do not claim statistically significant differences for these two variants. This is consistent with the ablation results in Figure 5, where hypergraph modeling and polarization regularization provide complementary but less dominant improvements.

Overall, the gain of HHC cannot be explained by the fallback mechanism alone. Instead, HHC achieves low fallback usage while maintaining overlapping, active, and hierarchical group structures.

E VISUALIZATION OF MULTI-MEMBERSHIP STRENGTH

Figure 8 further visualizes the learned membership strengths in crowding scenarios. Panels (a) and (c) show the agent-to-tactical-group memberships, while Panels (b) and (d) show the tactical-group-to-strategic-group memberships. Darker colors indicate stronger memberships.

The visualizations show the following: (1) At the tactical level, several agents have non-zero memberships to more than one group, indicating that HHC does not simply reduce to single-group argmax assignment. (2) The memberships are sparse and structured rather than uniformly dense, suggesting that HHC learns selective group participation instead of relying on all-to-all communication. (3) At the strategic level, tactical groups are also associated with multiple strategic groups, showing that overlapping structures are preserved across hierarchy levels. In the crowding hunting scenarios, the tactical



(a) Tact. Mem. of Zerg Crowding (b) Strat. Mem. of Zerg Crowding (c) Tact. Mem. of Stag Hunt Crowding (d) Strat. Mem. of Stag Hunt Crowding

Figure 8: Visualization of learned membership strength matrices $\bar{\mathbf{P}}^{(1)}$ and $\bar{\mathbf{P}}^{(2)}$ in crowding scenarios. Panels (a,c) show tactical memberships, and Panels (b,d) show strategic memberships. Darker colors indicate stronger memberships.

Table 7: Computational Overhead Comparison on Zerg and Zerg-Crowding. Inference latency is reported as milliseconds per environment step.

Metric	Task	QMIX	HAMS	HYGMA	HHC
Training time	Zerg	5h49min	7h01min	8h47min	7h15min
	Zerg-Crowding	11h19min	12h38min	14h04min	12h49min
Inference latency	Zerg	0.135 ms	0.216 ms	0.260 ms	0.233 ms
	Zerg-Crowding	0.159 ms	0.245 ms	0.285 ms	0.270 ms
GPU memory	Zerg	1526 MB	1690 MB	2518 MB	1950 MB
	Zerg-Crowding	3808 MB	6788 MB	10088 MB	8074 MB

layer can be interpreted as capturing local coordination units, while the strategic layer connects related tactical groups that need to share a higher-level coordination context. For example, groups that observe the target and groups that move toward the target can be associated at the strategic level to support coordinated capture. This suggests that the second hypergraph layer models relations among tactical groups rather than simply repeating agent-level aggregation.

F COMPUTATIONAL OVERHEAD

We analyze the additional overhead introduced by HHC in Table 7. Compared with independent value-decomposition methods, the extra computation mainly comes from hierarchical topology construction and hierarchical message passing. HHC does not enumerate dense agent–agent pairs for communication. Instead, it uses compact tactical and strategic group capacities M and L, together with Top-k sparsification.

At each timestep, membership inference and sparse topology construction require evaluating agent–group and group–supergroup memberships, with cost $O(NM + ML)$, where N is the number of agents. After Top-k sparsification, each agent connects to at most k tactical groups and each tactical group connects to at most k strategic groups. Therefore, the active hierarchical message-passing cost is $O(k(N + M))$, instead of dense all-to-all communication over $O(N^2)$ agent pairs. Since M, L, and k are compact in our experiments, the practical overhead remains moderate.

We further measure training time, inference latency, and peak GPU memory on Zerg and Zerg-Crowding under the same hardware and evaluation settings. Inference latency is measured as the average wall-clock time of action selection per environment step during evaluation, excluding environment stepping, replay storage, and logging. For parallel runners, the latency is normalized by the number of active parallel environments.

The results show that HHC introduces additional overhead compared with QMIX, which is expected because HHC performs hierarchical topology construction and two-level communication. However, compared with more related group or hypergraph communication baselines, the overhead is moderate. HHC has training time and inference latency close to HAMS and lower than HYGMA on both tasks. For memory usage, HHC requires more memory than QMIX and HAMS due to the additional hierarchy construction and communication modules, but remains lower than HYGMA.

G ADDITIONAL COMPARISON WITH COMMFORMER

To further compare HHC with recent learnable communication methods, we include CommFormer as an additional baseline on two representative high-uncertainty tasks, Zerg and Zerg-Crowding. As shown in Table 8, HHC outperforms CommFormer [Hu et al., 2026] on both tasks, supporting the benefit of hierarchical group-level coordination in complex cooperative scenarios.

Table 8: Additional Comparison with CommFormer. Results are reported as win rate (%).

Task	CommFormer	HHC
Zerg	56.1 ± 9.5	$\mathbf{64.2 \pm 8.8}$
Zerg-Crowding	60.2 ± 9.0	$\mathbf{65.3 \pm 13.0}$