Complex & Intelligent Systems (2026) 12:225
https://doi.org/10.1007/s40747-026-02356-7

ORIGINAL ARTICLE

LLM-guided graph neural coordination framework for cooperative multi-agent reinforcement learning

Zihao Kuang[1] · Zengmin Xu[1,2,4] · Linkai Wan[1] · Chunjiong Zhang[3]

Received: 25 November 2025 / Accepted: 5 May 2026 / Published online: 8 June 2026
© The Author(s) 2026

Abstract
Multi-Agent Reinforcement Learning (MARL) relies on accurate modeling of inter-agent interactions for effective coordination. Existing graph-based methods fail to infer dynamic coordination structures from high-level semantics and still rely on trajectory-based interaction patterns, leading to suboptimal policies and unstable credit assignment. Moreover, agents are often hindered by manually designed static rewards that fail to adapt to dynamic contexts and cannot address the challenges of sparse reward environments. To address these limitations, we propose the LLM-Guided Graph Neural Coordination Framework (LLM-GNCF), which establishes a systematic integration between Large Language Model semantic reasoning and Graph Neural Coordination. LLM-GNCF leverages an LLM to dynamically construct a Team-Adaptive Coordination Graph (TACG) based on real-time LLM-guided strategic semantics and actions, which are validated by the Tactical Critic model. This graph structure serves as an effective structural prior to guide effective localized information exchange at both the agent-level and the group-level, thus facilitating cooperative policy optimization. Crucially, we introduce an LLM-empowered latent reward shaping method supported by a Chain of Aggregation mechanism that aggregates multi-frame information to provide fine-grained, context-aware semantic feedback while managing computational overhead. This signal further enhances coordination and strategy refinement, particularly under sparse reward conditions. Furthermore, LLM-GNCF employs a two-stage training paradigm in which LLM guidance reduces blind exploration in MARL and accelerates training convergence. Experiments on the challenging StarCraft II micromanagement tasks suggest that the proposed integrated framework achieves competitive performance compared to representative baselines in terms of coordination efficiency and policy generalization.

Keywords Graph-based multi-agent reinforcement learning · LLM-guided graph neural coordination · LLM-empowered latent reward shaping · StarCraft II multi-agent challenge

✉ Zengmin Xu
xzm@guet.edu.cn

✉ Chunjiong Zhang
cjz@guet.edu.cn

Zihao Kuang
kzh@mails.guet.edu.cn

Linkai Wan
23072302011@mails.guet.edu.cn

1 School of Mathematics and Computing Science, Guangxi Colleges and Universities Key Laboratory of Data Analysis and Computation, Guilin University of Electronic Technology, Guilin 541004, China

2 Center for Applied Mathematics of Guangxi (GUET), Guilin 541004, China

3 School of Artificial Intelligence, Guilin University of Electronic Technology, Guilin 541004, China

4 Anview.ai, Guilin 541010, China

Introduction

Multi-Agent Reinforcement Learning(MARL) [1–3] is a promising framework for cooperative autonomy and has attracted increasing attention in recent years. Its applications span a wide range of domains, including autonomous driving [4, 5], wireless communications [6, 7], multiplayer gaming [8, 9] and urban traffic management [10, 11]. However, achieving effective coordination in complex cooperative tasks remains a formidable challenge. A common paradigm involves value decomposition methods (e.g., VDN [12] and QMIX [13]) and their extensions [14–17], which implicitly assume a coordination structure. Nonetheless, inferring the dynamic graph topology inherent in MARL remains a significant, open problem. Current methods for learning the

underlying graph in MARL can generally be categorized into four types, including constructing complete graphs by directly connecting all agents [18, 19], employing attention mechanisms to compute fully connected weighted graphs [20, 21], designing drop-edge criteria to generate sparse graphs [22–24] and developing approaches to infer group-aware coordination graphs [25, 26]. Despite this progress made, existing graph-based methods typically infer coordination structures from low-level features and thus cannot exploit high-level semantic information to construct truly dynamic interaction topologies. In addition, they fail to inject timely corrective feedback from external knowledge sources when erroneous behaviours are observed during training. Moreover, reward shaping still relies on manually designed, static functions, which generalize poorly across dynamic scenarios. From a broader perspective, these limitations are also consistent with the AutoML view that reducing hand-crafted design and automating key stages of the learning pipeline can lower development complexity, improve repeatability, and enhance efficiency in increasingly complex data-driven tasks [27]. These limitations often make it inefficient for agents to acquire task-specific knowledge from scratch and hinder their ability to capture high-level tactical intents, such as feints or baiting. Relying solely on low-level features, such as coordinate distances, agents cannot comprehend the underlying high-level strategy semantics.

Recent advances show that Large Language Model (LLM) excels at complex zero-shot reasoning and world knowledge integration [28, 29]. This positions LLM as a promising tool to address the core challenges of multi-agent coordination [30]. In recent years, research on the integration of LLM and MARL has mainly fallen into three categories: One line of work treats the LLM as a high-level planner that generates abstract action plans based on historical experiences and current environmental states, while low-level RL policies learn and implement the corresponding primitive control behaviours [31, 32]. Another direction leverages the reasoning and generative capabilities of LLM to transform linguistic knowledge into dynamic reward signals for designing, adjusting or redistributing reward functions [33, 34]. A further approach focuses on communication and coordination, utilizing LLM language understanding and generation abilities to enable natural language interactions among agents, thereby enhancing semantic alignment and cooperative generalization in multi-agent systems [35, 36]. Despite these advances, these three lines of work share several structural limitations. Critically, there is a pronounced gap between high-level strategic semantics and the underlying topology. The LLM is typically utilized as an external module without leveraging adaptive graph structures to use this high-level information for coordinating agent grouping and cooperation, making it impossible to map strategic intents at the semantic level onto the underlying coordination graph structure. Moreover, most existing LLM–based MARL frameworks operate in a fragmented manner, they either rely on one-off offline planning based on high-level strategic semantics or treat real-time queries as mutually independent, stateless instances of high-level strategic semantics to generate immediate instructions or reward signals. Crucially, these methods fail to establish temporal correlations among high-level strategic semantics, thereby lacking a mechanism to effectively integrate the long-term evolution of semantic guidance with the dynamic MARL training process. Moreover, they have rarely been validated on challenging multi-agent coordination benchmarks. StarCraft Multi-Agent Challenge (SMAC) [8, 37], with its complex and highly cooperative task settings, has evolved into a representative benchmark for evaluating MARL algorithms [38]. Although various MARL methods have proven effective, there is still a lack of widely recognized benchmarks and methods that integrate LLM with MARL. Only a few preliminary studies have attempted to convert game state into textual description and employ LLM as high-level strategic planner, without integrating them with reinforcement learning [39].

Motivated by the limitations of existing methods and their inadequate performance on SMAC, we propose the LLM-Guided Graph Neural Coordination Framework (LLM-GNCF) for Cooperative Multi-Agent Reinforcement Learning. To bridge the gap between high-level semantics and underlying topology, LLM-GNCF establishes a systematic integration between LLM reasoning and graph-based MARL. The LLM dynamically processes textual environmental observations to extract real-time, strategic semantic dependencies. Crucially, these dependencies are explicitly utilized to guide agent grouping and calculate inter-agent relationships, thereby constructing a context-aware Team-Adaptive Coordination Graph (TACG). To resolve the issue of temporal credit assignment in cooperative tasks, the framework introduces an LLM-empowered latent reward shaping method. This module leverages the Chain of Aggregation (CoA) mechanism to aggregate multi-frame information, ensuring temporal semantic consistency and providing fine-grained feedback under sparse reward settings. A key aspect of our methodology is that it adopts an LLM-guided two-stage MARL training framework. In the LLM-guided pre-training stage, the LLM acts as a high-level commander with a built-in evaluation and correction mechanism to detect and rectify potential strategy errors while mitigating hallucination risks. In the subsequent reinforcement learning stage, agents further optimize their cooperative strategies by inheriting the coordination graph structure constructed from the strategic semantic dependencies. This approach enables agents to acquire high-level cooperative experience and contextual understanding rather than relying solely on pure trial-and-error exploration.

Our main contributions are:

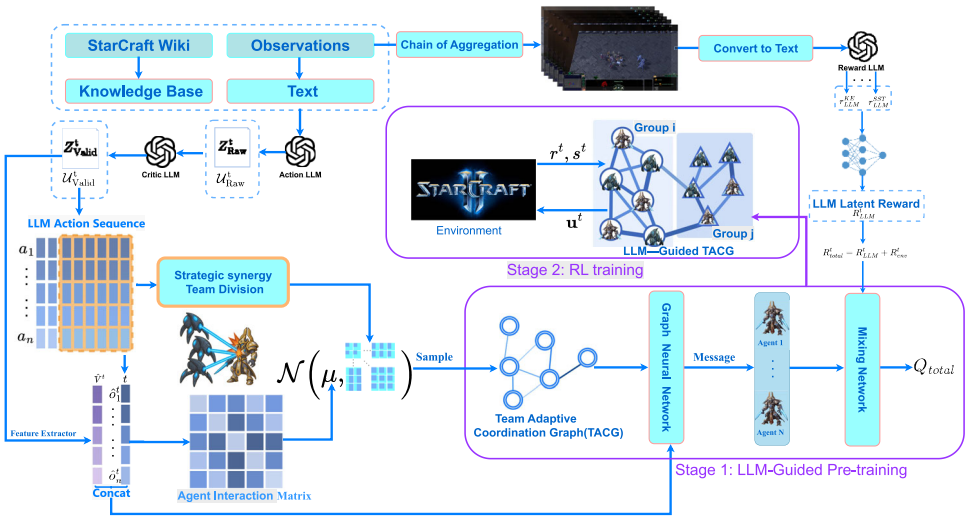

Fig. 1 The framework of our method consists of two training stages. In the first stage, the LLM provides valid tactical actions and temporal semantic information to infer cooperation demands and capture group-level dependencies, while generating map-specific latent rewards to guide strategy learning. In the second stage, the LLM-guided graph structure is refined through reinforcement learning to optimize agent policies and enhance coordinated behaviors

- We present an integrated framework that leverages LLM for dynamic coordination graph construction by extracting semantic dependencies to model pairwise cooperation demands and capture group-level relationships simultaneously. We also provide an evaluation setting on SMAC for studying LLM-integrated MARL frameworks.
- We introduce a principled two-stage training paradigm, allowing agents to benefit from high-level cooperative priors early in training. This, together with the critic mechanism, facilitates the construction of a richer cooperative relationship model in our framework.
- We develop an LLM-empowered dynamic reward function that promotes specialized cooperative strategies through targeted semantic feedback. The Chain of Aggregation mechanism is concurrently introduced to effectively manage computational overhead and reduce the token consumption of the LLM during training.

Related work

Dynamic topology inference and graph-based MARL

Recent advances in the StarCraft Multi-Agent Challenge (SMAC) leverage latent interaction graphs to improve coordination and scalability [40, 41]. Methods such as DGN [42], DICG [20], and HGAP [43] employ attention mechanisms or Graph Neural Network (GNN) [44–46] to infer inter-agent dependencies and guide information exchange, while variational approaches [47] further model interaction graphs as latent variables. Building upon these ideas, group-aware and hierarchical coordination frameworks capture higher-level relational structures and inter-group dependencies, thereby enhancing scalability and cooperative efficiency in decentralized MARL. VAST [48] factorizes value functions over subgroups to adapt to different contexts. GoMARL [25] introduces a select-and-kick-out mechanism to achieve automatic grouping without domain knowledge, thereby improving cooperation efficiency. GACG [26] jointly models pairwise cooperation needs and cross-trajectory group dependencies. The primary challenge of these approaches is their reliance on low-level feature similarity or heuristic rules to implicitly infer interaction structures. Consequently, they often do not fully incorporate explicit high-level strategic guidance and temporal coordination signals crucial for complex tasks. Furthermore, in decentralized learning scenarios, the inferred coordination topologies are often non-adaptive, hindering efficient and context-sensitive communication. In contrast, our approach integrates explicit strategic semantic information from an LLM into the graph construction process, enabling each agent to adaptively select task-relevant neighbors based on both physical graph topology and semantic context. This design facilitates communication-efficient and locally interpretable cooperation.

Integration of LLM in MARL

Large Language Model (LLM) have been increasingly applied to high-level planning in general reinforcement learning, providing semantic reasoning and task decomposition capabilities [49, 50].*MineDojo* [51] transforms natural language goals into structured task plans.*Voyager* [52] achieves autonomous task generation and skill reuse and *Plan4MC* [53] performs hierarchical planning to decompose long-horizon objectives into executable subtasks. These studies demonstrate the role of LLM as semantic policy generator that bridge natural language reasoning and reinforcement learning. However, these pioneering systems typically focus on single-agent control or lack specialized mechanisms for robust multi-agent coordination.

The application of LLM in MARL broadly falls into three categories: high-level planning, communication and reward modeling. For high-level planning, L2M2 [31] adopts a hierarchical architecture in which an LLM agent performs strategic planning and subtask allocation, while low-level MARL agents execute primitive actions.YOLO-MARL [32] uses a single LLM call to generate an environment-specific task-assignment planning function, which is then used to guide subsequent MARL training. For communication, in *LangGround* [35], agent communication is mapped to a human-interpretable linguistic space, enabling semantic alignment and cooperative generalization. Similarly, in social reasoning environments [36], LLM-based agents engage in dialogic reasoning to infer the intentions and latent beliefs of others, thereby promoting strategic coordination and belief propagation. For reward modeling, *LaRe* [33] utilizes LLM to generate interpretable latent semantic reward that improve credit assignment, whereas *LAMARL* [34] employs LLM to evaluate high-level cooperative behaviors, providing strategy-relevant reward signals for multi-agent systems. Despite the significant improvements brought by these methods, most existing LLM-based MARL approaches integrate an LLM into only a single component of the system and rarely provide a unified framework that simultaneously learns the coordination-graph structure and incorporates latent reward shaping. As a result, their effectiveness can be further enhanced in complex, real-time adversarial environments, where dynamically evolving topologies and precise, temporally grounded coordination are required.

Few efforts to date have explored the integration of LLM within the SMAC. In particular, *TextStarCraft* II [39] introduces a text-based benchmark and a macro-level reasoning framework. However, this approach has not been integrated with decentralized MARL. To address these challenges, our method introduces an integrated LLM guidance mechanism spanning the two-stage training framework, covering semantic coordination graph generation, LLM-guided action supervision and adaptive latent reward shaping. This coordinated design drives the dynamic evolution of the MARL interaction graph based on real-time LLM-guided strategic semantics. Within the SMAC, this framework enables agents to construct temporally-grounded semantic coordination patterns and dynamically adaptive decentralized cooperation topologies, thereby facilitating effective coordination and interaction modeling in complex adversarial scenarios.

Preliminaries

Decentralized partially observable MDP

We consider cooperative multi-agent tasks modeled as a Dec-POMDP [54], defined by the tuple

$$\langle \mathcal{A}, \mathcal{S}, \{\mathcal{U}_i\}_{i=1}^n, \mathcal{P}, \{\mathcal{O}_i\}_{i=1}^n, \{\pi_i\}_{i=1}^n, \mathcal{R}, \gamma \rangle,$$

where $\mathcal{A}$ denotes the finite set of n agents and $s \in \mathcal{S}$ represents the true state of the environment. The set of actions available to agent i is $\mathcal{U}_i$ and its partial observation space is $\mathcal{O}_i$. The joint action space is $\mathcal{U} = \mathcal{U}_1 \times \ldots \times \mathcal{U}_n$, while the joint observation is $\mathbf{o}^t = (o_1^t, \ldots, o_n^t)$. At each time step t, each agent a_i receives a partial observation $o_i^t \in \mathcal{O}_i$ and selects an action $u_i^t \in \mathcal{U}_i$ according to its individual policy π_i. The dynamics are governed by the transition probability $\mathcal{P}(s^{t+1} \mid s^t, \mathbf{u}^t)$, where $\mathbf{u}^t = (u_1^t, \ldots, u_n^t)$ is the joint action, leading to a shared global reward $r^t = \mathcal{R}(s^t, \mathbf{u}^t)$ for all agents. We consider episodic tasks that generate finite-length trajectories

$$(s^0, \{o_i^0\}_{i=1}^n, \mathbf{u}^0, r^0, \ldots, s^T, \{o_i^T\}_{i=1}^n),$$

where T denotes the episode length. The agents collectively seek to maximize the expected discounted cumulative return (Global Return):

$$Q_{\text{tot}}(s^0, \mathbf{u}^0) = \mathbb{E}_{s^{0:T}, \mathbf{u}^{0:T}} \left[\sum_{t=0}^{T} \gamma^t r^t \,\middle|\, s^0, \mathbf{u}^0 \right],$$

where $\gamma \in [0, 1)$ is the discount factor.

Coordination graph and group structure

In multi-agent cooperative tasks, the dependency and interaction structure among agents can be represented by a Coordination Graph(CG). Given n agents, the coordination graph is formally defined as

$$\mathcal{C} = \{\mathcal{A}, \mathcal{E}\},$$

where $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ denotes the set of agents (nodes) and $\mathcal{E}$ represents the set of edges or relational dependencies between agents. A key component is the adjacency matrix $\mathbf{C} \in \mathbb{R}^{n \times n}$, which concisely captures the global interaction structure among agents.

Building upon this, we further introduce a group structure to model the hierarchical cooperation relationships among agents. Given the agent set $\mathcal{A}$, the group set is denoted as

$$\mathcal{G} = \{g_1, g_2, \ldots, g_m\}, \quad 1 \le m \le n.$$

Each group g_i contains n_i agents, i.e.,

$$g_i = \{a_{i1}, a_{i2}, \ldots, a_{in_i}\} \subseteq \mathcal{A},$$

satisfying the properties of a disjoint partition: $g_i \cap g_j = \emptyset$ for $i \neq j$ and $\bigcup_{i=1}^m g_i = \mathcal{A}$. This grouping provides a hier-

archical organization for multi-agent systems, facilitating structured coordination and group-level reasoning in subsequent modeling.

Method

The proposed framework, illustrated in Fig. 1, is designed to achieve LLM-guided tactical coordination and decision-making in cooperative multi-agent reinforcement learning (MARL). To mitigate inefficient exploration in complex state spaces and accelerate the learning of cooperative policies, we adopt a two-stage training paradigm.

LLM-guided semantic and action reasoning

In multi-agent reinforcement learning (MARL), traditional policy models rely on local observations to directly generate low-level actions, which often limits their ability to capture overarching tactical semantics and high-level strategic intent. To address this limitation, and inspired by the emerging potential of LLM to provide strategic-level reasoning for MARL [31, 55–57], we introduce LLM as a high-level policy generator, leveraging its inherent semantic understanding and reasoning capabilities to produce high-level tactical action priors and strategic semantic guidance that serve as policy-level supervision for multi-agent coordination. This design effectively complements conventional local policy models by enhancing battlefield semantic modeling and cooperative reasoning.

To provide high-quality tactical demonstrations within MARL, we design an LLM-guided semantic and action reasoning module that transforms environmental observations into structured textual representations and produces executable tactical actions. At timestep t, the module takes the joint observation $\mathbf{o}^t$ and first extracts essential features, such as unit coordinates, spatial positions, health and shields, skill availability, and alliance relations. It then passes $\mathbf{o}^t$ through a unified semantic text conversion process, which sequentially transcribes critical numerical attributes into natural language, integrates domain-specific battlefield knowledge and generates a structured tactical description X^t:

$$X^t = NL(\mathbf{o}^t). \tag{1}$$

where $NL(\cdot)$ denotes the semantic textualization function, which transforms environmental state information into representations that can be understood and reasoned about by the LLM.

Based on this semantic description X^t, the LLM simultaneously generates the raw tactical action and its corresponding strategic semantic text as follows:

$$(\mathcal{U}^t_{\text{Raw}},\ Z^t_{\text{Raw}}) = LLM(X^t), \tag{2}$$

where $\mathcal{U}^t_{\text{Raw}}$ and Z^t_{Raw} are both generated by $LLM(\cdot)$, with $\mathcal{U}^t_{\text{Raw}}$ denoting the raw tactical action and Z^t_{Raw} representing the corresponding raw strategic semantic text, summarizing key situational information and high-level tactical intent under the current environment.

Inspired by the collaborative criticism paradigm of Table-Critic [58], which shows that structured criticism and refinement can enhance LLM reasoning, we introduce a rigorous three-level tactical critic mechanism into the LLM-based tactical action generation process to ensure that the generated tactical actions are executable in the environment, strategically sound and aligned with high-level tactical semantics.This mechanism imposes multi-dimensional constraints and corrections on the LLM outputs, enabling efficient coordination and tactical cooperation among multiple agents. This mechanism evaluates the LLM-generated results along three dimensions:

- **Numerical Feasibility Critic** (C_{num}): Verifies adherence to low-level physical constraints such as energy consumption, skill cooldown and effective range to prevent the generation of infeasible actions (e.g., attempting to release a skill without sufficient energy).
- **Semantic Consistency Critic** (C_{cons}): Validates the consistency between the LLM's textual reasoning and the actual high-level environmental observations, ensuring that the tactical description aligns with the real-world situation. For example, when generating the command "Zealots move forward to absorb damage and block the enemy, while Stalkers coordinate to focus fire on enemy Stalkers in rotation", the system further verifies whether it is physically feasible for Zealots to advance and absorb damage and for Stalkers to execute coordinated focus fire and rotational attacks.
- **Tactical reasoning Critic** (C_{reason}): Assesses the spatiotemporal coordination and cooperative logic of the actions from a global tactical perspective, ensuring that multiple agents form effective collaboration at the tactical layer (e.g., front-line units drawing enemy fire while back-line units concentrate damage output).

All three indicators are binary decision variables with values in {0, 1}, where 1 indicates that the criterion is satisfied and 0 otherwise. The three levels are evaluated simultaneously and independently. Only when all three conditions are simultaneously satisfied ($C_{num} = 1 \wedge C_{cons} = 1 \wedge C_{reason} = 1$) are the raw outputs retained as valid. Otherwise, the entire output is discarded and the *CriticLLM* regenerates a corrected output until all three critic levels are satisfied. An example of this validation pipeline is illustrated in the left half of Fig. 2.

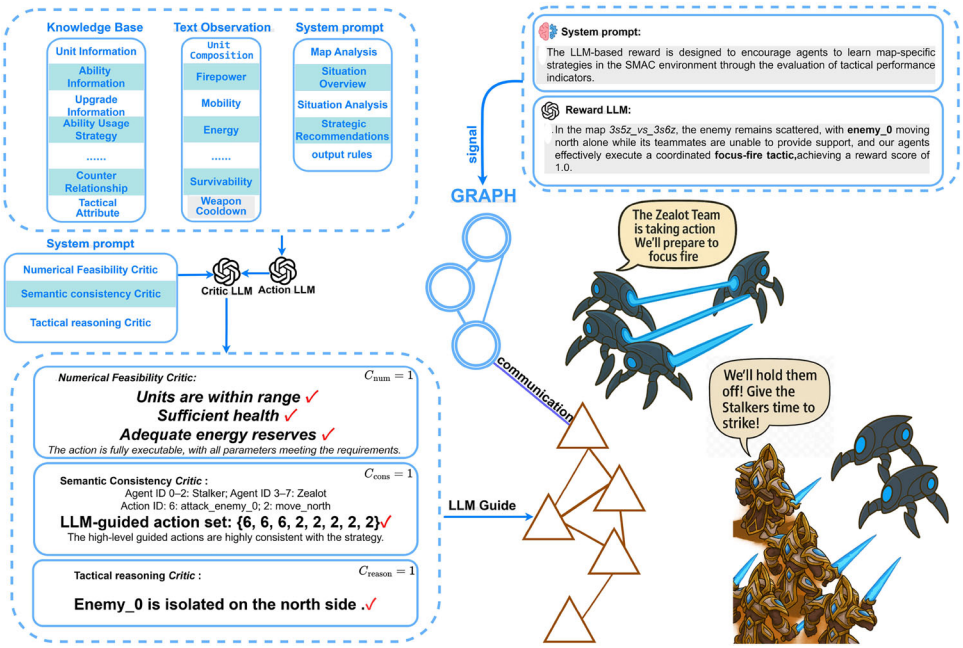

Fig. 2 The prompt is designed for obtaining valid LLM-guided actions and strategic semantics, along with examples of the three-level tactical critic and the LLM latent reward. Under LLM guidance and the LLM latent reward signal, the LLM-Guided Team-Adaptive Coordination Graph is constructed to enable efficient inter-agent communication and robust cooperative decision-making

Accordingly, the correction rule for a valid tactical command is formally defined as:

$$(\mathcal{U}^t_{\text{Valid}},\ Z^t_{\text{Valid}}) = \begin{cases} (\mathcal{U}^t_{\text{Raw}},\ Z^t_{\text{Raw}}), \text{ if } C_{num} = 1 \wedge C_{cons} = 1 \wedge C_{reason} = 1, \\ CriticLLM(\mathcal{U}^t_{\text{Raw}},\ Z^t_{\text{Raw}},\ X^t), \text{ otherwise.} \end{cases} \tag{3}$$

Both $\mathcal{U}^t_{\text{Valid}}$ and Z^t_{Valid} denote the LLM validated outputs that pass the three-level evaluation, ensuring that only executable tactical actions and consistent semantic information are retained. Here, $\mathcal{U}^t_{\text{Valid}}$ provides high-level action demonstrations tailored to different maps and battlefield situations. Acting as a teacher signal, these demonstrations guide the early-stage policy learning process and provide imitable tactical action priors for local agent policies during the LLM-guided pre-training phase. Z^t_{Valid} captures high-level semantic information of the battlefield, including agent roles, cooperative intentions and inter-agent relations. These high-level semantic cues are incorporated into the construction of the Team-Adaptive Coordination Graph, enabling it to capture group-level tactical dependencies that are often challenging to infer solely from raw observations.As the graph continually updates, the semantic guidance provides the time-evolving dependency structures.

To better capture the LLM's real-time high-level strategic intentions and transform them into tactical semantic representations that can be utilized by lower-level policy networks, we introduce a semantic encoding process that models the strategic intent and semantic dependencies embodied in the language model. The semantic information is modeled through a combination of self-attention mechanisms, which can be formally expressed as:

$$\hat{V}^t = \text{GRU}(\text{SelfAttn}(Z^{t-k:t}_{\text{Valid}})), \tag{4}$$

where SelfAttn$(\cdot)$ denotes the self-attention mechanism and GRU$(\cdot)$ represents the gated recurrent unit used to integrate temporal contextual information. The resulting latent semantic vector $\hat{V}^t$ represents the embedding feature of the current tactical description in the semantic space.

Team-adaptive coordination graph

The proposed Team-Adaptive Coordination Graph (TACG) dynamically partitions the agent team into several cooperative groups and optimizes the team-level coordination topology based on interactions among these groups. TACG integrates LLM-guided semantic tactical priors with the environment-driven feedback loop in MARL. While the LLM provides high-level strategic guidance, it inherently lacks the capacity for low-level execution and real-time adaptation necessary for complex environmental dynamics. Through a reinforcement learning-based feedback optimization process, TACG facilitates efficient learning of executable cooperative policies under resource constraints and accelerates policy convergence.

Unlike traditional approaches [3, 26] that rely solely on trajectory-level behavioral evidence or interaction patterns for group division in graph construction, TACG incorporates an LLM-Guided semantic context vector $\hat{V}^t$. This vector embeds high-level tactical information directly into the graph structure, enabling the model to jointly consider individual

observation features and the global strategic context during dependency construction. Consequently, the dynamic adjustment of edge weights and dependency structures contributes to an interpretable and semantically consistent coordination graph. To realize this, TACG employs an observation feature encoder and an agent-pair predictor to enrich feature representation and capture interaction intensity, formally defined as:

$$\hat{o}_i^t = f_{oe}(\mathbf{o}_i^t), \quad \mu_{ij}^t = f_{ap}(\hat{o}_i^t, \hat{o}_j^t, \hat{V}^t), \tag{5}$$

where $f_{oe}(\cdot)$ is implemented using a Multi-Layer Perceptron (MLP) to extract the observation feature representation of each agent, and $f_{ap}(\cdot)$ is an attention-based network used to estimate the interaction relationship between agent pairs. The matrix $\mu^t \in \mathbb{R}^{n^2 \times 1}$, which stacks all μ_{ij}^t in a fixed order, represents the edge weights between agent pairs, characterizing their interaction importance at time step t, thereby ensuring structural symmetry and context sensitivity in agent dependency modeling.

Due to partial observability in MARL, agents often exhibit analogous behavioral patterns when perceiving similar local observations over certain temporal windows. Leveraging this, we introduce a grouping model $f_t(\mathcal{A}) = \mathcal{G}$ designed to capture and exploit the inherent correlations and strategic convergence among agents throughout the temporal decision trajectory. To capture higher-level collaborative patterns effectively, a trajectory-based semantic grouping function is introduced:

$$\mathcal{G}^t = f_t(\mathbf{o}^{t-k:t}, \hat{V}^t), \tag{6}$$

where $\mathbf{o}^{t-k:t}$ denotes the joint observation sequence over the past k steps, and $\hat{V}^t$ is the shared semantic embedding provided by the LLM. The function $f_t(\cdot)$ partitions agents into semantically consistent groups based on trajectory-level behavioral and semantic features, thereby establishing structured cooperative relationships in the high-level semantic space. We define the agent–group matrix M^t to represent inter-agent grouping, where $M_{ij}^t = 1$ if agents a_i and a_j are in the same semantic group at time t:

$$M_{ij}^t = \begin{cases} 1, & \text{if agents } a_i \text{ and } a_j \text{ belong to the same} \\ & \quad \text{group and share semantic similarity,} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

This structure ensures intra-group consistency and inter-group separability, providing a crucial structural prior for subsequent coordination graph modeling (Table 1).

Building on this, we transform the agent–group matrix M^t into an edge-covariance structure, enabling direct incorporation of group information into the edge relation modeling. This transformation explicitly reflects the semantic coupling and interaction intensity across agent groups, providing a structural prior for subsequent distributed edge representation learning.This is described by the following operation:

$$\hat{M}^t = \mathrm{vec}(M^t)\mathrm{vec}(M^t)^\top, \tag{8}$$

where the covariance matrix $\hat{M}^t \in \mathbb{R}^{n^2 \times n^2}$ captures the semantic dependency and covariance structure among edges. Agents within the same semantic group are thus characterized by stronger covariance correlations, reflecting higher semantic consistency and cooperative alignment. Here, $\mathrm{vec}(\cdot)$ denotes the operation of flattening a matrix into a vector. Consequently, the dynamic coordination graph at time t is defined as:

$$\mathcal{C}^t = \{\mathcal{A}, \mathcal{E}^t\}, \quad \mathcal{E}^t \sim \mathcal{N}(\mu^t, \hat{M}^t), \tag{9}$$

Such LLM-guided group partitioning enables TACG to adaptively track the dynamic evolution of agent interaction patterns under semantic priors and imposes a hierarchical organization over the multi-agent system. This hierarchy allows the coordination mechanism to reason about both individual-level interactions and group-level cooperative dependencies. To operationalize this structural prior at the edge level, we follow GACG [26] and model pairwise relations with a Gaussian distribution, where the mean vector μ^t encodes the interaction importance of each edge and the covariance matrix $\hat{M}^t$ characterizes dependency relationships among different edges. In particular, when agents belong to the same semantic group, $\hat{M}^t$ emphasizes stronger intra-group correlations, providing a principled way to inject group-level semantic structure into the coordination graph.

Using the Gaussian distribution defined above, we sample the group-aware coordination graph structure $\mathcal{C}^t$ at each time step and reconstruct it into an adjacency matrix form for message passing. The sampled graph structure facilitates information exchange via a Graph Neural Network (GNN), enabling agents to efficiently share and integrate multi-source information for strategic adaptation in the dynamic MARL environment. The GNN is defined as follows:

$$H_l^t = \mathrm{ReLU}\big(\hat{\mathbf{C}}^t H_{l-1}^t W_{l-1}^t\big), \quad l = 1, \ldots, L. \tag{10}$$

where l is the index of GNN layers,$\hat{\mathbf{C}}^t = \widetilde{D}^{-1/2}\mathbf{C}^t\widetilde{D}^{-1/2}$ denotes the normalized adjacency matrix, and $\widetilde{D}_{ii} = \sum_j \mathbf{C}^t[i, j]$. The initial input of the GNN, H_0^t, is the set of observation feature vectors $\{\hat{o}_1^t, \ldots, \hat{o}_n^t\}$ extracted from Equation (5). After l layers of propagation, each agent i obtains a contextual embedding $m_i^t = H_l^t$, which successfully integrates local features, relational dependencies

Table 1 Comparison of experimental methods by graph type, group utilization and LLM-guided graph construction

Method	Graph type	Group coordination	LLM guidance
QMIX [13]	×	×	×
DICG [20]	Complete	×	×
CASEC [23]	Sparse	×	×
VAST [48]	×	✓	×
GACG [26]	Sparse	✓	×
LLM-GNCF	Sparse	✓	✓

and group-level tactical information.This contextual embedding is then utilized in the local action-value function $Q_i^t(\tau_i^t, u_i^t; m_i^t)$.

while the global action-value function is computed using a monotonic mixing network [13]:

$$Q_{\text{tot}}(\tau^t, \mathbf{u}^t; s^t) = f_{\text{mix}}(Q_1^t, \ldots, Q_n^t; s^t). \tag{11}$$

The joint action-value function is trained to satisfy the Bellman optimality equation:

$$Q_{\text{tot}}(s^t, \mathbf{u}^t) = \mathbb{E}\left[r^t + \gamma \max_{\mathbf{u}'} Q_{\text{tot}}(s^{t+1}, \mathbf{u}')\right]. \tag{12}$$

Moreover, the factorization is designed to satisfy the Individual–Global–Max (IGM) property:

$$\arg\max_{\mathbf{u}} Q_{\text{tot}}(\tau^t, \mathbf{u}; s^t) = \left[\arg\max_{u_i} Q_i(\tau_i^t, u_i; m_i^t)\right]_{i=1}^n. \tag{13}$$

Serving as the core bridge between high-level LLM-guided strategy and low-level MARL control, TACG leverages LLM-guided action supervision and semantic signals as teacher guidance to construct a dynamic coordination graph. Together with the LLM-empowered latent reward shaping module described in Sect. LLM-empowered latent reward shaping, these components jointly constitute *Stage I (LLM-guided pre-training)*.During this stage, the actions executed from the LLM-guided valid action set $\mathcal{U}_{\text{Valid}}^t$ generate trajectories that are used to pre-train both the TACG networks and the value decomposition.

In *Stage II (RL training)*, we no longer query the LLM. Instead, we reuse the LLM-guided TACG learned at the end of Stage I as a structural prior graph and continue to optimize only the policy and value networks through interaction with the environment. Concretely, the group assignments and agent interaction relations are computed without the semantic context vector, i.e.,

$$G^t = f_t\left(o_{t-k:t}\right), \qquad \mu_{ij}^t = f_{\text{ap}}\left(\hat{o}_i^t, \hat{o}_j^t\right), \tag{14}$$

The latent reward branch is disabled, so that the total reward reduces to the environment reward, i.e.,

$$R_{\text{total}}^t = R_{\text{env}}^t. \tag{15}$$

Although the semantic vector $\hat{V}^t$ is removed in Stage II, the TACG networks retain the parameters learned under semantic supervision in Stage I, which act as a structural prior for subsequent training. This initialization places the policy in a favorable region of the optimization landscape, enabling it to preserve high-level coordination patterns while being fine-tuned using only local observations. This two-stage design employs a relatively short LLM-guided pre-training phase, which both reduces the overall LLM resource cost and yields a strong graph-structural prior that accelerates the convergence of cooperative policies.

LLM-empowered latent reward shaping

In cooperative multi-agent reinforcement learning (MARL), RL agents operate at a significantly higher decision frequency than environmental feedback frequency, which makes accurate tactical quality assessment challenging based on instantaneous, single-frame information. Taking typical environments such as the *StarCraft Multi-Agent Challenge* as examples, the environment usually provides episodic win or loss signal. Such a reward structure, which relies on independent frames or single-step outcomes, fails to capture the strategic evolution of agents during continuous decision-making processes, thereby exacerbating the credit assignment problem.

To address this issue, we propose an *LLM-Empowered Latent Reward Shaping* method. Conceptually related to prior work such as [59, 60], it fundamentally differs in its temporal modeling mechanism and its ability to produce trajectory-level semantic rewards. Specifically, we employ a *Chain of Aggregation* (CoA) mechanism that temporally concatenates and semantically compresses multi-frame observations over a sliding window of length k, enabling the LLM to reason about fine-grained strategic evolution and cooperative dynamics over extended temporal segments.LLM inference is triggered only at key temporal checkpoints, which synchronizes the high-frequency decision rhythm of the game with the slower inference cycle of the LLM, reduces token

Algorithm 1 LLM-Guided Training Process in LLM-GNCF

Require: *StarCraft II* environment env
Ensure: Trained LLM-Guided TACG
Initialize batch size B, buffer $\mathcal{D} = \{\}$
Initialize sliding window k for CoA
for each timestep t **do**
  Obtain Raw LLM action $\mathcal{U}^t_{\text{Raw}}$ and Raw LLM semantics Z^t_{Raw} from Action LLM
  Obtain Valid LLM action $\mathcal{U}^t_{\text{Valid}}$ and Valid LLM semantics Z^t_{Valid} from Critic LLM ▷ Eq. (3)
  Obtain the LLM strategic semantic vector of the joint temporal segment $\hat{V}^t$
  LLM semantic-guided agent pairs: $f_{ap}(\hat{o}^t_i, \hat{o}^t_j, \hat{V}^t)$ ▷ Eq. (5)
  LLM semantic-guided agent grouping: $\mathcal{G}_t = f_t(\mathbf{o}^{t-k:t}, \hat{V}^t)$ ▷ Eq. (6)
  Construct TACG: $\mathcal{C}^t = \{\mathcal{A}, \mathcal{E}^t\}$
  Propagate messages via GNN
  for each agent i **do**
    Compute Q-values for all actions
    $\mathcal{U}^t_i = \mathcal{U}^t_{\text{Valid}}[i]$
  end for
  Execute $\{\mathcal{U}^t_i\}$ and observe next state and reward R^t_{env}
  Aggregate k-frame history S^t_{history} via CoA ▷ Eq. (17)
  Get r^t_{LLM} from LLM based on S^t_{history}
  Decode latent reward: $R^t_{\text{LLM}} = f_\psi(r^t_{\text{LLM}})$
  $R^t_{\text{total}} = R^t_{\text{env}} + R^t_{\text{LLM}}$
  Store transition into buffer $\mathcal{D}$
  if size of $\mathcal{D} \geq B$ **then**
    Sample batch from $\mathcal{D}$
    Compute Q_{tot} from QMIX mixing network
    Update parameters under TD loss
  end if
end for

consumption, and maintains semantic evaluation precision. In this way, the model yields strategically grounded latent reward signals that provide a high-level shaping signal over continuous decision sequences and compensate for the limitations of traditional frame-level reward functions. The core computational process of CoA can be formally expressed as follows. To capture temporal dependencies in tactical behaviors, the system maintains a sliding window of length k:

$$\mathbf{W}^t = \{X^{t-k+1}, \ldots, X^t\}, \tag{16}$$

It further aggregates the past k frames of structured tactical description text into a temporal summary $\mathbf{S}^t_{\text{history}}$, which integrates tactical transitions and semantic contextual information across consecutive frames, thus achieving semantic compression and high-level abstraction:

$$\mathbf{S}^t_{\text{history}} = f_{\text{agg}}(\mathbf{W}^t), \tag{17}$$

Here, $f_{\text{agg}}(\cdot)$ denotes the aggregation function that integrates multi-frame textual features within the semantic space to capture the key temporal segments of tactical semantics, generating semantically coherent and contextually complete temporal summary representations.

To enable task-specific strategic policy learning across diverse environments, we introduce a set of tactical strategy metrics that are semantically evaluated by the LLM. These metrics provide high-level tactical guidance and latent reward signals during training. Accordingly, we define the set of LLM-based reward evaluation metrics, where each metric are as follows:

- **Kiting effectiveness (KE):** Quantifies the proficiency of micro-control maneuvers, such as "hit-and-run" behaviors.
- **Focus-fire coordination (FFC):** Measures the synchronization and consistency of multi-agent target focusing.An example of this can be observed in the top-right panel of Fig. 2, where the LLM Reward signal enables agents to learn focus-fire tactical coordination more effectively.
- **Formation maintenance (FM):** Assesses the formation stability and spatial mutual support between agents.
- **Skill synergy timing (SST):** Evaluates the temporal accuracy and cooperative efficiency of combined skill usage.

Based on the temporal semantic summary $\mathbf{S}^t_{\text{history}}$, the LLM computes a multidimensional semantic metric vector:

$$r^t_{\text{LLM}} = [r^t_{\text{KE}}, r^t_{\text{FFC}}, r^t_{\text{FM}}, r^t_{\text{SST}}], \tag{18}$$

where each dimension quantifies the performance of the current tactical strategy in terms of temporal-semantic consistency. The semantic metric values satisfy:

$$r^t_{\text{KE}},\ r^t_{\text{FFC}},\ r^t_{\text{FM}},\ r^t_{\text{SST}} \in [-1, 1], \tag{19}$$

Here, positive values indicate alignment with the desired semantic intent, whereas negative values reflect deviation from the expected cooperative behavior. Subsequently, a neural aggregation function is applied to combine these tactical indicators, yielding the latent LLM-based reward:

$$R^t_{\text{LLM}} = f_\psi(\mathbf{r}^t_{\text{LLM}}), \tag{20}$$

where f_ψ denotes an MLP that captures semantic interactions and importance weighting across different dimensions of the tactical metrics.

Since the LLM output is non-differentiable, $\mathbf{r}^t_{\text{LLM}}$ is treated as a fixed numeric feature vector parsed from the LLM response, rather than as a differentiable latent representation. Such a design is consistent with prior LLM-based reward learning frameworks, where LLM feedback is used as external annotation and distilled into a learned intrinsic reward model, rather than treating the LLM itself as an end-to-end differentiable component, such as ONI [61]. Accordingly, the

LLM is excluded from gradient-based optimization, receives no gradient updates, and remains fixed throughout the entire training process.

Inspired by the decoder-based return-level supervision paradigm used in LaRe [33], we further introduce an explicit regression objective to train f_ψ separately from the value network. However, different from LaRe, which uses the decoded reward for return decomposition in episodic RL, our framework retains the original environmental reward during MARL optimization. To avoid double counting the environmental reward, the decoder is trained against a complementary residual target instead of the raw environmental return itself.

To train the reward decoder f_ψ separately from the value network, a k-step discounted environmental return is first computed as:

$$J_k^t = \sum_{j=0}^{k-1} \gamma^j R_{\text{env}}^{t+j}, \tag{21}$$

where k is set to be consistent with the CoA window length, γ is the discount factor, and R_{env}^{t+j} denotes the original environmental reward at step $t+j$. The residual target is then defined as:

$$\Delta J_k^t = J_k^t - R_{\text{env}}^t, \tag{22}$$

which represents the complementary return information beyond the immediate environmental reward at the current step.

The parameters ψ of the MLP are optimized over samples drawn from the replay distribution $\mathcal{D}$ using the following regression loss:

$$\mathcal{L}_{\text{MLP}}(\psi) = \mathbb{E}_{t\sim\mathcal{D}}\left[\left(f_\psi(\mathbf{r}_{\text{LLM}}^t) - \Delta J_k^t\right)^2\right], \tag{23}$$

This residual objective aligns the supervision target with the multi-frame temporal scope of CoA while preventing f_ψ from simply reproducing the immediate environmental reward, thereby encouraging it to capture complementary long-horizon semantic information.

To incorporate the LLM-empowered latent reward shaping signal into the learning process, we define the shaped overall reward as:

$$R_{\text{total}}^t = R_{\text{env}}^t + R_{\text{LLM}}^t, \tag{24}$$

where R_{env}^t denotes the original environmental reward, and R_{LLM}^t denotes the latent semantic reward derived from the LLM semantic metrics. During Stage I, the MLP f_ψ is trained with $\mathcal{L}_{\text{MLP}}$ using a separate optimizer, while the MARL control networks, including the TACG backbone and the QMIX value-decomposition network, are updated under the TD objective induced by R_{total}^t. The scalar reward R_{LLM}^t is treated as a detached shaping signal during TD optimization. Therefore, the reward decoder is not jointly optimized with the Q-network via back-propagation from the TD objective. Instead, R_{LLM}^t serves as an additional semantic shaping signal for MARL training. In Stage II, the latent reward branch is disabled, and only the environmental reward is used.

Experiment

In this section, we design experiments to answer the following research questions:

- Which large language model provides the most effective guidance during training?
- How well does LLM-GNCF perform on complex cooperative multi-agent tasks compared with other representative CG-based methods?
- Do the LLM-guided components, including action supervision, tactical reward shaping, and strategic synergy group division, improve the performance of LLM-GNCF?
- What is the influence of the number of groups on the performance of LLM-GNCF?
- How does the selected trajectory length affect the final results?
- Can the proposed CoA method effectively reduce LLM inference costs while maintaining robust performance?

All experiments are conducted on the *StarCraft II* micromanagement benchmark [8], which provides a diverse set of cooperative multi-agent tasks with varying map configurations and agent numbers. The benchmark encompasses scenarios involving at least eight agents and covers both homogeneous and heterogeneous unit compositions. To ensure consistent evaluation conditions, all environments are set to a difficulty level of 7 (Very Hard). Each experiment is repeated with five random seeds to guarantee statistical robustness of the reported results.

The experiments are divided into two stages. Following the curriculum-learning perspective [62] of starting from an easier, highly guided regime and then relaxing external supervision, we allocate the initial 0.25 million environment steps on the LLM-Guided Pre-training stage, during which the agents execute actions inferred by the LLM, perform strategic semantic group division, and incorporate context-aware tactical reward signals generated by the LLM. The second stage, Reinforcement Learning Training, proceeds without any LLM involvement and continues with standard MARL optimization.

Table 2 Performance comparison of different LLM models on various SMAC maps. Values represent win rates (WR) with standard deviations in parentheses

Map name	GPT-4o	DeepSeek-R1	Qwen−2.5
3s5z	0.96 (0.01)	0.94 (0.02)	0.94 (0.04)
3 s_vs_5z	0.95 (0.02)	0.93 (0.03)	0.90 (0.04)
1c3s5z	0.93 (0.02)	0.90 (0.03)	0.88 (0.04)
8 m_vs_9m	0.94 (0.03)	0.91 (0.04)	0.89 (0.04)
10 m_vs_11m	0.91 (0.03)	0.90 (0.05)	0.87 (0.01)
bane_vs_bane	0.98 (0.01)	0.97 (0.02)	0.96 (0.03)
MMM2	0.58 (0.03)	0.53 (0.05)	0.46 (0.02)
2c_vs_64zg	0.86 (0.03)	0.79 (0.04)	0.75 (0.05)
3s5z_vs_3s6z	0.09 (0.02)	0.06 (0.03)	0.04 (0.03)

Comparative experiment

We select nine representative maps from the SMAC benchmark, covering four key dimensions: unit composition, battle symmetry and force balance. The selection includes four homogeneous and five heterogeneous maps, designed to evaluate how LLM guidance enhances role coordination among agents. With respect to battle symmetry, two maps are symmetric while seven are asymmetric, the latter requiring more advanced tactical reasoning and coordination. Regarding force balance, the chosen maps span from completely balanced encounters to highly disadvantaged scenarios, capturing diverse combat dynamics and testing the adaptability of the proposed method under different strategic conditions. Based on this setup, we further compare the performance of different LLMs to identify the most effective one for SMAC. The best-performing model is then adopted for subsequent experiments, where we compare our proposed method with several representative baseline algorithms.

LLM model comparison

To further investigate the influence of different large language models on training effectiveness, we evaluate GPT-4o, DeepSeek-R1, and Qwen−2.5 under our proposed LLM-GNCF method on nine representative SMAC maps that cover a wide range of task difficulties and coordination complexities.

As shown in Table 2, on nine SMAC maps all configured with the Very Hard difficulty level, the three LLMs under the LLM-GNCF framework exhibit a stable performance ranking. GPT-4o (0.80) achieves the highest average win rate across all tasks, followed by DeepSeek-R1 (0.77) and Qwen−2.5 (0.74). Moreover, it attains the lowest average standard deviation, indicating that under identical training settings, GPT-4o provides not only better overall performance but also a more stable and reproducible learning process. A grouped analysis based on task structural complexity reveals a consistent pattern. On relatively simple maps such as $3s5z$ and $3s_vs_5z$, all three models achieve near-saturated win rates. In contrast, on *MMM2*, $2c_vs_64zg$ and $3s5z_vs_3s6z$, which exhibit extreme force imbalance or require fine-grained kiting and harassment tactics, the performance gap becomes much more pronounced. For example, *MMM2* is widely regarded as one of the most demanding SMAC scenarios in terms of multi-stage tactical coordination and long-horizon planning. On this map, GPT-4o (0.58) clearly outperforms DeepSeek-R1 (0.53) and Qwen−2.5 (0.46). In the highly unfavorable scenario $3s5z_vs_3s6z$, all three models obtain low absolute win rates, yet GPT-4o (0.09) still surpasses DeepSeek-R1 (0.06) and Qwen−2.5 (0.04).

These results suggest that, in complex cooperative scenarios that require timing decisions across multiple temporal scales, unit-position management and high-risk sacrifice strategies, the semantic descriptions and evaluation signals produced by GPT-4o provide more consistent semantic guidance in our experiments. Consequently, LLM-GNCF can acquire better structural priors and reward shaping already in Stage I pre-training, which is associated with stronger final performance during Stage II reinforcement-learning fine-tuning.

CG-based methods comparison

To the best of our knowledge, there is currently no widely adopted benchmark for LLM-based MARL on SMAC. The exclusion of other LLM-integrated MARL methods from direct comparison is justified on three grounds.First, existing LLM-MARL approaches such as YOLO-MARL [32] and LAMARL [34] are evaluated on cooperative navigation or assembly tasks within environments and have not been validated on SMAC under the Very Hard difficulty setting, making fair numerical comparison methodologically unsound. TextStarCraft II [39] similarly evaluates LLMs in single-agent macro-strategic decision-making within StarCraft II, and does not address multi-agent coordination under partial observability. Second, YOLO-MARL and LAMARL interact with the LLM only once before training as an offline planner, while TextStarCraft II focuses on sequential single-agent decision-making. None of these frameworks incorporate dynamic graph-based coordination structures, which is a central design focus of LLM-GNCF. Third, these methods operate under fundamentally different training paradigms and information access assumptions, making win-rate comparison across frameworks difficult to interpret without confounding factors.

To provide a broader performance reference, we additionally include MAPPO [63] as a strong non-graph CTDE baseline, and compare LLM-GNCF against the following CG-based baselines:

- **QMIX** [13]: Decomposes the joint action-value function into a monotonic mixing of individual value functions, ensuring Individual-Global-Maximum consistency.
- **DICG** [20]: Uses attention mechanisms to compute a weighted fully connected graph for inter-agent information passing.
- **CASEC** [23]: Drops edges on the weighted fully connected graph using a variance payoff function to reduce redundant communication.
- **VAST** [48]: Performs value factorization over predefined sub-teams; sub-team values are linearly decomposed for all members.
- **GACG** [26]: Achieves collaboration via group division, but lacks LLM-based action guidance, strategic semantics and tactical rewards.
- **MAPPO** [63]: A strong on-policy CTDE baseline without graph-based inter-agent communication structures.

In Fig. 3, we present the comprehensive results of our experiments conducted across nine diverse maps, highlighting the competitive performance of our LLM-Guided Graph Neural Coordination Framework for MARL method. LLM-GNCF achieves competitive win rates across the evaluated maps and often shows relatively rapid convergence, showing competitive performance compared to representative baselines as well as the strong non-graph-based baseline MAPPO in complex multi-agent scenarios. On the asymmetric confrontation maps $10m_vs_11m$ and $1c3s5z$, all CG-based methods display comparable convergence trajectories, reaching their maximum test win rates at nearly the same training stage at the end of 1.75 million training steps. Furthermore, on the hard maps $2c_vs_64zg$ and $MMM2$, LLM-GNCF efficiently learns effective combat strategies under LLM-guided training, achieving among the strongest final win rates on these difficult maps, with sustained improvements during training. Additionally, on the very hard map $3s5z_vs_3s6z$, LLM-GNCF shows a sustained advantage on this evaluated setting.

However, disparities become more evident on other maps. For instance, QMIX exhibits the weakest performance on the *bane_vs_bane* map, only starting to converge around 1.75 million steps to a peak win rate of 73%, struggling to match the effectiveness of other methods. Similarly, DICG falls behind on the $8m_vs_9m$ map, achieving a win rate approximately 0.2 lower than the next best-performing method. On the challenging $3s5z$ and $3s_vs_5z$ maps, CASEC shows a lower win rate and does not exhibit clear convergence within the tested training budget, suggesting that its strategy is less suited to the intricacies of this particular scenario.By contrast, MAPPO as a non-graph-based baseline exhibits distinct performance characteristics. On moderate-difficulty maps such as $8m_vs_9m$ (92.3%) and *bane_vs_bane* (93.1%), MAPPO converges rapidly and surpasses several graph-based baselines. On $1c3s5z$, MAPPO's win rate rises faster than LLM-GNCF in the early phase, yet its final win rate remains marginally lower, suggesting that graph-based communication may provide additional benefits during policy refinement in heterogeneous settings. On harder maps $3s_vs_5z$ and $2c_vs_64zg$, MAPPO's final win rate falls noticeably below LLM-GNCF, indicating weaker adaptability in sustained coordination-intensive scenarios. On $3s5z_vs_3s6z$, MAPPO steadily approaches LLM-GNCF in later training but never surpasses it, suggesting that LLM-provided semantic priors may provide a sustained advantage in this setting under extreme force imbalance.

The weaker performance of these methods may be related to slower correction when suboptimal strategies are learned, requiring extensive trial-and-error before gradually correcting their behavior. In contrast, our method, LLM-GNCF, initially exhibits slow strategy learning and a slight decline between 1 and 1.3 million training steps on the $MMM2$ map. However, it quickly recovers, resulting in a rapid increase in win rate of approximately 0.2 at 1.4 million steps. This pattern is consistent with the possibility that LLM guidance helps the model recover more quickly from suboptimal early-stage behaviors.

Through comparative experiments, the limitations of existing approaches become apparent. QMIX does not explicitly incorporate graph structures. DICG introduces an attention mechanism to capture the strength of graph dependencies, but does not directly capture group-level dynamics. CASEC reduces redundant communication, but does not explicitly model collective behavioral patterns. VAST investigates sub-team interactions but does not effectively exploit dynamic graph structures. GACG facilitates agent collaboration by group division, but it lacks the ability to provide fine-grained tactical semantic guidance for dynamically constructing graph relationships and conducting adaptive group division. MAPPO achieves strong performance via centralized value estimation, but does not explicitly model dynamic inter-agent coordination.

LLM-GNCF integrates large language model–driven tactical reasoning with graph-based cooperative learning. In the first-stage LLM-guided pre-training, the LLM injects tactical semantics and action guidance into the coordination process, dynamically inferring inter-agent dependencies and, together with the LLM-based reward signal, promoting adaptive group formation and semantically consistent coordination graphs. Across nine *StarCraft II* micromanagement scenarios, the win rate continues to increase during the RL fine-tuning phase after a pre-training stage of 0.25 million environment steps. In *Stage II*, the policies often converge faster and reach stronger final performance than several graph-based baselines and MAPPO on the evaluated tasks,

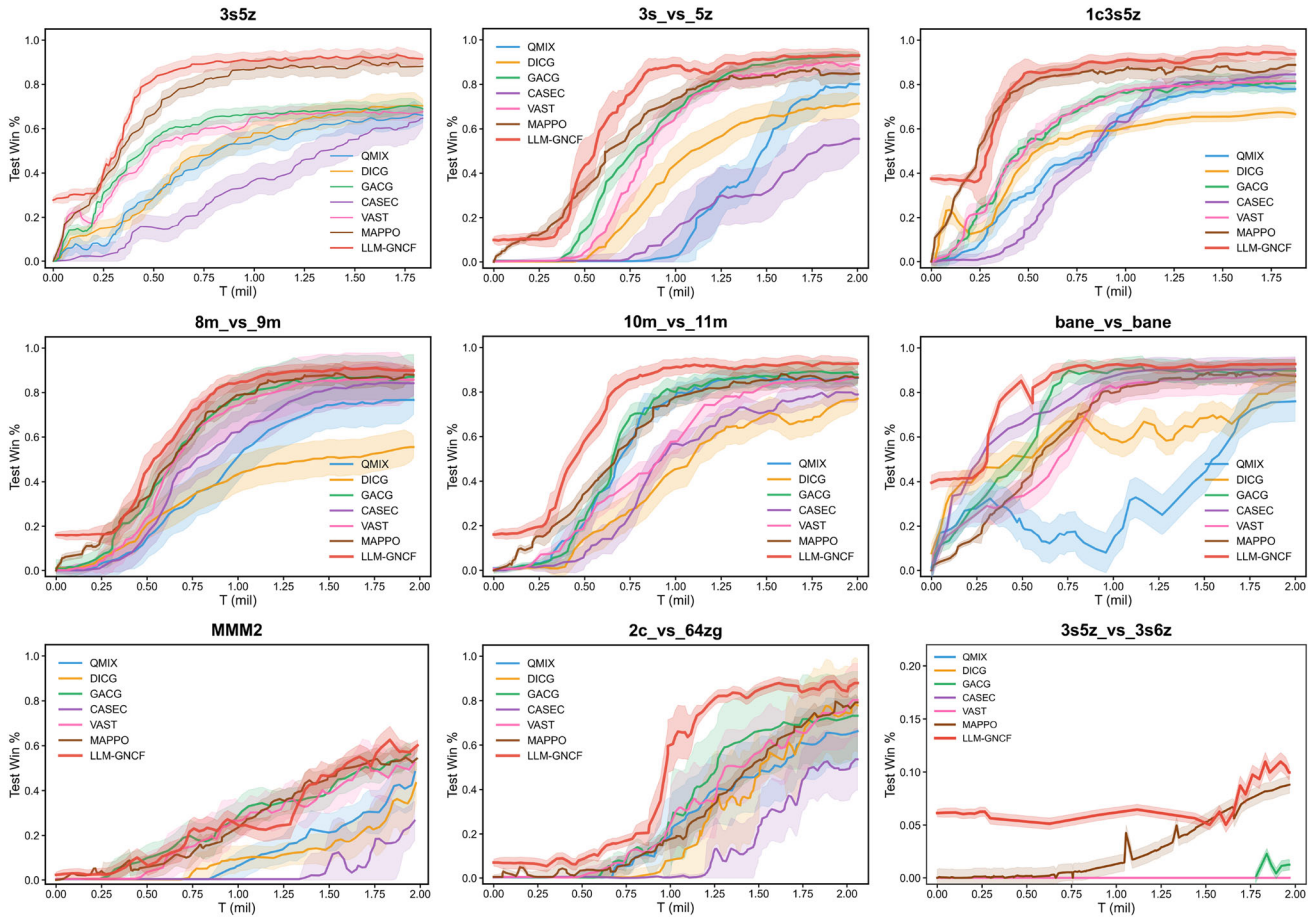

Fig. 3 Performance of LLM-GNCF and baselines on nine maps of the SMAC. The x-axis represents the time steps (in millions), while the y-axis quantifies the test win rate in the games

suggesting that *Stage II* can retain useful coordination patterns from *Stage I* while further refining the policy, without clear evidence of severe drift in our experiments.

Ablation study

Different LLM Guidance Modules

In this section, we experiment on two maps, aiming to provide insights into the unique aspects of the LLM's guiding role in LLM-GNCF and its effectiveness in capturing group-level tactical semantic information and exploring context-aware LLM rewards for cooperative multi-agent tasks. We conduct ablation experiments to isolate the contribution of each LLM functional module in LLM-GNCF. The detailed settings are shown below:

- **(w/o LLM-Reward):** In this setting, the LLM-based latent reward branch is removed: the LLM still provides tactical semantics and teacher actions for TACG construction, but its scalar reward R^t_{LLM} is no longer added to the environment reward. This setting is used to isolate the contribution of LLM-based reward shaping and to assess whether agents can learn effective coordination purely from LLM-guided structure and demonstrations.
- **(w/o LLM-Action):** Here the LLM-Guided action module is disabled, so no expert action demonstrations are supplied. The framework still employs semantic guidance for TACG and the latent reward shaping module. This ablation focuses on the role of LLM-based action guidance and quantifies how much expert demonstrations accelerate and stabilize policy learning when semantic and reward-level LLM signals remain available.
- **(w/o LLM-Semantics):** In this configuration, the semantic guidance to the grouping network is removed: the semantic latent variable generated by the LLM is not injected into the TACG divider and group division is driven only by trajectory-based features. LLM-based teacher actions and latent reward shaping are still retained. This variant is designed to examine how much the semantic prior in TACG contributes to group-

structure modeling and cooperative policy learning beyond action- and reward-level LLM supervision.

- **TACG (w/o LLM):** All LLM-related components are removed and the framework reduces to a purely trajectory-based adaptive coordination graph trained with standard environment rewards. This non-LLM TACG baseline serves as a reference to quantify the overall performance gains brought by LLM guidance across action supervision, reward shaping and semantic grouping.

The results are shown in Fig. 4. Across various maps, each ablation approach exhibits similar patterns and the full LLM-GNCF achieves the strongest or near-strongest win rates in these ablations, supporting the effectiveness of the proposed framework. In the setting that retains the LLM-Action module, the first 0.25 million steps correspond to the LLM-guided pre-training phase, during which agents execute actions generated by the LLM and the win rate remains nearly constant. After switching to the reinforcement-learning–only phase, the learning curve rises rapidly and the convergence speed becomes almost identical to that of the full model. Furthermore, the LLM-Reward and LLM-Semantics ablations show distinct effects. Removing the LLM-Reward leads to a slight degradation in cooperative learning performance because the lack of temporally coherent, context-aware shaping signals prevents agents from effectively refining tactical behaviours, resulting in a marginally lower win rate compared with the full model. When LLM-Semantics is removed, the performance is close to that of TACG without any LLM guidance, indicating that LLM-guided high-level strategic semantics are important for group division and structuring cooperative relationships among agents.

Number of groups

We further investigate the impact of different group division numbers under maps with different unit compositions. Specifically, we select $3s5z$ and $8m_vs_9m$ as test scenarios, each containing eight agents. The former map involves two types of units, while the latter consists of a single unit type. The number of group divisions, denoted as m, is set to 0, 2, 4 and 8 in the experiments.

When $m = 0$, it implies the absence of group divisions between agents, indicating that the training process does not incorporate any LLM-guided strategic semantic priors introduced through the group division process. In contrast, when $m = 8$, each agent is regarded as an independent group. Although agents no longer share group communication, each one can still receive LLM-guided strategic semantics that are conditioned on its local state and tactical context. This may support more context-aware decisions, resulting in flexible yet coordinated behaviors at the global level.

Table 3 Average token consumption comparison between CoA and frame-by-frame methods under different window lengths

Method / Window Length (k)	1	5	10	20
CoA	1685	3057	4730	7314
Frame-by-frame	1265	6325	12,650	25,281

Under these two extreme group division settings, even in the absence of explicit group structures, the introduction of LLM-based action guidance during the early pre-training phase provides agents with strong high-level tactical decision guidance. By executing actions inferred by the LLM, agents are exposed to advanced tactical behaviors and cooperative patterns. This guidance facilitates the emergence of preliminary coordination capabilities, allowing the model to maintain stable learning dynamics and coherent cooperative behaviors even without structured group collaboration.

The experimental results are presented in Fig. 5. Optimal performance is observed when $m = 2$, which suggests that moderately LLM-guided semantic group divisions play an important role in enhancing cooperative efficiency. In the comparison across the two maps, although the win rates remain stable and nearly identical during the LLM-action execution stage, the training efficiency differs substantially due to the varying group division settings. The configurations with group divisions ($m = 2$ and $m = 4$) outperform those with no group divisions or a high number of them ($m = 0$ and $m = 8$). This result suggests that, under the guidance of LLM strategic semantics, an appropriate degree of group division can effectively coordinate individual behaviors and collective cooperation, enabling agents to maintain flexibility while developing a higher level of collaborative capability.

Notably, in the 3s5z map, which contains two distinct agent types, the convergence speed under $m = 8$ is significantly faster than that under $m = 0$. This acceleration effect may be attributed to the importance of policy diversity in multi-agent settings.

Length of trajectory for group division

This experiment aims to investigate the impact of different observation trajectory lengths on the effectiveness of group division, particularly evaluating the optimal trajectory length at which our method best captures strategic semantics. The observation trajectory uses a window length k, which specifies the temporal span of each agent's observation, with $k \in \{1, 5, 10, 20\}$.

The setting $k = 1$ corresponds to single-step observations, whereas $k = 5$, $k = 10$ and $k = 20$ allow the evaluation of semantic and relational dependencies over longer temporal horizons. By varying k, we can examine the model's perception range of strategic semantics under different trajectory

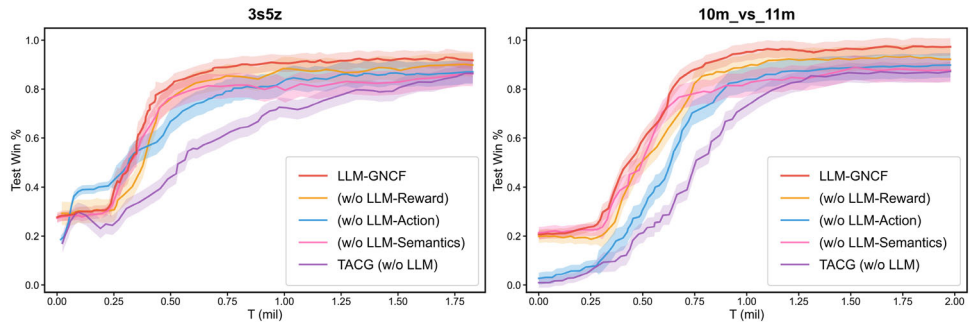

Fig. 4 Ablation study on different LLM-guided models

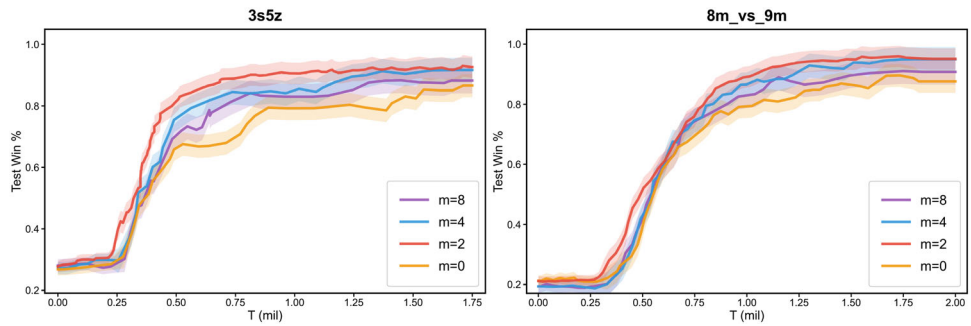

Fig. 5 Experiment of dividing agents into m groups in *3s5z* and *8m_vs_9m*. The former involves two agent types, whereas the latter contains only one

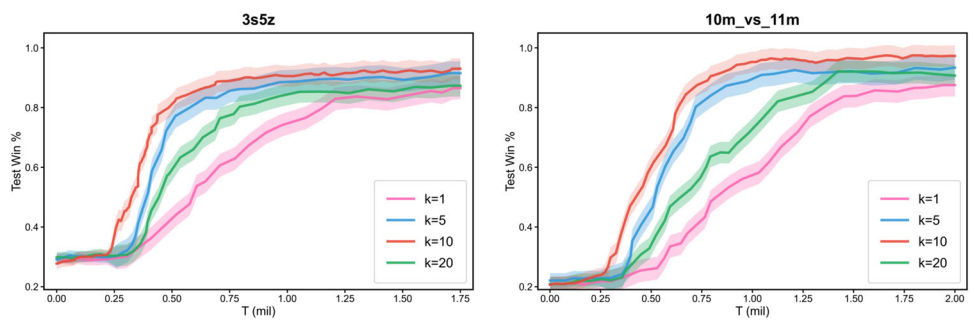

Fig. 6 Experiment of varying observation window lengths (k) on the group divider model f_t

lengths and assess how this variation influences the quality of group division.

The experimental results are illustrated in Fig. 6, showing that a moderate trajectory length of $k = 10$ yields the strongest performance among the tested settings across both maps. This setting provides agents with sufficient historical information to collaborate effectively with the LLM-based reward during training, thereby enabling the model to better capture strategic semantic information for group division. In contrast, excessively short or long trajectories limit the balance between semantic precision and temporal generalization, highlighting the importance of appropriately capturing dynamic contextual dependencies in multi-agent coordination.

More specifically, when $k = 1$, the lack of adequate historical context prevents the LLM semantic signals from being properly grounded in the agents' behavioral histories, resulting in poor group division and decision-making performance. As the trajectory length increases from $k = 1$ to $k = 5$, the amount of available contextual information gradually increases, improving semantic embedding and group division capability. However, it does not surpass the performance achieved with $k = 10$.

Efficiency analysis of the CoA method

This section analyzes the token efficiency of the Chain of Aggregation (CoA) method and examines how its per-

Table 4 Average reward comparison between CoA and frame-by-frame methods under different observation window lengths

Method / Window Length (k)	1	5	10	20
CoA	0.10	0.34	0.42	0.39
Frame-by-frame	−0.02	−0.01	−0.01	−0.01

Table 5 Computational cost comparison between the LLM-guided pre-training stage and standard MARL baseline. Wall-clock times are reported as average values over multiple runs, while token counts are obtained from API usage statistics

Metric	Stage I (0.25M)	Stage II (1.75M)	TACG (2 M)
Wall-clock time (hours)	5.9±0.21	13.77±0.32	15.74±0.48
GPU	RTX 5090 ×1	RTX 5090 ×1	RTX 5090 ×1
Action LLM tokens/step	2,300	0	0
Critic LLM tokens/step	1,950	0	0
Reward LLM tokens/step	473 (4,730 per 10 steps)	0	0
Total LLM tokens/step	4,723	0	0

formance varies with the observation window length k, consistent with the setting in Section (Length of trajectory for group division), where $k \in \{1, 5, 10, 20\}$. In the CoA method, the LLM evaluates the current frame by incorporating the semantic historical information from the preceding k frames. This design allows the model to integrate temporal semantic context and perform more context-aware evaluations across time. In contrast, the frame-by-frame approach computes LLM-based indicator scores solely from the current frame's observation, without leveraging any historical information for contextual evaluation or decision-making, which limits its temporal reasoning capability and overall evaluation efficiency. Experiments are conducted on the same nine maps as in Sect. (Comparative experiment) and the average token consumption is used to assess the effectiveness of the CoA method.

Table 3 reports the token consumption of the CoA method and the frame-by-frame baseline under different observation window lengths. As the window size increases from 1 to 20, the token usage of the frame-by-frame method grows rapidly from 1265 to 25,281, whereas CoA shows a much slower increase from 1685 to 7314. This contrast suggests that CoA provides a more favorable trade-off between temporal context and token cost through semantic aggregation and sparsely queries the LLM only at critical time steps, achieving a more favorable trade-off between semantic reasoning and computational cost, especially under larger observation windows.

Although the above analysis demonstrates that the CoA method effectively reduces LLM token consumption compared to the frame-by-frame approach, we further evaluate its impact on task performance by computing the **average reward** to verify whether the method not only conserves computational resources but also provides meaningful strategic guidance. Specifically, experiments are conducted on the same nine maps under different observation window lengths to identify the most effective window size that balances computational efficiency and decision-making performance.
The results are shown in Table 4. It can be observed that the CoA method consistently outperforms the frame-by-frame approach across all window sizes, while the latter produces reward values that remain almost constant and very close to zero (slightly negative) in all settings. This suggests that introducing temporal semantic context is crucial for effective LLM-based reward estimation. As the window size increases from 1 to 10, the average latent reward rises markedly from 0.10 to 0.42, indicating that aggregating a longer trajectory enables the LLM to better capture team-level tactical quality. When the window is further expanded to 20, the average reward no longer improves, revealing a point of diminishing returns where additional distant history brings limited extra semantic benefit. Therefore, selecting an observation window of $k = 10$ provides a good trade-off, allowing the LLM to integrate sufficient temporal information for effective yet efficient evaluation.

To assess the practical feasibility of LLM-GNCF, we provide a detailed breakdown of computational costs across training stages, as summarized in Table 5. Stage I (0.25M steps) involves three LLM modules operating at different query frequencies. The Action LLM and Critic LLM are queried at every step, consuming approximately 2,300 and 1,950 tokens per step, respectively. The Reward LLM, operating under the CoA mechanism with a window size of $k = 10$, is queried once every 10 steps, yielding an average of 473 tokens per step. The combined LLM consumption across all three modules is therefore approximately 4,723 tokens per step. Under this setting, Stage I completes in 5.9±0.21 hours on a single RTX 5090 GPU.

Stage II (1.75M steps) involves no LLM queries and proceeds as standard MARL optimization, with a wall-clock time of 13.77 ± 0.32 hours. The total training time of LLM-GNCF is approximately 19.67 h, compared with 15.74±0.48 hours for the pure RL baseline TACG over 2 M steps, corresponding to an overhead of roughly 25%. Once Stage I is completed, the LLM is fully decoupled from the system, and Stage II proceeds with comparable per-step computational cost to the pure RL baseline, since it no longer requires LLM queries.

Table 6 reports the overall computation time of different methods on map $10m_vs_11m$. LLM-GNCF requires 35.41 ± 0.69 seconds per 1k steps and 9.84 ± 0.38 hours per 1 M steps, making it more computationally expensive than most pure RL baselines. However, its overall training cost remains within the same practical order of magnitude

Table 6 Computation time on map $10m_vs_11m$

Method	1k steps time (s)	1 m steps time (h)
QMIX	18.12 ± 3.23	6.11 ± 0.33
DICG	30.21 ± 4.19	10.47 ± 0.58
CASEC	27.45 ± 1.83	9.11 ± 0.46
VAST	19.15 ± 3.23	5.59 ± 0.50
GACG	20.10 ± 3.80	7.06 ± 0.44
MAPPO	21.87 ± 3.51	7.38 ± 0.45
LLM-GNCF	35.41 ± 0.69	9.84 ± 0.38

as existing graph-based methods and is still lower than the 1 M-step runtime of DICG. This suggests that, although LLM-GNCF introduces additional overhead, its overall training cost on this map remains within the same practical order of magnitude as existing graph-based methods.

We acknowledge that the frequent LLM queries in Stage I introduce a non-negligible computational overhead. As an early exploration of LLM–MARL integration, this overhead mainly reflects the current implementation cost of incorporating semantic guidance, rather than a weakness in the framework's effectiveness. An important direction for future work is to use LLMs more efficiently during pre-training while preserving the benefits of semantic guidance.

Empirical validation of key mechanisms

Prompt sensitivity and the LLM-critic mechanism

Since LLM-GNCF relies on LLM-guided strategic semantics, an important question is whether the framework is overly sensitive to prompt design. A more fundamental issue is whether different semantic prompts induce genuinely different coordination graphs, rather than merely causing superficial performance fluctuations. To investigate this, we construct three prompt variants with different levels of strategic specificity and compare both their learning curves and the graph structures induced during training.In this subsection, we focus on prompt-level sensitivity while holding the observation-to-text conversion pipeline fixed, so that the effect of semantic prompting on graph induction and policy learning can be isolated.

To examine how prompt design affects the semantic prior provided to the framework, three prompt variants with different levels of semantic specificity are considered, as illustrated in Fig. 7. The Vague Prompt removes most domain knowledge and strategic reasoning constraints from the Standard Strategic Prompt. The Standard Strategic Prompt, used as the default in all main experiments, provides the complete set of domain knowledge, tactical principles, and unit-specific guidance. The Explicit Grouping Prompt extends the standard version with mandatory functional group assignments and inter-group coordination dependencies. The full template of the Standard Strategic Prompt is provided in Appendix A (Fig. 14), with the Vague Prompt and Explicit Grouping Prompt derived from it by removing or adding semantic constraints, respectively.

Table 7 Experiment of sensitivity to text-conversion granularity on `3s5z` and `10 m_vs_11m`

Conversion Variant	3s5z	10 m_vs_11m
Full conversion	0.96 ± 0.01	0.91 ± 0.03
Compressed conversion	0.94 ± 0.02	0.89 ± 0.03
w/o StarCraft Wiki	0.92 ± 0.02	0.87 ± 0.03
w/o spatial description	0.89 ± 0.03	0.82 ± 0.04

Before analyzing prompt-level sensitivity, we first examine whether the observation-to-text conversion itself affects the robustness of the semantic prior. Since prompt variation can only be meaningfully isolated after the text-conversion pipeline is fixed, we compare the full conversion pipeline with several simplified variants on two representative maps, namely `3s5z` and `10 m_vs_11m`.

Table 7 shows that the full conversion achieves the best overall performance on both maps, while the compressed conversion remains highly competitive. This indicates that the effectiveness of the framework does not depend on textual verbosity itself, provided that the main tactical structure is preserved. Removing the StarCraft Wiki descriptions causes only moderate degradation, suggesting that auxiliary unit knowledge is helpful but not the dominant source of performance. By contrast, removing spatial descriptions leads to the largest drop on both 3s5z and 10 m_vs_11m, indicating that explicit spatial structure appears to be the most influential component among the tested text-conversion factors for downstream decision quality. Based on this result, all subsequent prompt-sensitivity experiments are conducted using the full conversion as the default textual input format.

With the text-conversion pipeline fixed to the full setting, we next examine prompt-level sensitivity by analyzing how different prompt designs affect the semantic prior generated by the LLM.Using the 3s5z map as an illustrative example, the three prompt variants produce qualitatively different LLM outputs under the same observation input. Under the Vague Prompt, the absence of unit knowledge and reasoning constraints leads to both semantic impoverishment and logical inconsistency: the LLM cannot infer that Stalkers have an attack range of 6 or that Zealots are melee units suited for frontline pressure, and may therefore produce tactically weak or logically inconsistent actions, such as assigning attack commands to units still under cooldown (e.g., Agent 6) or failing to withdraw critically low-HP units (e.g., Agent 2). In contrast, under the Standard Strategic Prompt, the

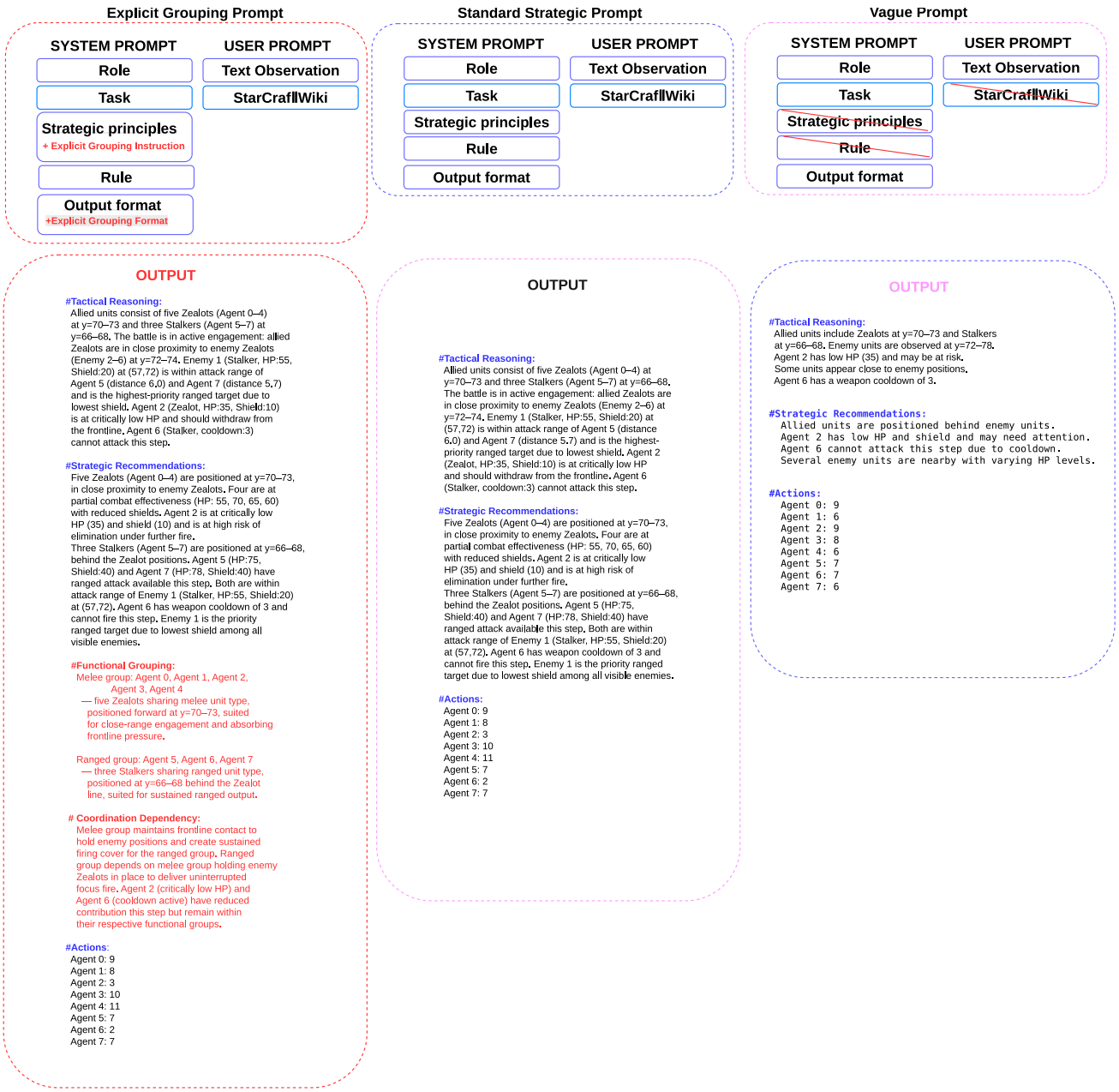

Fig. 7 Structural comparison of the three prompt variants used in the prompt sensitivity study: Explicit Grouping Prompt, Standard Strategic Prompt and Vague Prompt

LLM generates substantially more precise tactical reasoning. It correctly identifies Enemy 1 as the priority ranged target based on verified attack distances (e.g., 6.0 and 5.7 for Agents 5 and 7), and its strategic recommendations accurately describe the spatial layering of Zealots at y=70–73 and Stalkers at y=66–68 together with attack availability and cooldown states. The Explicit Grouping Prompt preserves this tactical precision while further requiring the LLM to infer functional groups from unit type and position, explicitly assigning agents to melee and ranged groups and describing the coordination dependencies between them.

To isolate the direct effect of semantic prompt design on graph induction, we examine the coordination structures learned under the three prompt variants before introducing Critic-based filtering into the analysis. Figure 8 visualizes the induced coordination graphs at the end of Stage I under the Critic-removed setting, thereby highlighting how different semantic priors alone shape the topology.

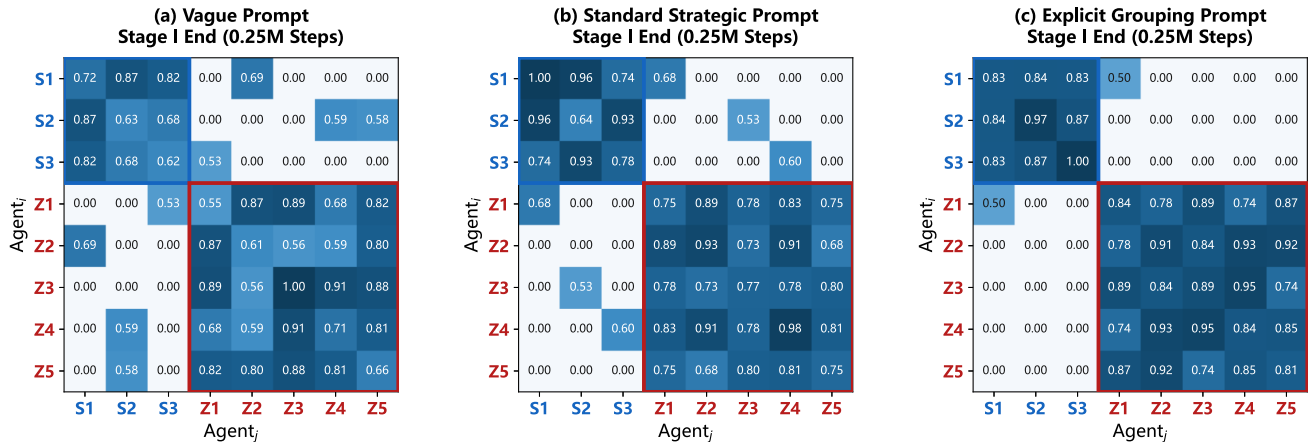

Fig. 8 Induced coordination graphs under the three prompt variants at the end of Stage I without LLM-Critic filtering

In Fig. 8, the upper-left blue block corresponds to the intra-group connectivity among Stalker units (S1–S3), and the lower-right red block corresponds to the intra-group connectivity among Zealot units (Z1–Z5), whereas the off-diagonal regions represent cross-group coordination between the two unit types. From left to right, the three prompt variants exhibit a clear structural progression. Under the Vague Prompt, the off-diagonal regions contain the largest number of nonzero entries, indicating relatively diffuse and weakly organized cross-group couplings, while the overall intra-group cohesion remains weakest. Under the Standard Strategic Prompt, the two diagonal blocks become more coherent while only a small number of cross-group edges remain active, yielding a more balanced topology. Under the Explicit Grouping Prompt, the diagonal blocks are strongest and the off-diagonal regions are nearly suppressed, revealing a more rigid coordination prior. Quantitatively, cross-group connectivity decreases monotonically from left to right, with 4, 3 and 1 nonzero cross-group edges, respectively, while the mean intra-group strength increases from 0.760 to 0.817 and 0.859. This monotonic reduction in cross-group connectivity indicates that the prompt increasingly constrains the freedom of inter-group coordination, pushing the induced topology from an under-structured regime with scattered cross-group couplings toward a more rigid regime in which inter-group interaction is progressively suppressed. At the same time, the increase in mean intra-group strength shows that stronger semantic priors reinforce within-group cohesion.

Importantly, these structural patterns are qualitatively consistent with the tactical content of the corresponding prompt outputs: the Vague Prompt yields under-specified battlefield reasoning and weak role differentiation, matching the most diffuse topology; the Standard Strategic Prompt explicitly separates frontline Zealots from backline Stalkers and identifies concrete target priorities, matching the most balanced coordination structure; and the Explicit Grouping Prompt further appends functional grouping and coordination dependency information, matching the strongest intra-group blocks and the sparsest cross-group links. Taken together, these two trends reveal a structural tradeoff controlled by prompt design: richer and more prescriptive semantics improve group coherence, but can also reduce the flexibility of cross-group coordination when the prior becomes overly rigid. These results are consistent with the interpretation that prompt semantics are reflected in interpretable graph structures, rather than merely affecting the wording of the LLM outputs.

Although the LLM-guided strategic semantics, actions, and latent rewards produced in Stage I all rely on LLM-based modules, each component operates under its own dedicated prompt with distinct objectives and output specifications. As detailed in Appendix A (Figs. 14 and 15), the Action prompt is proposal-oriented in that it receives structured battlefield observations and domain knowledge as input and generates tactical reasoning, strategic recommendations, and executable actions. The Critic prompt, by contrast, is verification-oriented in that it evaluates the generated action–semantic pair against explicit criteria of numerical feasibility, semantic consistency, and tactical soundness. Importantly, these criteria are grounded in structured environment observations and hard constraints derived from observable quantities, including unit states, cooldown availability, and spatial relations, rather than free-form self-justification. The Critic is therefore designed to serve as an environment-grounded verifier, rather than merely repeating the Action LLM's semantic outputs of the Action LLM. The two modules thus constitute a functionally decoupled generator–verifier architecture with distinct objectives, output formats, and decision criteria. The experiments that follow examine whether this functional separation provides nontrivial filtering value beyond a trivial

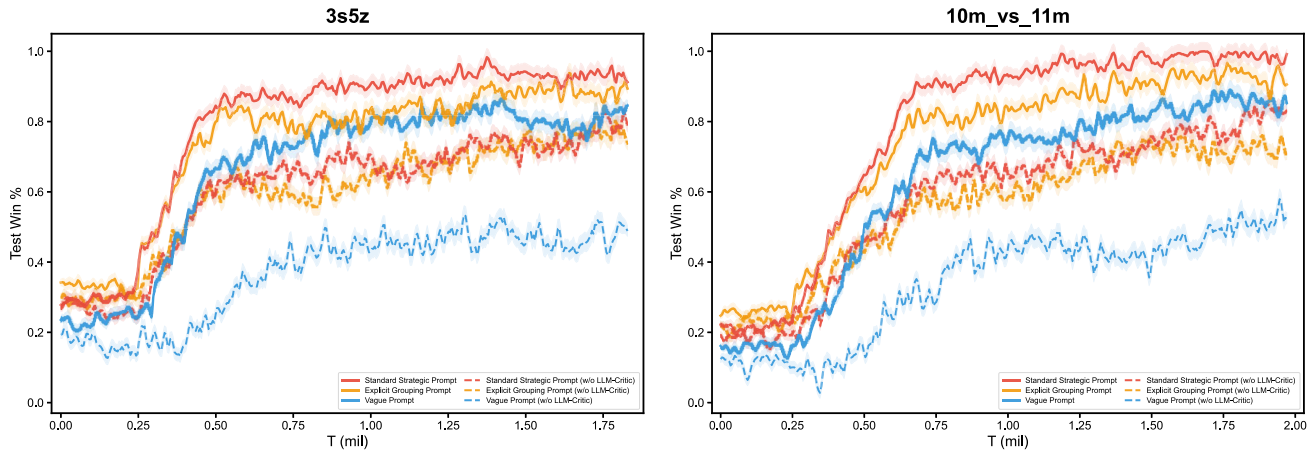

Fig. 9 Experiment of LLM-Critic ablation on 3s5z and $10m_vs_11m$. Solid curves denote the results obtained by the full LLM-GNCF framework under three prompt variants, while dashed curves denote the corresponding ablated settings without the LLM-Critic (w/o LLM-Critic)

self-reinforcing loop, and whether strategically suboptimal guidance introduces recoverable or destabilizing bias into downstream policy learning.

Figure 9 evaluates three prompt variants under the full two-stage pipeline, with and without the LLM-Critic, on 3s5z and $10m_vs_11m$. The six conditions form a consistent performance hierarchy rather than exhibiting catastrophic collapse. Across both maps, the Standard Strategic Prompt with LLM-Critic achieves the best final performance, while the Vague Prompt without LLM-Critic performs worst. Importantly, even the most degraded setting remains trainable and continues to improve during Stage II, indicating that low-quality semantic guidance weakens the learned prior but does not lead to collapse in the tested setting.

The effect of the LLM-Critic is consistent across all three prompt variants. For each prompt, the Critic-enabled setting outperforms its Critic-removed counterpart, and this gain is most pronounced under the weakest prompt condition. In particular, the final Critic gap under the Vague Prompt reaches 33.3 percentage points on $3s5z$ and 36.4 percentage points on $10m_vs_11m$, substantially larger than the corresponding gain under the Standard Strategic Prompt. This pattern is less consistent with a trivial self-reinforcing validation loop. Instead, it suggests that the Critic provides meaningful filtering value, especially when the upstream semantic guidance is unreliable. An additional observation is that, during Stage I, both the Standard Strategic Prompt and the Explicit Grouping Prompt without LLM-Critic still outperform the Critic-enabled Vague Prompt on both maps. This suggests that prompt quality appears to play a stronger role than Critic-based filtering in these settings in the proposed framework: informative semantic guidance can establish stronger tactical priors than a weak prompt, even when the latter is supplemented by the LLM-Critic. The Critic therefore serves as a complementary verifier rather than a substitute for high-quality semantic conditioning.

The Explicit Grouping Prompt provides a direct test of strategically coherent but potentially suboptimal LLM guidance. During Stage I, it reaches a higher plateau than the Standard Strategic Prompt on both maps (33.0% vs. 29.8% on 3s5z, and 26.1% vs. 21.6% on $10m_vs_11m$), indicating that explicit grouping instructions can improve short-term coordination consistency. However, once Stage II fine-tuning begins, the Standard Strategic Prompt overtakes Explicit Grouping at approximately 0.35M steps on 3s5z and 0.5M steps on $10m_vs_11m$, and then maintains a widening lead throughout the rest of training. It ultimately converges to a higher final win rate (92.9% vs. 88.7% on 3s5z, and 98.5% vs. 92.6% on 10m_vs_11m), corresponding to final gains of 4.2 pp and 5.9 pp, respectively. This crossover pattern suggests that over-prescriptive semantic guidance introduces a recoverable form of bias in later refinement: it improves early consistency, but constrains the flexibility of the learned structural prior and limits subsequent policy refinement.

Taken together, these results suggest that LLM-GNCF does not require LLM guidance to be strategically optimal. Rather, LLM outputs act as high-level but imperfect priors, whose quality affects early coordination behavior, while Stage II reinforcement learning remains responsible for later correction and refinement.

Semantic-to-Topology Mapping and Structural-Prior Persistence

Although the previous subsection shows that prompt-cond itioned semantic guidance induces distinguishable coordination graphs, and further demonstrates that the LLM-Critic provides meaningful filtering value rather than a trivial self-

reinforcing loop, it remains unclear whether the structural prior learned in Stage I is preserved and reused after the LLM is removed in Stage II. This subsection therefore focuses on whether the Stage-I structural prior is retained as a transferable coordination structure that persists across the two-stage transition, rather than being reduced to a transient warm-start effect.

To directly examine whether the structural prior learned in Stage I persists over time, we compare the final coordination graphs at two checkpoints, namely the end of Stage I and the end of Stage II, as shown in Fig. 10. The comparison is conducted on two maps, namely 3s5z and $10m_vs_11m$. To quantify structural persistence, we report three measurements: the mean within-block and cross-block connectivity, the cosine similarity between the Stage I and Stage II graphs and the edge preservation rate, defined as the percentage of nonzero Stage-I graph edges that are still present in Stage II.

On 3s5z, Fig. 10a–c shows that Stage I already forms a clear block structure, with strong within-block connectivity in the Stalker block (mean = 0.907) and the Zealot block (mean = 0.859), while cross-block connectivity remains limited (mean = 0.110, with 3 nonzero cross-block edges of magnitude 0.518−0.598). After 1.75M additional steps of LLM-free RL fine-tuning, the two within-block means change only marginally: the Stalker block mean decreases from 0.907 to 0.901, and the Zealot block mean decreases from 0.859 to 0.850. By contrast, cross-block connectivity increases substantially: its mean rises from 0.110 to 0.265, and the number of nonzero cross-block edges rises from 3 to 6, with the largest value reaching 0.911. The cosine similarity between the Stage I and Stage II graphs is 0.936, and 95% of nonzero Stage-I edges remain nonzero in Stage II. The corresponding difference map shows that the largest changes are concentrated in the cross-block region, while the dominant diagonal blocks remain largely preserved.

A similar pattern appears on $10m_vs_11m$ in Fig. 10(d)–(f). Stage I exhibits two cohesive reference blocks with mean within-block strengths of 0.826 and 0.798, while cross-block connectivity remains weak (mean = 0.062, with 3 nonzero cross-block edges). After Stage II fine-tuning, the two within-block means again change only slightly: the first block mean decreases from 0.826 to 0.814, and the second block mean decreases from 0.798 to 0.784. Meanwhile, cross-block connectivity increases to 0.106 and the number of nonzero cross-block edges rises from 3 to 4, with strengthened values up to 0.852. The cosine similarity reaches 0.952, and the edge preservation rate is 93%.

Taken together, these structural observations help clarify why the two-stage framework yields sustained gains. The high edge preservation rate (93–95%) and the minimal shifts in within-block means indicate that the coordination graph learned in Stage I remains largely stable during Stage II, suggesting that the benefit of the second stage is unlikely to be fully explained by a simple warm-start effect based only on favorable initialization. In addition, the difference maps do not reveal a uniform graph-wide reorganization after further training; instead, the observed changes are concentrated in a small number of cross-block edges superimposed on an otherwise preserved Stage-I structure, implying that the improvement is not merely a consequence of longer training time. Finally, the consistently high cosine similarity between the Stage I and Stage II graphs provides quantitative support for the view that the structural prior acquired during LLM-guided pre-training persists throughout the subsequent LLM-free fine-tuning phase. Overall, Fig. 10 suggests that Stage II appears to build upon the coordination structure established in Stage I, with selective refinement instead of full relearning from scratch.

Grounding and credit assignment quality of the LLM Latent Reward

This subsection examines whether the semantic criteria underlying the LLM latent reward are grounded in objectively measurable environment-level quantities, rather than remaining purely semantic scoring heuristics without independent environmental validation. To this end, for each map we analyze the four dimensions of the latent reward, namely r^t_{KE}, r^t_{FFC}, r^t_{FM}, and r^t_{SST}, using a compact set of four representative environment-derived tactical proxies: Kiting Effectiveness (KE), Focus-Fire Coordination (FFC), Formation Maintenance (FM), and Skill Synergy Timing (SST). These proxies are computed directly from the underlying environment state without using any LLM outputs. They are not intended to exhaust all environment-level evidence involved in the semantic reward construction; rather, they serve as four salient and independently measurable reference criteria for testing the objectivity and consistency of the LLM-reward mechanism. We further contrast these step-level relations with the return-to-go:

$$J^t_{\mathrm{RTG}} = \sum_{k=0}^{T-t} \gamma^k R^{t+k}_{\mathrm{env}}, \tag{25}$$

where T denotes the episode length. Unlike J^t_k in Eq. (21), J^t_{RTG} extends to the end of the episode and thus captures trajectory-level outcome rather than step-level tactical quality. The precise definitions of the four tactical proxies are provided in Appendix B.

Figure 11 shows that the LLM latent reward is systematically aligned with observable tactical state. On both 3s5z and 2c_vs_64zg, each latent-reward dimension exhibits a strong positive correlation with its matched environment-derived tactical proxy. Across the eight map-dimension pairs shown in the figure, the Pearson correlation ranges from 0.630 to

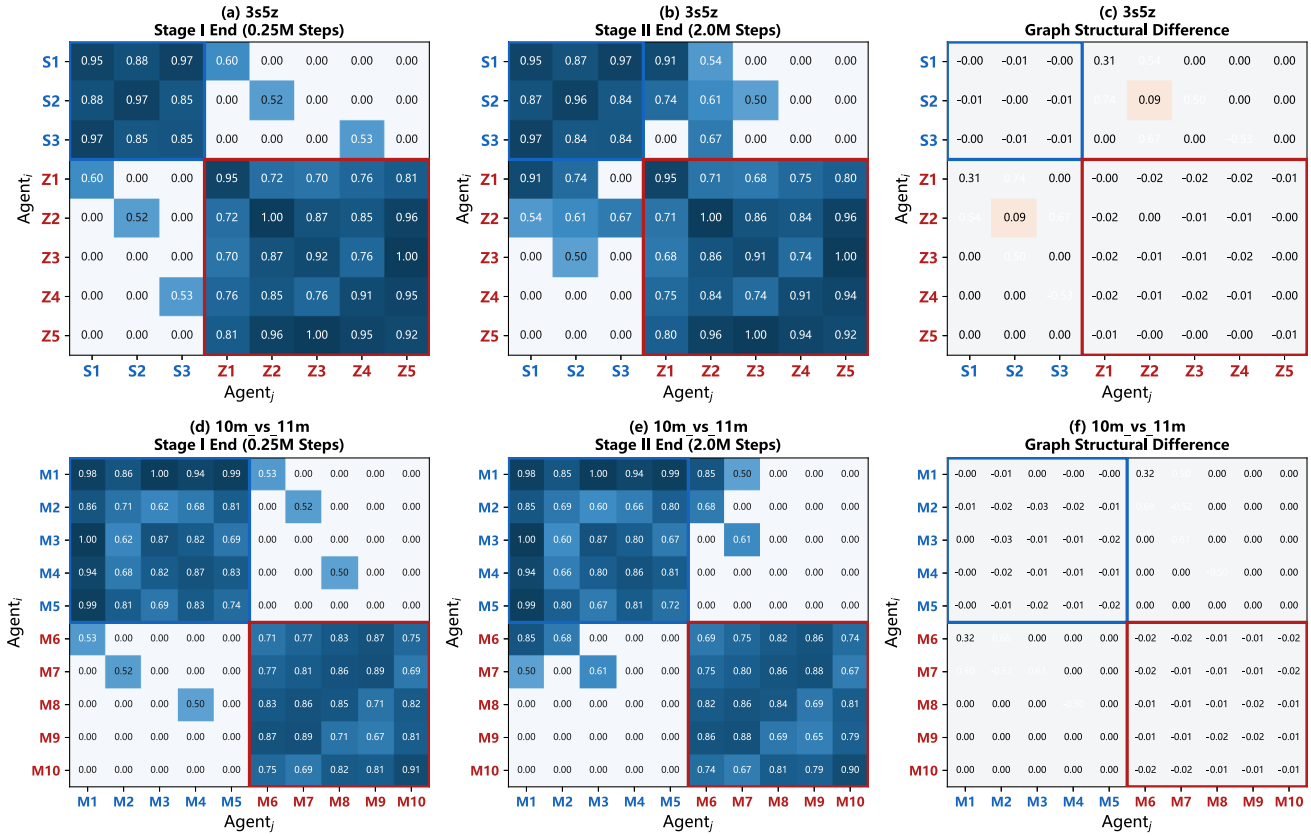

Fig. 10 Coordination graph structures at Stage I end (0.25M steps) and Stage II end (2.0M steps), together with their difference maps, on 3s5z (a–c) and $10m_vs_11m$ (d–f)

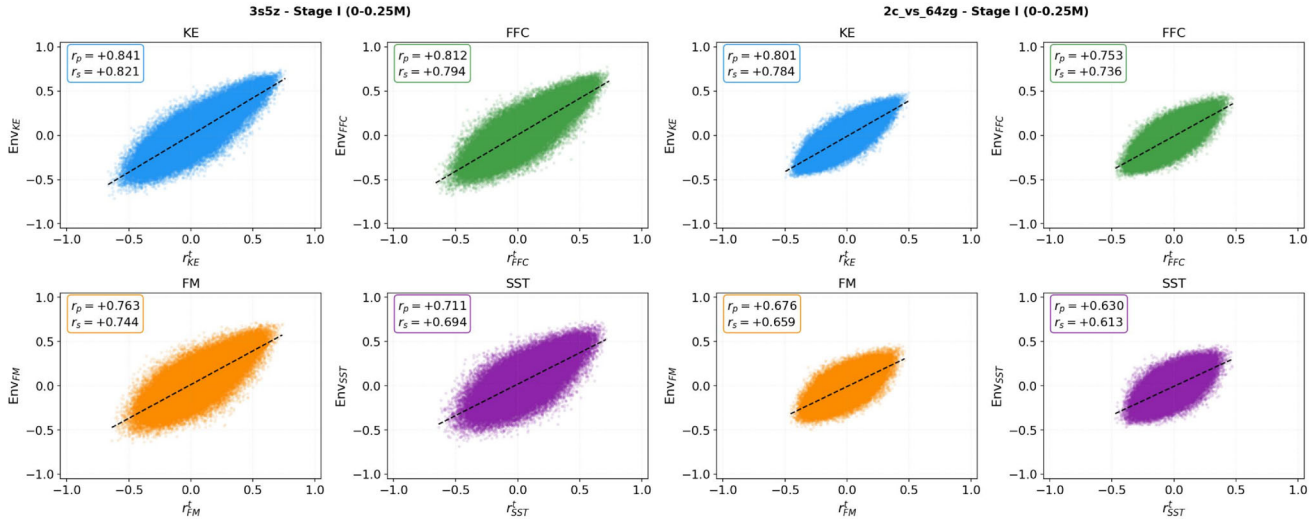

Fig. 11 Alignment between latent reward dimensions and matched environment-derived tactical proxies during Stage I (0–0.25M steps). Each panel shows the correlation between one latent reward dimension and its corresponding proxy on 3s5z and 2c_vs_64zg. Pearson and Spearman coefficients are reported in each subplot

0.841, and the Spearman correlation ranges from 0.613 to 0.821, indicating that the alignment is strong not only in magnitude but also in rank order. The strongest correlations consistently appear on KE and FFC, while FM and SST remain substantially positive on both maps. Notably, the correlations are uniformly higher on 3s5z than on 2c_vs_64zg,

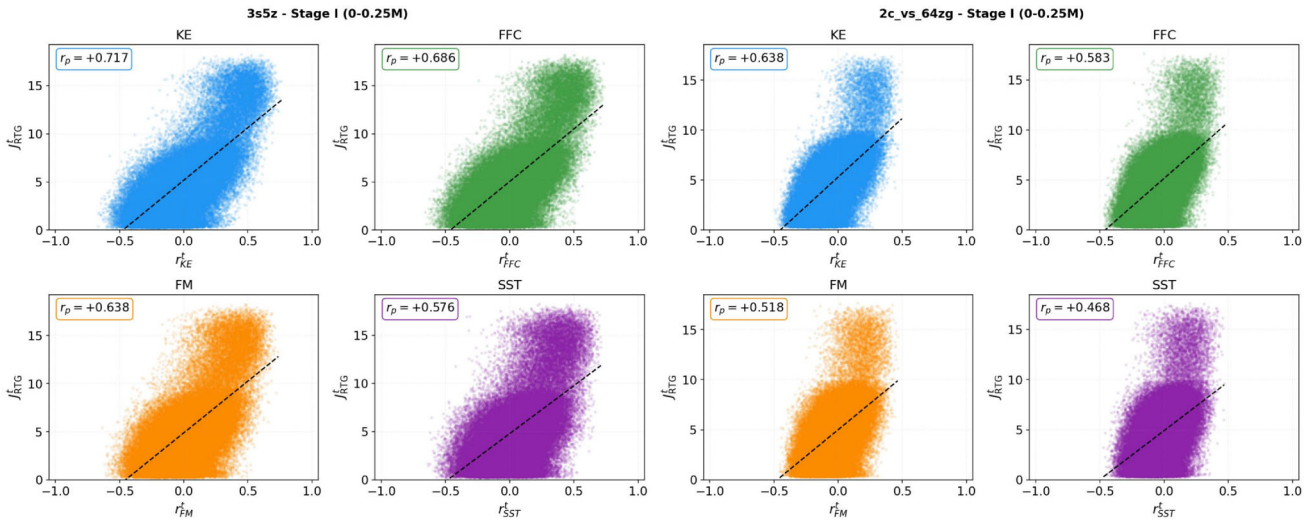

Fig. 12 Control comparison between latent reward dimensions and return-to-go during Stage I (0–0.25M steps). For both maps, the correlation with matched tactical proxies is consistently stronger than the correlation with J^t_{RTG}, indicating that the latent reward is more closely tied to immediate tactical quality than to trajectory-level outcome alone

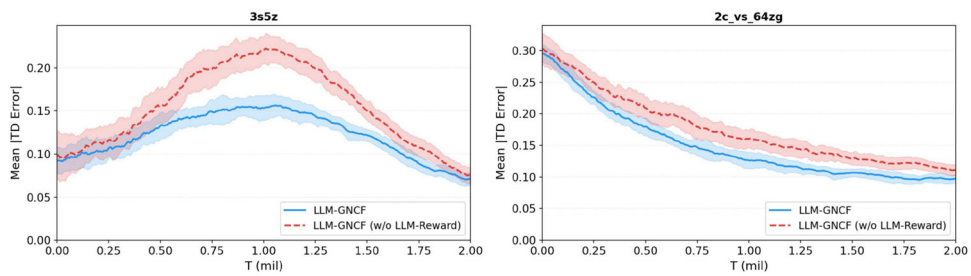

Fig. 13 Mean |TD Error| over training for LLM-GNCF and its variant without LLM reward. The curves illustrate the reduction in value-estimation error across training on 3s5z and 2c_vs_64zg

especially for FM and SST, which is consistent with the richer coordination structure of 3s5z and the broader tactical variation reflected in the latent reward. Because these proxies are computed solely from low-level environment variables and contain no LLM outputs, the observed alignment suggests that the latent reward dimensions appear to be well aligned with independently measurable tactical properties rather than arbitrary semantic scores.

Figure 12 presents the key control comparison against trajectory-level return. Across all maps and all reward dimensions shown in the figure, the correlation between each latent reward dimension and its matched tactical proxy is consistently stronger than that with the return-to-go. On 3s5z, for example, the KE and SST dimensions exhibit correlations of 0.841 and 0.711 with their matched tactical proxies, compared with 0.717 and 0.576, respectively, with J^t_{RTG}. On $2c_vs_64zg$, the corresponding correlations are 0.801 and 0.638 for KE, and 0.630 and 0.468 for SST. This consistent separation suggests that the latent reward is more responsive to immediate tactical quality than to trajectory-level outcome alone. Moreover, relative to the broader and more discriminative patterns observed on 3s5z, the lower and more concentrated distributions on $2c_vs_64zg$ suggest that the latent reward remains grounded even under a more compressed early-stage tactical regime.

Figure 13 shows that the reward-augmented setting maintains lower mean absolute temporal-difference (TD) error than the ablated baseline throughout training on both maps. On 3s5z, both variants exhibit an inverted-U trajectory, but the curve under LLM-GNCF remains consistently below that of the w/o LLM-reward variant, with the separation becoming most visible in the middle stage of training. This pattern is consistent with more stable Bellman fitting during the phase in which coordinated behavior is still being formed. On 2c_vs_64zg, by contrast, both curves decrease monotonically from the beginning, and the reward-augmented model remains below the ablated baseline over the entire training horizon, indicating smoother reduction of value-estimation error in a setting where informative damage signals are available early. Taken together, these trends suggest that the LLM-reward provides denser and more locally aligned supervision for value learning, which is consistent with improved step-level credit assignment, rather than only affecting final returns.

Conclusion

This paper presents the Large Language Model-Guided Graph Neural Coordination Framework (LLM-GNCF), an integrated framework for cooperative multi-agent reinforcement learning that links high-level semantic reasoning to low-level graph-based control. Unlike prior approaches that construct coordination structures primarily from the trajectory-level or observation-based statistics, LLM-GNCF integrates natural language policy semantics into hierarchical inter-agent coordination. The proposed Team-Adaptive Coordination Graph provides agents with high-level action demonstrations as informative priors and captures the semantic dependencies over consecutive frames to model the trajectory-level group behavior. LLM-GNCF adopts a two-stage training paradigm. In Stage I, LLM-based semantic reasoning guides the transfer from strategic interpretation to coordination graph construction. In Stage II, the learned structure is further optimized using environment rewards, which reduces inefficient exploration and improves convergence stability. Experiments on the SMAC benchmark indicate that LLM-GNCF achieves competitive results compared to representative graph-based baselines across the evaluated scenarios. Ablation studies support the effectiveness of LLM-guided action supervision, semantic grouping via TACG and the latent reward shaping mechanism, and indicate that LLM-based semantic guidance is important for dynamic agent grouping and coordination. In addition, the Chain of Aggregation mechanism effectively manages token usage while preserving semantic consistency, and comparisons across different LLM backbones suggest that stronger reasoning capabilities further enhance multi-agent coordination. Overall, this work suggests that combining LLM-based strategic reasoning with decentralized policy optimization is a promising direction for scalable, interpretable and semantically aligned multi-agent reinforcement learning.

Appendix A Prompt templates

See Figs. 14, 15.

Appendix B Environment proxy definitions

B.1 Environment-derived tactical proxy metrics

To examine whether the LLM latent reward is aligned with independently measurable tactical behavior, we construct four *environment-derived proxy metrics* directly from raw SMAC trajectories, without using any LLM output or semantic annotation. These quantities are not intended to serve as exhaustive ground-truth descriptions of tactical quality. Rather, each proxy operationalizes one measurable aspect of the corresponding tactical concept using only environment states and executed actions.

All proxies are computed over a discrete Chain-of-Aggregation (CoA) window

$$W = \{t_0, t_0 + 1, \ldots, t_0 + K - 1\}, \qquad K = 10,$$

where t_0 is the starting environment step of the window, and K is the number of environment transitions aggregated in one window. Since some formulas use $t + 1$, the trajectory segment must include state information up to step $t_0 + K$.

At each step t, let $\mathcal{I}_t$ denote the set of alive allied unit indices and $\mathcal{J}_t$ denote the set of alive enemy unit indices. For any unit i, let

$$\mathbf{p}_i(t) = (x_i(t), y_i(t))$$

be its 2D position at step t, where $x_i(t)$ and $y_i(t)$ are the environment-provided horizontal and vertical coordinates. Let

$$h_i(t) = \text{health}_i(t) + \text{shield}_i(t)$$

denote its effective hit points, where $\text{health}_i(t)$ is the unit's health value from the environment and $\text{shield}_i(t)$ is its shield value, with $\text{shield}_i(t) = 0$ for non-Protoss units.

For allied unit i, let u_i^t denote the executed discrete action ID at step t. In the SMAC action space used here, movement and stop actions occupy the first six action IDs, while attack actions satisfy $u_i^t \geq 6$. We use $\mathbf{1}[\cdot]$ for the indicator function, $\|\cdot\|_2$ for the Euclidean norm, and

$$\text{clip}(z, 0, 1) = \min(1, \max(0, z)).$$

We set $\varepsilon = 10^{-6}$ to avoid division by zero. Unless otherwise stated, if a quantity becomes undefined because the corresponding alive-unit set is empty, its per-step contribution is set to zero.

For consistency with the main text, all four proxies are normalized to the common range $[-1, 1]$, thereby facilitating relative comparison across maps and training stages.

B.1.1 Env_{KE}

Env_{KE} is designed to capture the *damage-exchange efficiency* associated with effective kiting, i.e., whether the allied team inflicts more damage than it receives within a short time window. It does not directly measure movement-level kiting behavior itself; instead, it quantifies the resulting local damage trade-off.

We first define the one-step damage suffered by unit i as

$$d_i(t) = \max\big(0,\ h_i(t) - h_i(t+1)\big).$$

Fig. 14 Complete prompt template of the Action LLM, illustrating the components provided to generate tactical reasoning, strategic recommendations, and executable actions

Standard Strategic Prompt

SYSTEM PROMPT

#Role
You are a high-level strategic commander for cooperative combat decision-making in StarCraft II.

#Task
Given a structured battlefield observation, you must produce TWO outputs in strict order:

Tactical Reasoning and Strategic Recommendations:

Tactical Reasoning:
- Spatial layout of allied and enemy units
- Current HP and shield status of each unit
- Immediate threats and vulnerable units
- Tactical opportunities in the current state

Strategic Recommendations:
- Objective state of each visible allied unit type
- Position relative to enemies and allies
- Combat effectiveness and attack availability
- Coordination-relevant spatial relationships
- No group labels, team names, or role instructions

Actions:
Generate executable actions for allied units, consistent with the situation described above.

#Strategic principles:
1. Situational Inference
All judgments must be derived strictly from the provided observation. Do not assume any tactical configuration before reading the unit states.

2. Spatial Awareness
Identify which allied units are closer to enemies, which are further back, and what spatial relationships exist between unit types.

3. Combat Phase Assessment
Determine the current combat phase: engage, hold, reposition, or retreat, based on unit states and positions.

#Rules:
- All tactical judgments must be inferred strictly from the observation.
- Use only actions from the provided valid action set.
- Do not reference units, tags, or abilities absent from the observation.
- Actions assigned to the same unit must not contradict each other.
- List actions separately for each allied unit type.

#Output format:

Tactical Reasoning:
<Battlefield analysis derived from observation: spatial layout, unit states, threats, opportunities>

Strategic Recommendations:
<Objective description of each visible allied unit type: position, HP, shield, capability, cooldown, and spatial relationship to allies and enemies. No group labels. No role assignments.>

Please select the best action for each agent from the following discrete action space:

Index Action
--- ---
0 no-op (only when unit is dead)
1 Stop
2 Move North
3 Move South
4 Move East
5 Move West
6 Attack enemy 0
7 Attack enemy 1
8 Attack enemy 2
9 Attack enemy 3
10 Attack enemy 4
11 Attack enemy 5
12 Attack enemy 6
13 Attack enemy 7

Actions:
Agent 0: <action_index>
Agent 1: <action_index>
...
Agent 7: <action_index>

USER PROMPT

#Text Observation
Game Info
Time: 0:55

Map :
Map Name: 3s5z
Terrain: Open flat terrain, no obstacles.
Combat style: Close-range symmetric engagement.
Allied Composition: 3 Stalkers, 5 Zealots
Enemy Composition: 3 Stalkers, 5 Zealots

Allied Units:
Agent 0 Zealot Tag: 0x100140001 Pos: (55, 72)
Health: 55 Shield: 20
Agent 1 Zealot Tag: 0x100140002 Pos: (53, 73)
Health: 70 Shield: 30
Agent 2 Zealot Tag: 0x100140003 Pos: (51, 70)
Health: 35 Shield: 10
Agent 3 Zealot Tag: 0x100140004 Pos: (49, 72)
Health: 65 Shield: 25
Agent 4 Zealot Tag: 0x100140005 Pos: (47, 71)
Health: 60 Shield: 30
Agent 5 Stalker Tag: 0x100200001 Pos: (57, 66)
Health: 75 Shield: 40 Weapon_cooldown: 0
Agent 6 Stalker Tag: 0x100200002 Pos: (55, 67)
Health: 65 Shield: 30 Weapon_cooldown: 3
Agent 7 Stalker Tag: 0x100200003 Pos: (53, 68)
Health: 78 Shield: 40 Weapon_cooldown: 0

Observed Enemy Units:
Enemy 0 Stalker Tag: 0x101340001 Pos: (53, 73)
Health: 70 Shield: 50 Weapon_cooldown: 2
Enemy 1 Stalker Tag: 0x101280001 Pos: (57, 72)
Health: 55 Shield: 20 Weapon_cooldown: 0
Enemy 2 Zealot Tag: 0x1012c0001 Pos: (54, 73) Health: 80
Enemy 3 Zealot Tag: 0x1014c0001 Pos: (52, 72) Health: 90
Enemy 4 Zealot Tag: 0x1014c0002 Pos: (50, 74) Health: 85
Enemy 5 Zealot Tag: 0x1014c0003 Pos: (48, 73) Health: 95
Enemy 6 Zealot Tag: 0x1014c0004 Pos: (56, 74) Health: 88
Enemy 7 Stalker Tag: 0x101340002 Pos: (62, 78)
Health: 60 Shield: 45 Weapon_cooldown: 1

#StarCraftIIWiki:

Protoss.Stalker
- Type: Ranged ground unit capable of attacking both ground and air targets

- HP: 80 Shield: 80 Attack range: 6
- Tactical Attribute:
  Effective at ranged damage when safe distance is maintained.
  More vulnerable in close-range trades, especially after shield loss.
- Matchup Tendency:
  Performs better against Zealots when spacing is preserved.
  Can be pressured by Zealots once they close the distance.

Protoss.Zealot
- Type: Melee ground unit
- HP: 100 Shield: 50 Attack range: 0.1
- Tactical Attribute:
  Durable frontline unit suited for absorbing damage and applying pressure.
  Less effective when ranged enemies continuously maintain distance.
- Matchup Tendency:
  Performs well against Stalkers once in close contact.
  Can be kited by ranged units if distance is preserved.

#Task Instruction

Generate Tactical Reasoning,StrategicRecommendations and Actions strictly following the output format.

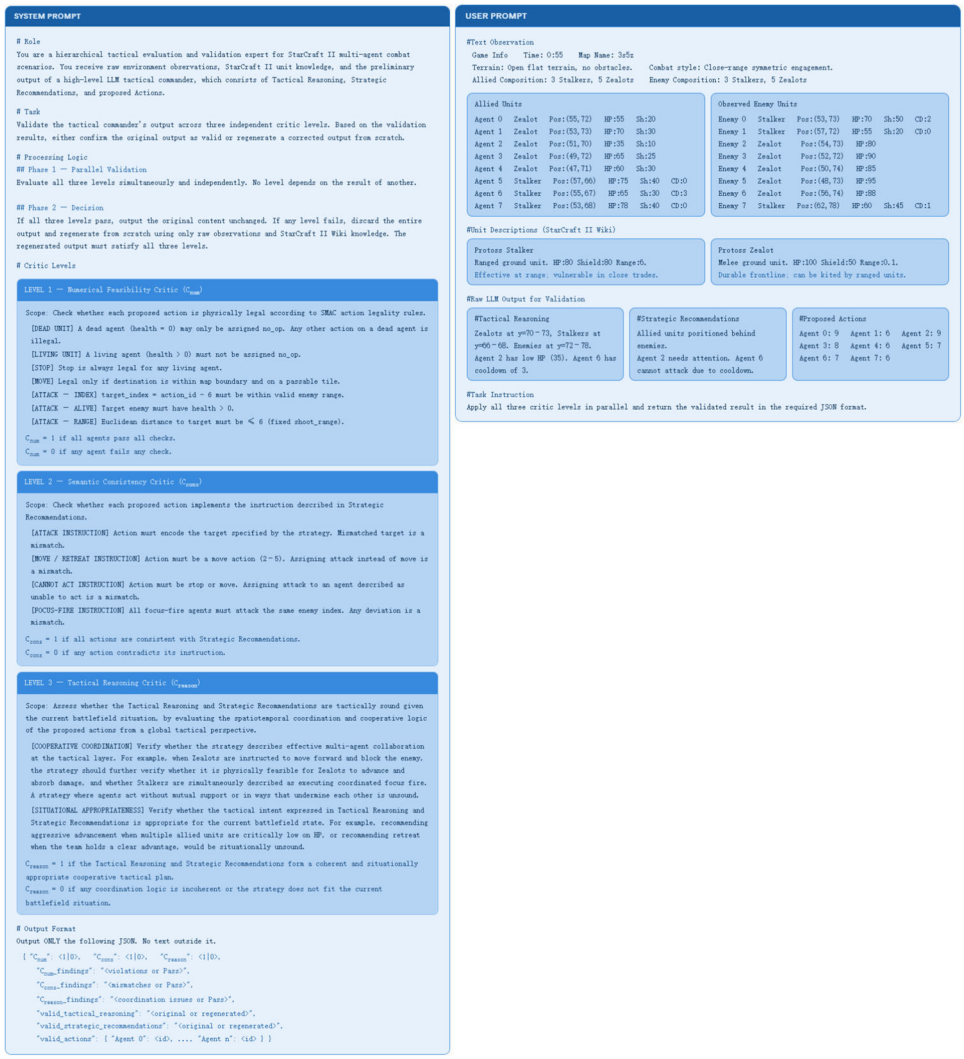

Fig. 15 Complete prompt template of the LLM-Critic, illustrating the verification criteria applied to evaluate the generated action-semantic pair for numerical feasibility, semantic consistency, and tactical soundness

Here, $d_i(t)$ is the decrease in effective hit points of unit i from step t to step $t+1$. If the unit gains shield or receives healing between the two steps, the decrement is truncated at zero by the $\max(\cdot, 0)$ operator.

The total enemy and allied damage accumulated within window W are then defined as

$$D_{\text{enemy}}(W) = \sum_{t \in W} \sum_{j \in \mathcal{J}_t} d_j(t), \qquad D_{\text{ally}}(W) = \sum_{t \in W} \sum_{i \in \mathcal{I}_t} d_i(t).$$

In the above equations, $D_{\text{enemy}}(W)$ is the total damage dealt by allies to all alive enemy units across the window, and $D_{\text{ally}}(W)$ is the total damage received by all alive allied units across the same window.

The resulting proxy is

$$\text{Env}_{\text{KE}}(W) = \frac{D_{\text{enemy}}(W) - D_{\text{ally}}(W)}{D_{\text{enemy}}(W) + D_{\text{ally}}(W) + \varepsilon} \in [-1, 1].$$

The numerator measures the net damage advantage of allies over enemies within the window, while the denominator normalizes this difference by the total amount of damage exchanged. A higher Env_{KE} indicates a more favorable local damage trade-off.

B.1.2 Env_{FFC}

Env_{FFC} measures how strongly allied agents concentrate their attacks on the same enemy target. It captures *target concentration* rather than the tactical optimality of the selected target.

Let u_i^t denote the discrete action ID executed by allied agent i at step t. In the SMAC action space, attacking enemy target j corresponds to action ID $6+j$, where j is the enemy index used by the environment.

For each enemy target j, define the number of allied agents attacking that target at step t as

$$n_j(t) = \left|\left\{i \in \mathcal{I}_t : u_i^t = 6 + j\right\}\right|.$$

Here, $n_j(t)$ counts how many alive allied agents are issuing attack actions toward enemy j at step t.

Let the total number of allied agents issuing attack actions at step t be

$$N_{\text{atk}}(t) = \sum_{j \in \mathcal{J}_t} n_j(t).$$

Thus, $N_{\text{atk}}(t)$ is the number of alive allied agents whose executed action at step t is an attack action toward some currently alive enemy.

The per-step focus-fire concentration is defined as

$$c(t) = \begin{cases} \dfrac{\max_{j \in \mathcal{J}_t} n_j(t)}{N_{\text{atk}}(t) + \varepsilon}, & \text{if } |\mathcal{J}_t| > 0, \\ 0, & \text{otherwise}, \end{cases} \qquad c(t) \in [0, 1].$$

In this expression, $\max_{j \in \mathcal{J}_t} n_j(t)$ is the largest number of allied agents focusing on the same enemy target at step t, and $c(t)$ is the fraction of all current attack actions assigned to the most-focused target.

We then define the window-level proxy as

$$\text{Env}_{\text{FFC}}(W) = 2 \cdot \left(\frac{1}{K} \sum_{t \in W} c(t)\right) - 1 \in [-1, 1].$$

The term $\frac{1}{K} \sum_{t \in W} c(t)$ is the average target-concentration level over the window, and the affine transformation $2(\cdot) - 1$ rescales it from $[0, 1]$ to $[-1, 1]$. A higher Env_{FFC} indicates stronger concentration of allied fire on a common target.

B.1.3 Env_{FM}

Env_{FM} is intended to quantify the temporal stability of allied spatial dispersion within a short window. It does not explicitly encode geometric formation shape; rather, it measures whether the team's overall spread remains stable over time.

For each step t, if $|\mathcal{I}_t| > 0$, we define the allied centroid as

$$\mathbf{c}(t) = \frac{1}{|\mathcal{I}_t|} \sum_{i \in \mathcal{I}_t} \mathbf{p}_i(t).$$

Here, $|\mathcal{I}_t|$ is the number of alive allied units at step t, and $\mathbf{c}(t)$ is the average allied position at that step.

The instantaneous formation spread is then defined as

$$s(t) = \begin{cases} \dfrac{1}{|\mathcal{I}_t|} \sum_{i \in \mathcal{I}_t} \|\mathbf{p}_i(t) - \mathbf{c}(t)\|_2, & \text{if } |\mathcal{I}_t| > 0, \\ 0, & \text{otherwise}. \end{cases}$$

In this expression, $\|\mathbf{p}_i(t) - \mathbf{c}(t)\|_2$ is the Euclidean distance between allied unit i and the allied centroid at step t, and $s(t)$ is the mean allied distance to the centroid, i.e., the average team spread at that step.

Its window average is

$$\bar{s}(W) = \frac{1}{K} \sum_{t \in W} s(t),$$

and the fluctuation of spread within the window is

$$v(W) = \frac{1}{K} \sum_{t \in W} |s(t) - \bar{s}(W)|.$$

Here, $\bar{s}(W)$ is the average allied spread over the window, and $v(W)$ measures how strongly that spread changes from step to step within the same window.

We normalize this fluctuation using a map-dependent reference distance

$$d_{\text{ref}} = 0.1\sqrt{L_x^2 + L_y^2}.$$

In this definition, L_x and L_y are the horizontal and vertical extents of the playable combat area provided by the environment. Therefore, $\sqrt{L_x^2 + L_y^2}$ is the diagonal length of the playable area, and d_{ref} is set to 10% of that diagonal as a normalization scale.

The final proxy is defined as

$$\text{Env}_{\text{FM}}(W) = 1 - 2 \cdot \text{clip}\left(\frac{v(W)}{d_{\text{ref}} + \varepsilon},\ 0,\ 1\right) \in [-1, 1].$$

The ratio $\frac{v(W)}{d_{\text{ref}} + \varepsilon}$ measures the spread fluctuation relative to the map size, the clip operator bounds it to $[0, 1]$, and the transformation $1 - 2(\cdot)$ maps lower fluctuation to higher scores. A higher Env_{FM} indicates more stable allied spatial dispersion and, correspondingly, stronger short-horizon formation maintenance.

B.1.4 Env_{SST}

To maintain notation consistency with the main text, we retain the symbol Env_{SST}. Since the environment trajectory does not provide a direct ground-truth measure of semantic *skill synergy timing*, we operationalize Env_{SST} here as a *shot-selection-timing proxy*: whether agents issue attack actions promptly when at least one enemy is already within attack range.

For allied agent i, define the attack-opportunity indicator

$$o_i(t) = \mathbf{1}\left[\exists\, j \in \mathcal{J}_t \text{ s.t. } \left\|\mathbf{p}_i(t) - \mathbf{p}_j(t)\right\|_2 \le r_i\right],$$

where $\mathbf{p}_i(t)$ and $\mathbf{p}_j(t)$ are the positions of allied agent i and enemy unit j at step t, respectively, and r_i is the shooting range of allied agent i. In the present SMAC environment implementation, this shooting-range threshold is provided by the environment and is fixed when evaluating the availability of attack actions.

Define the actual attack indicator as

$$\eta_i(t) = \mathbf{1}\left[u_i^t \geq 6\right].$$

Thus, $\eta_i(t) = 1$ means that allied agent i actually executes an attack action at step t, and $\eta_i(t) = 0$ means that it executes a non-attack action.

We then compute the raw window score

$$\mathrm{SST}_{\mathrm{raw}}(W) = \frac{\sum_{t\in W}\sum_{i\in\mathcal{I}_t} o_i(t)\,\eta_i(t)}{\sum_{t\in W}\sum_{i\in\mathcal{I}_t} o_i(t) + \varepsilon} \in [0, 1].$$

In the numerator, $o_i(t)\eta_i(t) = 1$ only when agent i both has an in-range attack opportunity and actually takes an attack action at step t. The denominator counts the total number of such in-range attack opportunities over all alive allied agents in the window. Therefore, $\mathrm{SST}_{\mathrm{raw}}(W)$ is the fraction of available attack opportunities that are converted into immediate attacks.

Finally, we map it to

$$\mathrm{Env}_{\mathrm{SST}}(W) = 2 \cdot \mathrm{SST}_{\mathrm{raw}}(W) - 1 \in [-1, 1].$$

A higher $\mathrm{Env}_{\mathrm{SST}}$ indicates that allied agents more consistently convert available in-range attack opportunities into immediate attack actions.

B.1.5 Summary

The four proxies above are computed solely from environment states and executed actions, without any semantic feedback from the LLM. They therefore provide independently measurable references for evaluating whether the LLM latent reward is grounded in observable tactical behavior.

Author Contributions • Zihao Kuang designed the overall LLM-GNCF framework, implemented the algorithms, conducted the experiments and took primary responsibility for drafting the manuscript. • Zengmin Xu supervised the research, contributed to the conceptual design, and critically reviewed the manuscript. • Linkai Wan contributed to the experimental analysis and manuscript preparation. • Chunjiong Zhang revised the manuscript. All authors read and approved the final manuscript.

Funding This research was supported in part by the Guangxi Natural Science Foundation (2024GXNSFAA010493), National Natural Science Foundation of China (61862015), Science and Technology Project of Guangxi (AD25069086).

Code availability statement The code for LLM-GNCF is currently not publicly available, as it is being used in ongoing research projects. It will be released upon completion of these projects.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethics approval Not applicable. This article does not contain any studies with human participants or animals performed by any of the authors.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Busoniu L, Babuska R, De Schutter B (2008) A comprehensive survey of multi-agent reinforcement learning. IEEE Trans Syst Man Cybern C Appl Rev 38(2):156–172
2. Du W, Ding S (2021) A survey on multi-agent deep reinforcement learning: from the perspective of challenges and applications. Artif Intell Rev 54(5):3215–3238
3. Duan, W., Lu, J., Xuan, J.: Bayesian ego-graph inference for networked multi-agent reinforcement learning. In: Proc. 39th Annual Conf. on Neural Information Processing Systems (NeurIPS 2025). Curran Associates Inc., San Diego, CA, USA (2025)
4. Zhou, M., Luo, J., Villella, J., Yang, Y., Rusu, D., Miao, J., Zhang, W., Alban, M., Fadakar, I., Chen, Z., *et al.:* SMARTS: An open-source scalable multi-agent RL training school for autonomous driving. In: Proc. 4th Conf. on Robot Learning (CoRL 2020), vol. 155, pp. 264–285. PMLR, Cambridge, MA, USA (2021)
5. Yeh, J.-C., Soo, V.-W.: Toward socially friendly autonomous driving using multi-agent deep reinforcement learning. In: Proc. 23rd Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2024), pp. 2573–2575. IFAAMAS, Auckland, New Zealand (2024)
6. Naderializadeh N, Sydir JJ, Simsek M, Nikopour H (2021) Resource management in wireless networks via multi-agent deep reinforcement learning. IEEE Trans Wireless Commun 20(6):3507–3523
7. Lv Z, Xiao L, Du Y, Zhu Y, Han S, Liu Y-J (2024) Efficient communications in multi-agent reinforcement learning for mobile applications. IEEE Trans Wireless Commun 23(9):12440–12454
8. Whiteson, S., Samvelyan, M., Rashid, T., De Witt, C.S., Farquhar, G., Nardelli, N., Rudner, T.G.J., Hung, C.-M., Torr, P.H.S., Foerster, J.: The StarCraft multi-agent challenge. In: Proc. 18th Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2019), pp. 2186–2188. IFAAMAS, Montreal, Canada (2019)

9. Shao J, Lou Z, Zhang H, Jiang Y, He S, Ji X (2022) Self-organized group for cooperative multi-agent reinforcement learning. Advances in Neural Information Processing Systems 35: Proc. 36th Annual Conference on Neural Information Processing Systems (NeurIPS 2022). Curran Associates Inc., New Orleans, LA, USA, pp 5711–5723
10. Chu, T., Chinchali, S., Katti, S.: Multi-agent reinforcement learning for networked system control. In: Proc. 8th Int. Conf. on Learning Representations (ICLR 2020). OpenReview.net, Addis Ababa, Ethiopia (2020)
11. Zhang Y, Zhou Y, Fujita H (2024) Distributed multi-agent reinforcement learning for cooperative low-carbon control of traffic network flow using cloud-based parallel optimization. IEEE Trans Intell Transp Syst 25(12):20715–20728
12. Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W.M., Zambaldi, V.F., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J.Z., Tuyls, K., Graepel, T.: Value-decomposition networks for cooperative multi-agent learning based on team reward. In: Proc. 17th Int. Conf. on Autonomous Agents and MultiAgent Systems (AAMAS 2018), pp. 2085–2087. IFAAMAS, Stockholm, Sweden (2018)
13. Rashid, T., Samvelyan, M., De Witt, C.S., Farquhar, G., Foerster, J.N., Whiteson, S.: QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In: Proc. 35th Int. Conf. on Machine Learning (ICML 2018), vol. 80, pp. 4292–4301. PMLR, Stockholm, Sweden (2018)
14. Son K, Kim D, Kang WJ, Hostallero DE, Yi Y (2019) QTRAN: learning to factorize with transformation for cooperative multi-agent reinforcement learning. Proc. 36th Int. Conf. on Machine Learning (ICML 2019), vol 97. Long Beach, CA, USA, pp 5887–5896 (**PMLR**)
15. Rashid T, Farquhar G, Peng B, Whiteson S (2020) Weighted QMIX: expanding monotonic value function factorisation for deep multi-agent reinforcement learning. Advances in Neural Information Processing Systems 33: Proc. 34th Annual Conference on Neural Information Processing Systems (NeurIPS 2020). Curran Associates Inc., Vancouver, Canada, pp 10199–10210
16. Wang J, Ren Z, Liu T, Yu Y, Zhang C (2021) QPLEX: Duplex dueling multi-agent Q-learning. Proc. 9th Int. Conf. on Learning Representations (ICLR 2021). OpenReview.net, Virtual Event. Austria,
17. Yu X, Lin Y, Wang X, Han S, Lv K (2024) GHQ: grouped hybrid Q-learning for cooperative heterogeneous multi-agent reinforcement learning. Complex Intell Syst 10:5261–5280
18. Liu, I.-J., Yeh, R.A., Schwing, A.G.: PIC: permutation invariant critic for multi-agent deep reinforcement learning. In: Proc. 3rd Conf. on Robot Learning (CoRL 2019), pp. 590–602. PMLR, Osaka, Japan (2020)
19. Böhmer W, Kurin V, Whiteson S (2020) Deep coordination graphs. Proc. 37th Int. Conf. on Machine Learning (ICML 2020), vol 119. PMLR, Virtual Event, pp 980–991
20. Li, S., Gupta, J.K., Morales, P., Allen, R.E., Kochenderfer, M.J.: Deep implicit coordination graphs for multi-agent reinforcement learning. In: Proc. 20th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2021), pp. 764–772. IFAAMAS, Virtual Event, United Kingdom (2021)
21. Liu Y, Wang W, Hu Y, Hao J, Chen X, Gao Y (2020) Multi-agent game abstraction via graph attention neural network. Proc. 34th AAAI Conf. on Artificial Intelligence (AAAI 2020). AAAI Press, New York, NY, USA, pp 7211–7218
22. Yang, Q., Dong, W., Ren, Z., Wang, J., Wang, T., Zhang, C.: Self-organized polynomial-time coordination graphs. In: Proc. 39th Int. Conf. on Machine Learning (ICML 2022), vol. 162, pp. 24963–24979. PMLR, Baltimore, MD, USA (2022)
23. Wang T, Zeng L, Dong W, Yang Q, Yu Y, Zhang C (2022) Context-aware sparse deep coordination graphs. Proc. 10th Int. Conf. on Learning Representations (ICLR 2022). OpenReview.net, Virtual Event
24. Duan W, Lu J, Xuan J (2025) Inferring latent temporal sparse coordination graph for multiagent reinforcement learning. IEEE Trans Neural Netw Learn Syst 36(8):14358–14370
25. Zang Y, He J, Li K, Fu H, Fu Q, Xing J, Cheng J (2023) Automatic grouping for efficient cooperative multi-agent reinforcement learning. Advances in Neural Information Processing Systems 36: Proc. 37th Annual Conference on Neural Information Processing Systems (NeurIPS 2023). Curran Associates Inc., New Orleans, LA, USA, pp 46105–46121
26. Duan, W., Lu, J., Xuan, J.: Group-aware coordination graph for multi-agent reinforcement learning. In: Proc. 33rd Int. Joint Conf. on Artificial Intelligence (IJCAI 2024), pp. 3926–3934. IJCAI Organization, Jeju, South Korea (2024)
27. Addula, S.R., Sajja, G.S.: Automated machine learning to streamline data-driven industrial application development. In: 2024 Second International Conference Computational and Characterization Techniques in Engineering & Sciences (IC3TES), pp. 1–4. IEEE, Lucknow, India (2024)
28. Durante, Z., Huang, Q., Wake, N., Gong, R., Park, J.S., Sarkar, B., Taori, R., Noda, Y., Terzopoulos, D., Choi, Y., et al.: Agent AI: Surveying the horizons of multimodal interaction. Preprint at https://arXiv.org/abs/2401.03568 (2024)
29. Huang, J., Chang, K.C.-C.: Towards reasoning in large language models: A survey. In: Proc. Assoc. Comput. Linguist.: Findings Of ACL 2023, pp. 1049–1065. ACL, Toronto, Canada (2023)
30. Cao Y, Zhao H, Cheng Y, Shu T, Chen Y, Liu G, Liang G, Zhao J, Yan J, Li Y (2025) Survey on large language model-enhanced reinforcement learning: concept, taxonomy, and methods. IEEE Trans Neural Netw Learn Syst 36(6):9737–9757
31. Geng, M., Pateria, S., Subagdja, B., Li, L., Zhao, X., Tan, A.-H.: L2M2: a hierarchical framework integrating large language model and multi-agent reinforcement learning. In: Proc. 34th Int. Joint Conf. on Artificial Intelligence (IJCAI 2025), pp. 99–107. IJCAI Organization, Montreal, Canada (2025)
32. Zhuang Y, Shen Y, Zhang Z, Chen Y, Miao F (2025) YOLO-MARL: You only LLM once for multi-agent reinforcement learning. 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, Hangzhou, China, pp 11283–11290
33. Qu, Y., Jiang, Y., Wang, B., Mao, Y., Wang, C., Liu, C., Ji, X.: Latent reward: LLM-empowered credit assignment in episodic reinforcement learning. In: Proc. 39th AAAI Conf. on Artificial Intelligence (AAAI 2025), pp. 20095–20103. AAAI Press, Philadelphia, PA, USA (2025)
34. Zhu G, Zhou R, Ji W, Zhao S (2025) LAMARL: LLM-aided multi-agent reinforcement learning for cooperative policy generation. IEEE Robot Autom Lett 10(7):7476–7483
35. Li H, Nourkhiz Mahjoub H, Chalaki B, Tadiparthi V, Lee K, Moradi Pari E, Lewis C, Sycara K (2024) Language grounded multi-agent reinforcement learning with human-interpretable communication. Advances in Neural Information Processing Systems 37: Proc. 38th Annual Conference on Neural Information Processing Systems (NeurIPS 2024). Curran Associates Inc., Vancouver, 1970 Canada, pp 87908–87933
36. Sarkar, B., Xia, W., Liu, C.K., Sadigh, D.: Training language models for social deduction with multi-agent Reinforcement Learning. In: Proc. 24th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2025), pp. 1830–1839. IFAAMAS, Detroit, MI, USA (2025)
37. Vinyals O, Babuschkin I, Czarnecki WM, Mathieu M, Dudzik A, Chung J, Choi DH, Powell R, Ewalds T, Georgiev P et al (2019) Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature 575(7782):350–354
38. Vinyals, O., Ewalds, T., Bartunov, S., Georgiev, P., Vezhnevets, A.S., Yeo, M., Makhzani, A., Küttler, H., Agapiou, J., Schrittwieser,

J., et al.: StarCraft II: A new challenge for reinforcement learning. Preprint at https://arXiv.org/abs/1708.04782 (2017)
39. Ma W, Mi Q, Zeng Y, Yan X, Lin R, Wu Y, Wang J, Zhang H (2024) Large language models play StarCraft II: Benchmarks and a chain of summarization approach. Advances in Neural Information Processing Systems 37: Proc. 38th Annual Conference on Neural Information Processing Systems (NeurIPS 2024). Curran Associates Inc., Vancouver, Canada, pp 133386–133442
40. Du W, Ding S, Zhang C, Shi Z (2023) Multiagent reinforcement learning with heterogeneous graph attention network. IEEE Trans Neural Netw Learn Syst 34(10):6851–6860
41. Ding S, Du W, Ding L, Zhang J, Guo L, An B (2025) Multi-agent reinforcement learning with graphical mutual information maximization. IEEE Trans Neural Netw Learn Syst 36(11):19515–19524
42. Jiang, J., Dun, C., Huang, T., Lu, Z.: Graph convolutional reinforcement learning. In: Proc. 8th Int. Conf. on Learning Representations (ICLR 2020). OpenReview.net, Addis Ababa, Ethiopia (2020)
43. Lin B-J, Lee C-Y (2024) HGAP: boosting permutation invariant and permutation equivariant in multi-agent reinforcement learning via graph attention network. Proc. 41st Int. Conf. on Machine Learning (ICML 2024), vol 235. PMLR, Vienna, Austria, pp 30189–30210
44. Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS (2021) A comprehensive survey on graph neural networks. IEEE Trans Neural Netw Learn Syst 32(1):4–24
45. Duan W, Xuan J, Qiao M, Lu J (2022) Learning from the dark: boosting graph convolutional neural networks with diverse negative samples. Proc. 36th AAAI Conf. on Artificial Intelligence (AAAI 2022). AAAI Press, Virtual Event, pp 6550–6558
46. Duan W, Xuan J, Qiao M, Lu J (2024) Graph convolutional neural networks with diverse negative samples via decomposed determinant point processes. IEEE Trans Neural Netw Learn Syst 35(12):18160–18171
47. Du, Y., Liu, B., Moens, V., Liu, Z., Ren, Z., Wang, J., Chen, X., Zhang, H.: Learning correlated communication topology in multi-agent reinforcement learning. In: Proc. 20th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2021), pp. 456–464. IFAAMAS, Virtual Event, United Kingdom (2021)
48. Phan T, Ritz F, Belzner L, Altmann P, Gabor T, Linnhoff-Popien C (2021) VAST: Value function factorization with variable agent subteams. Advances in Neural Information Processing Systems 34: Proc. 35th Annual Conference on Neural Information Processing Systems (NeurIPS 2021). Curran Associates Inc., Virtual Event, pp 24018–24032
49. Yang, Y., Zhou, T., Li, K., Tao, D., Li, L., Shen, L., He, X., Jiang, J., Shi, Y.: Embodied multi-modal agent trained by an LLM from a parallel TextWorld. In: Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR 2024), pp. 26265–26275. IEEE, Seattle, WA, USA (2024)
50. Kannan SS, Venkatesh VLN, Min B (2024) SMART-LLM: Smart multi-agent robot task planning using large language models. Proc. 2024 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2024). IEEE, Abu Dhabi, United Arab Emirates, pp 12140–12147
51. Fan L, Wang G, Jiang Y, Mandlekar A, Yang Y, Zhu H, Tang A, Huang D-A, Zhu Y, Anandkumar A (2022) MineDojo: Building open-ended embodied agents with internet-scale knowledge. Advances in Neural Information Processing Systems 35: Proc. 36th Annual Conference on Neural Information Processing Systems (NeurIPS 2022). Curran Associates Inc., New Orleans, LA, USA, pp 18343–18362
52. Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., Anandkumar, A.: Voyager: An open-ended embodied agent with large language models. Paper presented at the Intrinsically-Motivated and Open-Ended Learning Workshop, NeurIPS 2023; available at https://openreview.net/forum?id=nfx5IutEed (2023)
53. Yuan, H., Zhang, C., Wang, H., Xie, F., Cai, P., Dong, H., Lu, Z.: Plan4MC: Skill reinforcement learning and planning for open-world Minecraft tasks. Preprint at https://arxiv.org/abs/2303.16563
54. Oliehoek FA, Amato C (2016) A Concise Introduction to Decentralized POMDPs, vol 1. Springer, Cham
55. Zhang N, Zhao Y, Yang M, Dai S (2025) LLMs augmented hierarchical reinforcement learning with action primitives for long-horizon manipulation tasks. Sci Rep 15(1):36779
56. Wei Q, Li Y, Zhang J, Wang FY (2024) VGN: Value decomposition with graph attention networks for multiagent reinforcement learning. IEEE Trans Neural Netw Learn Syst 35(1):182–195
57. Li, Z., Ke, Z., Zhao, P.: Injecting structured knowledge into LLMs via graph neural networks. In: Proc. 1st Joint Workshop on Large Language Models and Structure Modeling (XLLM 2025), pp. 16–25. ACL, Vienna, Austria (2025)
58. Yu, P., Chen, G., Wang, J.: Table-critic: A multi-agent framework for collaborative criticism and refinement in table reasoning. In: Proc. 63rd Annual Meeting of the Assoc. Comput. Linguist. (ACL 2025), pp. 17432–17451. ACL, Vienna, Austria (2025)
59. Xie, T., Zhao, S., Wu, C.H., Liu, Y., Luo, Q., Zhong, V., Yang, Y., Yu, T.: Text2Reward: reward shaping with language models for reinforcement learning. In: Proc. 12th Int. Conf. on Learning Representations (ICLR 2024). OpenReview.net, Vienna, Austria (2024)
60. Ma, Y.J., Liang, W., Wang, G., Huang, D.-A., Bastani, O., Jayaraman, D., Zhu, Y., Fan, L., Anandkumar, A.: Eureka: human-level reward design via coding large language models. In: Proc. 12th Int. Conf. on Learning Representations (ICLR 2024). OpenReview.net, Vienna, Austria (2024)
61. Zheng Q, Henaff M, Zhang A, Grover A, Amos B (2025) Online intrinsic rewards for decision making agents from large language model feedback. Reinforcement Learning J 6:139–164
62. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: Proc. 26th Int. Conf. on Machine Learning (ICML 2009), pp. 41–48. ACM, Montreal, Canada (2009)
63. Yu C, Velu A, Vinitsky E, Gao J, Wang Y, Bayen A, Wu Y (2022) The surprising effectiveness of PPO in cooperative multi-agent games. Advances in Neural Information Processing Systems 35: Proc. 36th Annual Conference on Neural Information Processing Systems (NeurIPS 2022). Curran Associates Inc., New Orleans, LA, USA, pp 24611–24624

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.