Many Minds, One Path: LLM-Augmented Consensus Decision for Distributed Control in Multi-Agent Collaborative Stable Scenarios

Zhuohao Yu*, Zhe Liu*, Tao Ren†, Chenxue Wang, Junjie Wang, Qing Wang†

State Key Laboratory of Complex System Modeling and Simulation Technology, Beijing, China
Institute of Software Chinese Academy of Sciences, Beijing, China
University of Chinese Academy of Sciences, Beijing, China
{yuzhuohao2023, liuzhe2020, rentao22, wangchenxue2024, junjie, wq}@iscas.ac.cn

Abstract

Distributed multi-agent systems are increasingly deployed in dynamic and high-stakes environments such as power grids, intelligent traffic systems, and collaborative robotics. In these systems, long-term stability, the ability to maintain coherent and safe system behavior over time, is critical but underexplored in existing research. This paper presents LLMASC, a framework designed to enhance long-term stability in multi-agent collaboration by combining semantic reasoning with decentralized control. LLMASC comprises three key components: a *Semantic Perception Encoder* that transforms heterogeneous agent observations into structured natural language; an *LLM-Guided Consensus Decision* module that enables strategic alignment through proposal exchange and voting; and a *Policy Execution Controller* that maps high-level plans to executable actions via reinforcement learning. We evaluate LLMASC across three representative simulation domains (Multi-Walker, Simulation of Urban Mobility and Power Grid Stabilization), spanning both physical and cyber-physical systems. Experiments show that LLMASC consistently outperforms the best baselines, improving stability rates by up to 44% and long-term success by 31%. Further analysis confirms its decision-making efficiency and robustness under varying agent populations and model choices.

Code — https://github.com/yuzh2001/LLMASC

Introduction

Distributed multi-agent systems are increasingly deployed in mission-critical tasks and infrastructure control. Existing research focuses on task-driven collaboration scenarios (Rashid et al. 2020; Liu et al. 2021a; Yu et al. 2022) (e.g., winning a game) and short-term performance (Son et al. 2019; Peng et al. 2021), where multiple agents cooperate to complete a single, well-defined objective within a bounded time horizon. Unlike short-term tasks, many real-world systems, such as power grids (Wang et al. 2021; Chen et al. 2023) and city-scale traffic light networks (Wang, Cao, and Hussain 2021; Antonio and Maria-Dolores 2022; Zhou et al. 2022) require agents to continuously maintain system stability over the long term, often weeks, months, or even years. In a smart power grid (Lu et al. 2023; Yan and Xu 2024), maintaining long-term stability requires ongoing coordination among decentralized control units that react to fluctuating energy sources like wind or solar. Similarly, adaptive traffic lights must dynamically regulate flows to avoid long-term congestion or deadlock (Zhou et al. 2022; Mukhtar et al. 2023), responding to unpredictable events while preserving overall system reliability. While short-term task performance is important, ensuring stable, coherent, and sustainable collaboration over time introduces unique challenges that remain underexplored.

Considering the difference in objectives between short-term task execution (Carroll et al. 2019; Peng et al. 2021) and long-term stability maintenance, this paper focuses on enabling multi-agent systems to sustain reliable and coordinated behavior over extended time horizons. Long-term stability tasks (Aotani, Kobayashi, and Sugimoto 2021; Ma et al. 2023; Berducci et al. 2024) require agents to continuously adapt to environmental dynamics and maintain global coherence, often without a well-defined terminal condition. Agents must not only optimize their local actions, but also anticipate delayed effects, recover from transient inconsistencies, and preserve overall system integrity under partial observability and evolving conditions. This requires designing new ways of decision-making, communication, and coordination, shifting the focus from occasional optimization to sustained and flexible collaboration.

Achieving long-term stability in distributed multi-agent systems requires overcoming three fundamental challenges. First, agents often operate under *limited perception and semantic understanding*. These raw observation signals often lack the structured, knowledge-level representations necessary for consistent coordination. Agents may interpret similar situations differently, leading to behavioral inconsistencies. Second, agents face *inflexible decision-making*. In dynamic environments, agents must be capable of rapidly adjusting their behavioral modes in response to shifting conditions, yet a lack of contextual understanding and strategy-level flexibility makes such adaptation difficult. Finally, there is the issue of *inaccurate response*. Inconsistency in interpretation across agents may lead to coordination mismatches, oscillations, or cumulative errors that undermine system-wide stability. These challenges compound over time in long-horizon tasks, making it difficult to sustain

*These authors contributed equally.
†Corresponding authors.

robust collaboration. Addressing them requires new mechanisms that enable agents to share semantically meaningful information, reach consensus dynamically, and adjust behavior proactively in response to emerging conditions.

The inspiration comes from the operation of human teams in dynamic environments. Humans actively interpret environmental cues, share information, anticipate contextual shifts, and adjust behaviors through collaborative reasoning. The recent advances in Large Language Models(LLMs) offer a promising opportunity to bridge the semantic and decision-making gaps in such systems. LLMs possess strong generalization capabilities, allowing agents to generate high-level strategic insights from raw contextual information and to align decisions based on shared semantic abstractions.

We propose LLMASC, a structured framework designed to enhance the stability and interpretability of multi-agent coordination through knowledge-augmented consensus. The architecture of LLMASC consists of three interconnected modules, each addressing a specific challenge identified earlier. First, to mitigate observation heterogeneity, the *Semantic Perception Encoder* module converts each agent's localized, modality-specific observations into a structured natural language representation. This process standardizes how agents perceive and interpret their surroundings, serving as the semantic input for further decision-making. Second, to address the problem of inconsistent or conflicting strategies, the *LLM-Guided Consensus Decision* module enables agents to individually infer appropriate operational modes based on semantic representations and then participate in a voting-based consensus process to agree on a joint plan. Finally, to ensure that these high-level strategies can be enacted reliably, the *Policy Execution Controller* module translates the shared plan into executable low-level actions using reinforcement learning. By integrating symbolic reasoning, semantic abstraction, and data-driven control, LLMASC provides a novel framework toward stable and decentralized multi-agent collaboration.

We evaluate LLMASC across three diverse simulation environments: (1) *Multi-Walker* (Gupta, Egorov, and Kochenderfer 2017), where agents physically coordinate to transport a shared payload over uneven terrain; (2) *Simulation of Urban Mobility (SUMO)* (Krajzewicz et al. 2012; Alegre 2019; Ault and Sharon 2021), where decentralized agents manage city-wide intersections to avoid congestion; and (3) *Distributed Power Grid Stabilization* (Wang et al. 2021), where energy nodes collaboratively regulate voltages under fluctuating conditions. These domains span both embodied and cyber-physical systems, enabling a comprehensive assessment of long-term stability. Experiments show that LLMASC consistently outperforms the best baselines, improving stability rates by up to 44% and long-term success rates rates by 31%. Further analysis confirms that LLMASC's consensus mechanism operates efficiently within system time constraints. LLMASC also generalizes well to varying agent population sizes and underlying model choices, demonstrating robustness and adaptability across tasks.

The contributions of this paper are as follows:

- We formulate the long-term stability problem in distributed multi-agent systems (dynamic, partially observable environments), expanding the scope of traditional MARL research.
- We propose LLMASC, a structured framework that enhances multi-agent coordination through semantic perception, consensus-driven decision-making, and robust policy execution. Our implementation is open-sourced to facilitate reproducibility and extension.
- We evaluate LLMASC across three diverse simulation environments involving physical and cyber-physical control tasks. Results show that LLMASC achieves significant improvements in long-term stability and coordination performance, validating the LLMASC's generality and effectiveness.

Related Work

Stability in Multi-Agent Systems. Stability has long been a core concern in distributed control and cooperative robotics (Zhang, Dong, and Pan 2020; Wang et al. 2022; Li et al. 2025), often formulated as system convergence, consensus, or regulation. Classical approaches design control laws under known system dynamics to satisfy Lyapunov or frequency-domain stability conditions. However, such methods assume strong structural priors, limiting their applicability to complex, black-box, or data-driven multi-agent environments. In the multi-agent reinforcement learning (MARL) community, most efforts center around optimizing task-level objectives such as win rate (Samvelyan et al. 2019; Kurach et al. 2020), episodic return (Peng et al. 2021; Liu et al. 2021b; Towers et al. 2024), or exploration coverage (Christianos, Schäfer, and Albrecht 2020; Zheng et al. 2021; Hao et al. 2023), without directly modeling long-term dynamic stability. Our work differs by treating steady-state regulation as an explicit optimization goal. We propose a generalizable, variance-based formulation that integrates smoothly into standard reinforcement learning pipelines. We address scenarios where agents must adaptively switch policies in response to subtle changes in environmental conditions while preserving global consistency.

Language Models for Agent Coordination and Reasoning. LLMs have recently shown promise in enhancing multi-agent systems by enabling communication, strategy generation, and common-sense reasoning (Agashe et al. 2023; Li et al. 2024; Guo et al. 2024). Several works have explored using LLMs to generate executable plans (Kannan, Venkatesh, and Min 2024; Zhai et al. 2025) or suggest cooperative behavior in simulated environments (Bo et al. 2024; Yang et al. 2025). However, these approaches often assume a centralized planner or rely on prompt engineering without integrating structured perception or policy execution. In the multi-agent setting, recent studies have proposed frameworks where LLMs simulate autonomous agents engaging in natural language dialogue. These works emphasize human-like communication and deliberation but are not grounded in real-time physical environments with action and reward feedback. Our work differs in three key aspects: (1) we target decentralized physical control tasks rather than pure dialogue; (2) we introduce a structured perception layer that translates observations into semantic

prompts; and (3) we develop a lightweight language-guided consensus mechanism that integrates naturally with the Reinforcement Learning pipelines.

Problem Formulation

We consider a general setting where a group of N autonomous agents interact in a dynamic, partially observable environment to achieve a shared task. The environment is modeled as a *Partially Observable Multi-Agent Markov Decision Process* (PO-MAMDP), defined by the tuple $\langle \mathcal{S}, \{\mathcal{A}^i\}_{i=1}^N, \mathcal{T}, \{\mathcal{O}^i\}_{i=1}^N, \mathcal{R}, \gamma \rangle$. Here, $\mathcal{S}$ is the global state space, and $\mathcal{A}^i$ and $\mathcal{O}^i$ denote the action and observation spaces of agent i, respectively. The transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A}^1 \times \cdots \times \mathcal{A}^N \to \mathcal{P}(\mathcal{S})$ defines the probability distribution over next states. Each agent receives a local observation $o_t^i \in \mathcal{O}^i$ based on the observation function $\mathcal{O} : \mathcal{S} \to \mathcal{O}^1 \times \cdots \times \mathcal{O}^N$. The reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A}^1 \times \cdots \times \mathcal{A}^N \to \mathbb{R}$ provides a shared reward signal. The objective is to learn a joint policy $\pi = (\pi^1, \ldots, \pi^N)$ that maximizes the expected cumulative return with discount factor $\gamma \in (0, 1)$.

Unlike conventional MARL tasks that focus primarily on short-term reward maximization or adversarial competition, our objective is to achieve *stable long-horizon collaboration* under non-stationary and partially observable conditions. Stability, in this context, refers to the system's ability to maintain coherent and coordinated behavior over extended time horizons, despite environmental changes, heterogeneous observations, and asynchronous agent decisions.

To capture this notion, we define a stability-oriented utility function:

$$\mathcal{J}_{\text{stab}}(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \mathcal{L}_{\text{stab}}(s_t, a_t^1, \ldots, a_t^N) \right], \quad (1)$$

where $\mathcal{L}_{\text{stab}}$ is a task-specific stability evaluator that quantifies deviations from coordinated behavior, policy switching smoothness, or inter-agent consistency.

Approach

To achieve long-horizon stability in distributed multi-agent collaboration, we propose a novel framework named LLMASC, which integrates semantic perception, consensus-driven coordination, and reinforcement-based execution into a unified decision-making process. The core idea is to bridge the semantic gap between low-level, heterogeneous agent observations and high-level collaborative planning by introducing an intermediate LLM-based consensus layer. As shown in Figure 1, the LLMASC is composed of three interdependent components, each designed to address a key challenge of distributed collaboration: (1) *Semantic Perception Encoder*, which transforms raw agent observations into structured natural language descriptions that capture contextual semantics and temporal patterns; (2) *LLM-Guided Consensus Decision*, which enables agents to generate strategy proposals using LLMs and aggregate them into a globally consistent collaborative plan; and (3) *Policy Execution Controller*, which translates the consensus plan into executable low-level actions by optimizing for both task performance and system stability.

Semantic Perception Encoder

The Semantic Perception Encoder is responsible for transforming low-level, modality-specific agent observations into structured LLM-based representations that capture salient semantic features of the local environment. This component addresses the inherent heterogeneity of partial observations in distributed systems and serves as the foundation for LLM-guided coordination in LLMASC.

Observation Abstraction via Feature Projection. At each timestep t, agent i receives a raw observation vector $o_t^i \in \mathcal{O}^i$, which may contain sensory readings, agent states, or environmental signals. To construct a semantically meaningful representation, we define a hierarchical abstraction function:

$$x_t^i = f(o_t^i, \mathcal{H}_t^i) = \text{GenStruct}(\phi(o_t^i), \psi(\mathcal{H}_t^i)) \quad (2)$$

where $\mathcal{H}_t^i = \{o_{t-K}^i, \ldots, o_{t-1}^i\}$ is a temporal context window of size K, ϕ is a feature extractor applied to the current observation, and ψ is a temporal summarization operator. The function $\text{GenStruct}(\cdot)$ denotes a structured representation generator that outputs a sentence-level semantic description.

Temporal Contextualization of Local Observations. We design $\phi(o_t^i)$ to project the raw observation into a latent representation space:

$$\phi(o_t^i) = W_o o_t^i + b_o \quad (3)$$

where W_o and b_o are learnable parameters. Similarly, the temporal encoding $\psi(\mathcal{H}_t^i)$ is defined as:

$$\psi(\mathcal{H}_t^i) = \sum_{k=1}^{K} \alpha_k \cdot \phi(o_{t-k}^i), \quad \alpha_k = \frac{\exp(-\lambda k)}{\sum_{j=1}^{K} \exp(-\lambda j)} \quad (4)$$

which captures historical context with exponential decay, controlled by the hyperparameter $\lambda > 0$.

Structured Semantic Prompt Construction. The final semantic input to the generation function is constructed as the concatenation of current and contextual embeddings:

$$h_t^i = [\phi(o_t^i) \,\|\, \psi(\mathcal{H}_t^i)] \quad (5)$$

where $[\cdot \,\|\, \cdot]$ denotes vector concatenation. The structured representation x_t^i is then generated as:

$$x_t^i = \text{Decoder}(h_t^i; \Theta) \quad (6)$$

where $\text{Decoder}(\cdot)$ is a symbolic rule-based template engine with parameters Θ. In practice, we implement this component using prompt-conditioned template filling for controlled interpretability, while allowing extensibility to generative neural models.

The output x_t^i resides in a shared semantic space $\mathcal{X}$, which serves as the communication and reasoning substrate for downstream consensus generation. This transformation ensures that heterogeneous raw observations are projected into a unified, interpretable representation amenable to large language model processing. By encoding not only spatial features but also temporal patterns, this module enables each agent to reason about local dynamics and system evolution in a semantically aligned manner.

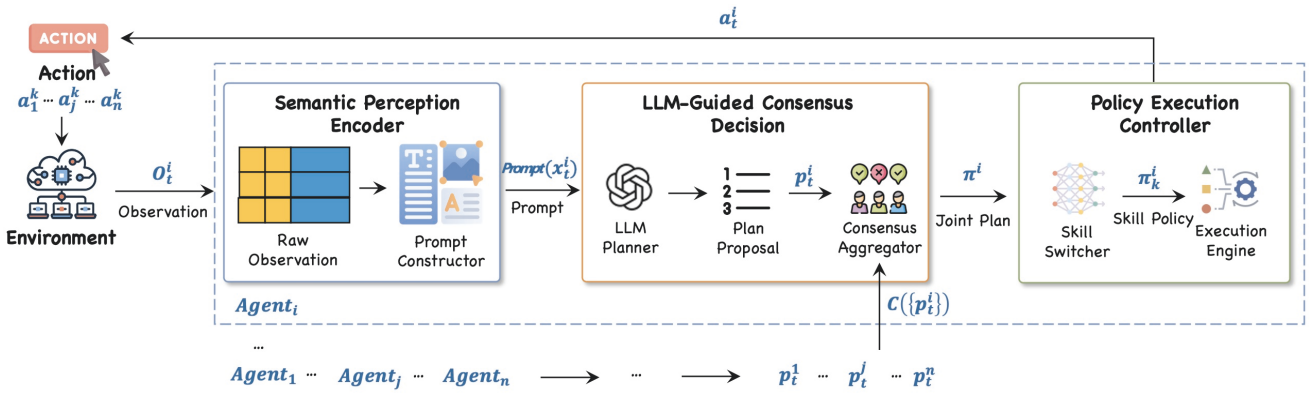


Figure 1: Overview of LLMASC

LLM-Guided Consensus Decision

In distributed systems with partial observability and non-stationary dynamics, individual agents may form divergent interpretations of the environment, leading to inconsistent or suboptimal coordination. To overcome this, we introduce the *LLM-Guided Consensus Decision* module, which enables each agent to independently reason over its semantic observation using a dedicated large language model (LLM), and to reach a globally consistent high-level plan through a structured consensus process. This module is the key innovation in LLMASC, enabling decentralized agents to align their strategies via interpretable, semantic-level communication rather than implicit policy synchronization.

Prompt-Based Strategic Reasoning with LLM. Each agent i is equipped with an LLM instance, denoted $\mathcal{L}_\theta^i$, parameterized by θ, which serves as a local strategic reasoner. At time t, the LLM takes as input the agent's semantic state $x_t^i \in \mathcal{X}$ and a domain-specific prompt template $\mathcal{P}_{\text{task}}$, yielding a policy proposal p_t^i:

$$p_t^i = \mathcal{L}_\theta^i \left(\text{Prompt}(x_t^i; \mathcal{P}_{\text{task}})\right) \tag{7}$$

where $\text{Prompt}(\cdot)$ is a deterministic formatting function that concatenates task, constraints, and the agent's local state description. The output p_t^i belongs to a structured strategy space $\mathcal{P}$:

$$\mathcal{P} = \{\text{maintain}, \text{adjust}, \text{switch\_to\_mode\_}k, \ldots\}$$

To enhance consistency and reasoning reliability, we design prompts to follow a structured, chain-of-thought style format: $Prompt(x_t^i; \mathcal{P}_{task})$ = *Instruction* + *"Given the following observation:"* + x_t^i + *"What is the appropriate strategy? Explain your reasoning first, then output the plan."*

The output of $\mathcal{L}_\theta^i$ is parsed to extract both the final proposal p_t^i and an intermediate reasoning trace r_t^i for interpretability and optional verification.

Consensus Aggregation Through Voting Mechanisms. At each timestep, the system collects all N agent proposals:

$$\mathcal{P}_t = \{(p_t^1, r_t^1), (p_t^2, r_t^2), \ldots, (p_t^N, r_t^N)\} \tag{8}$$

where each element consists of a strategy token p_t^i and its supporting explanation r_t^i. These proposals can be interpreted as individual agent intents.

To aggregate proposals into a unified collaborative plan, we define a consensus function $\mathcal{C} : \mathcal{P}^N \to \mathcal{P}$ that computes a representative consensus strategy Π_t^{cons}:

$$\Pi_t^{\text{cons},i} = \mathcal{C}^i(p_t^1, \ldots, p_t^N) \tag{9}$$

A deterministic selection of the majority proposal:

$$\Pi_t^{\text{cons},i} = \arg\max_{p \in \mathcal{P}} \sum_{j=1}^{N} \mathbb{I}[p_t^j = p] \tag{10}$$

with tie-breaking by priority rules or predefined preferences.

When LLMs produce confidence scores c_t^j along with proposals, we can define a soft aggregation:

$$\Pi_t^{\text{cons},i} = \sum_{j=1}^{N} \alpha_t^j p_t^j, \quad \alpha_t^j = \frac{c_t^j}{\sum_{k=1}^{N} c_t^k} \tag{11}$$

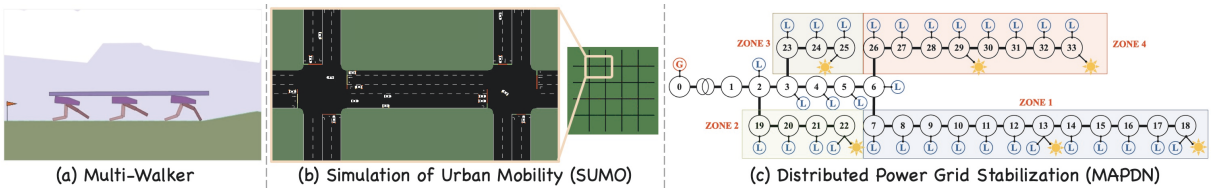
This representation is suitable when $\mathcal{P}$ is embedded in a continuous latent space or parameterized via vectors. In both cases, the consensus function ensures that the system transitions from a collection of local intentions to a globally aligned plan $\Pi_t^{\text{cons},i}$, which will be subsequently executed.

To reinforce alignment between local reasoning and the final plan, we introduce a regularization term during execution that penalizes divergence between the agent's proposal and the adopted consensus:

$$\mathcal{L}_{\text{cons}}^i = \text{KL}(p_t^i \parallel \Pi_t^{\text{cons},i}) \tag{12}$$

where the Kullback-Leibler divergence is evaluated over a categorical distribution derived from proposal embeddings. This term encourages agents to produce proposals that are not only locally optimal but also globally coherent.

Notably, the consensus module avoids direct parameter sharing or gradient synchronization among agents. Instead, it operates at the symbolic level via LLM-generated proposals. This design enables plug-and-play adaptation of agents, modular training of policy components, and scalability to large agent populations, provided that communication latency and reasoning delays are bounded. In implementation, proposals and reasoning traces can be compressed or pruned based on token budgets or criticality assessments.



(a) Multi-Walker (b) Simulation of Urban Mobility (SUMO) (c) Distributed Power Grid Stabilization (MAPDN)

Figure 2: Example of 3 different simulation environments

Policy Execution Controller

Given the global collaborative plan Π_t^{cons} obtained from the LLM-Guided Consensus Decision module, the role of the Policy Execution Controller is to generate low-level executable actions for each agent that are both behaviorally consistent with the consensus and robust to local variations in the environment. This module provides a mechanism to decouple high-level reasoning from dynamic policy execution, allowing agents to adaptively specialize their actions while preserving system-wide coherence.

Policy Switching Mechanism. Each agent maintains a parameterized policy $\pi^i : \mathcal{O}^i \times \mathcal{P} \to \mathcal{A}^i$, which conditions on both the local observation o_t^i and the consensus plan Π_t^{cons} to produce an action distribution:

$$a_t^i \sim \pi^i(a \mid o_t^i, \Pi_t^{\text{cons}}) \tag{13}$$

This structure enables agents to specialize their behavior based on their own sensory input, while aligning with the collective strategic intent captured in Π_t^{cons}.

To enable dynamic adaptation under diverse environmental conditions, we assume that each agent is equipped with a library of K operational sub-policies:

$$\{\pi_k^i\}_{k=1}^K, \quad \text{where } \pi_k^i : \mathcal{O}^i \to \mathcal{A}^i \tag{14}$$

A policy selector function g^i is trained to select the most suitable sub-policy based on the current context:

$$k_t^i = g^i(o_t^i, \Pi_t^{\text{cons}}) = \arg\max_k Q_k^i(o_t^i, \Pi_t^{\text{cons}}) \tag{15}$$

where Q_k^i is a context-conditioned value function that estimates the long-term utility of executing sub-policy π_k^i. The selected action is then from the corresponding sub-policy:

$$a_t^i \sim \pi_{k_t^i}^i(a \mid o_t^i) \tag{16}$$

This mechanism supports robust mode-switching and dynamic reconfiguration under non-stationary dynamics.

Stability-Oriented Reinforcement Learning Objective. The policy is trained to optimize a composite objective that combines task-specific reward $\mathcal{R}$ and a stability-oriented auxiliary signal $\mathcal{L}_{\text{stab}}$, introduced in Section 2:

$$\mathcal{J}_{\text{stab}} = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \left(\mathcal{R}(s_t, \vec{a}_t) - \beta \cdot \mathcal{L}_{\text{stab}}(s_t, \vec{a}_t) \right) \right] \tag{17}$$

where $\vec{a}_t = (a_t^1, \ldots, a_t^N)$ denotes the joint action vector and β is a tunable coefficient that balances task success with behavioral stability.

To provide supervision at the level of consensus plan execution, we introduce a centralized critic V_ω that estimates the expected long-term return given the current state and the consensus plan:

$$V_\omega(s_t, \Pi_t^c) \approx \mathbb{E} \left[\sum_{\tau=t}^{\infty} \gamma^{\tau - t} \left(\mathcal{R}(s_\tau, \vec{a}_\tau) - \beta \cdot \mathcal{L}_s(s_\tau, \vec{a}_\tau) \right) \right] \tag{18}$$

This critic serves two purposes: (1) to guide policy updates via actor-critic training; (2) to enable long-term validation of the consensus plan's effectiveness, which may feed back into prompt refinement or plan re-evaluation.

The policy parameters θ^i, sub-policy ensemble $\{\pi_k^i\}$, selector g^i, and centralized critic V_ω are jointly trained using an actor-critic optimization algorithm. The gradient of the stability-augmented return is computed as:

$$\nabla_{\theta^i} \mathcal{J}_{\text{stab}} = \mathbb{E}_\pi \left[\nabla_{\theta^i} \log \pi^i(a_t^i \mid o_t^i, \Pi_t^{\text{cons}}) \cdot \hat{A}_t \right] \tag{19}$$

where $\hat{A}_t$ is the advantage estimate based on the critic and incorporates both reward and stability feedback.

Experiments

- **RQ1: Effectiveness.** Does LLMASC improve long-term success rate and system stability compared to baselines?
- **RQ2: Decision Efficiency.** How efficient is the LLM-Guided Consensus Decision module in responding to disturbances and adapting decisions across agents?
- **RQ3: Module Ablation.** What is the impact of each key component in LLMASC?
- **RQ4: Environmental Robustness.** How robust is our LLMASC to changes in system configuration?

Experimental Setup

Environments. We evaluate LLMASC across three different simulation environments, which are implemented using HARL (Zhong et al. 2024) and PettingZoo (Terry et al. 2021), which provide the APIs for reproducibility. **(1) Multi-Walker.** (Gupta, Egorov, and Kochenderfer 2017) Multiple agents carry a shared payload across uneven terrain, as shown in Figure 2(a). The stability is measured by maintaining the payload angle and the uninterrupted walk. **(2) Simulation of Urban MObility (SUMO).** (Krajzewicz et al. 2012; Alegre 2019; Ault and Sharon 2021) A large-scale city traffic simulation in Figure 2(b), each agent manages a traffic light. Purpose is to reduce long-term gridlock. **(3) Distributed Power Grid Stabilization.** (Wang et al.

	MultiWalker		SUMO		PowerGrid	
Algorithm	Stability Rate	Long-term SR	Stability Rate	Long-term SR	Stability Rate	Long-term SR
MAPPO	54%	61%	83%	89%	64%	66%
HAPPO	23%	25%	78%	88%	62%	65%
HASAC	37%	43%	68%	75%	58%	64%
Heuristic	56%	60%	82%	87%	56%	58%
LLMASC	**78%**(+44%)	**80%**(+31%)	**93%**(+12%)	**98%**(+10%)	**75%**(+17%)	**78%**(+18%)

Table 1: Comprehensive performance comparison with metric breakdown (RQ1)

2021) A decentralized voltage control task where agents adjust reactive power generation under time-varying weather and load conditions to stabilize bus voltages within target thresholds, as shown in Figure 2(c). More details regarding the environment configurations and hyperparameter settings are provided in the Appendix.

Baselines. We compare LLMASC with 3 state-of-the-art RL approaches, **MAPPO** (Yu et al. 2022), **HAPPO** (Zhong et al. 2024) and **HASAC** (Liu et al. 2024). We also compare LLMASC with **Heuristic Rules**, which are manually crafted policies based on domain knowledge and rule-based action switching. We trained these algorithms by sweeping more than 80 sets of training hyper-parameters each and selected the checkpoints with the highest rewards for the following evaluations. Details including computing infrastructure and tuned hyper-parameters can be found in the Appendix file.

Metrics. We select 2 metrics to assess both effectiveness and coordination efficiency. **Long-term Success Rate ($\mathcal{LTSR}$)**: The percentage of episodes where agents complete the task without triggering failure or violating system constraints. **Stability Rate ($\mathcal{SR}$)**: The proportion of episodes where all critical stability indicators remain within safe bounds (e.g., payload tilt $< 5^{\circ}$, cars' waiting time $< 2000s$, voltage safe region $0.95\ p.u. \leq v \leq 1.05\ p.u.$, where $p.u.$ is a unit to measure voltage). For each evaluation setting, we conducted 100 independent test episodes with randomized initial states and environmental perturbations to ensure statistical robustness. Each baseline was run three times for each evaluation setting, and we averaged these metrics to mitigate potential bias.

Results and Analysis

RQ1: Long-Term Stability

We first evaluate whether LLMASC improves long-term multi-agent coordination stability. Table 1 shows that compared to the best baseline (MAPPO), LLMASC improves stability rate by 44% and Long-term success rate by 31% in the MultiWalker environment. The largest improvements are observed in the Multi-Walker task, where traditional methods often fail after 300–500 steps due to payload imbalance or agent falls. In contrast, LLMASC maintains synchronized behavior and precise pose control across all agents for over 1000 steps in test runs.

	MultiWalker	SUMO	PowerGrid
Per-Step	0.1s	180s	5s
LLM (Ratio)	1.65s	1.62s	1.66s

Table 2: Excution Time Efficiency (RQ2)

We also observe that similar gains are seen in SUMO and PowerGrid. In SUMO, LLMASC reduces gridlocks by adjusting control modes based on emerging traffic patterns, improving the stability rate by 12% and long-term success rate by 10%. In PowerGrid, agents coordinate voltage adjustments more coherently, resisting fluctuations caused by sudden weather changes or load disturbances. These improvements demonstrate that semantic reasoning and consensus aggregation mechanisms help prevent cascading errors and sustain stable collaboration over long durations. We also add the case analysis of LLMASC in the Appendix file.

RQ2: Decision Efficiency

We next examine whether the use of LLM-based consensus introduces latency that may hinder real-time decision-making. Table 2 reports the average environment step time alongside the average LLM response time when using GPT-4o. Across all environments, the LLM response time remains around 1.6 seconds. In SUMO and PowerGrid, where each step requires approximately 180 and 5 seconds, respectively, the additional 1.6 seconds of LLM reasoning are negligible relative to the intrinsic simulation dynamics. More importantly, the consensus output arrives well before the next control cycle, ensuring no disruption to system performance.

The MultiWalker environment presents a different case, since each simulation step is only 0.1 seconds. However, every agent is equipped with a radar sensor that can detect terrain variations up to 25 steps in advance. This predictive horizon provides a buffer of 2.5 seconds, which is sufficient to accommodate the 1.6 seconds LLM decision process. Therefore, consensus decisions are delivered in time for agents to adapt their modes before destabilizing events occur. Results show that integration of LLM-based reasoning into control pipeline doesn't bottleneck real-time coordination.

Variants	MultiWalker SR	MultiWalker LTSR	SUMO SR	SUMO LTSR	PowerGrid SR	PowerGrid LTSR
I. Semantic Perception Encoder						
w/o Semantic	36%	54%	74%	76%	44%	52%
II. LLM-Guided C.D.:Variant Base LLM Models						
Gemini-2.5pro	72%	74%	87%	97%	66%	76%
Claude-3.7	72%	74%	89%	97%	67%	78%
Claude-3.5	74%	76%	89%	97%	67%	78%
DeepSeekV3	70%	72%	87%	96%	62%	76%
GPT-4o-mini	66%	70%	90%	98%	64%	76%
Llama-3.2-3B	68%	72%	90%	97%	63%	78%
Qwen3-0.6B	70%	74%	87%	97%	62%	72%
Qwen3-1.7B	70%	74%	90%	98%	64%	76%
III. Policy Exec. Controller:Variant Base RL Models						
(HAPPO)	60%	64%	79%	84%	71%	76%
(HASAC)	63%	64%	80%	85%	72%	74%
LLMASC	**78%**	**80%**	**93%**	**98%**	**75%**	**78%**

Table 3: Ablations on the variation of module (RQ3)

RQ3: Module Ablation

To understand the contribution of each module in LLMASC, we conduct ablation studies as shown in Table 3.

(1) *Semantic Perception Encoder*. Removing the semantic encoder and directly feeding raw observation values to the LLM leads to the most severe performance drop. Stability rate decreases by 36%–74%, and long-horizon success rate drops by 52%–76% across environments. Without structured abstraction, the LLM struggles to form coherent interpretations of agent contexts, resulting in erratic decisions and misaligned coordination. This confirms that semantic grounding is crucial for effective language-based reasoning in multi-agent settings.

(2) *LLM-Guided Consensus Decision*. We replace the GPT-4o with several alternative LLMs, including both proprietary (e.g., Claude-3.7, Gemini-2.5 Pro, DeepSeek V3) and open-source models (e.g., LLaMA 3B, Qwen3). Larger models maintain comparable performance (within 0%–8% gap), while smaller models (e.g., LLaMA 1B, Qwen3-0.6B) experience a moderate degradation of 6%–10% in long-horizon metrics. This shows that smaller models may serve as viable alternatives in latency-sensitive or on-device deployment scenarios.

(3) *Policy Execution Controller*. We substitute MAPPO with HAPPO and HASAC as the base reinforcement learner. While the long-horizon success rate drops by 2%–16% depending on the task, overall performance remains stable. The symbolic coordination layer decouples policy execution from semantic reasoning, enabling flexible integration with existing MARL algorithms.

Overall, the ablation results confirm that all three modules are essential to the performance of LLMASC. Among them, semantic abstraction has the most pronounced effect, while the LLM and RL backbones offer deployment flexibility without significantly compromising effectiveness.

Variants	MultiWalker n=2	**n=3**	n=5	n=7	n=8	n=9
SR	90%	78%	76%	75%	74%	72%
LTSR	94%	80%	80%	77%	76%	74%

Table 4: Robustness on the variation of environments (RQ4)

RQ4: Environmental Robustness

To evaluate the robustness of LLMASC under varying agent population sizes, we conduct experiments in the MultiWalker environment and vary the number of agents from the default of 3 to 2, 5, 7, 8 and 9. Table 4 summarizes the results. Across all settings, LLMASC maintains high stability and long-term success, with only marginal variation in performance. Notably, even in the 9-agent configuration, where coordination becomes significantly more challenging due to increased interaction complexity, LLMASC achieves a stability rate of 72%.

The best results are observed in the 2-agent case, where the system reaches a stability rate of 90% and a long-term success rate of 94%. In the default 3-agent setup, these values are 78% and 80%, respectively. As the number of agents increases to 5–9, the performance remains stable with only moderate drops: the stability rate decreases gradually from 76% to 72%, and the long-term success rate from 80% to 74%. Results show that LLMASC generalizes well to teams of varying sizes, without requiring any reconfiguration.

Due to space limitations, we omit detailed results for SUMO and PowerGrid. However, we observe similar trends: LLMASC exhibits stable performance across varying numbers of traffic lights or power nodes, consistently outperforming baselines. These findings confirm that LLMASC's architecture generalizes well to larger agent populations and diverse task topologies.

Conclusion

We propose LLMASC, a framework designed to enhance long-term stability in multi-agent collaboration by combining semantic reasoning with decentralized control. Semantic Perception Encoder transforms heterogeneous agent observations into structured natural language. LLM-Guided Consensus Decision enables strategic alignment through proposal exchange and voting. Policy Execution Controller maps high-level plans to executable actions via reinforcement learning. We evaluate LLMASC across three representative simulation domains, spanning both physical and cyber-physical systems. Experiments show that LLMASC consistently outperforms the best baselines.

In future work, we plan to explore more scalable consensus mechanisms, extend LLMASC to more real-world systems, and investigate how continual language feedback can further enhance adaptability in non-stationary settings.

Acknowledgements

Supported by the Strategic Priority Research Program of Chinese Academy of Sciences, Grant No. XDB0900000, the National Natural Science Foundation of China Grant No. 62402483, 62232016, 62072442, 62402484, Major Program of ISCAS Grant No. ISCAS-ZD-202401 and ISCAS-ZD-202302, Innovation Team 2024 ISCAS (No. 2024-66), Basic Research Program of ISCAS Grant No. ISCAS-JCZD-202304.

References

Agashe, S.; Fan, Y.; Reyna, A.; and Wang, X. E. 2023. Llm-coordination: evaluating and analyzing multi-agent coordination abilities in large language models. *arXiv preprint arXiv:2310.03903*.

Alegre, L. N. 2019. SUMO-RL. https://github.com/LucasAlegre/sumo-rl. Accessed: 2025-07-01.

Antonio, G.-P.; and Maria-Dolores, C. 2022. Multi-agent deep reinforcement learning to manage connected autonomous vehicles at tomorrow's intersections. *IEEE Transactions on Vehicular Technology*, 71(7): 7033–7043.

Aotani, T.; Kobayashi, T.; and Sugimoto, K. 2021. Bottom-up multi-agent reinforcement learning by reward shaping for cooperative-competitive tasks. *Applied Intelligence*, 51(7): 4434–4452.

Ault, J.; and Sharon, G. 2021. Reinforcement Learning Benchmarks for Traffic Signal Control. In *Proceedings of the Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS 2021) Datasets and Benchmarks Track*.

Berducci, L.; Yang, S.; Mangharam, R.; and Grosu, R. 2024. Learning adaptive safety for multi-agent systems. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2859–2865. IEEE.

Bo, X.; Zhang, Z.; Dai, Q.; Feng, X.; Wang, L.; Li, R.; Chen, X.; and Wen, J.-R. 2024. Reflective multi-agent collaboration based on large language models. *Advances in Neural Information Processing Systems*, 37: 138595–138631.

Carroll, M.; Shah, R.; Ho, M. K.; Griffiths, T.; Seshia, S.; Abbeel, P.; and Dragan, A. 2019. On the utility of learning about humans for human-ai coordination. *Advances in neural information processing systems*, 32.

Chen, P.; Liu, S.; Wang, X.; and Kamwa, I. 2023. Physics-guided multi-agent deep reinforcement learning for robust active voltage control in electrical distribution systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 71(2): 922–933.

Christianos, F.; Schäfer, L.; and Albrecht, S. 2020. Shared experience actor-critic for multi-agent reinforcement learning. *Advances in neural information processing systems*, 33: 10707–10717.

Guo, T.; Chen, X.; Wang, Y.; Chang, R.; Pei, S.; Chawla, N. V.; Wiest, O.; and Zhang, X. 2024. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*.

Gupta, J. K.; Egorov, M.; and Kochenderfer, M. 2017. Cooperative multi-agent control using deep reinforcement learning. In *International conference on autonomous agents and multiagent systems*, 66–83. Springer.

Hao, J.; Yang, T.; Tang, H.; Bai, C.; Liu, J.; Meng, Z.; Liu, P.; and Wang, Z. 2023. Exploration in deep reinforcement learning: From single-agent to multiagent domain. *IEEE Transactions on Neural Networks and Learning Systems*, 35(7): 8762–8782.

Kannan, S. S.; Venkatesh, V. L.; and Min, B.-C. 2024. Smart-llm: Smart multi-agent robot task planning using large language models. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 12140–12147. IEEE.

Krajzewicz, D.; Erdmann, J.; Behrisch, M.; Bieker, L.; et al. 2012. Recent development and applications of SUMO-Simulation of Urban MObility. *International journal on advances in systems and measurements*, 5(3&4): 128–138.

Kurach, K.; Raichuk, A.; Stanczyk, P.; Zajkac, M.; Bachem, O.; Espeholt, L.; Riquelme, C.; Vincent, D.; Michalski, M.; Bousquet, O.; et al. 2020. Google research football: A novel reinforcement learning environment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 4501–4510.

Li, X.; Wang, S.; Zeng, S.; Wu, Y.; and Yang, Y. 2024. A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*, 1(1): 9.

Li, Z.; Peng, X. B.; Abbeel, P.; Levine, S.; Berseth, G.; and Sreenath, K. 2025. Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control. *The International Journal of Robotics Research*, 44(5): 840–888.

Liu, B.; Liu, Q.; Stone, P.; Garg, A.; Zhu, Y.; and Anandkumar, A. 2021a. Coach-player multi-agent reinforcement learning for dynamic team composition. In *International Conference on Machine Learning*, 6860–6870. PMLR.

Liu, I.-J.; Jain, U.; Yeh, R. A.; and Schwing, A. 2021b. Cooperative exploration for multi-agent deep reinforcement learning. In *International conference on machine learning*, 6826–6836. PMLR.

Liu, J.; Zhong, Y.; Hu, S.; Fu, H.; Fu, Q.; Chang, X.; and Yang, Y. 2024. Maximum Entropy Heterogeneous-Agent Reinforcement Learning. In *The Twelfth International Conference on Learning Representations*. 595–596.

Lu, Y.; et al. 2023. Deep Reinforcement Learning Based Optimal Scheduling of Active Distribution System Considering Distributed Generation, Energy Storage and Flexible Load. *Energy*, 271: n. pag. Web.

Ma, Y. J.; Liang, W.; Wang, G.; Huang, D.-A.; Bastani, O.; Jayaraman, D.; Zhu, Y.; Fan, L.; and Anandkumar, A. 2023. Eureka: Human-level reward design via coding large language models. *arXiv preprint arXiv:2310.12931*.

Mukhtar, H.; Afzal, A.; Alahmari, S.; and Yonbawi, S. 2023. CCGN: Centralized Collaborative Graphical Transformer Multi-Agent Reinforcement Learning for Multi-Intersection Signal Free-Corridor. *Neural Networks*, 166: 396–409. Epub 2023 Jul 26, PMID: 37549608.

Peng, B.; Rashid, T.; Schroeder de Witt, C.; Kamienny, P.-A.; Torr, P.; Böhmer, W.; and Whiteson, S. 2021. Facmac: Factored multi-agent centralised policy gradients. *Advances in Neural Information Processing Systems*, 34: 12208–12221.

Rashid, T.; Samvelyan, M.; De Witt, C. S.; Farquhar, G.; Foerster, J.; and Whiteson, S. 2020. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178): 1–51.

Samvelyan, M.; Rashid, T.; De Witt, C. S.; Farquhar, G.; Nardelli, N.; Rudner, T. G.; Hung, C.-M.; Torr, P. H.; Foerster, J.; and Whiteson, S. 2019. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*.

Son, K.; Kim, D.; Kang, W. J.; Hostallero, D. E.; and Yi, Y. 2019. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International conference on machine learning*, 5887–5896. PMLR.

Terry, J.; Black, B.; Grammel, N.; Jayakumar, M.; Hari, A.; Sullivan, R.; Santos, L. S.; Dieffendahl, C.; Horsch, C.; Perez-Vicente, R.; et al. 2021. Pettingzoo: Gym for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 15032–15043.

Towers, M.; Kwiatkowski, A.; Terry, J.; Balis, J. U.; De Cola, G.; Deleu, T.; Goulão, M.; Kallinteris, A.; Krimmel, M.; KG, A.; et al. 2024. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*.

Wang, J.; Xu, W.; Gu, Y.; Song, W.; and Green, T. C. 2021. Multi-agent reinforcement learning for active voltage control on power distribution networks. *Advances in neural information processing systems*, 34: 3271–3284.

Wang, T.; Cao, J.; and Hussain, A. 2021. Adaptive Traffic Signal Control for large-scale scenario with Cooperative Group-based Multi-agent reinforcement learning. *Transportation research part C: emerging technologies*, 125: 103046.

Wang, W. Z.; Shih, A.; Xie, A.; and Sadigh, D. 2022. Influencing towards stable multi-agent interactions. In *Conference on robot learning*, 1132–1143. PMLR.

Yan, R.; and Xu, Y. 2024. Multi-Objective and Multi-Agent Deep Reinforcement Learning for Real-Time Decentralized Volt/VAR Control of Distribution Networks Considering PV Inverter Lifetime. *IEEE Transactions on Power Systems*.

Yang, Y.; Chai, H.; Shao, S.; Song, Y.; Qi, S.; Rui, R.; and Zhang, W. 2025. Agentnet: Decentralized evolutionary coordination for llm-based multi-agent systems. *arXiv preprint arXiv:2504.00587*.

Yu, C.; Velu, A.; Vinitsky, E.; Gao, J.; Wang, Y.; Bayen, A.; and Wu, Y. 2022. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in neural information processing systems*, 35: 24611–24624.

Zhai, W.; Liao, J.; Chen, Z.; Su, B.; and Zhao, X. 2025. A Survey of Task Planning with Large Language Models. *Intelligent Computing*, 4: 0124.

Zhang, Q.; Dong, H.; and Pan, W. 2020. Lyapunov-based reinforcement learning for decentralized multi-agent control. In *International Conference on Distributed Artificial Intelligence*, 55–68. Springer.

Zheng, L.; Chen, J.; Wang, J.; He, J.; Hu, Y.; Chen, Y.; Fan, C.; Gao, Y.; and Zhang, C. 2021. Episodic multi-agent reinforcement learning with curiosity-driven exploration. *Advances in Neural Information Processing Systems*, 34: 3757–3769.

Zhong, Y.; Kuba, J. G.; Feng, X.; Hu, S.; Ji, J.; and Yang, Y. 2024. Heterogeneous-agent reinforcement learning. *Journal of Machine Learning Research*, 25(32): 1–67.

Zhou, W.; Chen, D.; Yan, J.; Li, Z.; Yin, H.; and Ge, W. 2022. Multi-agent reinforcement learning for cooperative lane changing of connected and autonomous vehicles in mixed traffic. *Autonomous Intelligent Systems*, 2(1): 5.