CONTROLLING LARGE LANGUAGE MODEL-BASED AGENTS FOR LARGE-SCALE DECISION-MAKING: AN ACTOR-CRITIC APPROACH

Bin Zhang[1,2], **Hangyu Mao**[3,*], **Jingqing Ruan**[1,2], **Ying Wen**[4], **Yang Li**[5], **Shao Zhang**[4], **Zhiwei Xu**[1,2], **Dapeng Li**[1,2], **Ziyue Li**[3], **Rui Zhao**[3], **Guoliang Fan**[1,2], **Lijuan Li**[1,2,*]
[1]Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]SenseTime Research [4]Shanghai Jiao Tong University [5]The University of Manchester

ABSTRACT

The remarkable progress in Large Language Models (LLMs) opens up new avenues for addressing planning and decision-making problems in Multi-Agent Systems (MAS). However, as the number of agents increases, the issues of hallucination in LLMs and coordination in MAS have become increasingly prominent. Additionally, the efficient utilization of tokens emerges as a critical consideration when employing LLMs to facilitate the interactions among a substantial number of agents. In this paper, we develop a modular framework called LLaMAC to mitigate these challenges. LLaMAC implements a value distribution encoding similar to that found in the human brain, utilizing internal and external feedback mechanisms to facilitate collaboration and iterative reasoning among its modules. Through evaluations involving system resource allocation and robot grid transportation, we demonstrate the considerable advantages afforded by our proposed approach.

1 INTRODUCTION

Relying on training from massive datasets to capture extensive common knowledge and having demonstrated certain reasoning capabilities, Large Language Models (LLMs) have been widely applied and explored across various domains, rapidly emerging as powerful tools Brown et al. (2020); Kojima et al. (2022); Ruan et al. (2023a); Yang et al. (2023). The utilization of prompting techniques, such as chain-of-thought (CoT) Wei et al. (2022), has played a pivotal role in further augmenting the reasoning and planning capabilities of LLMs. This approach eliminates the need for training from scratch by providing an acceptable initial strategy based on common knowledge. Examples of such applications include question-answering systems Mallen et al. (2023), commonsense reasoning Hao et al. (2023), programming Tian et al. (2023), and embodied intelligence Driess et al. (2023).

Recently, in the fields of natural language processing (NLP) and multi-agent systems (MAS), numerous research endeavors are dedicated to exploring the collaborative task-solving potential facilitated by the cooperation of multiple agents grounded in LLMs. These efforts leverage the role-playing Li et al. (2023) and debate Chan et al. (2023) to facilitate synergy and effective coordination among the agents involved. However, most existing works focus on coordinating a limited number of agents as shown in Table 1. The application of LLMs for effective coordination in large-scale agent scenarios has received limited attention. This is attributed mainly to the significant increase in complexity and difficulty associated with applying LLMs to large-scale multi-agent decision-making tasks.

In this paper, we direct our attention to the following key challenges: (1) As the number of agents increases, the joint action space grows exponentially, amplifying the difficulty of exploration and exploitation in complex MAS. (2) The limitations of LLMs themselves, as highlighted by the issue of hallucinations Zhang et al. (2023e), can affect the reliability of decision-making. (3) Effectively

*Corresponding author: {hy.mao@pku.edu.cn, maohangyu@sensetime.com}, lijuan.li@ia.ac.cn

Table 1: Comprehensive comparison of LLM-based multi-agent methods. All approaches rely on either multi-agent debate or role-playing to accomplish decision-making tasks and solve NLP problems (task solver), or simulate collective behavior (community simulator).

Type	Method	Target	Agent Configuration	Agents Num.
Muti-Agent Debate	Debate (Du et al.)	Task Solver	2 debaters	2
	MAD (Liang et al.)		1 judge + 2 debaters	3
	ChatEval (Chan et al.)		multi debaters	5
Role Playing	CAMEL (Li et al.)	Task Solver	1 assistant + 1 user	2
	AgentVerse (Chen et al.)		1 role assigner + 2-4 experts + 1 evaluater	6
	Proagent (Zhang et al.)		2 cooks	2
	LLaMAC (ours)		**3 critic + 1-50 actors**	**50**
	Generative Agents (Park et al.)	Community Simulator	25 agents	25
	Werewolf Agents (Xu et al.)		7 players	7
	ReCon (Wang et al.)		6 players	6

managing tokens or communication resources presents a substantial challenge in scenarios involving large-scale LLM-based agents. We prioritize these challenges due to their inherent and widespread nature in large-scale settings, and the absence of comprehensive solutions. By focusing on these general challenges, we try to contribute insights and solutions that hold broad relevance, offering a foundational framework for tackling intricacies in diverse real-world scenarios.

To this end, we present **L**arge **La**nguage **M**odel-based **A**ctor-**C**ritic (**LLaMAC**), a novel framework for achieving a comprehensive decision-making process in collaborative tasks involving large-scale LLM-based agents, drawing inspiration from the classical actor-critic reinforcement learning (RL) approach Konda & Tsitsiklis (1999). Within LLaMAC, we design a centralized critic which takes on the role of a coordinator, making suggestions to each actor based on their decision memory. Subsequently, the actors engage in interactions with the environment, receiving assigned tasks, conducting analyses, and performing corresponding actions. Specifically, our primary contributions are as follows:

- To attain a viable and robust initial strategy and tackle the exploration-exploitation trade-off inherent in the decision-making process, we introduce the TripletCritic structure, which is inspired by the distributional code for value in the brain Dabney et al. (2020). This architecture effectively coordinates multiple critics with shared objectives but varying preferences through internal feedback, thereby providing dependable action suggestion for each actor involved.
- We also establish an external feedback mechanism between the LLM-based actors (i.e., agents) and the TripletCritic. This mechanism serves to not only reduce the access cost of the LLM but also allows each actor to maintain independent exploration and decision-making capabilities.
- We propose a modular and token-efficient framework for augmenting the decision-making capabilities of LLM-based agents in large-scale multi-agent environments. This framework enables autonomous iterative cooperation among a large number of agents.

We first evaluate the performance of our method on a system resource allocation task to demonstrate its ability to strike a balance between exploration and exploitation, as well as its capability in large-scale multi-agent decision-making tasks. We further deploy our method in a more complex robot grid transportation scenario to validate its planning and decision-making capabilities. Experimental results demonstrate that our method outperforms existing approaches in terms of final performance, token utilization efficiency, and policy stability. To the best of our knowledge, we are the first to apply LLMs to large-scale multi-agent decision-making tasks involving more than 50 agents, as indicated in Table 1.

2 Related Work

2.1 Multi-Agent Cooperation

Extensive research has been conducted to explore collaborative control among agents in MAS, with the objective of acquiring optimal strategies to accomplish ultimate goals. Game theory and RL

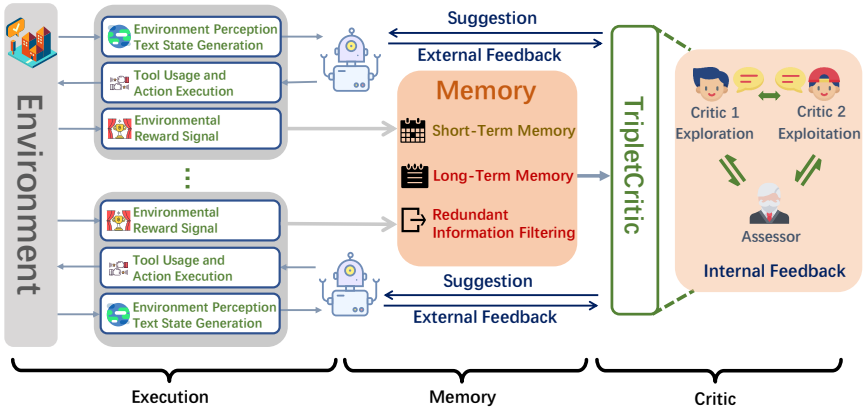

Figure 1: The overall framework of LLaMAC. The LLM-based agents achieve autonomous and continuous decision-making and interaction through the utilization of the execution, memory, and critic modules.

serve as essential theoretical and practical foundations for this research Yang & Wang (2020); Zhang et al. (2021), leading to the development of several novel collaborative training frameworks that effectively address challenges such as equilibrium strategy solving Kuba et al. (2021); Zhang et al. (2023a;b), credit assignment Zhou et al. (2020), non-stationarity of the environment, and partial observability Rashid et al. (2020). Among these approaches, the Actor-Critic method Lowe et al. (2017), widely recognized as one of the classical RL techniques, has found extensive application within the context of MAS. Within this framework, a centralized critic estimates the value function to evaluate the quality of policies, while decentralized actors employ gradient ascent based on these assessments to improve their policies, thereby maximizing the expected cumulative return. However, these methods often suffer from limitations in generalization and require exploration of a large number of irrelevant trajectories, resulting in low training efficiency. Moreover, strategies generated by such black-box optimization methods often lack interpretability. In contrast, our approach enables optimal strategy formulation through a stable and efficient framework based on natural language interaction, providing a transparent and interpretable decision-making process.

2.2 Planning and Reasoning with LLM

Learning in massive corpora gives LLMs certain commonsense reasoning capabilities Kojima et al. (2022). Although there are still challenges in solving complex decision tasks, a large amount of work has proven that their methods can effectively improve the planning ability of LLMs Zelikman et al. (2022); Creswell et al. (2022). One line of research focuses on decomposing complex queries into sequential intermediate steps, known as Chain-of-Thought (CoT) Wei et al. (2022), to achieve accurate solutions. Another direction involves incorporating feedback mechanisms, showcasing their extensive capabilities in tackling complex decision-making challenges Wang et al. (2023b). Moreover, recent studies have begun to address this issue employing multiple LLMs. These approached are enhanced in their planning capabilities through techniques such as debate Chan et al. (2023); Liang et al. (2023) or role-playing Li et al. (2023); Hong et al. (2023). In the domain of decision-making, a subset of research utilizes prompting techniques to construct comprehensive processes covering perception, planning, and action, in cluding video games Zhang et al. (2023c), robot contro l Zhang et al. (2023d); Mandi et al. (2023), and open-world tasks Zhu et al. (2023); Gong et al. (2023). There are some studies about task planning and external tool usage Ruan et al. (2023b); Kong et al. (2023). However, it is worth noting that the existing studies, as outlined in Table 1, have predominantly concentrated on tasks involving a limited number of agents. The involvement of a larger number of agents has primarily been observed in methods that analyze community simulations, wherein task-solving is not a requirement. In light of this observation, our work uniquely emphasizes the application of language models in the realm of decision-making within large-scale multi-agent systems.

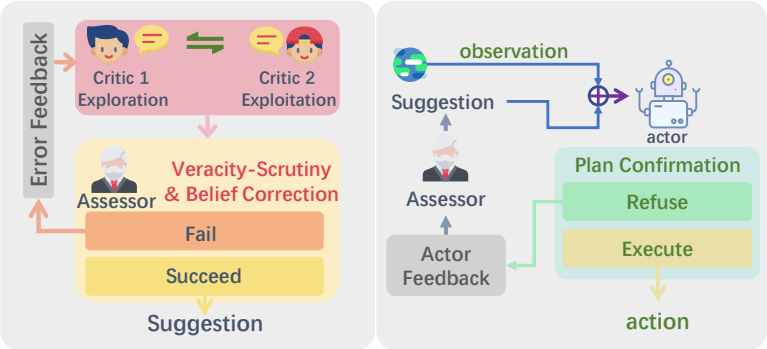

Figure 2: Internal Feedback within the TripletCritic (*Left*) and External Feedback mechanism from actor to critic (*Right*).

3 LLaMAC

In this section, we formally present a systematic and modular framework designed for LLM-based agents, namely Large Language Model-based Actor-Critic (LLaMAC), with a specific emphasis on their suitability for large-scale decision-making contexts.

3.1 Problem Formulation

This study focuses on the collaborative task solving of MAS, which can be formalized as a Goal-Augmented Decentralized Partially Observable Markov Decision Process (GA-Dec-POMDP) Spaan (2012). It is defined by a tuple: $\Gamma \triangleq \langle \mathcal{I}, \mathcal{S}, \mathcal{G}, \{\mathcal{O}^i\}_{i\in\mathcal{I}}, \{\mathcal{A}^i\}_{i\in\mathcal{I}}, \mathcal{P}, R\rangle$, where $\mathcal{I}$, $\mathcal{S}$, $\mathcal{G}$, $\mathcal{O}$, and $\mathcal{A}$ represent the sets of agents, state space, goal space, observation space, and action space, respectively. $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ denotes the dynamic transition function, and $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ represents the reward function. Within a given state $s \in \mathcal{S}$, each agent $i \in \mathcal{I}$ possesses its own local observations $o^i \in \mathcal{O}^i$ within its field of view and performs action $a^i \in \mathcal{A}^i$ accordingly. Formally, this problem requires each agent i to learn a decision policy $\pi^i : \mathcal{O}^i \to \mathcal{A}^i$ to solve the task with a goal, which is equivalent to maximizing cumulative rewards.

3.2 Overall Framework

As illustrated in Figure 1, LLaMAC introduces the Centralized Critic with Decentralized Actor (CCDA) structure, where actors and critics are LLM-based agents. The system incorporates three fundamental modules to facilitate a comprehensive decision-making process, enabling iterative reasoning, planning, and continuous interaction between the agents and the environment. The functionalities of each module are as follows:

Execution Module. The execution module fulfills the vital function of converting the original state information obtained from the environment into text-based descriptions that can be comprehended and processed by the language model. The actions performed by each actor encompass a broad spectrum, ranging from intricately detailed actions like adjusting the joint movement angles of a robot to more abstract and higher-level actions such as issuing instructions for the utilization of a specific tool.

Memory Module. The memory module serves to store crucial information needed during the decision-making process to aid the accumulation of useful knowledge and enhance the agent's decision-making capabilities. Specifically, the short-term memory is used to store the most recent state. In contrast, the historical trajectory and experiential information learned from interactions are stored in the long-term memory. The memory module also incorporates a mechanism for filtering redundant information. During long-term planning processes, it retains only the most recent L steps of state transitions $< s_{t-L+1}, a_{t-L+1}, r_{t-L+1}, s_{t-L+2}, ..., s_t >$. This assists the agent in comprehending the relationship between actions and changes in environmental states.

Critic Module. The critic module assumes a central role within the workflow of LLaMAC. It receives the present *state* and extracts pertinent details from the memory module, enabling evaluation and learning from the actors' historical trajectories. Functioning as a centralized coordinator, the

critic module engages in reasoning and planning activities to formulate potential high-reward and reliable plan suggestions. These suggestions then serve as guides for the interaction between the actor and the environment.

Furthermore, we devise a comprehensive feedback mechanism along with a token-efficient solution to address the challenges posed by the increase in the number of agents, such as exacerbation of hallucinatory phenomena, escalation of the access cost, and the trade-offs involved in exploration and exploitation. By coordinating the functionalities of each module and incorporating the feedback mechanism, we have the coherent decision-making workflow:

(1) The environment produces a new *state*, denoted as s, which is presented in textual format to enable processing by the language model-based agent.

(2) The critic receives the state and extracts the relevant information from the memory module. Utilizing these inputs, it facilitates a three-critic dialogue (*Internal Feedback*) and subsequently generates the textual *suggestion* denoted as su for each actor.

(3) Each actor is provided with the *observation* denoted as o from the environment, as well as the *suggestion* su from the TripletCritic. Subsequently, actors engage in a process called *External Feedback*.

  (3.1) If all actors reach a consensus that the suggestion is correct, each actor generates an action a based on the information $< o, su >$ and executes the action a in the environment. The environment provides a reward r to the agents, indicating the quality of the action. The entire state transition process is stored in the memory module. Subsequently, a new round of interaction commences, signifying a return to step (1).

  (3.2) If an actor identifies that the suggestion is incorrect, an external feedback signal is generated. Subsequently, the TripletCritic receives this external feedback information and formulates a new suggestion for the actor based on the three-critic dialogue history and the recently received feedback information. The TripletCritic then transmits the revised suggestion to the respective actor, and the workflow resumes at step (3).

(4) The task concludes either when the goal is successfully achieved or when the maximum iteration limit is reached, at which point the final task results are returned.

3.3 TripletCritic with Internal Feedback

The increasing number of agents presents formidable challenges to the accuracy and efficiency of task evaluation and planning conducted by the critic module. The expansion of coordinated action spaces and the growing inter-dependencies in decision-making among agents significantly amplify the complexity of decision-making for language models. Moreover, these factors intensify the already challenging issue of hallucinations.

To this end, we develop the TripletCritic, which incorporates an internal feedback mechanism. The design of TripletCritic is inspired by the distributed encoding of reward and value by dopamine neurons in the brain Dabney et al. (2020). Each dopamine neuron contains partial information about the actual reward, and different dopamine neurons utilize different value predictions, enabling the brain to model the value distribution of reward events. Similarly, as depicted in Figure 2, the TripletCritic framework encompasses a dual-critic structure, each with the same objective but distinct preferences, alongside the third critic, called the assessor, who assumes the responsibility of reconciling these preferences. One critic exhibits a proclivity for exploration, prioritizing long-term gains, while the other gravitates towards exploitation, emphasizing short-term gains. The assessor fulfills two primary roles. Firstly, it makes *Veracity Scrutiny* to check the strategies employed by the dual-critic, offering internal feedback in the event of errors. Secondly, it undertakes *Belief Correction* in order to establish a harmonious equilibrium between exploration and exploitation within the planners. Additionally, the assessor collaborates with the actors to transmit the final suggestion assignment, informed by these assessments and corrections.

3.4 External Feedback from Actor to Critic

The TripletCritic provides each actor with a potential initial feasible solution. To facilitate the iterative long-term planning process and achieve the ultimate goal, as well as to reduce the access costs

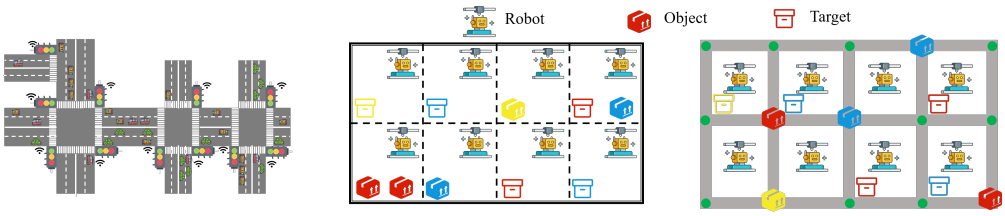

Figure 3: Multi-agent task planning environments. *Left*: System resource allocation, exemplified by addressing traffic congestion. *Middle*: Grid Transportation-Easy. *Right*: Grid Transportation-Hard.

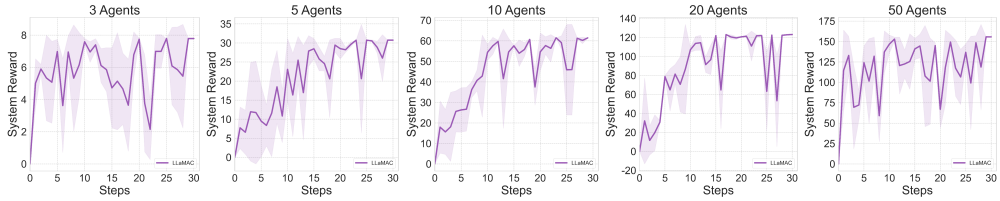

Figure 4: The evaluation performance of LLaMAC in system resource allocation scenarios with different number of agents.

of decision-making for a large number of intelligent agents, we additionally incorporate an external feedback mechanism from actor to critic.

Initially, as depicted in Figure 2, the TripletCritic sends *suggestions* $\{su^i\}_{i\in\mathcal{I}}$ to each actor, and all actors pass the proposed plans through an external *Plan Confirmation* to determine their feasibility. If further improvements are deemed necessary, the corresponding LLM is accessed. The LLM takes as input the agent's *observation* o^i and the corresponding *suggestion* su^i, providing insights into the underlying issues and potential enhancement strategies. Once feedback is received from all actors, the information is aggregated and sent back to the Assessor within the TripletCritic. The Assessor utilizes the internal feedback dialogue information and the actors' external feedback to further update the suggestions for actors with identified issues, returning new *suggestions* to the respective actors. This iterative process continues until all actors determine that no further improvements are necessary, at which point actions are executed directly.

The coordination among various modules is facilitated by both internal and external feedback, thus forming a comprehensive and automated iterative planning process. TripletCritic enhances the viability and robustness of the initial policy by incorporating an internal feedback mechanism and an evaluation mechanism that balances different preferences. Additionally, it effectively reduces the occurrence of hallucination issues. It is important to highlight that the reliability of TripletCritic reduces the actors' opportunity to provide external feedback, thereby minimizing access costs and promoting the development of token-efficient solutions. The occasional external feedback process further improves the performance of the ultimate strategy.

4 Evaluation

In this section, we employ the state-of-the-art large language model, namely GPT-4 OpenAI (2023), to conduct a comprehensive evaluation of the effectiveness of our method within two distinct categories of scenarios, as illustrated in Figure 3. Firstly, we examine system resource allocation scenarios to primarily assess the performance of the TripletCritic. Secondly, we explore robot grid transportation scenarios to showcase the performance of LLaMAC in long-term iterative decision-making throughout the entire process.

4.1 System Resource Allocation

4.1.1 Experimental Settings

System resource allocation HolmesParker et al. (2014) can be viewed as a single-step decision and optimization problem that require mathematical reasoning capabilities of LLMs. It has numerous practical applications, such as addressing traffic congestion. In this context, the primary objective is

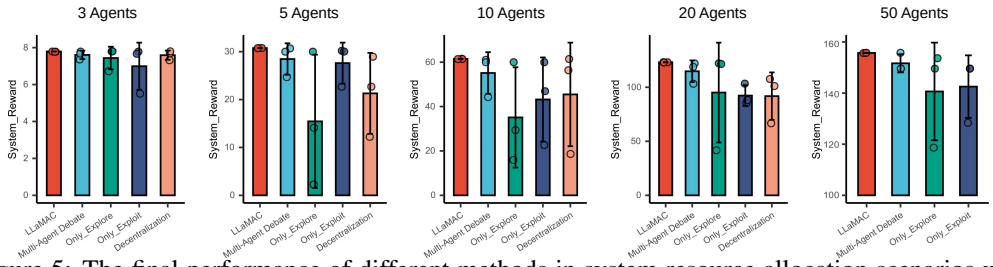

Figure 5: The final performance of different methods in system resource allocation scenarios with different number of agents.

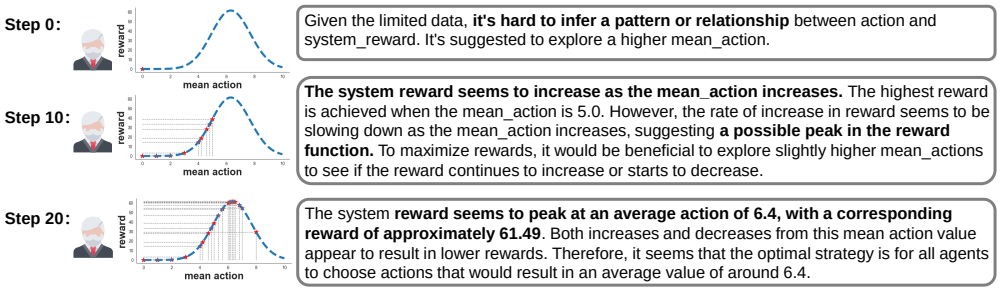

Figure 6: The Assessor in system resource allocation scenario undertakes the crucial tasks of data collection and cognitive analysis. The blue dashed line represents the reward function, while the red dots indicate the explored actions.

to achieve effective system resource allocation among multiple traffic controllers acting as agents. These agents play a crucial role in directing vehicles onto the main road, optimizing the utilization of the main route while mitigating congestion.

In our experimental setup, the system objective function is defined as the Gaussian squeeze function: $R(x) = xe^{-\frac{(x-\mu)^2}{\sigma^2}}$, where $x = \sum_{i \in \mathcal{I}} a^i$ represents the sum of actions chosen by all agents, μ and σ are inherent parameters of the system representing the mean and variance, respectively.

In this scenario, each agent is capable of selecting an integer between 0 and 9 as their action, with no knowledge of the choices made by other agents. The objective for the agents is to synthesize their experiences from multiple decision rounds and infer the allocation scheme that leads to the maximum rewards. Centralized critic possesses the authority to access the actions taken by all agents and the corresponding average values of these actions. This particular scenario is highly suited for validating the capabilities of the TripletCritic.

Specifically, we consider scenarios with different numbers of agents, namely 3, 5, 10, 20, and 50. As the number of agents increases, the difficulty of decision-making escalates. We examine several comparative experimental setups, including the *Multi-agent Debate* method Chan et al. (2023), which has recently been utilized in the field of NLP to alleviate hallucinations and enhance mathematical reasoning abilities. Additionally, we explore the *Only_Explore* approach that solely utilizes a critic biased towards exploration, the *Only_Exploit* approach that employs a critic biased towards exploitation, and the *Decentralization* method where each agent independently makes decisions based on its own observation history. Due to limitations in terms of access costs, we solely test the *Decentralization* method for scenarios involving fewer than 20 agents.

4.1.2 Results

As shown in Figure 4, it is evident that within a limited number of steps, LLaMAC demonstrates the ability to explore and learn through continuous interaction with the environment. The final performance of all methods is presented in Figure 5. The TripletCritic approach within LLaMAC exhibits a similar structure to the Multi-agent Debate method, and compared to other approaches, these two methods display relatively stable performance. However, debate-based methods often suffer from excessive or insufficient exploration, resulting in a tendency to converge to local optima. On the other hand, approaches that emphasize exploration and exploitation struggle to maintain stable performance. The former exhibits significant oscillations due to excessive exploration, while the latter

Table 2: Evaluation results under different grid settings in the Grid Transportation scenarios include metrics such as the success rate (*Success*), time steps (*Steps*) taken to execute tasks, and the count of feedback instances (*Feedback*). The values in parentheses correspond to a single standard deviation over 10 trials.

		Grid Transportation-Easy			Grid Transportation-Hard		
		Success	**Steps**	**Feedback**	**Success**	**Steps**	**Feedback**
2×2	**HMAS-2**	**100%**	9.9(2.74)	3.3(2.05)	80%	7.0(5.0)	6.0(9.74)
	LLaMAC	**100%**	**7.0(1.79)**	**2.0(1.26)**	**100%**	**4.7(1.35)**	**3.6(2.80)**
2×4	**HMAS-2**	80%	15.5(6.09)	12.3(5.83)	20%	17.0(9.0)	24.0(20.0)
	LLaMAC	**100%**	**7.6(1.36)**	**4.3(1.42)**	**90%**	**7.44(2.95)**	**10.56(7.54)**
4×8	**HMAS-2**	60%	30.6(9.70)	26.1(13.59)	0%	-	-
	LLaMAC	**100%**	**12.9(2.70)**	**10.7(3.35)**	**90%**	**8.44(1.57)**	**12.11(2.51)**

prematurely converges to local optima after only a few simple exploratory steps, aligning with the expected characteristics of these methods. Distributed approach incurs the highest access cost , as each agent is required to independently access the LLM. Nevertheless, the lack of collaboration among the agents still hinders the capture of true relationships.

4.1.3 Case Study

We explicitly depict the cognitive process of the assessor after continuous data collection, as illustrated in Figure 6. It can be observed that LLaMAC is capable of providing insightful recommendations based on the current state of data collection, aiding in further inference of the relationship between actions and rewards. At step 10, the collected data only reveals a positive correlation between actions and rewards. However, remarkably, the Assessor accurately identifies the non-linear growth pattern of rewards and infers the existence of a potential peak in the objective function. After 20 decision rounds, the Assessor successfully identifies the optimal value and conducts thorough exploration near the peak to avoid getting trapped in local optima.

4.2 Grid Transportation

4.2.1 Experimental Settings

The robot grid transportation task is relatively more complex as it simulates the automatic control system of robots in factory assembly line operations. It can be considered as a multi-step decision problem that requires the spatial reasoning and logical reasoning capabilities of LLMs. Additionally, it puts the long-term planning ability to the test. We consider two environmental configurations:

Grid Transportation-Easy. The environment consists of a grid of size $N \times M$, with one intelligent agent assigned to each grid cell. Different types of objects and targets are unevenly distributed across the grid. The objective of the intelligent agents is to transport all objects to their respective targets. The available actions for each agent include moving an object to a horizontally or vertically adjacent grid cell, or placing an object into the target location if both the object and target are in the same grid cell.

Grid Transportation-Hard. The task goals are the same as in the easy scenario, with the key difference being that objects can only move along the grid boundaries. Each robot's available actions include moving an object located at one of the four corners of its grid cell to one of the other three corners, or to the target location if the object's target position is within the grid. In this scenario, the interdependent coordination among agents becomes more complex. Objects located at a particular corner may be moved simultaneously by multiple agents, leading to conflicts. Additionally, adjacent agents may attempt to move different objects to the same corner, resulting in collisions.

Our objective is to ensure the smooth execution of tasks and the successful accomplishment of goals by LLM-based agents. When an agent experiences hallucinations that persist beyond the specified iteration limit, the task is deemed unsuccessful. This includes instances where the output grammar format fails to meet the requirements even after reaching the maximum number of iterations, when the dialogue context exceeds the token length limit, and when the decision time steps surpass the designated limit.

4.2.2 RESULTS

We conduct a comparative analysis between our method and the state-of-the-art solution, HMAS-2 Chen et al. (2023b). For each scenario, we conduct tests on grid configurations of 2×2, 2×4, and 4×8, respectively. Table 2 presents a comprehensive performance comparison between the two methods, clearly demonstrating the overall superiority of our approach. In complex scenarios involving long-term iterative decision-making, LLaMAC exhibits a significantly higher success rate compared to HMAS-2. Furthermore, LLaMAC consistently achieves task completion in fewer interaction steps, highlighting the performance advantages of its employed strategies. Additionally, as shown in Figure 7 the TripletCritic facilitates the generation of superior initial suggestions, thereby reducing the need for feedback iterations and greatly enhancing token utilization efficiency.

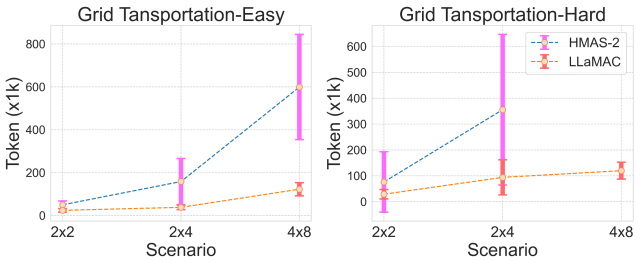

Figure 7: Token usage of LLaMAC and HMAS-2 in the Grid Transportation scenarios.

4.2.3 CASE STUDY

During the experimental process, we observe that LLaMAC effectively enhances the capabilities of LLM in long-term planning and execution, spatial reasoning, and learning from interactions or errors. For example, spatial reasoning poses a significant challenge for LLMs, as they are more prone to hallucinations when determining whether an object is closer to the target. This issue becomes more pronounced in the Hard scenario. As shown in Figure 8, in the HMAS-2 method, agents often move objects to positions far from the target and may repeatedly move them between two particular locations. In contrast, in LLaMAC, such occurrences are often corrected during the external feedback phase. The actor only needs to focus on its own task, and when it receives suggestions from the critic, the difficulty of determining the effectiveness of individual agent tasks is significantly reduced compared to joint policies. This makes spatial reasoning errors more easily detected, reflected and corrected.

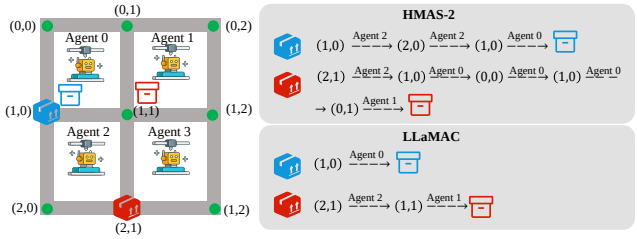

Figure 8: The performance of LLaMAC and HMAS-2 in the 2x2 robotic grid transportation scenario. To enhance visualization, non-essential objects and targets within the scene are concealed.

5 CONCLUSION

In this study, we present a novel framework called LLaMAC to enhance the collaborative performance of large-scale multi-agent systems based on Large Language Models. Building upon the commonsense reasoning capabilities exhibited by LLMs, we effectively augment the planning and coordination abilities among agents through stable reasoning mechanisms and comprehensive feedback mechanisms, facilitating continuous interaction between agents and the environment. LLaMAC demonstrates remarkable performance in coordinated scenarios involving a large number of agents. Notably, it exhibits exceptional capabilities in long-term planning, mathematical reasoning and optimization problems, spatial reasoning, and learning from mistakes. Additionally, LLaMAC reduces the access costs associated with large-scale multi-agent collaboration. We believe that with further enhancements in LLMs and the emergence of more collaboration frameworks, the field of multi-agent collaboration will experience new opportunities for advancement.

6 ACKNOWLEDGMENTS

This project was supported by the Strategic Priority Research Program of the Chinese Academy of Sciences, Grant No. XDA27050102

REFERENCES

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*, 2023.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *arXiv preprint arXiv:2308.10848*, 2023a.

Yongchao Chen, Jacob Arkin, Yang Zhang, Nicholas Roy, and Chuchu Fan. Scalable multi-robot collaboration with large language models: Centralized or decentralized systems? *arXiv preprint arXiv:2309.15943*, 2023b.

Antonia Creswell, Murray Shanahan, and Irina Higgins. Selection-inference: Exploiting large language models for interpretable logical reasoning. *arXiv preprint arXiv:2205.09712*, 2022.

Will Dabney, Zeb Kurth-Nelson, Naoshige Uchida, Clara Kwon Starkweather, Demis Hassabis, Rémi Munos, and Matthew Botvinick. A distributional code for value in dopamine-based reinforcement learning. *Nature*, 577(7792):671–675, 2020.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

Ran Gong, Qiuyuan Huang, Xiaojian Ma, Hoi Vo, Zane Durante, Yusuke Noda, Zilong Zheng, Song-Chun Zhu, Demetri Terzopoulos, Li Fei-Fei, et al. Mindagent: Emergent gaming interaction. *arXiv preprint arXiv:2309.09971*, 2023.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*, 2023.

Chris HolmesParker, M Taylor, Yusen Zhan, and Kagan Tumer. Exploiting structure and agent-centric rewards to promote coordination in large multiagent systems. In *Adaptive and learning agents workshop*, 2014.

Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, et al. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 2023.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

Yilun Kong, Jingqing Ruan, Yihong Chen, Bin Zhang, Tianpeng Bao, Shiwei Shi, Guoqing Du, Xiaoru Hu, Hangyu Mao, Ziyue Li, Xingyu Zeng, and Rui Zhao. Tptu-v2: Boosting task planning and tool usage of large language model-based agents in real-world systems. *arXiv preprint arXiv:2311.11315*, 2023.

Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. Trust region policy optimisation in multi-agent reinforcement learning. *arXiv preprint arXiv:2109.11251*, 2021.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for "mind" exploration of large language model society. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*, 2023.

Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, 2023.

Zhao Mandi, Shreeya Jain, and Shuran Song. Roco: Dialectic multi-robot collaboration with large language models. *arXiv preprint arXiv:2307.04738*, 2023.

OpenAI. Gpt-4 technical report, 2023.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22, 2023.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *The Journal of Machine Learning Research*, 21(1):7234–7284, 2020.

Jingqing Ruan, Yihong Chen, Bin Zhang, Zhiwei Xu, Tianpeng Bao, Guoqing Du, Shiwei Shi, Hangyu Mao, Xingyu Zeng, and Rui Zhao. Tptu: Task planning and tool usage of large language model-based ai agents. *arXiv preprint arXiv:2308.03427*, 2023a.

Jingqing Ruan, YiHong Chen, Bin Zhang, Zhiwei Xu, Tianpeng Bao, Hangyu Mao, Xingyu Zeng, Rui Zhao, et al. Tptu: Task planning and tool usage of large language model-based ai agents. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023b.

Matthijs TJ Spaan. Partially observable markov decision processes. In *Reinforcement learning: State-of-the-art*, pp. 387–414. Springer, 2012.

Haoye Tian, Weiqi Lu, Tsz On Li, Xunzhu Tang, Shing-Chi Cheung, Jacques Klein, and Tegawendé F Bissyandé. Is chatgpt the ultimate programming assistant–how far is it? *arXiv preprint arXiv:2304.11938*, 2023.

Shenzhi Wang, Chang Liu, Zilong Zheng, Siyuan Qi, Shuo Chen, Qisen Yang, Andrew Zhao, Chaofei Wang, Shiji Song, and Gao Huang. Avalon's game of thoughts: Battle against deception through recursive contemplation. *arXiv preprint arXiv:2310.01320*, 2023a.

Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. Mint: Evaluating llms in multi-turn interaction with tools and language feedback. *arXiv preprint arXiv:2309.10691*, 2023b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

Yuzhuang Xu, Shuo Wang, Peng Li, Fuwen Luo, Xiaolong Wang, Weidong Liu, and Yang Liu. Exploring large language models for communication games: An empirical study on werewolf. *arXiv preprint arXiv:2309.04658*, 2023.

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin, and Xia Hu. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *arXiv preprint arXiv:2304.13712*, 2023.

Yaodong Yang and Jun Wang. An overview of multi-agent reinforcement learning from game theoretical perspective. *arXiv preprint arXiv:2011.00583*, 2020.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Bin Zhang, Lijuan Li, Zhiwei Xu, Dapeng Li, and Guoliang Fan. Inducing stackelberg equilibrium through spatio-temporal sequential decision-making in multi-agent reinforcement learning. *arXiv preprint arXiv:2304.10351*, 2023a.

Bin Zhang, Hangyu Mao, Lijuan Li, Zhiwei Xu, Dapeng Li, Rui Zhao, and Guoliang Fan. Stackelberg decision transformer for asynchronous action coordination in multi-agent systems. *arXiv preprint arXiv:2305.07856*, 2023b.

Ceyao Zhang, Kaijie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei Zhang, Anji Liu, Song-Chun Zhu, et al. Proagent: Building proactive cooperative ai with large language models. *arXiv preprint arXiv:2308.11339*, 2023c.

Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B Tenenbaum, Tianmin Shu, and Chuang Gan. Building cooperative embodied agents modularly with large language models. *arXiv preprint arXiv:2307.02485*, 2023d.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control*, pp. 321–384, 2021.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*, 2023e.

Meng Zhou, Ziyu Liu, Pengwei Sui, Yixuan Li, and Yuk Ying Chung. Learning implicit credit assignment for cooperative multi-agent reinforcement learning. *Advances in neural information processing systems*, 33:11853–11864, 2020.

Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, et al. Ghost in the minecraft: Generally capable agents for open-world enviroments via large language models with text-based knowledge and memory. *arXiv preprint arXiv:2305.17144*, 2023.

A ENVIRONMENT DETAILS

A.1 SYSTEM RESOURCE ALLOCATION

The system resource allocation environment can be regarded as an optimization problem or a single-step decision problem, where the available actions of all agents are fixed at each decision-making instance. The memory stores the observation history of the agents in the form of a dictionary: *[{action:[], system_reward:[]}, ..., {action:[], system_reward:[]}]*. Additionally, we require the decision-makers to simultaneously output *thoughts* and *actions* to enhance the reasoning capability of the language model.

A.2 GRID TRANSPORTATION

Grid transportation tasks are inherently more complex and demand higher decision-making capabilities. They involve language models assuming different roles to collaborate through continuous dialogue and interaction, generating long-term action trajectories, and ultimately achieving the final objectives.

In this environment, the *Veracity Scrutiny* within the Internal Feedback involves policy checks of the joint strategy and is set to evaluate (1) whether the output grammar conforms to the specified format and (2) whether the joint actions result in conflicts. The *Plan Confirmation* within the External Feedback involves policy checks specific to each agent and is set to evaluate (1) the availability of actions and (2) whether the suggestions result in a shorter Manhattan distance between objects and targets. Taking the Hard scenario as an example, the variables utilized in the decision-making process of the intelligent agents are depicted in Figure A.1.

Example of Environmental Information

State:

```
{"0.5_0.5": ["target_blue", "target_green"], "0.5_1.5": ["target_red", "target_purple",
"target_orange"], "1.5_0.5": [], "1.5_1.5": [], "0.0_0.0": ["box_red"], "0.0_1.0": ["box_purple"],
"0.0_2.0": [], "1.0_0.0": ["box_orange"], "1.0_1.0": ["box_green"], "1.0_2.0": [], "2.0_0.0":
["box_blue"], "2.0_1.0": [], "2.0_2.0": []}
```

Action:

```
{"Agent[0.5, 0.5]":"move(box_green, target_green)", "Agent[0.5, 1.5]":"move(box_purple,
target_purple)", "Agent[1.5, 0.5]":"move(box_orange, position[1.0, 1.0])"}
```

Observation & Available Actions:

```
Agent[0.5, 0.5]: I am in square[0.5, 0.5], I can observe ['target_blue'], I can do ['move(box_red,
position(0.0, 0.0))', 'move(box_red, position(0.0, 1.0))', 'move(box_red, position(1.0, 1.0))',
'move(box_blue, position(0.0, 0.0))', 'move(box_blue, position(0.0, 1.0))', 'move(box_blue,
position(1.0, 0.0))', 'move(box_blue, target_blue)']
Agent[0.5, 1.5]: I am in square[0.5, 1.5], I can observe ['target_red'], I can do ['move(box_blue,
position(0.0, 1.0))', 'move(box_blue, position(0.0, 2.0))', 'move(box_blue, position(1.0, 2.0))']
Agent[1.5, 0.5]: I am in square[1.5, 0.5], I can observe [], I can do ['move(box_red, position(1.0,
1.0))', 'move(box_red, position(2.0, 0.0))', 'move(box_red, position(2.0, 1.0))', 'move(box_blue,
position(1.0, 0.0))', 'move(box_blue, position(2.0, 0.0))', 'move(box_blue, position(2.0, 1.0))']
Agent[1.5, 1.5]: I am in square[1.5, 1.5], I can observe [], I can do ['move(box_blue, position(1.0,
2.0))', 'move(box_blue, position(2.0, 1.0))', 'move(box_blue, position(2.0, 2.0))']
```

Figure A.1: Text description of states, observations, and actions in the grid transport environment.

B IMPLEMENTATION DETAILS

B.1 PSEUDO-CODE

Algorithm 1 Execution Procedure for LLaMAC

Hyperparameters: Length of episode T, number of agents N, trajectory length of Memory L, maximum number of internal and external feedback iterations IF, EF
Initialize: Memory $\mathcal{M}$, Environmental initial state s_0 and observation $\{o_0^i\}_{i\in\mathcal{I}}$, timestep $t = 0$

while $t \leq T$ **do**
  TripletCritic receives the memory information m_t and the current state s_t
  Generate suggestion for all actors $su = \{su_t^i\}_{i\in\mathcal{I}}$ through Internal Feedback (Algorithm 2)
  Genearate the joint action $\mathbf{a}_t = \{a_t^1, a_t^2, \ldots, a_t^n\}$ through External Feedback (Algorithm 3)
  Execution the joint action $\mathbf{a}_t$, obtain reward r_t^i and environmental state s_{t+1}
  Collect trajectories τ^i, push transitions $\{(s_t, a_t^i, r_t^i, s_{t+1}\}$ into $\mathcal{M}$
end while

Algorithm 2 Internal Feedback

Input: Maximum number of internal feedback iterations IF, current iteration number$f_i = 0$, feedback information$F_{if} = None$, state s_t, memory m_t

while $f_i \leq IF$ **do**
  for critic $j = 1$ **to** 2 **do**
    Generate actions $\mathbf{a}_j = \{a_t^i\}_{i\in\mathcal{I}}$ corresponding to preference $\mathbf{a}_j \sim \mathbf{LLM}_{critic_j}(m_t, s_t, F_{if})$
  end for
  Assessor makes *Veracity Scrutiny*, get results vs
  if vs is True **then**
    Assessor makes *Belief Correction*, generate final action suggestion for all actors $su = \{su_t^i\}_{i\in\mathcal{I}}$, where $su \sim \mathbf{LLM}_{assessor}(m_t, s_t, \mathbf{a}_1, \mathbf{a}_2)$
    break
  else
    Generate feedback information F_{if}
  end if
  $f_i = f_i + 1$
end while

B.2 PROMPT EXAMPLE IN LLAMAC

As shown in Section B.1, the entire process of LLaMAC's iterative decision-making is facilitated by internal and external feedback mechanisms, enabling seamless collaboration among its modules to accomplish decision tasks in large-scale intelligent agent systems. As shown in Figure B.1 and Figure B.2, we demonstrate the core content of the prompt used by LLaMAC to access LLM during the decision-making process.

Algorithm 3 External Feedback

Input: Maximum number of Enternal feedback iterations IF, current iteration number $f_e = 0$, feedback information $F_{ef} = []$, suggestion from TripletCritic su, observation $\{o_t^i\}_{i \in \mathcal{I}}$, state s_t

while $f_e \leq EF$ **do**
  for agent $i = 1$ **to** N **do**
    Actor i makes *Plan Confirmation*
    if Execute **then**
      $a_t^i = su_t^i$
    else
      Generate actor feedback Information $F_{ef}^i \sim \mathbf{LLM}_{actor_i}(o_t^i, su_t^i)$
      $F_{ef} = F_{ef} + F_{ef}^i$
    end if
  end for
  if F_{ef} is not [] **then**
    Assessor regenerates action suggestions $su \sim \mathbf{LLM}_{assessor}(m_t, s_t, F_{ef})$
  else
    break
  end if
  $f_e = f_e + 1$
end while

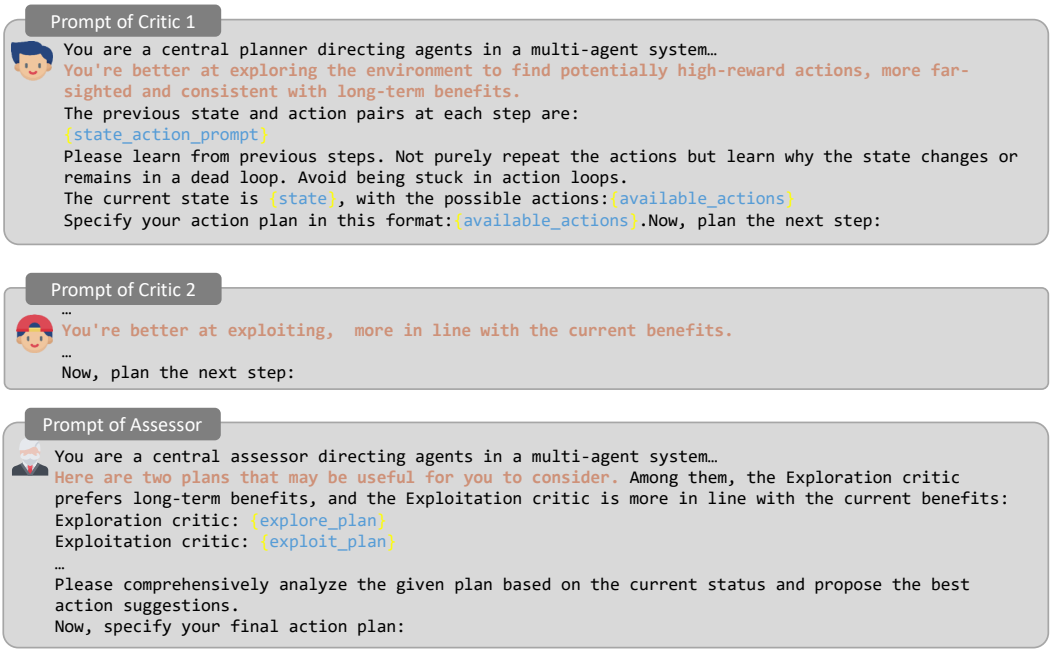

Figure B.1: Prompt for the large language model in the internal feedback mechanism.

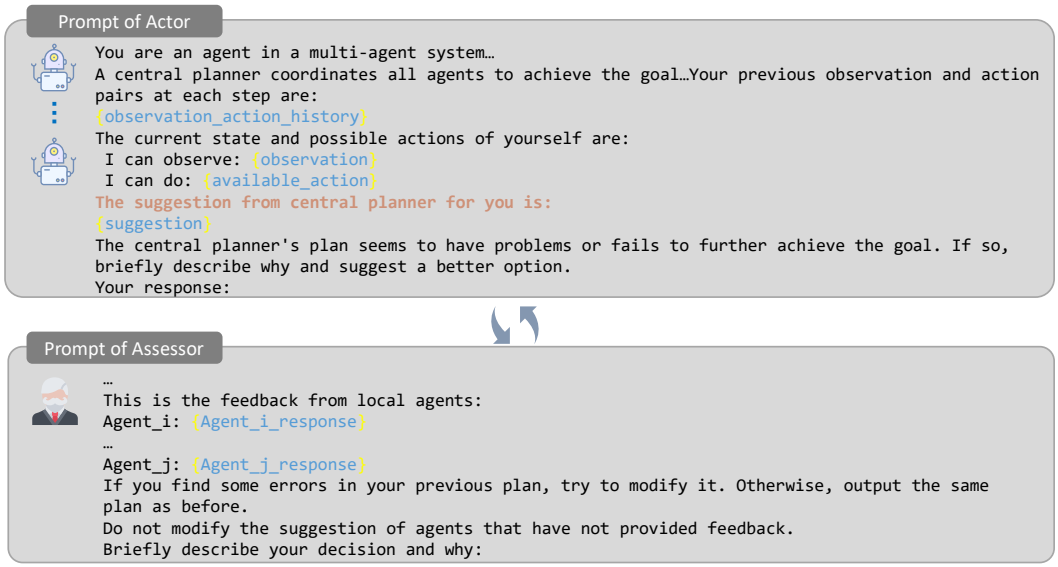

Figure B.2: Prompt for the large language model in the external feedback mechanism. Only actors that are determined by the plan confirmation to be reused will execute this access.