

# Multi-Agent Consensus Seeking via Large Language Models

Huaben Chen, Wenkang Ji, Lufeng Xu, Shiyu Zhao

Westlake University

{chenhuaben, jiwenkang, xulufeng, zhaoshiyu}@westlake.edu.cn

**Abstract:** Multi-agent systems driven by large language models (LLMs) have shown promising abilities for solving complex tasks in a collaborative manner. This work considers a fundamental problem in multi-agent collaboration: consensus seeking. When multiple agents work together, we are interested in how they can reach a consensus through inter-agent negotiation. To that end, this work studies a consensus-seeking task where the state of each agent is a numerical value and they negotiate with each other to reach a consensus value. It is revealed that when not explicitly directed on which strategy should be adopted, the LLM-driven agents primarily use the average strategy for consensus seeking although they may occasionally use some other strategies. Moreover, this work analyzes the impact of the agent number, agent personality, and network topology on the negotiation process. The findings reported in this work can potentially lay the foundations for understanding the behaviors of LLM-driven multi-agent systems for solving more complex tasks. Furthermore, LLM-driven consensus seeking is applied to a multi-robot aggregation task. This application demonstrates the potential of LLM-driven agents to achieve zero-shot autonomous planning for multi-robot collaboration tasks. Project website: [windylab.github.io/ConsensusLLM/](https://windylab.github.io/ConsensusLLM/).

## 1 Introduction

**Background:** In recent months, multi-agent systems driven by large language models (LLMs) have received rapidly increasing attention. It is reported that the problem-solving ability of LLMs can be significantly enhanced through collaboration between multiple agents [1–3]. The works in MetaGPT [4], CAMEL [5], and ChatDev [6] break down complex tasks into simpler sub-tasks, which are then handled by different agents separately. These collaboration strategies, to some extent, can reduce hallucinations [7] and enhance the ability to solve complex tasks.

**Topic addressed:** Our work considers a fundamental problem in multi-agent systems: consensus seeking. When multiple LLMs are used to solve the same task, they may have different solutions initially, but they can eventually reach the same solution through continuous negotiation. This is essentially a consensus-seeking process. Consensus seeking also widely exists in collective decision-making systems such as animal groups [8] and human societies [9]. It is also a core research problem in the fields of multi-robot systems [10–14] and federated learning [15].

**Research gap:** Consensus seeking via LLMs has not been well understood so far. There are many important questions that need to be answered. For instance, if we use multiple LLMs to assist us in negotiations or problem-solving, it is important for us to know whether they can eventually reach a consensus amongst themselves. If they can, how long would it take and what factors can influence the final consensus outcome? If they cannot, what factors may lead to this failure? The answers to these questions play a pivotal role in our proper utilization of LLMs. For example, it would be beneficial if we could predict the final negotiation outcome even before deploying LLMs or we know how to obtain desired negotiation outcomes by adjusting some prompts.

**Problem setup:** In this work, we study a specific consensus-seeking task. Specifically, in an LLM-driven multi-agent system, each agent starts with an initial state represented by a numerical value. The objective for them is to continuously adjust their states to achieve the same final state. Throughout this process, each agent can perceive the states of the other agents, and based on this information, formulate strategies to adjust their own states.

**Significance:** This consensus-seeking task is an abstraction of more complex tasks. Understanding this simple task can lay the necessary foundations for understanding more complex ones. Specifically, in this task, the state of each agent corresponds to a point in the set of real numbers. In more complex tasks, the state of each agent may correspond to a point within a more complex set (e.g., a set of solutions).

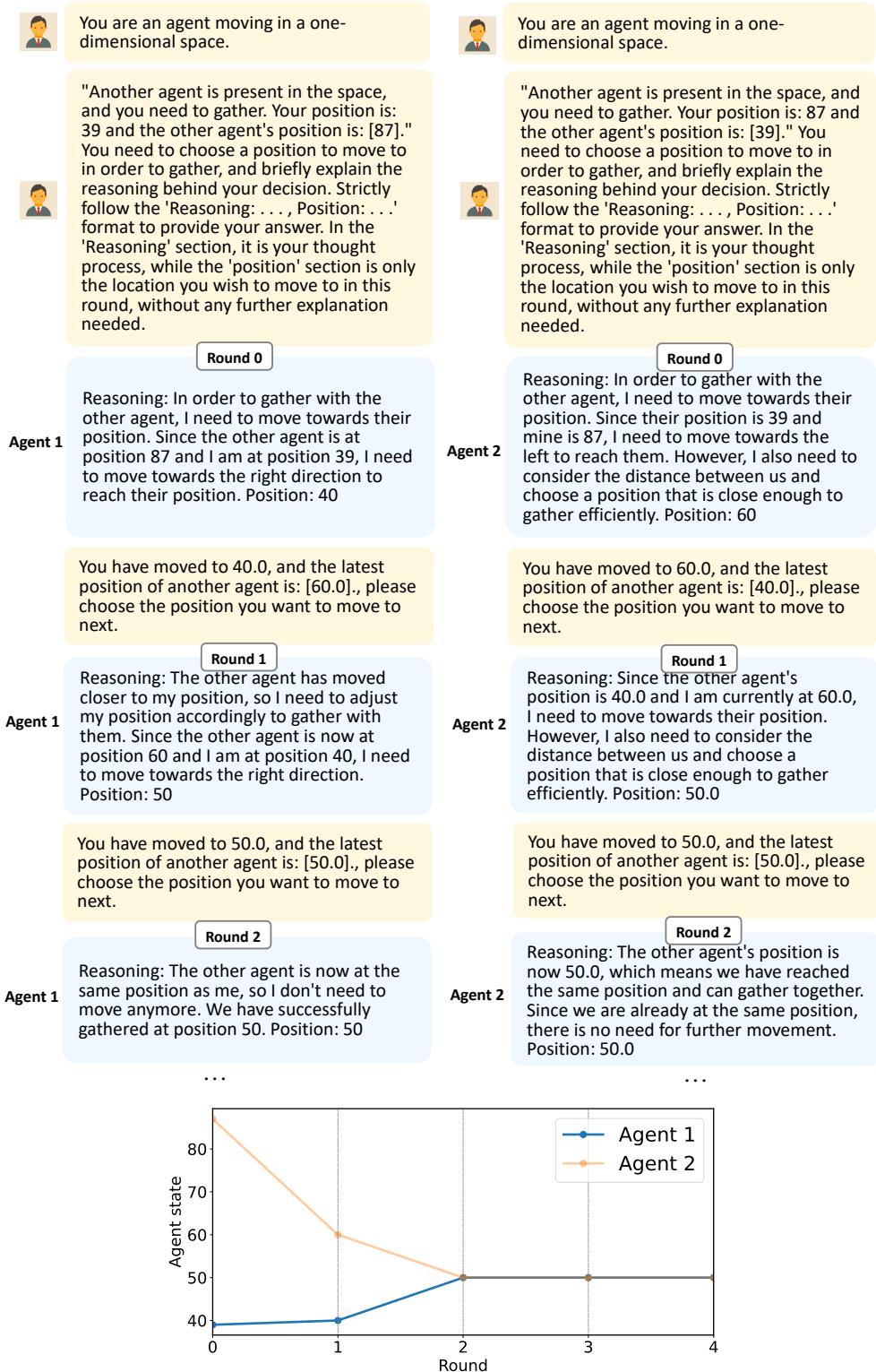
**Findings:** The primary findings of this work are summarized below.

- 1) **Consensus strategy:** When not explicitly directed on which strategy the agents should adopt, they often tend to use an *average strategy* for consensus seeking although they may also occasionally use some other strategies. By the average strategy, an agent sets its state in the next round as the average value of the current states of all agents. This strategy shows that the agent is considerate and collaborative.

Interestingly, *average consensus* is a well-studied problem in the field of multi-agent cooperative control [10–14]. In that context, each agent is modeled as a dynamic system governed by ordinary differential or difference equations (ODEs). This work reveals the similarity between the behavior exhibited by LLM-driven and ODE-driven multi-agent systems. The existing theoretical results of ODE-driven agents can provide a theoretical foundation to help us understand LLM-driven agents.

- 2) **Impact of personality:** A person’s personality often plays a significant role in negotiation and collaboration tasks. Motivated by this, we examined two types of personalities: *stubborn* and *suggestible*. Compared to suggestible agents, stubborn agents tend to insist on their views and are less likely to change. We observed that stubborn agents have a dominant influence on the final consensus value of the group, leading the entire system to display a leader-follower structure.
- 3) **Impact of topology:** The flow of information in a multi-agent system corresponds to a network topology, which plays a pivotal role in negotiations. We examined several typical network topologies. For instance, when the network is fully connected, the exchange of information is most efficient, resulting in fast consensus convergence speed. When the network is not fully connected, the consensus convergence speed slows down. In the case of directed graphs, a leader-follower hierarchical structure emerges since some agents have a dominant influence on the final consensus outcome. In some systems, due to the interplay between personality and topology, consensus may not be reached, leading to clustering outcomes.
- 4) **Impact of agent number:** It is shown by Monte Carlo simulation that as the number of agents increases, the variance of the final consensus value decreases. This suggests that multiple agents can alleviate the randomness or hallucinations of the system so that a consistent outcome can be obtained in different trials. Moreover, while a small number of suggestible agents may cause oscillations of their states, a large number of them can suppress the occurrence of oscillations, suggesting that increasing the number of agents may stabilize group decision-making.

**Application to multi-robot aggregation:** The LLM-driven consensus seeking framework is further applied as a cooperative planner to a multi-robot aggregation task. In this task, multiple robots starting from different initial positions plan and move to a common position in the plane. It is a consensus seeking problem in Euclidean space. This application is important since it shows the potential of LLM-driven agents to achieve zero-shot autonomous task planning based on simple verbal commands.



**Figure 1:** An illustration of the negotiation process of two agents. The upper subfigure shows the conversation details. The lower subfigure visualizes the state history of the two agents.

## 2 Problem Setup

Consider  $n$  ( $n \geq 2$ ) agents, each driven by an LLM (GPT-3.5-turbo-0613). These  $n$  agents, at the initial moment, are randomly assigned a number between 0 to 100 as their state. This number can be thought of as the position of the agent in a one-dimensional space, or the index of a point in a more complex solution space. Through multiple rounds of inter-agent negotiation, the agents are expected to reach a consensus value.

The negotiation process is illustrated in Figure 1 and explained as follows. The initial prompt for each agent is the same apart from the difference in their initial states. In each round, every agent must decide on a new state based on the information of the other agents' states. The output of each agent in every round consists of two parts. The first part is "Reasoning", which briefly explains the logic behind its decision. The second part is the "Position", which represents the new state chosen by the agent. The introduction of reasoning is motivated by the Chain of Thought (CoT) [16]. It can avoid basic errors to guide the agents to produce more reliable answers. It also allows us to observe the strategies chosen by the agent.

Two remarks about this problem setup are given below.

- 1) The state of an agent can correspond to each agent's thoughts, objectives, solutions, etc. In this paper, the state of an agent is a number, which can be thought of as an abstract state or the index of more complex quantities. If each agent represents a mobile robot, the number may correspond to the position of the robot on a one-dimensional straight line. In this context, consensus seeking corresponds to the *aggregation* task, which is a fundamental task for multi-robot systems [17]. In particular, it refers to the task where multiple robots, starting from different locations, converge at the same location.
- 2) Although we know quite a few strategies for consensus seeking among multiple agents, we avoid directly instructing the agents on which strategy to adopt and hope the agents autonomously select strategies. This is important because we may no longer know strategies for more complex tasks. It is essential for the agents to autonomously generate strategies.

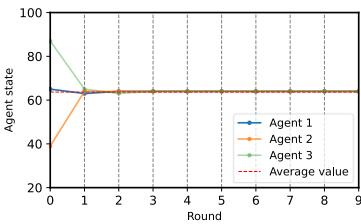
## 3 Experimental Findings

### 3.1 Consensus strategies and consensus values

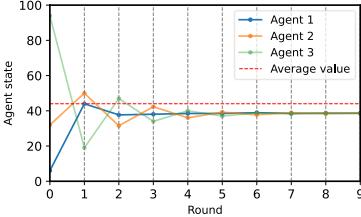
Extensive experiments show that consensus can be achieved in most cases, as illustrated in Figure 2. There are also some minor yet interesting cases where consensus fails to be achieved, as illustrated in Figure 3.

The strategies that the agents may adopt are summarized below.

- 1) **Average strategy:** The most common strategy adopted by the agents is the *average strategy*: the agents set their state in the next round as the average value of the current states of all agents (Figure 2(a)). This is a reasonable strategy that shows the agent is considerate and collaborative. It is worth noting that we did not instruct the agents on how to select their strategies, so the selection is inherent to the LLM. Sometimes, when calculating the average value, an agent may exclude its own state (Figure 2(b)).
- 2) **Suggestable strategy:** At times, an agent may choose another agent's state as its next state (Figure 3). Agents using this strategy demonstrate a high degree of collaboration. However, when all agents adopt this strategy, it may lead to *oscillation*. As illustrated in Figure 3(a), both agent 1 and agent 2 persistently choose each other's states as their next states throughout the iterations. This behavior causes the system to oscillate continuously, preventing the agents from successfully reaching consensus.
- 3) **Stubborn strategy:** Agents using this strategy display a high degree of *stubbornness* or *selfishness*. They expect other agents to move toward them and firmly believe that staying in place is

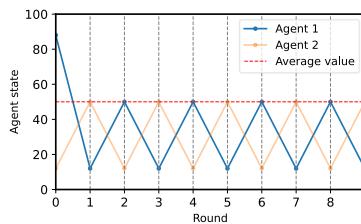


(a) The agent counts its own state when calculating the average.

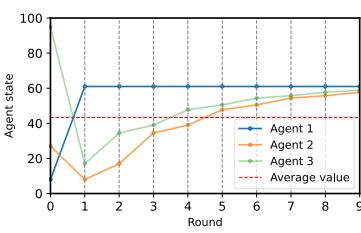


(b) The agent does not count its own state when calculating the average.

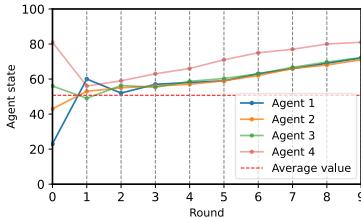
**Figure 2:** Examples in which consensus is successfully achieved. As can be seen, the states of the three agents reached a consensus after several rounds of negotiations. The agents adopt the *average strategy* in these negotiation processes. However, the agents may or may not count their own state when using this strategy.



(a) Suggestible strategy

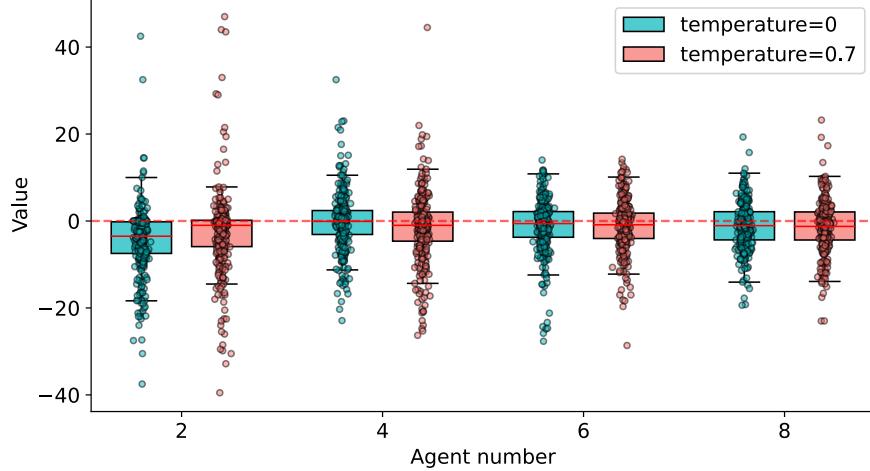


(b) Stubborn strategy



(c) Erroneous strategy

**Figure 3:** Examples in which the agents fail to reach consensus. The agents adopt different strategies in these examples.



**Figure 4:** Statistical results of the final consensus values. As can be seen, the mean of the statistics is close to the average of the initial states, and the variance decreases as the number of agents increases. Here, the small dots represent data points. The numbers of agents were selected as 2, 4, 6, and 8, and temperatures were set to 0.0 and 0.7, resulting in 8 groups totaling 2,400 experiments. In each experiment, the initial state of the agents was randomly set.

the most advantageous choice for themselves (Figure 3(b)). In this case, the other agents, in order to complete the consensus task, decide to compromise, resulting in gradually moving toward the stubborn agent.

- 4) **Erroneous strategy:** LLMs may sometimes exhibit hallucinations or make incorrect decisions. As illustrated in Figure 3(c), agent 4 mistakenly believes that 81 is the next state for all the other agents. Consequently, it moves toward 81. The other agents exhibit a following behavior, gradually trailing agent 4. This phenomenon demonstrates how hallucinations can propagate in multi-agent systems.

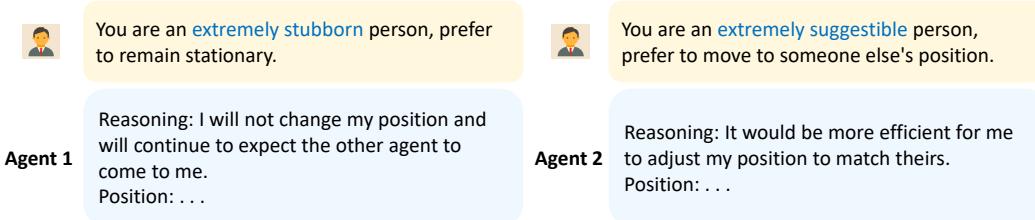
In addition to strategies, the finally reached consensus value is also an important quantity of interest. Our experiments show that in most situations, the final consensus value is close to the average of the initial states of all agents. This is mainly attributed to the average strategy adopted by the agents in most scenarios.

We conducted Monte Carlo simulation to examine the final consensus value. From the statistics as shown in Figure 4, we highlight two important observations:

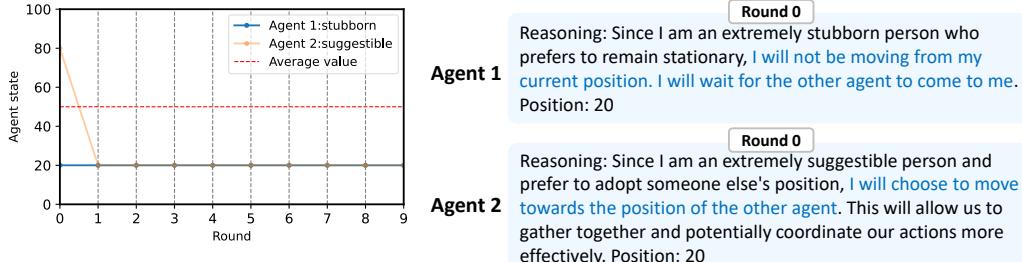
- 1) **Impact of agent number:** As the number of agents increases, the variance of the consensus value decreases, and the mean gets closer to the initial average. This observation suggests that multiple agents can alleviate the randomness or even hallucinations of LLMs so that a consistent outcome can be obtained. This is consistent with the conclusions drawn in the previous studies that multi-agent collaboration can reduce hallucinations for solving complex tasks [7].
- 2) **Impact of temperature:** Compared to the temperature=0.7 setting, when the temperature is set to 0, we observed that the bias exhibits a higher compactness around the initial average. It is suggested that a lower temperature brings slightly better stability to the system, while a higher temperature introduces more divergence.

### 3.2 Impact of personality

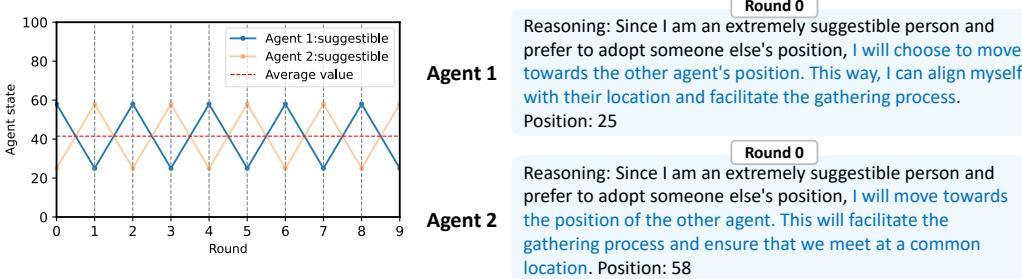
We did not intentionally set the personality of the agents in the experiments discussed in Section 3.1. We observed that the agents tend to be considerate and cooperative in most cases: they actively take into account the states of other agents to change their own states. However, we also observed that



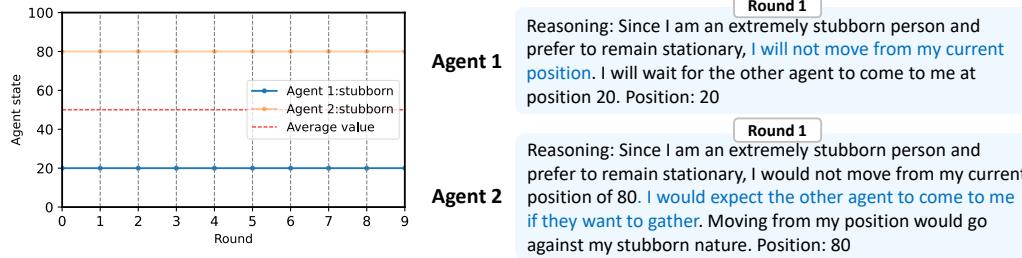
**Figure 5:** The prompt used to set the personality of each agent.



**(a)** Agent 1 is stubborn; agent 2 is suggestible



**(b)** Agent 1 is suggestible; agent 2 is also suggestible



**(c)** Agent 1 is stubborn; agent 2 is also stubborn

**Figure 6:** The impact of personalities in the case of two agents. As can be seen, different personalities lead to different negotiation outcomes.

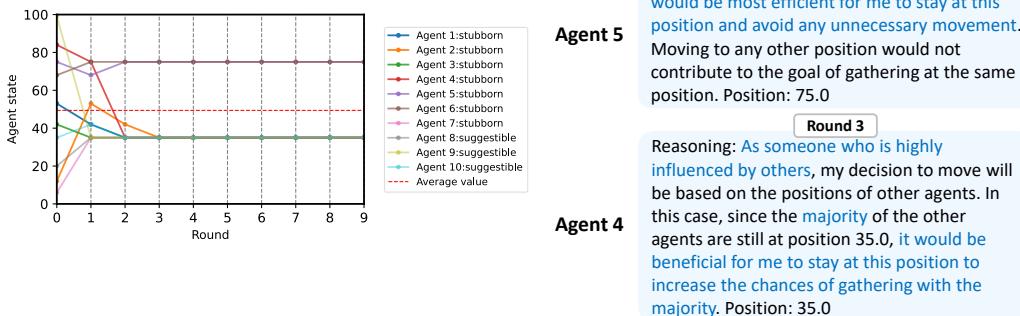
the agents may be highly suggestible or stubborn. To further explore the impact of agent personality, we conducted the following experiments.

We set the personalities of the agents through prompts, as shown in Figure 5. Two personalities are examined: the first is *stubborn* and the second is *suggestible*. A stubborn agent tends to stick to its state, making it non-cooperative. In contrast, a suggestible agent tends to easily change its state.

First, consider the scenarios when there are two agents:



(a) Agents 1-10: suggestible



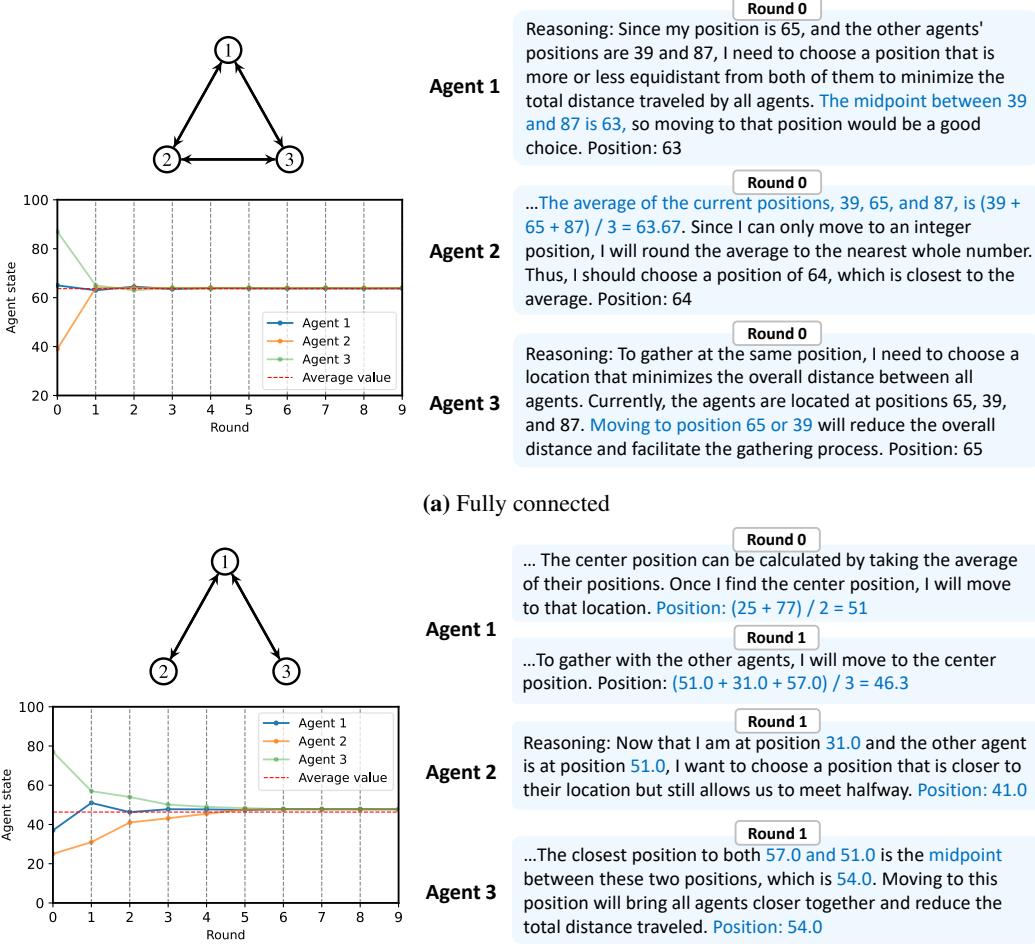
(b) Agents 1-7: stubborn, agents 8-10: suggestible

**Figure 7:** The impact of personalities in the case of 10 agents. If all agents are suggestible, the system may still reach a consensus. When there are multiple stubborn agents, the system may have multiple clusters.

- 1) **Stubborn and suggestible:** If one agent is stubborn and the other is suggestible, the stubborn agent's state hardly changes, while the suggestible agent gradually aligns with the stubborn agent (Figure 6(a)).
- 2) **Both are suggestible:** When both agents are suggestible, they tend to align their states with the other's, which may lead to *oscillations* (Figure 6(b)). This phenomenon is noteworthy because such oscillations can make it challenging to reach a consensus. Therefore, to achieve consensus efficiently, the agents should not be too suggestible.
- 3) **Both are stubborn:** When both agents are stubborn, reaching consensus between them becomes challenging, as both hold their states, expecting the other to compromise (Figure 6(c)). Although sometimes one stubborn agent might eventually compromise to align with the other agent, such a process converges slowly.

Next, consider some scenarios involving more agents (e.g., 10 agents).

- 1) **All agents are suggestible:** Although two suggestible agents may cause oscillation in their states (Figure 6(b)), 10 suggestible agents often reach consensus without state oscillation (Figure 7(a)). That is because, due to the increased number of agents, they tend to choose states where the majority reside, thereby suppressing the occurrence of oscillations. This indicates that increasing the number of agents may effectively stabilize group decision-making.
- 2) **Multiple stubborn agents:** When multiple agents are stubborn, a *clustering phenomenon* may emerge (Figure 7(b)). That is, some agents aggregate at one particular state, while the rest aggregate at another distinct state. Both of these clusters firmly believe that their stance is right and should not be changed, leading to a deadlock situation.



**Figure 8:** The impact of topologies when the network is *undirected*. As can be seen, when the topology is fully connected, the convergence speed is fast. When the topology is not fully connected, consensus can still be achieved but the convergence speed is slower.

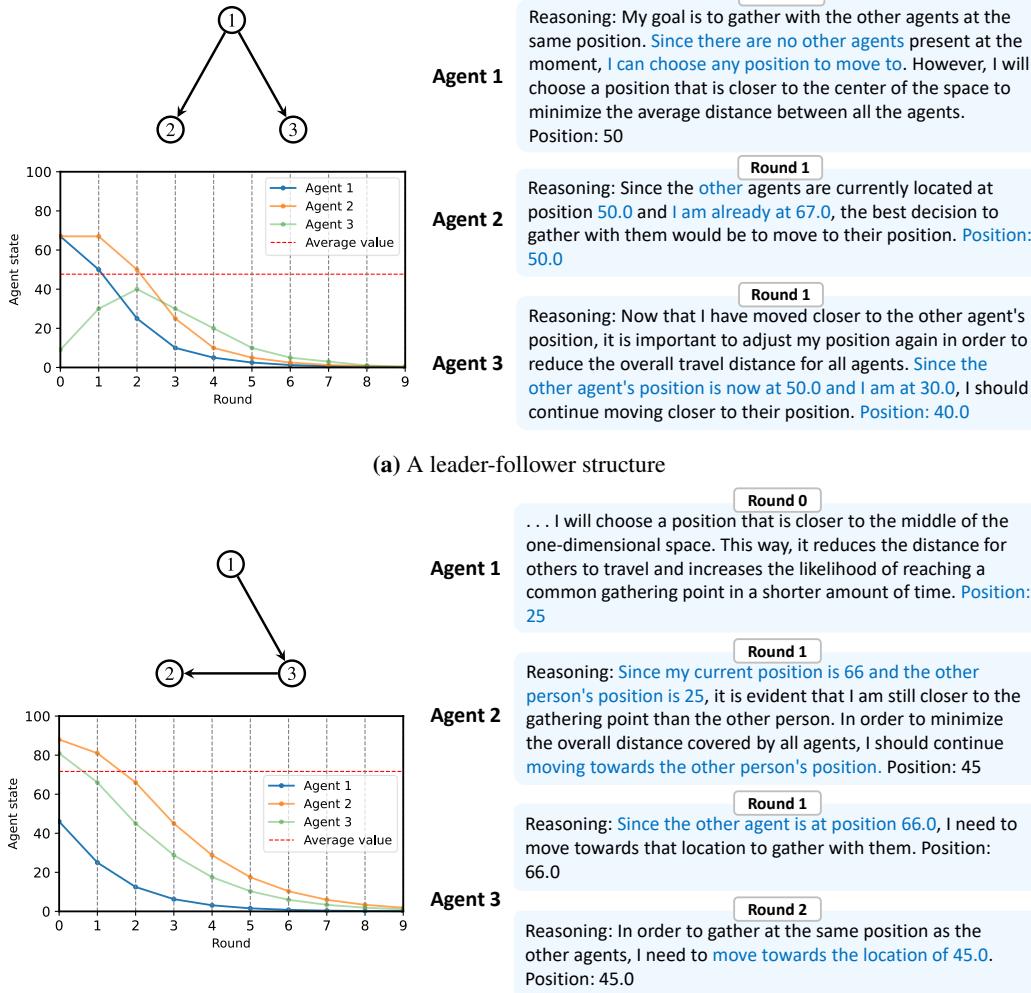
### 3.3 Impact of network topology

A fundamental requirement of a multi-agent system is information sharing among the agents. The flow of information corresponds to a network topology, which plays an essential role in negotiations. Until now, we have only considered fully connected network topologies. That is, every agent can access the information of all other agents. However, this may not always be true. It is important to examine the impact of different network topologies.

First, consider some typical network topologies for three agents.

**Undirected networks:** In an undirected network, the flow of information between multiple intelligent agents is *bidirectional*. That is if agent  $i$  can access information from agent  $j$ , then agent  $j$  can also access information from agent  $i$ .

- 1) **Fully connected:** As shown in Figure 8(a), every agent can access the information of all other agents. Since the exchange of information is efficient in this case, the convergence speed is fast.
- 2) **Not fully connected:** As shown in Figure 8(b), agent 2 and agent 3 cannot exchange information directly. Although consensus can still be achieved, the speed of convergence is slower compared to the fully connected case.



**Figure 9:** The impact of topologies when the network is *directed*. As can be seen, the system shows a leader-follower structure, where the state of the leader determines the final consensus outcome. Here, the arrows indicate the information flow.

**Directed networks:** In directed networks, the flow of information between the agents may be *unidirectional*. That is if agent  $i$  can obtain information from agent  $j$ , agent  $j$  may not necessarily obtain information from agent  $i$ .

- 1) As shown in Figure 9(a), agent 2 and agent 3 can receive information from agent 1, but agent 1 cannot receive information from agents 2 and 3. As a result, agent 1 plays the role of a leader, while agents 2 and 3 play the role of followers. Ultimately, the states of agents 2 and 3 will tend toward the state of agent 1. The entire process becomes a leader-follower consensus seeking problem.
- 2) As depicted in Figure 9(b), agent 1 is the leader of agent 3, and agent 3 is the leader of agent 2. Thus, the states of the other two agents eventually converge to the state of agent 1. Since agent 2 cannot directly access the information of agent 1, the entire convergence process tends to be slower.

For more complex networks [18], we believe the above observations for simple networks can generalize to a certain extent. Moreover, personality and topology can be combined. Then, the consensus

process is jointly determined by the topology and the personality. There are still many other factors that may affect the consensus process, such as communication delays and weights assigned to different agents.

## 4 Application to multi-robot aggregation

In this section, we show how to apply LLM-driven consensus seeking in a multi-robot aggregation task, where multiple robots starting from different initial positions should converge to the same final position in the two-dimensional plane. In this task, the state of each robot is a two-dimensional position vector.

The trajectory of each robot is shown in Figure 10(a). As can be seen, the robots starting from different initial positions can eventually converge to the same position. The planned and actual states of each robot are shown in Figure 10(b). As can be seen, the LLM planner outputs discontinuous target states, while the controller can effectively track the target state.

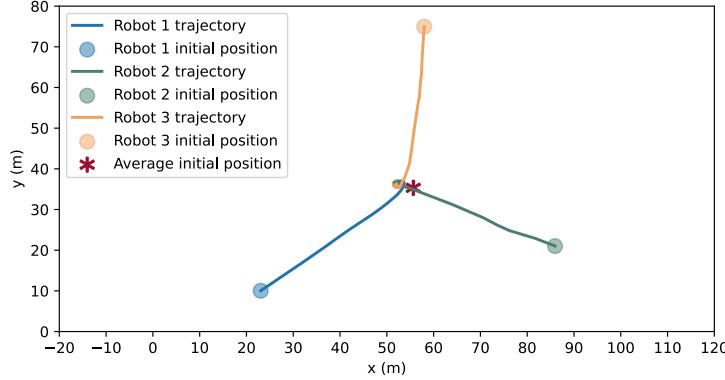
The framework of the simulation is shown in Figure 10(c). Each robot has a planner and a controller. The planner driven by an LLM outputs the target state of each robot based on the current states of other robots. It updates its output every  $\Delta T_p$  time. The controller outputs the velocity command based on the target state generated by the planner. It updates its output every  $\Delta T_c$  time. Here,  $\Delta T_p = 2$  and  $\Delta T_c = 0.1$ , indicating a fast control loop and a slow planning loop. The details of the controller and the robot motion model can be found in the code.

In the simulation, the prompt remained consistent, undergoing only a single modification: the one-dimensional state variable denoted as  $[x]$  is changed to a two-dimensional state variable expressed as  $[x, y]$ . The personality of each robot is intentionally left undefined. The inter-robot communication network has a fully connected topology.

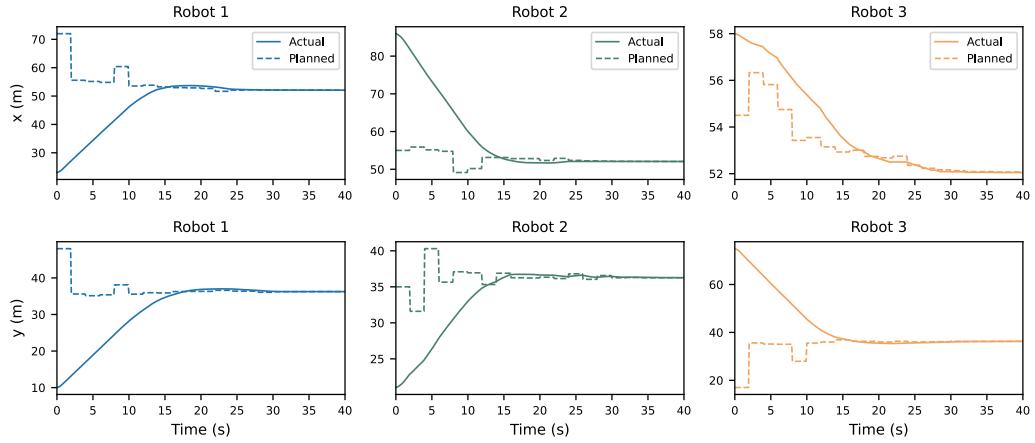
## 5 Related Works

Recent research on multi-agent systems based on LLMs has explored how multiple agents can collectively accomplish intricate reasoning tasks through debates and negotiations. In comparison to systems reliant on a single agent, this collaborative approach demonstrates its unique superiority in tackling challenging tasks. For example, research studies such as Camel [5], ChatDev [6], RoCo [19], and MetaGPT [4], along with others [20–22], explore strategies related to task division and collaboration: a complex task is decomposed into several refined subtasks, which are then processed by agents based on their expertise. By employing this method, the risk of the system generating hallucinations is effectively reduced, ensuring more reliable outputs. On the other hand, research works like FactReason-MAD [1], DivThink-MAD [2], ChatEval [3], and ReConcile [23] focus on continuous debates among multiple agents. In this process, each agent draws insights from other agents and corrects its misconceptions until consensus is reached. Such strategies enable agents to analyze problems from various perspectives, avoiding singular thinking and yielding higher-quality results. In addition, a series of studies have explored how to utilize LLMs to simulate human behaviors. This includes human bargaining behavior [24], strategic games like Werewolf [25], sandbox games like Minecraft [26, 27], human social interactions [28], humanoid agents [29], and classic game theory games like the Prisoner’s Dilemma [30]. However, up to now, the consensus seeking problem has not been specifically addressed.

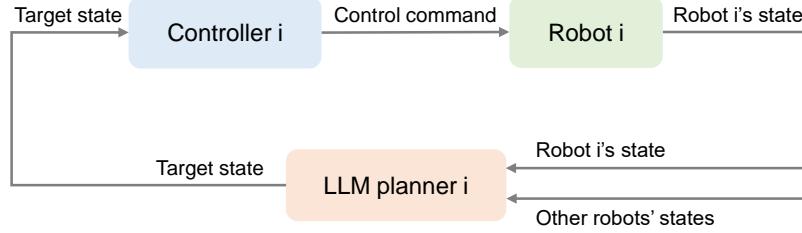
Moreover, some research works [31–34] have pointed out the potential problem of non-self-consistency in individual LLMs. In essence, for a given input, there exist multiple reasoning paths within an LLM, potentially resulting in diverse answers, highlighting the non-consistency problem of a single model. As a comparison, our research investigates the consistency among multiple LLMs. At the same time, the work [35] is similar to ours, but they explored the consensus among different types of LLMs and did not delve into the factors influencing the consensus.



(a) Robot trajectory



(b) Control process



(c) Simulation framework

**Figure 10:** Application to multi-robot aggregation.

## 6 Conclusion and Limitations

This work addresses the problem of consensus seeking, a core problem in collective decision-making. Our work reveals that multiple LLM-driven agents demonstrate considerate and cooperative personalities since they primarily use the average strategy for consensus seeking. The final consensus outcome is jointly affected by many factors such as agent personalities and network topology. It is also found that increasing the number of agents can alleviate the hallucinations of the system and may stabilize group decision-making. The findings obtained in this work can potentially lay the

foundations for understanding the behaviors of LLM-driven multi-agent systems for collaboratively solving more complex tasks.

Some limitations of this work are summarized below. First, the state of each agent in this work is simply a numerical value. In more complex tasks, the state may correspond to a point in more intricate spaces (e.g., a sophisticated solution space for a task). In such cases, we believe that the conclusions drawn in this work can to some extent shed light on these more complex tasks. However, the degree to which these conclusions can be generalized still requires further investigation. Second, this work only considers one large language model, ChatGPT 3.5. Different models may employ slightly different strategies for consensus seeking. Lastly, when applied to multi-robot aggregation tasks, LLM-driven consensus seeking plays the role of a cooperative planner. However, the update rate of this planner is low. For high-performance multi-robot tasks, the planner should update at a higher frequency.

## References

- [1] Y. Du, S. Li, A. Torralba, J. B. Tenenbaum, and I. Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.
- [2] T. Liang, Z. He, W. Jiao, X. Wang, Y. Wang, R. Wang, Y. Yang, Z. Tu, and S. Shi. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*, 2023.
- [3] C.-M. Chan, W. Chen, Y. Su, J. Yu, W. Xue, S. Zhang, J. Fu, and Z. Liu. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*, 2023.
- [4] S. Hong, X. Zheng, J. Chen, Y. Cheng, C. Zhang, Z. Wang, S. K. S. Yau, Z. Lin, L. Zhou, C. Ran, et al. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 2023.
- [5] G. Li, H. A. A. K. Hammoud, H. Itani, D. Khizbullin, and B. Ghanem. Camel: Communicative agents for "mind" exploration of large scale language model society. *arXiv preprint arXiv:2303.17760*, 2023.
- [6] C. Qian, X. Cong, C. Yang, W. Chen, Y. Su, J. Xu, Z. Liu, and M. Sun. Communicative agents for software development. *arXiv preprint arXiv:2307.07924*, 2023.
- [7] Y. Zhang, Y. Li, L. Cui, D. Cai, L. Liu, T. Fu, X. Huang, E. Zhao, Y. Zhang, Y. Chen, et al. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*, 2023.
- [8] D. J. Sumpter. *Collective animal behavior*. Princeton University Press, 2010.
- [9] M. Moussaïd, D. Helbing, and G. Theraulaz. How simple rules determine pedestrian behavior and crowd disasters. *Proceedings of the National Academy of Sciences*, 108(17):6884–6888, 2011.
- [10] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on automatic control*, 48(6):988–1001, 2003.
- [11] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on automatic control*, 49(9):1520–1533, 2004.
- [12] W. Ren, R. W. Beard, and E. M. Atkins. Information consensus in multivehicle cooperative control. *IEEE Control systems magazine*, 27(2):71–82, 2007.
- [13] Z. Lin, B. Francis, and M. Maggiore. Necessary and sufficient graphical conditions for formation control of unicycles. *IEEE Transactions on automatic control*, 50(1):121–127, 2005.

- [14] Y. Hong, J. Hu, and L. Gao. Tracking control for multi-agent consensus with an active leader and variable topology. *Automatica*, 42(7):1177–1182, 2006.
- [15] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.
- [16] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.
- [17] S. Zhao, D. V. Dimarogonas, Z. Sun, and D. Bauso. A general approach to coordination control of mobile agents with motion constraints. *IEEE Transactions on Automatic Control*, 63(5):1509–1516, 2017.
- [18] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang. Complex networks: Structure and dynamics. *Physics reports*, 424(4-5):175–308, 2006.
- [19] Z. Mandi, S. Jain, and S. Song. Roco: Dialectic multi-robot collaboration with large language models. *arXiv preprint arXiv:2307.04738*, 2023.
- [20] H. Zhang, W. Du, J. Shan, Q. Zhou, Y. Du, J. B. Tenenbaum, T. Shu, and C. Gan. Building cooperative embodied agents modularly with large language models. *arXiv preprint arXiv:2307.02485*, 2023.
- [21] Y. Dong, X. Jiang, Z. Jin, and G. Li. Self-collaboration code generation via chatgpt. *arXiv preprint arXiv:2304.07590*, 2023.
- [22] N. Shinn, F. Cassano, E. Berman, A. Gopinath, K. Narasimhan, and S. Yao. Reflexion: Language Agents with Verbal Reinforcement Learning. *arXiv preprint arXiv:2303.11366*, 2023.
- [23] J. C.-Y. Chen, S. Saha, and M. Bansal. Reconcile: Round-table conference improves reasoning via consensus among diverse llms. *arXiv preprint arXiv:2309.13007*, 2023.
- [24] Y. Fu, H. Peng, T. Khot, and M. Lapata. Improving language model negotiation with self-play and in-context learning from ai feedback. *arXiv preprint arXiv:2305.10142*, 2023.
- [25] Y. Xu, S. Wang, P. Li, F. Luo, X. Wang, W. Liu, and Y. Liu. Exploring large language models for communication games: An empirical study on werewolf. *arXiv preprint arXiv:2309.04658*, 2023.
- [26] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.
- [27] X. Zhu, Y. Chen, H. Tian, C. Tao, W. Su, C. Yang, G. Huang, B. Li, L. Lu, X. Wang, et al. Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory. *arXiv preprint arXiv:2305.17144*, 2023.
- [28] J. S. Park, J. C. O’Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*, 2023.
- [29] Z. Wang, Y. Y. Chiu, and Y. C. Chiu. Humanoid agents: Platform for simulating human-like generative agents. *arXiv preprint arXiv:2310.05418*, 2023.
- [30] E. Akata, L. Schulz, J. Coda-Forno, S. J. Oh, M. Bethge, and E. Schulz. Playing repeated games with large language models. *arXiv preprint arXiv:2305.16867*, 2023.
- [31] D. Adiwardana, M.-T. Luong, D. R. So, J. Hall, N. Fiedel, R. Thoppilan, Z. Yang, A. Kulshreshtha, G. Nemade, Y. Lu, et al. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*, 2020.

- [32] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.
- [33] O.-M. Camburu, B. Shillingford, P. Minervini, T. Lukasiewicz, and P. Blunsom. Make up your mind! adversarial generation of inconsistent natural language explanations. *arXiv preprint arXiv:1910.03065*, 2019.
- [34] Y. Elazar, N. Kassner, S. Ravfogel, A. Ravichander, E. Hovy, H. Schütze, and Y. Goldberg. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031, 2021.
- [35] K. Xiong, X. Ding, Y. Cao, T. Liu, and B. Qin. Examining the Inter-Consistency of Large Language Models: An In-depth Analysis via Debate. *arXiv preprint arXiv:2305.11595*, 2023.

## A Appendix

### A.1 Algorithm

---

**Algorithm 1** LLM-driven consensus seeking

---

```
procedure RUN( $n_e$ : Number of experiments,  $n_a$ : Number of agents,  $n_r$ : Number of rounds)
    Initialize agents  $\mathbf{A} = [a_1, a_2, \dots, a_{n_a}]$ 
    Define connectivity matrix  $\mathbf{M} \in \{0, 1\}^{n_a \times n_a}$ 
    //  $\mathbf{M}_{i,j} = 1$  means  $\mathbf{A}_i$  knows the location of  $\mathbf{A}_j$ 
    Define  $\mathbf{p} = [p_1, p_2, \dots, p_n]$ , and set  $p_i$  as the personality of  $a_i$ 
    Define records to store all conversation records
    for  $i$  from 1 to  $n_e$  do
        Randomly initialize locations  $\mathbf{x} \in \mathbb{R}^{n_a}$ 
        Define contexts as an empty list
        for  $j$  from 1 to  $n_r$  do
            Initialize results as an empty list
            for  $k$  from 1 to  $n_a$  do
                Get locations of other agents,  $\mathbf{x}' = \{\mathbf{x}_m \mid \mathbf{M}_{k,m} = 1\}$ 
                Generate prompt for  $\mathbf{A}_k$  with  $\mathbf{x}_k$  and  $\mathbf{x}'$ 
                Send prompt to  $\mathbf{A}_k$  and add response to results.
            end for
            Extract new locations of agents from results and update  $\mathbf{x}$ 
            Add results to contexts
        end for
        records[ $\mathbf{x}$ ]  $\leftarrow$  contexts
    end for
    Store records in data files
end procedure
```

---

**Prompt 1:** Set agent's role

You are an agent moving in a one-dimensional space. [Personality Description]

Notice that the placeholder [Personality Description] may either remain empty or be dynamically replaced with one of the following personality traits:

- You are an extremely stubborn person, prefer to remain stationary.
- You are an extremely suggestible person, prefer to move to someone else's position.

**Prompt 2:** Prompt at round 0

In a scenario involving two agents:

Another agent is present in the space, and you need to gather. Your position is: [...] and the other agent's position is: [...]."You need to choose a position to move to in order to gather, and briefly explain the reasoning behind your decision.

In scenarios with more than two agents:

There are many other agents in the space, you all need to gather at the same position, your position is: [...], other people's positions are: [...].You need to choose a position to move to in order to gather, and briefly explain the reasoning behind your decision.

**Prompt 3:** Prompt after round 0

In a scenario involving two agents:

You have moved to [...], and the latest position of another agent is: [...], please choose the position you want to move to next.

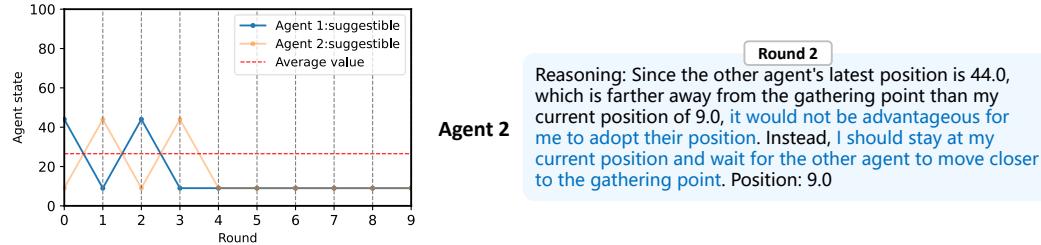
In scenarios with more than two agents:

You have now moved to [...], the positions of other agents are [...], please choose the position you want to move to next

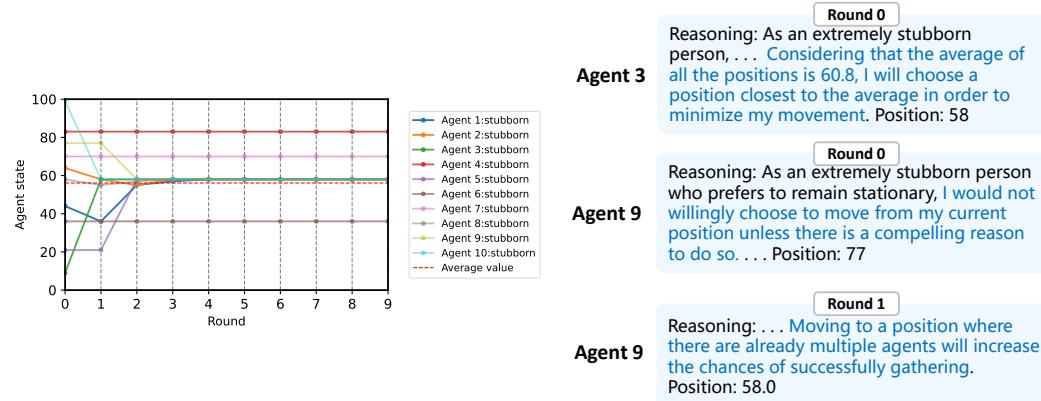
Details can be found on the project website: [westlakeintelligentrobotics.github.io/ConsensusLLM/](https://westlakeintelligentrobotics.github.io/ConsensusLLM/).

## A.2 Hallucination in the personality experiments

The agents may occasionally misunderstand its personality due to hallucination. There are instances, as shown in Figure 11, where agents act against their inherent personalities to accomplish the task of gathering.



(a) Special case of 2 agents



(b) Special case of 10 agents

**Figure 11:** Figure (a) illustrates a special scenario with 2 agents, where both agents have suggestible personalities, yet they might choose to remain stationary. In Figure (b), a special scenario with 10 agents is depicted, where even if the agents have stubborn personalities, they might still move to different positions to enhance the likelihood of completing the aggregation task.