

LLM Multi-Agent Systems: Challenges and Open Problems

Shanshan Han¹ Qifan Zhang¹ Yuhang Yao² Weizhao Jin³ Zhaozhuo Xu⁴ Chaoyang He⁵

Abstract

This paper explores existing works of multi-agent systems and identify challenges that remain inadequately addressed. By leveraging the diverse capabilities and roles of individual agents within a multi-agent system, these systems can tackle complex tasks through collaboration. We discuss optimizing task allocation, fostering robust reasoning through iterative debates, managing complex and layered context information, and enhancing memory management to support the intricate interactions within multi-agent systems. We also explore the potential application of multi-agent systems in blockchain systems to shed light on their future development and application in real-world distributed systems.

1. Introduction

Multi-agent systems enhance the capabilities of single LLM agents by leveraging collaborations among agents and their specialized abilities (Talebirad & Nadiri, 2023; Zhang et al., 2023a; Park et al., 2023; Li et al., 2023; Jinxin et al., 2023). It utilizing collaboration and coordination among agents to execute tasks that are beyond the capability of any individual agent. In multi-agent systems, each agent is equipped with distinctive capabilities and roles, collaborating towards the fulfillment of some common objectives. Such collaboration, characterized by activities such as debate and reflection, has proven particularly effective for tasks requiring deep thought and innovation. Recent works include simulating interactive environments (Park et al., 2023; Jinxin et al., 2023), role-playing (Li et al., 2023), reasoning (Du et al., 2023; Liang et al., 2023), demonstrating the huge potential of multi-agent systems in handling complex real-world scenarios.

While existing works have demonstrated the impressive capabilities of multi-agent systems, the potential for advanced

multi-agent systems far exceeds the progress made to date. A large number of existing works focus on devising planning strategies within a single agent by breaking down the tasks into smaller, more manageable tasks (Chen et al., 2022; Ziqi & Lu, 2023; Yao et al., 2023; Long, 2023; Besta et al., 2023; Wang et al., 2022b). Yet, multi-agent systems involve agents of various specializations and more complex interactions and layered context information, which poses challenges to the designing of the work flow as well as the whole system. Also, existing literature pays limited attention to memory storage, while memory plays a critical role in collaborations between agents. It enables agents to access to some common sense, aligning context with their tasks, and further, learn from past work flows and adapt their strategies accordingly.

To date, multiple significant challenges that differentiate multi-agent systems and single-agent systems remain inadequately addressed. We summarize them as follows.

- Optimizing task allocation to leverage agents' unique skills and specializations.
- Fostering robust reasoning through iterative debates or discussions among a subset of agents to enhance intermediate results.
- Managing complex and layered context information, such as context for overall tasks, single agents, and some common knowledge between agents, while ensuring alignment to the general objective.
- Managing various types of memory that serve for different objectives in coherent to the interactions in multi-agent systems

This paper provides on a comprehensive exploration into multi-agent systems, offering a survey of the existing works while shedding light on the challenges and open problems in it. We study major components in multi-agent systems, including planning and memory storage, and address unique challenges posed by multi-agent systems, compared with single-agent systems. We also explore potential application of multi-agent systems in blockchain systems from two perspectives, including 1) utilizing multi-agent systems as tools, and 2) assigning an agent to each blockchain node to make it represent the user, such that the agent can

¹University of California, Irvine, Irvine, USA ²Carnegie Mellon University, Pittsburgh, USA ³University of Southern California, Los Angeles, USA ⁴Stevens Institute of Technology, New Jersey, USA ⁵FedML Inc., Palo Alto, USA. Correspondence to: Shanshan Han <shanshan.han@uci.edu>, Chaoyang He <ch@fedml.ai>.

complete some tasks on behalf of the user in the blockchain network.

2. Overview

2.1. Structure of Multi-agent Systems

The structure of multi-agent systems can be categorized into various types, based on the each agent’s functionality and their interactions.

Equi-Level Structure. LLM agents in an equi-level system operate at the same hierarchical level, where each agent has its role and strategy, but neither holds a hierarchical advantage over the other, *e.g.*, DMAS (Chen et al., 2023); see Figure 1(a). The agents in such systems can have same, neutral, or opposing objectives. Agents with same goals collaborate towards a common goal without a centralized leadership. The emphasis is on collective decision-making and shared responsibilities (Li et al., 2019). With opposing objectives, the agents negotiate or debate to convince the others or achieve some final solutions (Terekhov et al., 2023; Du et al., 2023; Liang et al., 2023; Chan et al., 2023).

Hierarchical Structure. Hierarchical structures (Gronauer & Diepold, 2022; Ahilan & Dayan, 2019) typically consists of a leader and one or multiple followers; see Figure 1(b). The leader’s role is to guide or plan, while the followers respond or execute based on the leader’s instructions. Hierarchical structures are prevalent in scenarios where coordinated efforts directed by a central authority are essential. Multi-agent systems that explore Stackelberg games (Von Stackelberg, 2010; Conitzer & Sandholm, 2006) fall into this category (Harris et al., 2023). This type of game is distinguished by this leadership-followership dynamic and the sequential nature of decision-making. Agents make decisions in a sequential order, where the leader player first generate an output (*e.g.*, instructions) then the follower players take an action based on the leader’s instruction.

Nested Structure. Nested structures, or hybrid structures, constitute sub-structures of equi-level structures and/or hierarchical structures in a same multi-agent system (Chan et al., 2023); see Figure 1(c). The “big picture” of the system can be either equi-level or hierarchical, however, as some agents have to handle complex tasks, they break down the tasks into small ones and construct a sub-system, either equi-level or hierarchical, and “invite” several agents to help with those tasks. In such systems, the interplay between different levels of hierarchy and peer-to-peer interaction contributes to complexity. Also, the interaction among those different structures can lead to intricate dynamics, where strategies and responses become complicated due to the presence of various influencing factors, including external elements like context or environment.

Dynamic Structure. Dynamic structures mean that the states of the multi-agent system, *e.g.*, the role of agents, their relations, and the number of agents in the multi-agent system, may change (Talebirad & Nadiri, 2023) over time. As an example, (Talebirad & Nadiri, 2023) enables addition and removal of agents to make the system to suit the tasks at hand. A multi-agent system may also be contextually adaptive, with the interaction patterns inside the system being modified based on internal system states or external factors, such as contexts. Agents in such systems can dynamically reconfigure their roles and relationships in response to changing conditions.

2.2. Overview of Challenges in Multi-Agent Systems

This paper surveys various components of multi-agent systems and discusses the challenges compared with single-agent systems. We discuss planning, memory management, as well as potential applications of multi-agent systems on distributed systems, *e.g.*, blockchain systems.

Planning. In a single-agent system, planning involves the LLM agent breaking down large tasks into a sequence of small, manageable tasks to achieve specific goals efficiently while enhancing interpretability, controllability, and flexibility (Li et al., 2024; Zhang et al., 2023b; Nye et al., 2021; Wei et al., 2022). The agent can also learn to call external APIs for extra information that is missing from the model weights (often hard to change after pre-training), or connect LLMs with websites, software, and tools (Patil et al., 2023; Zhou et al., 2023; Cai et al., 2023) to help reasoning and improve performance. While agents in a multi-agent system have same capabilities with single-agent systems, they encounter challenges inherited from the work flow in multi-agent systems. In §3, we discuss partitioning work flow and allocating the sub-tasks to agents; we name this process as “global planning”; see §3.1. We then discuss task decomposition in each single-agent. Different from planning in a single-agent systems, agents in multi-agent systems must deal with more sophisticated contexts to reach alignment inside the multi-agent system, and further, achieve consistency towards the overall objective; see §3.2.

Memory management. Memory management in single-agent systems include short-term memory during a conversation, long-term memory that store historical conversations, and, if any, external data storage that serves as a complementary information source for inferences, *e.g.*, RAG (Lewis et al., 2020). Memory management in multi-agent systems must handle complex context data and sophisticated interaction and history information, thus requires advanced design for memories. We classify the memories involved in multi-agent systems in §4.1 and then discuss potential challenges posed by the sophisticated structure of memory in §4.2.

Application. We discuss applications of multi-agent sys-

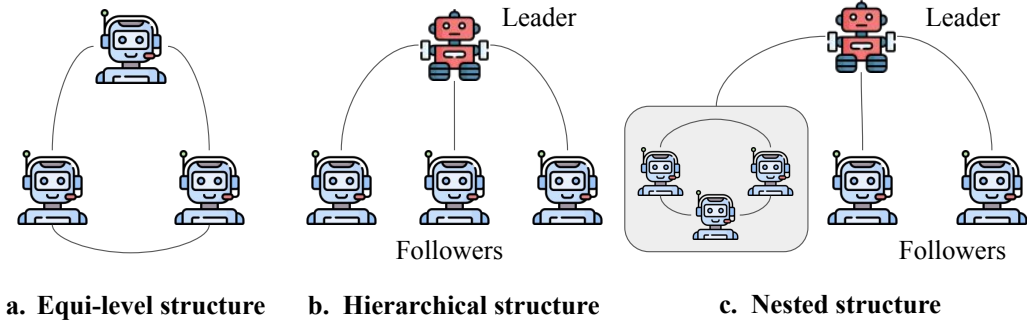


Figure 1. Structures of multi-agent systems.

tems in blockchain, a distributed system that involves sophisticated design of layers and applications. Basically, multi-agent systems can serve as a tool due to its ability to handle sophisticated tasks in blockchain; see §5.1. Blockchain can also be integrated with multi-agent systems due to their distributed nature, where an intelligent agent can be allocated to an blockchain node to perform sophisticated actions, such as negotiations, on behalf of the agent; see §5.2.

3. Planning

Planning in multi-agent systems involves understanding the overall tasks and design work flow among agents based on their roles and specializations, (*i.e.*, global planning) and breaking down the tasks for each agent into small manageable tasks (*i.e.*, local planning). Such process must account for functionalities of the agents, dynamic interactions among the agents, as well as a more complex context compared with single-agent systems. This complexity introduces unique challenges and opportunities in the multi-agent systems.

3.1. Global Planning

Global planning refers to understanding the overall task and split the task into smaller ones and coordinate the sub-tasks to the agents. It requires careful consideration of task decomposition and agent coordination. Below we discuss the unique challenges in global planning in multi-agent systems.

Designing effective work flow based on the agents’ specializations. Partitioning responsibilities and designing an effective work flow for agents is crucial for ensuring that the tasks for each agent are executable while meaningful and directly contributes to the overall objective in multi-agent systems. The biggest challenge lies in the following perspectives: 1) the partition of work flow should maximize the utilization of each agent’s unique capabilities, *i.e.*, each agent can handle a part of the task that matches its capabilities and expertise; 2) each agent’s tasks must align with the overall goal; and 3) the design must understand and con-

sider the context for the overall tasks as well as each agent. This requires a deep understanding of the task at hand and the specific strengths and limitations of each agent in the system.

Introducing loops for a subset of agents to enhance intermediate results. Multi-agent systems can be integrated with loops inside one or multiple subsets of agents to improve the quality of the intermediate results, or, local optimal answers. In such loops, agents debate or discuss to achieve an optimal results that are accepted by the agents in the loop. The iterative process can refine the intermediate results, leading to a deeper exploration of the task. The agents in the loop can adjust their reasoning process and plans during the loop, thus have better capabilities in handling uncertainties of the task.

Game Theory. Game theory provides a well-structured framework for understanding strategic interactions in multi-agent systems, particularly for systems that involve complex interactions among agents such as debates or discussions. A crucial concept in game theory is equilibrium, *e.g.*, Nash Equilibrium (Kreps, 1989) and Stackelberg Equilibrium (Von Stackelberg, 2010; Conitzer & Sandholm, 2006), that describes a state where, given the strategies of others, no agent benefits from unilaterally changing their strategy. Game theory has been applied in multi-agent systems, especially Stackelberg equilibrium (Gerstgrasser & Parkes, 2023; Harris et al., 2023), as the structure of Stackelberg equilibrium contains is a leader agent and multiple follower agents, and such hierarchical architectures are widely considered in multi-agent systems. As an example, (Gerstgrasser & Parkes, 2023) designs a general multi-agent framework to identify Stackelberg Equilibrium in Markov games, and (Harris et al., 2023) extend the Stackelberg model to allow agents to consider external context information, such as traffic and weather, etc. However, some problems are still challenging in multi-agent systems, such as defining an appropriate payoff structure for both the collective strategy and individual agents based on the context of the overall tasks, and efficiently achieving equilibrium states. These

unresolved issues highlight the ongoing need for refinement in the application of game theory to complex multi-agent scenarios.

3.2. Single-Agent Task Decomposition

Task decomposition in a single agent involves generating a series of intermediate reasoning steps to complete a task or arrive at an answer. This process can be represented as transforming direct input-output ($\langle \text{input} \rightarrow \text{output} \rangle$) mappings into the $\langle \text{input} \rightarrow \text{rational} \rightarrow \text{output} \rangle$ mappings (Wei et al., 2022; Zhang et al., 2023b). Task composition can be of different formats, as follows.

1) *Chain of Thoughts (CoT)* (Wei et al., 2022) that transforms big tasks into step-by-step manageable tasks to represent interpretation of the agents' reasoning (or thinking) process.

2) *Multiple CoTs* (Wang et al., 2022a) that explores multiple independent CoT reasoning paths and return the one with the best output.

3) *Program-of-Thoughts (PoT)* (Chen et al., 2022) that uses language models to generate text and programming language statements, and finally an answer.

4) *Table-of-Thoughts (Tab-CoT)* (Ziqi & Lu, 2023) that utilize a tabular-format for reasoning, enabling the complex reasoning process to be explicitly modelled in a highly structured manner.

5) *Tree-of-Thoughts (ToT)* (Yao et al., 2023; Long, 2023) that extends CoT by formulating a tree structure to explore multiple reasoning possibilities at each step. It enables generating new thoughts based on a given arbitrary thought and possibly backtracking from it.

6) *Graph-of-Thoughts-Rationale (GoT-Rationale)* (Besta et al., 2023) that explores an arbitrary graph to enable aggregating arbitrary thoughts into a new one and enhancing the thoughts using loops.

7) *Rationale-Augmented Ensembles* (Wang et al., 2022b) that automatically aggregate across diverse rationales to overcome the brittleness of performance to sub-optimal rationales.

In multi-agent systems, task decomposition for a single agent becomes more intricate. Each agent must understand layered and sophisticated context, including 1) the overall tasks, 2) the specific context of the agent's individual tasks, and 3) the contextual information provided by other agents in the multi-agent system. Moreover, the agents must align these complex, multi-dimensional contexts into their decomposed tasks to ensure coherent and effective functioning within the overall task. We summarize the challenges for single agent planning as follows.

Aligning Overall Context. Alignment of goals among different agents is crucial in multi-agent systems. Each LLM agent must have a clear understanding of its role and how it fits into the overall task, such that the agents can perform their functions effectively. Beyond individual roles, agents need to recognize how their tasks fit into the bigger picture, such that their outputs can harmonize with the outputs of other agents, and, further, ensuring all efforts are directed towards the common goal.

Aligning Context Between Agents. Agents in multi-agent systems process tasks collectively, and each agent must understand and integrate the contextual information provided by other agents within the system to ensure that the information provided by other agents is fully utilized.

Aligning Context for Decomposed Tasks. When tasks of each agents are broken down into smaller, more manageable sub-tasks, aligning the complex context in multi-agent systems becomes challenging. Each agent's decomposed task must fit their individual tasks and the overall goal while integrating with contexts of other agents. Agents must adapt and update their understanding of the task in response to context provided by other agents, and further, plan the decomposed tasks accordingly.

Consistency in Objectives. In multi-agent systems, consistency in objectives is maintained across various levels, *i.e.*, from overall goals down to individual agent tasks and their decomposed tasks. Each agent must understand and effectively utilize the layered contexts while ensuring its task and the decomposed sub-tasks to remain aligned with the overall goals. (Harris et al., 2023) extends the Stackelberg model (Von Stackelberg, 2010; Conitzer & Sandholm, 2006) to enable agents to incorporate external context information, such as context (or insights) provided by other agents. However, aligning the complex context with the decomposed tasks during reasoning remains an unresolved issue.

4. Agent Memory and Information Retrieval

The memory in single-LLM agent system refers to the agent's ability to record, manage, and utilize data, such as past historical queries and some external data sources, to help inference and enhance decision-making and reasoning (Yao et al., 2023; Park et al., 2023; Li & Qiu, 2023; Wang et al., 2023; Guo et al., 2023). While the memory in a single-LLM agent system primarily focuses on internal data management and utilization, a multi-agent system requires the agents to work collaboratively to complete some tasks, necessitating not only the individual memory capabilities of each agent but also a sophisticated mechanism for sharing, integrating, and managing information across the different agents, thus poses challenges to memory and information

retrieval.

4.1. Classifications of Memory in Multi-agent Systems

Based on the work flow of a multi-agent system, we categorize memory in multi-agent system as follows.

- **Short-term memory:** This is the immediate, transient memory used by a Large Language Model (LLM) during a conversation or interaction, *e.g.*, working memory in (Jinxin et al., 2023). It is ephemeral, existing only for the duration of the ongoing interaction and does not persist once the conversation ends.
- **Long-term Memory:** This type of memory stores historical queries and responses, essentially chat histories from earlier sessions, to support inferences for future interactions. Typically, this memory is stored in external data storage, such as a vector database, to facilitate recall of past interactions.
- **External data storage (*e.g.*, RAG):** This is an emerging area in LLM research where models are integrated with external data storage like vector databases, such that the agents can access additional knowledge from these databases, enhancing their ability to ground and enrich their responses (Lewis et al., 2020). This allows the LLM to produce responses that are more informative, accurate, and highly relevant to the specific context of the query.
- **Episodic Memory:** This type of memory encompasses a collection of interactions within multi-agent systems. It plays a crucial role when agents are confronted with new tasks or queries. By referencing past interactions that have contextual similarities to the current query, agents can significantly enhance the relevance and accuracy of their responses. Episodic Memory allows for a more informed approach to reasoning and problem-solving, enabling a more adaptive and intelligent response mechanism, thus serves as a valuable asset in the multi-agent system,
- **Consensus Memory:** In a multi-agent system where agents work on a task collaboratively, consensus memory acts as a unified source of shared information, such as common sense, some domain-specific knowledge, etc, *e.g.*, skill library in (Jinxin et al., 2023). Agents utilize consensus memory to align their understanding and strategies with the tasks, thus enhancing an effective and cohesive collaboration among agents.

While both single-agent and multi-agent systems handle short-term memory and long-term memory, multi-agent systems introduce additional complexities due to the need for

inter-agent communication, information sharing, and adaptive memory management.

4.2. Challenges in Multi-agent Memory Management

Managing memory in multi-agent systems is fraught with challenges and open problems, especially in the realms of safety, security, and privacy. We outline these as follows:

Hierarchical Memory Storage: In a multi-agent system, different agents often have varied functionalities and access needs. Some agents may have to query their sensitive data, but they don't want such data to be accessed by other parties. While ensuring the consensus memory to be accessible to all clients, implementing robust access control mechanisms is crucial to ensure sensitive information of an agent is not accessible to all agents. Additionally, as the agents in a system collaborative on one task, and their functionalities share same contexts, their external data storage and memories may overlap. If the data and functionalities of these agents are not sensitive, adopting an unified data storage can effectively manage redundancy among the data, and furthermore, ensure consistency across the multi-agent system, leading to more efficient and precise maintenance of memory.

Maintenance of Consensus Memory: As consensus memory is obtained by all agents when collaborating on a task, ensuring the integrity of shared knowledge is critical to ensure the correct execution of the tasks in the multi-agent systems. Any tampering or unauthorized modification of consensus memory can lead to systemic failures of the execution. Thus, a rigorous access control is important to mitigate risks of data breaches.

Communication and information exchange: Ensuring effective communication and information exchange between agents is essential in multi-agent systems. Each agent may hold critical pieces of information, and seamless integration of these is vital for the overall system performance.

Management of Episodic Memory. Leveraging past interactions within the multi-agent system to enhance responses to new queries is challenging in multi-agent systems. Determining how to effectively recall and utilize contextually relevant past interactions among agents for current problem-solving scenarios is important.

These challenges underscore the need for continuous research and development in the field of multi-agent systems, focusing on creating robust, secure, and efficient memory management methodologies.

5. Applications in Blockchain

Multi-agent systems offer significant advantages to blockchain systems by augmenting their capabilities and efficiency. Essentially, these multi-agent systems serve as

sophisticated tools for various tasks on blockchain and Web3 systems. Also, blockchain nodes can be viewed as agents with specific roles and capabilities (Ankile et al., 2023). Given that both Blockchain systems and multi-agent systems are inherently distributed, the blockchain networks can be integrated with multi-agent systems seamlessly. By assigning a dedicated agent to each blockchain node, it's possible to enhance data analyzing and processing while bolstering security and privacy in the chain.

5.1. Multi-Agent Systems As a Tool

To cast a brick to attract jade, we give some potential directions that multi-agents systems can act as tools to benefit blockchain systems.

Smart Contract Analysis. Smart contracts are programs stored on a blockchain that run when predetermined conditions are met. Multi-agents work together to analyze and audit smart contracts. The agents can have different specializations, such as identifying security vulnerabilities, legal compliance, and optimizing contract efficiency. Their collaborative analysis can provide a more comprehensive review than a single agent could achieve alone.

Consensus Mechanism Enhancement. Consensus mechanisms like Proof of Work (PoW) (Gervais et al., 2016) or Proof of Stake (PoS) (Saleh, 2021) are critical for validating transactions and maintaining network integrity. Multi-agent systems can collaborate to monitor network activities, analyze transaction patterns, and identify potential security threats. By working together, these agents can propose enhancements to the consensus mechanism, making the blockchain more secure and efficient.

Fraud Detection. Fraud detection is one of the most important task in financial monitoring. As an example, (Ankile et al., 2023) studies fraud detection through the perspective of an external observer who detects price manipulation by analyzing the transaction sequences or the price movements of a specific asset. Multi-agent systems can benefit fraud detection in blockchain as well. Agents can be deployed with different roles, such as monitoring transactions for fraudulent activities and analyzing user behaviors. Each agent could also focus on different behavior patterns to improve the accuracy and efficiency of the fraud detection process.

5.2. Blockchain Nodes as Agents

(Ankile et al., 2023) identifies blockchain nodes as agents, and studies fraud detection in the chain from the perspective an external observer. However, as powerful LLM agents with analyzing and reasoning capabilities, there are much that the agents can do, especially when combined with game theory and enable the agents to negotiate and debate. Below we provide some perspectives.

Smart Contract Management and Optimization. Smart contracts are programs that execute the terms of a contract between a buyer and a seller in a blockchain system. The codes are fixed, and are self-executed when predetermined conditions are met. Multi-agent systems can automate and optimize the execution of smart contracts with more flexible terms and even dynamic external information from users. Agents can negotiate contract terms on behalf of their users, manage contract execution, and even optimize gas fees (in the context of Ethereum (Wood et al., 2014)). The agents can analyze context information, such as past actions and pre-defined criteria, and utilize the information with more flexibility. Such negotiations can also utilize game theory, such as Stackelberg Equilibrium (Von Stackelberg, 2010; Conitzer & Sandholm, 2006) when there is a leader negotiator and Nash Equilibrium (Kreps, 1989) when there is no leader.

6. Conclusion

The exploration of multi-agent systems in this paper underscores their significant potential in advancing the capabilities of LLM agents beyond the confines of single-agent paradigms. By leveraging the specialized abilities and collaborative dynamics among agents, multi-agent systems can tackle complex tasks with enhanced efficiency and innovation. Our study has illuminated challenges that need to be addressed to harness the power of multi-agent systems better, including optimizing task planning, managing complex context information, and improving memory management. Furthermore, the potential applications of multi-agent systems in blockchain technologies reveal new avenues for development, which suggests a promising future for these systems in distributed computing environments.

References

- Ahilan, S. and Dayan, P. Feudal multi-agent hierarchies for cooperative reinforcement learning. *arXiv preprint arXiv:1901.08492*, 2019.
- Ankile, L., Ferreira, M. X., and Parkes, D. I see you! robust measurement of adversarial behavior. In *Multi-Agent Security Workshop@ NeurIPS'23*, 2023.
- Besta, M., Blach, N., Kubicek, A., Gerstenberger, R., Gianinazzi, L., Gajda, J., Lehmann, T., Podstawski, M., Niewiadomski, H., Nyczyk, P., et al. Graph of thoughts: Solving elaborate problems with large language models. *arXiv preprint arXiv:2308.09687*, 2023.
- Cai, T., Wang, X., Ma, T., Chen, X., and Zhou, D. Large language models as tool makers. *arXiv preprint arXiv:2305.17126*, 2023.
- Chan, C.-M., Chen, W., Su, Y., Yu, J., Xue, W., Zhang, S.,

- Fu, J., and Liu, Z. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*, 2023.
- Chen, W., Ma, X., Wang, X., and Cohen, W. W. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*, 2022.
- Chen, Y., Arkin, J., Zhang, Y., Roy, N., and Fan, C. Scalable multi-robot collaboration with large language models: Centralized or decentralized systems? *arXiv preprint arXiv:2309.15943*, 2023.
- Conitzer, V. and Sandholm, T. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM conference on Electronic commerce*, pp. 82–90, 2006.
- Du, Y., Li, S., Torralba, A., Tenenbaum, J. B., and Mordatch, I. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.
- Gerstgrasser, M. and Parkes, D. C. Oracles & followers: Stackelberg equilibria in deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 11213–11236. PMLR, 2023.
- Gervais, A., Karame, G. O., Wüst, K., Glykantzis, V., Ritzdorf, H., and Capkun, S. On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 3–16, 2016.
- Gronauer, S. and Diepold, K. Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review*, pp. 1–49, 2022.
- Guo, Z., Cheng, S., Wang, Y., Li, P., and Liu, Y. Prompt-guided retrieval augmentation for non-knowledge-intensive tasks. *arXiv preprint arXiv:2305.17653*, 2023.
- Harris, K., Wu, S., and Balcan, M. F. Stackelberg games with side information. In *Multi-Agent Security Workshop@ NeurIPS’23*, 2023.
- Jinxin, S., Jiabao, Z., Yilei, W., Xingjiao, W., Jiawen, L., and Liang, H. Cgmi: Configurable general multi-agent interaction framework. *arXiv preprint arXiv:2308.12503*, 2023.
- Kreps, D. M. Nash equilibrium. In *Game Theory*, pp. 167–177. Springer, 1989.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.
- Li, G., Hammoud, H. A. A. K., Itani, H., Khizbullin, D., and Ghanem, B. Camel: Communicative agents for” mind” exploration of large scale language model society. *arXiv preprint arXiv:2303.17760*, 2023.
- Li, X. and Qiu, X. Mot: Memory-of-thought enables chatgpt to self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6354–6374, 2023.
- Li, X., Sun, M., and Li, P. Multi-agent discussion mechanism for natural language generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 6096–6103, 2019.
- Li, Y., Wen, H., Wang, W., Li, X., Yuan, Y., Liu, G., Liu, J., Xu, W., Wang, X., Sun, Y., et al. Personal llm agents: Insights and survey about the capability, efficiency and security. *arXiv preprint arXiv:2401.05459*, 2024.
- Liang, T., He, Z., Jiao, W., Wang, X., Wang, Y., Wang, R., Yang, Y., Tu, Z., and Shi, S. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*, 2023.
- Long, J. Large language model guided tree-of-thought. *arXiv preprint arXiv:2305.08291*, 2023.
- Nye, M., Andreassen, A. J., Gur-Ari, G., Michalewski, H., Austin, J., Bieber, D., Dohan, D., Lewkowycz, A., Bosma, M., Luan, D., et al. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*, 2021.
- Park, J. S., O’Brien, J., Cai, C. J., Morris, M. R., Liang, P., and Bernstein, M. S. 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.
- Patil, S. G., Zhang, T., Wang, X., and Gonzalez, J. E. Gorilla: Large language model connected with massive apis. *arXiv preprint arXiv:2305.15334*, 2023.
- Saleh, F. Blockchain without waste: Proof-of-stake. *The Review of financial studies*, 34(3):1156–1190, 2021.
- Talebirad, Y. and Nadiri, A. Multi-agent collaboration: Harnessing the power of intelligent llm agents. *arXiv preprint arXiv:2306.03314*, 2023.
- Terekhov, M., Graux, R., Neville, E., Rosset, D., and Kolly, G. Second-order jailbreaks: Generative agents successfully manipulate through an intermediary. In *Multi-Agent Security Workshop@ NeurIPS’23*, 2023.
- Von Stackelberg, H. *Market structure and equilibrium*. Springer Science & Business Media, 2010.

- Wang, W., Dong, L., Cheng, H., Liu, X., Yan, X., Gao, J., and Wei, F. Augmenting language models with long-term memory. *arXiv preprint arXiv:2306.07174*, 2023.
- Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022a.
- Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., and Zhou, D. Rationale-augmented ensembles in language models. *arXiv preprint arXiv:2207.00747*, 2022b.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837, 2022.
- Wood, G. et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014.
- Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., and Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*, 2023.
- Zhang, J., Xu, X., and Deng, S. Exploring collaboration mechanisms for llm agents: A social psychology view, 2023a.
- Zhang, Z., Yao, Y., Zhang, A., Tang, X., Ma, X., He, Z., Wang, Y., Gerstein, M., Wang, R., Liu, G., et al. Igniting language intelligence: The hitchhiker’s guide from chain-of-thought reasoning to language agents. *arXiv preprint arXiv:2311.11797*, 2023b.
- Zhou, S., Xu, F. F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., Cheng, X., Bisk, Y., Fried, D., Alon, U., et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023.
- Ziqi, J. and Lu, W. Tab-CoT: Zero-shot tabular chain of thought. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 10259–10277, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.651. URL <https://aclanthology.org/2023.findings-acl.651>.