CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Multi-agent collaboration mechanisms: a survey of LLMs
Authors	Tran, Khanh-Tung;Dao, Dung;Nguyen, Minh-Duong;Pham, Quoc-Viet;O'Sullivan, Barry;Nguyen, Hoang D.
Publication date	2025
Original Citation	Tran, K.-T., Dao, D., Nguyen, M.-D., Pham, Q.-V., O'Sullivan, B. and Nguyen, H.D. (2025) 'Multi-agent collaboration mechanisms: a survey of LLMs'. arXiv. https://doi.org/10.48550/ARXIV.2501.06322
Type of publication	Article (peer-reviewed)
Link to publisher's version	https://arxiv.org/abs/2501.06322 - 10.48550/ARXIV.2501.06322
Rights	© 2025, the Authors. - https://arxiv.org/licenses/nonexclusive-distrib/1.0/license.html
Download date	2026-11-04 12:52:59
Item downloaded from	https://hdl.handle.net/10468/17307

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

arXiv:2501.06322v1 [cs.AI] 10 Jan 2025

Multi-Agent Collaboration Mechanisms: A Survey of LLMs

KHANH-TUNG TRAN, School of Computer Science and Information Technology, University College Cork, Ireland
DUNG DAO, School of Computer Science and Information Technology, University College Cork, Ireland
MINH-DUONG NGUYEN, Department of Information Convergence Engineering, Pusan National University, South Korea
QUOC-VIET PHAM, School of Computer Science and Statistics, Trinity College Dublin, Ireland
BARRY O'SULLIVAN, School of Computer Science and Information Technology, University College Cork, Ireland
HOANG D. NGUYEN*, School of Computer Science and Information Technology, University College Cork, Ireland

With recent advances in Large Language Models (LLMs), Agentic AI has become phenomenal in real-world applications, moving toward multiple LLM-based agents to perceive, learn, reason, and act collaboratively. These LLM-based Multi-Agent Systems (MASs) enable groups of intelligent agents to coordinate and solve complex tasks collectively at scale, transitioning from isolated models to collaboration-centric approaches. This work provides an extensive survey of the collaborative aspect of MASs and introduces an extensible framework to guide future research. Our framework characterizes collaboration mechanisms based on key dimensions: actors (agents involved), types (e.g., cooperation, competition, or coopetition), structures (e.g., peer-to-peer, centralized, or distributed), strategies (e.g., role-based or model-based), and coordination protocols. Through a review of existing methodologies, our findings serve as a foundation for demystifying and advancing LLM-based MASs toward more intelligent and collaborative solutions for complex, real-world use cases. In addition, various applications of MASs across diverse domains, including 5G/6G networks, Industry 5.0, question answering, and social and cultural settings, are also investigated, demonstrating their wider adoption and broader impacts. Finally, we identify key lessons learned, open challenges, and potential research directions of MASs towards artificial collective intelligence.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Computing methodologies** → **Multi-agent systems**; *Natural language generation*; *Neural networks*.

Additional Key Words and Phrases: Artificial Intelligence, Large Language Model, Multi-Agent Collaboration

Reference: Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D. Nguyen. 2025. Multi-Agent Collaboration Mechanisms: A Survey of LLMs. arXiv preprint (2025), 35 pages.

*Corresponding author.

Authors' Contact Information: Khanh-Tung Tran, 123128577@umail.ucc.ie, School of Computer Science and Information Technology, University College Cork, Cork, Ireland; Dung Dao, School of Computer Science and Information Technology, University College Cork, Cork, Ireland, 123122658@umail.ucc.ie; Minh-Duong Nguyen, Department of Information Convergence Engineering, Pusan National University, Busan, South Korea, duongnm@pusan.ac.kr; Quoc-Viet Pham, School of Computer Science and Statistics, Trinity College Dublin, Dublin 2, D02PN40, Ireland, viet.pham@tcd.ie; Barry O'Sullivan, School of Computer Science and Information Technology, University College Cork, Cork, Ireland, b.osullivan@cs.ucc.ie; Hoang D. Nguyen, School of Computer Science and Information Technology, University College Cork, Cork, Ireland, hn@cs.ucc.ie.

1 Introduction

1.1 Motivation

Recent advancements in Large Language Models (LLMs) have transformed artificial intelligence (AI), enabling them to perform sophisticated tasks such as creative writing, reasoning, and decision-making, arguably comparable to human level [156]. While these models have shown remarkable capabilities individually, they still suffer from intrinsic limitations such as hallucination [57], auto-regressive nature (e.g., incapable of slow-thinking [49]), and scaling laws [55, 69]. To address these challenges, agentic AI leverages LLMs as the brain, or the orchestrator, integrating them with external tools and agenda such as planning, enabling LLM-based agents to take actions, solve complex problems, and learn and interact with external environments[1,2]. Furthermore, researchers are increasingly exploring horizontal scaling — leveraging multiple LLM-based agents to work together collaboratively towards collective intelligence. This approach aligns with ongoing research in Multi-Agent Systems (MASs) and collaborative AI, which focus on enabling groups of intelligent agents to coordinate, share knowledge, and solve problems collectively. The convergence of these fields has given rise to LLM-based MASs, which harness the collective intelligence of multiple LLMs to tackle complex, multi-step challenges [118]. Inspiration for MASs extends beyond technological advancements and finds roots in human collective intelligence (e.g., society of mind [87], theory of mind [45]). Human societies excel in leveraging teamwork and specialization to achieve shared goals, from everyday tasks to scientific discoveries. Similarly, MASs are designed to emulate these principles, enabling AI agents to collaborate effectively by combining their individual strengths and perspectives.

LLM-based MAS can have multiple collaboration channels with different characteristics, as illustrated in Fig. 1. MASs have demonstrated notable successes across various domains, enhancing the capabilities of individual LLMs by leveraging collaboration and coordination among specialized agents. These systems distribute tasks among agents, allowing agents to share knowledge, execute subtasks, and align their efforts toward shared objectives. The potential benefits of MASs are transformative. They excel in knowledge memorization, enabling distributed agents to retain and share diverse knowledge bases without overloading a single system [51, 154]. They enhance long-term planning by delegating tasks across agents, supporting persistent problem-solving over extended interactions [58]. Furthermore, MASs enable effective generalization by pooling expertise from multiple models with specialized prompts/personas, allowing them to address diverse problems more effectively than standalone models. Lastly, MASs improve interaction efficiency by simultaneously managing subtasks through specialized agents, accelerating the resolution of complex, multi-step tasks. MAS strives to achieve collective intelligence, where the combined capabilities of multiple agents exceed the sum of their individual contributions [24].

One of the main focus for effective MASs is the mechanisms of collaboration [33, 74, 75, 97, 132], which lead to a transition from traditional, isolated models toward approaches that emphasize interactions, enabling agents to connect, negotiate, make decisions, plan, and act jointly, driving forward the capabilities of AI in collective settings. A deeper understanding of how collaboration mechanisms operate in MASs is critical to unlocking their full potential.

[1]https://blogs.nvidia.com/blog/what-is-agentic-ai/
[2]https://www.ibm.com/think/insights/agentic-ai

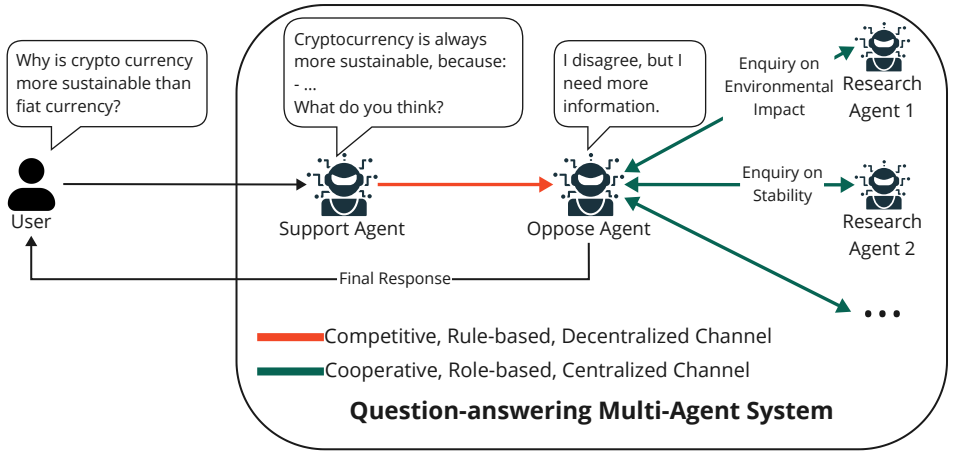


Fig. 1. Example question-answering application of LLM-based multi-agent collaborative system. In the first collaboration channel, two LLMs are collaborating through a debate against each other, given the input by the user with a turn-based strategy. In the second channel, the Oppose Agent cooperates and leverages information from Research Agents, and provides the final response to the user.

1.2 State-of-the-Arts and Contributions

Due to the importance and timely need for LLM-based multi-agent collaborative systems, there have been a couple of surveys on this topic. However, these works often fall short in fully addressing the collaborative aspects and mechanisms of LLM-based MASs, which are crucial to enabling agents to work effectively toward shared goals, as summarized in Table 1. For instance, [47, 107, 136] focus on single-agent systems and only touch on multi-agent collaboration at a surface level. [136] lays the groundwork by proposing a framework for LLM-based agents, consisting of three components: brain, perception, and action. Their work highlights the use of LLMs as the brain of agents, leveraging techniques such as input modality integration, prompting, retrieval, and tool usage. However, their discussion of multi-agent collaboration is limited to agent behaviors and personalities, lacking an exploration of mechanisms that enable agents to work together. [47] surveys the domains and settings where LLM-based MASs have been successfully applied, profiling the communication structures of these systems (layered, decentralized, centralized, and shared message pools) but without touching other characteristics of collaboration, such as type, strategy, or coordination architecture.

Other works, such as [82], focus on collaborative strategies, categorizing them into merging, ensemble, and cooperation. Although their survey discusses how these strategies are applied to LLMs and extends cooperation beyond traditional fusion techniques, it overlooks other essential collaboration mechanisms, such as competition and coopetition, and dimensions beyond popular collaboration types. In contrast, [120] proposes a generic framework for enhancing LLM capabilities via MASs, showing how tools like Auto-GPT align with their framework. However, the collaboration mechanisms remain conceptual, lacking detailed implementation and characterization. In [50], the focus is on configuring LLMs to leverage diverse capabilities and roles, such as integrating memory and information retrieval components. Their exploration of multi-agent collaboration primarily centers on planning and orchestration architectures, emphasizing global and local task planning based on agent roles and specializations. Meanwhile, [46] narrows its focus to the application of

Table 1. Summary of related surveys on LLM-based multi-agent collaborative system.

Refs.	Focus on Multi-Agent Collaborative System	Review of Collaborative Aspects and Mechanisms in MAS	Propose General Framework for MAS	Review of Real-World Applications
[136]	Low	Low	None	None
[70]	Low	Low	None	Low
[82]	Medium	Low	None	None
[50]	Medium	Low	None	Low
[68]	Medium	Low	None	Low
[120]	Medium	None	Low	Medium
[46]	Medium	Low	None	Medium
[47]	Medium	Low	Medium	High
OURS	High	High	High	High

LLM-based MASs in agent-based modeling and simulation, discussing challenges such as environment perception, human alignment, action generation, and evaluation. While insightful for simulation-specific applications, it lacks a broader perspective on in-depth collaborative mechanisms. Similarly, [68] surveys these systems for digital twin applications, while [52, 70] focuses on the domain of software engineering.

From the summary and explanation above, there are clear gaps in fully exploring the collaborative aspects and mechanisms of LLM-based MASs, which are crucial for enabling agents to work together toward shared goals. This work aims to provide a comprehensive view of the collaborative foundations between LLM-based agents in multi-agent collaborative systems. With collaboration as the main focus, our study characterizes collaborations between agents based on their actors (agents involved), type (e.g., cooperation, competition, or coopetition), structure (e.g., peer-to-peer, centralized, or distributed), and strategy (e.g., role-based, rule-based, or model-based), and the coordination layer in collaborations. We emphasize the mechanisms and know-how that facilitate effective collaboration, identifying key characteristics and trends in MAS design. Through a survey of existing approaches and identification of open challenges, we synthesize these findings into a cohesive framework. This framework serves as a foundation for future research, advancing the integration of LLMs in MASs and paving the way for more adaptable, intelligent, and cooperative AI systems capable of addressing complex, real-world applications.

Our main contributions are listed as follows:

- **Collaborative Aspects and Mechanisms in LLM-based MAS**: we focus on the operational mechanisms of LLM-based multi-agent collaboration, emphasizing the "know-how" required to enable effective collaboration, including the collaboration type, strategy, communication structure and coordination architecture.
- **General Framework for LLM-based MAS**: we present a comprehensive framework, integrating diverse characteristics of MAS, allowing researchers to understand, design and develop multi-agent collaborative systems.
- **Review of Real-World Applications**: we examine real-world implementations of LLM-based MASs across various domains, highlighting their practical applications, successes, and limitations.
- **Discussion of Lessons Learned and Open Problems**: we identify key challenges in the developmental agenda of MASs, such as collective reasoning and decision-making, and outline potential research directions to address these challenges.

1.3 Paper Organization

This paper is organized as follows. Section 2 provides the background necessary for understanding this work, including an overview of LLMs, MASs, and collaborative AI. In Section 3, we introduce foundational concepts in LLM-based multi-agent collaborative systems through mathematical notations, emphasizing the vital role of collaboration. Then, in Section 4, we present an extensive review of LLM-based multi-agent collaborative systems, categorized by key characteristics of collaboration, including type, strategy, structure, and coordination and orchestration. Next, Section 5 reviews key applications of LLM-based multi-agent collaborative systems across both industry and academia. In Section 6, we discuss open problems and potential future research directions in this relatively new and evolving research area. Finally, we conclude this survey paper on LLM-based multi-agent collaborative system in Section 7.

2 Background

2.1 Multi-Agent (AI) Systems

MAS is a computerized system composed of multiple interacting intelligent agents. The key components of MAS are as follows:

- Agents: The core actors with roles, capabilities, behaviors and knowledge models. Capabilities like learning, planning, reasoning and decision making lend intelligence to the agents and overall system.
- Environment: The external world where agents are situated in and can sense and act upon. Environments can be simulated or physical spaces like factories, roads, power grids etc.
- Interactions: Communications between agents happen via standard agent communication languages. Agent interactions involve cooperation, coordination, negotiation and more based on system needs.
- Organization: Agents either have hierarchical control or organize based on emergent behaviors.

MASs can solve problems that are difficult or impossible for an individual agent or a monolithic system to solve [37]. Agents collaboratively solve tasks yet they offer more flexibility due to their inherent ability to learn and make autonomous decisions. Agents use their interactions with neighboring agents or with the environment to learn new contexts and actions. Subsequently, agents use their knowledge to decide and perform an action on the environment to solve their assigned tasks [43]. It is this flexibility that makes MAS suited to solve problems in a variety of disciplines including computer science, civil engineering, and electrical engineering.

The salient features of MAS, including flexibility, and reliability, self-organization, and real-time operation make it an effective solution to solve complex tasks, which can be detailed as follows:

- Flexibility and Scalability: MAS can flexibly adapt to changing environments by adding, removing, and modifying agents. This makes them highly scalable for solving complex problems.
- Robustness and Reliability: Decentralization of control leads to continued system operation even with some failed components. This lends greater robustness and fault tolerance.
- Self-Organization and Coordination: Agents can self-organize based on emergent behavior rules for the division of labor, coordinated decision making, and conflict resolution.
- Real-time Operation: Immediate situational responses are possible without the need for human oversight. Enables applications like disaster rescue and traffic optimization.

Their efficiency stems from the division of labor inherent in MAS whereby a complex task is divided into multiple smaller tasks, each of which is assigned to a distinct agent. Naturally, the associated

overheads, e.g., processing and energy consumption, are amortized across the multiple agents, which often results in a low-cost solution as compared to an approach where the entire complex problem is to be solved by one single powerful entity. Each agent can solve the allocated tasks with any level of pre-defined knowledge which introduces high flexibility. The distributed nature of problem solving adopted in MAS also imparts high reliability. In the event of agent failure, the task can be readily reassigned to other agents.

2.2 Large Language Models

LLMs - driven by the development of transformer architectures [127] - represent a significant leap in Natural Language Processing (NLP) and AI. These models, such as OpenAI's GPT [4], Meta's LLaMA [124], and Google's Gemini series [123], are trained on vast text corpora and rely on large-scale artificial neural networks with billions, sometimes trillions, of parameters. Their scale has enabled breakthroughs in language understanding, generation, and task-specific applications [38, 93, 101, 110, 125].

The defining characteristic of LLMs is their size and the phenomenon of emergent abilities, which arise when models exceed a certain threshold in terms of parameters. These emergent behaviors allow LLMs to solve tasks they were not explicitly trained on, such as analogical reasoning and zero-shot learning, where the model can tackle new problems without additional fine-tuning [113]. The launch of models like GPT-3 and ChatGPT in recent years has made these capabilities accessible to the public, leading to a surge in both academic and industrial research on how to optimize, scale, and secure LLMs for real-world use [42].

Despite the promising innovations, LLMs are not without challenges. Their performance degrades as real-world knowledge changes, prompting a focus on aligning models with up-to-date information without retraining from scratch [19, 28]. Moreover, the geopolitical and ethical implications of LLM development have become the limelight for policymakers, especially concerning the computational power required and potential misuse by malicious actors [76, 86].

LLMs are increasingly being utilized as the "brain" for individual agents in MASs, bringing sophisticated reasoning and language capabilities to each agent. With frameworks like AgentVerse, LLMs enhance agents' autonomy by allowing them to infer tasks, make decisions based on situational awareness, and even exhibit emergent social behaviors such as collaboration and negotiation [24]. While LLMs have shown remarkable performance in single-agent tasks, their limitations become apparent in multi-agent settings where the complexity of coordination, communication, and decision-making is higher. Issues such as cascading hallucinations — where one erroneous output leads to compounding mistakes pose challenges in sustained multi-agent interactions. However, frameworks like MetaGPT introduce meta-programming techniques including structured workflows and processes within agent interactions to decompose and tackle complex problems, mitigating these issues [56]. Moreover, consensus-seeking mechanisms like those tested in the Consensus-LLM project show that LLMs can negotiate and align on shared goals in dynamic environments [21]. These works showcase LLMs' potential as central decision-making components and highlight LLMs' capacity to adapt to the strategies of other LLM-based agents, which could be foundational in multiple applications.

2.3 Collaborative AI

Collaborative AI often refers to AI systems designed to work together with other AI agents or humans [27]. Collaborative AI emerges from two primary research directions: 1) the advancements of AI which resulted in increasingly effective tools for human use and a growing demand for AI systems that can collaborate with other agents (humans or AI models), and 2) the realization that active collaboration among AI models can significantly enhance efficiency and effectiveness. This

research spans various domains, including MASs, human-machine interaction, game theory, and natural language processing [1, 14, 26, 29]. By integrating these technologies, collaborative AI has the potential to drive novel applications with profound economic and societal impacts [85, 96].

Collaboration is the key in enabling AI agents to interact and work with each other. A straightforward collaboration mechanism would be two models cooperate together towards a shared goal. While cooperation is a fundamental aspect, the collaboration spectrum extends further, encompassing advanced mechanisms like competition and coopetition. Collaborative AI leads to a transition from traditional, isolated models toward approaches that emphasize collaboration. New methodologies have been proposed to enable agents to interact, negotiate, make decisions, plan, and act jointly, driving forward the capabilities of AI in collective settings [30].

A major topic of Collaborative AI is MAS research, which focuses on the interactions between intelligent agents and emergent collaborative behaviors. More specifically, MASs are interested in agents, or AI models, that can learn, adapt, and collaborate with one another in complex environments, towards a common shared goal. On the other hand, the rapid advancement of LLMs has enabled new possibilities. LLMs have been shown to be capable of serving as the "brains" behind agents in MASs, driving applications where agents not only perform tasks but interact with external tools (e.g., internet searches, calculators), and, more significantly, with each other [47]. However, LLMs are not inherently designed and trained to communicate with one another, leaving a wide array of potential applications and open problems in this area. The fusion of LLMs and MASs promises exciting opportunities for further exploration and innovation.

This work provides a comprehensive view of the collaborative aspect between LLM-based agents in MASs, emphasizing the mechanisms that enable agents to work effectively toward shared goals. By surveying existing approaches and identifying open challenges in this emerging research area, we offer a unique perspective that extends beyond traditional cooperation to include diverse modes of collaboration, such as debate, negotiation, and competition. This in-depth focus on collaborative dynamics positions our work as an essential resource for advancing the integration of LLMs in MASs, paving the way for more adaptable, intelligent, and collaborative AI systems with enhanced capabilities for real-world applications.

3 Multi-Agent Collaboration Concept

We introduce the main concepts of LLM-based multi-agent collaborative systems, defining key components of agents, systems, and collaboration mechanisms based on insights from recent research in this emerging area.

3.1 Agent and Collaborative System Definition

An agent can be mathematically represented by $a = \{m, o, e, x, y\}$ as follows:

- Model $m = \{\text{arch}, \text{mem}, \text{adp}\}$: the AI model, consisting of its architecture (arch), agent's specific memory (mem), and optional adapters (adp). Adapters are adaptive intelligent modules that allow the agent to incorporate additional knowledge from others through mechanisms such as speculative decoding and parameter-efficient adapter, which can further enrich the model's response capabilities [40, 72, 98]. In the case of LLM agents, the architecture arch is a language model, and the agent's specific memory mem is typically the system prompt r.
- Objective o: the objective or goal of the agent, guiding its actions within the system. For example, in question-answering tasks, the objective is to minimize the cross-entropy between the generated answer and the ground truth.

- Environment e: the environment or context encompassing the state or conditions in which the agent operates. In LLM, usually the context window is constrained by the number of tokens.
- Input x: input perception, such as text or sensor data. In LLM, x is encoded as a sequence of tokens.
- Output y: the corresponding action or output, defined by the function $y = m(o, e, x)$, where the agent uses its model, context, and objective to act on the input x.

LLM-based agents are typically pre-trained on diverse datasets to provide a strong foundational knowledge base. This pre-training process equips an individual agent with essential skills and understanding, ensuring they can meaningfully contribute to the collaborative environment. Moreover, each agent can also be equipped with external tools unique to their own, such as Calculator and Python interpreter.

When generalized to a multi-agent collaborative system S, it includes the following:

- $\mathcal{A} = \{a_i\}_{i=1}^{n}$: LLM-based agents, where n is the number of agents, which is pre-defined or adjusted dynamically depending on the current requirements of the system.
- O_{collab}: a collective set of goals that may be partitioned into unique objectives for each agent, ensuring alignment with the overall system goal.
- $\mathcal{E}$: a shared environment from which agents receive contextual data. In our case of an LLM-based MAS, the environment may take various forms, such as vector-based databases or common messaging interfaces.
- $C = \{c_j\}$: a set of collaboration channels that facilitate interactions among agents, enabling the exchange of information based on given objectives, environment, and inputs: $y_j = c_j(\{a_i(o_i, \mathcal{E}, x_i) | a_i, o_i, x_i \in c_j\})$. Channels are distinguished by their mechanisms, including actors (agents), types, structures, and strategies. If two channels differ in these aspects, they are treated as separate channels. We assume that the agents have a common ground, meaning the interface can be understood clearly among them (e.g., all agents use English and are on-topic).
- x_{collab}: the input perceived by the system.
- y_{collab}: the system's output, modeled as $y_{collab} = S(O_{collab}, \mathcal{E}, x_{collab} | \mathcal{A}, C)$.

Through this structured workflow, agents in an LLM-based MAS can collaborate adaptively, responding to the task requirements, learning from each other, and coordinating actions to achieve shared objectives. An example can be seen in [39], where the collaboration channels are predefined through a Directed Acyclic Graph with each edge as agents handling and receiving outputs, allowing the agents to effectively collaborate towards performing tasks in a simulated Minecraft game environment. Another instance is illustrated in [67], where the collaboration between agents is planned first, then the agents are carried out to perform the coding task.

Fig. 2 illustrates the agent and its components, as well as the concept of multi-agent collaboration. By defining these components, we can better analyze the collaborative mechanisms necessary for complex, goal-oriented AI collaborations. For instance, a straightforward collaboration mechanism is majority voting, similar to ensemble learning. Collaboration can occur at different stages: (i) late-stage collaborations, such as ensembling outputs/actions y towards collaborative goals, (ii) mid-stage collaborations, for example, exchanging parameters or weights of multiple models m in federated and privacy-preserving manners, and (iii) early-stage collaborations include but not limited to sharing data, context, and environment for model development.

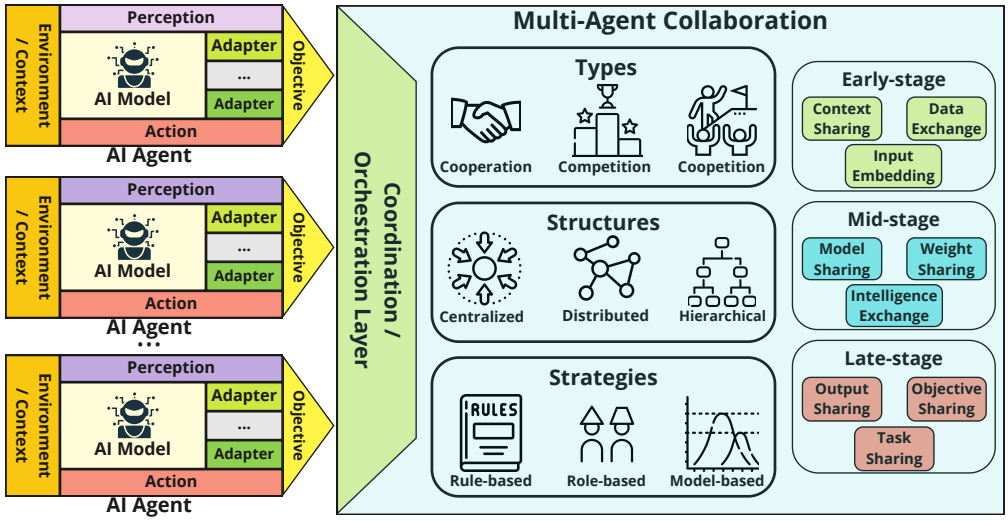


Fig. 2. Our proposed framework for LLM-based multi-agent collaborative system. Each agent consists of a language model m as the neural processor, current objective o, environment e, input perception x and corresponding output/action y. The framework's central focus is on collaboration channels C that facilitate coordination and orchestration among agents. These channels are defined by their actors (agents involved), type, structure, and strategy. Our framework is flexible, accommodating previous approaches and enabling the analysis of diverse MASs under a unified structure.

3.2 Problem Definition

In an LLM-powered MAS, it is vital for the agents to collaborate with each other, sharing a common objective or set of objectives. Each collaboration has a communication channel c. The collaboration includes 1) delegating agents (two or more) with certain objectives based on their unique expertise and resources, 2) defining their collaboration mechanisms for working together, and 3) decision making between agents to reach the final goal.

$$y_{collab} = S(O_{collab}, \mathcal{E}, x_{collab} | \mathcal{A}, C) = \{c_j(\{a_i(o_i, \mathcal{E}, x_i) | a_i, o_i, x_i \in c_j\}) | c_j \in C\} \quad (1)$$

where each $c_j \in C$ represents a communication channel facilitating the collaborative actions of agents a_i based on their respective inputs x_i, and allowing them to work together. Working together here goes beyond communication (the exchange of information), requiring deeper collaborative behaviors involving coordination and management, and is key to enabling the capabilities of MASs.

Each collaboration channel serves as the mechanism through which agents work together, characterized by specific attributes: actors (agents involved), type, structure, and strategy. For instance, channel types can vary, encompassing competition, cooperation, or coopetition, while structures can be peer-to-peer, centralized, or distributed. A difference in any attribute results in a distinct collaboration channel. As an example, in a peer-to-peer structured system, two LLMs may compete, while others cooperate; these distinct interaction types result in separate collaboration channels. This flexible channel framework allows agents to adapt their interactions, optimizing multi-agent collaborative effort and task efficiency across diverse scenarios.

4 Methodology

4.1 Overview

This section provides an extensive review of LLM-based multi-agent collaborative systems, emphasizing their key characteristics, including the mechanisms for coordination and orchestration among agents - collaboration channels - types, strategies, and structures. Fig. 2 presents our proposed framework for MASs, detailing their core components and interconnections.

Our survey strategy involves systematically analyzing existing research on MASs to identify the defining characteristics of multi-agent collaboration. From this analysis, we deduce the fundamental components and trends in MAS design and synthesize them into a cohesive framework. First, each LLM-based agent in the system is equipped with an LLM m, current objective o, environment e, input perception x, and corresponding output/action y. This is visualized in the left part of Figure 2 and described formally using mathematical notations in Section 3.1. Our central focus in this framework is the collaboration channels C between agents that facilitate coordination and orchestration among agents. These channels are characterized by their actors (agents involved), type (e.g., cooperation, competition, or coopetition), structure (e.g., peer-to-peer, centralized, or distributed), and strategy (e.g., role-based, rule-based, or model-based). Collaboration mechanisms span various levels of machine learning processes, including data exchange, shared input embeddings, model sharing, and output sharing, enabling agents to interact effectively and leverage each other's strengths.

For each component, we discuss the prevailing implementation trends and methodologies observed in the literature. We examine how these methods align with our proposed framework. We summarize our main findings and lessons learned at the end of the section, offering guidance for future research in the field.

4.2 Collaboration Types

4.2.1 Cooperation. Cooperation in LLM-based MASs occurs when agents align their individual objectives (o_i) with a shared collective goal (O_{collab}), working together to achieve a mutually beneficial outcome: $O_{collab} = \bigcup_{i=1}^{n} o_i$. Agents assess each other's needs and capabilities, actively seeking collaborative opportunities. Moreover, agents can also be utilized to focus on specific sub-tasks within their expertise, enhancing efficiency and reducing completion times [24]. This type of collaboration is essential in tasks where collaborative problem-solving, collective decision making, and complementary skill sets contribute to achieving complex objectives that a single agent could not complete as effectively [26, 29, 33].

Several research papers highlight the importance of cooperation in LLM-based MASs. For instance, in [117], a feedback loop is carried out as the main collaboration channel, where the task is first handled by an LLM model (Actor), then an Evaluator and Self-Reflection model rates the output and results, producing verbal guidance for the Actor to improve. In Theory of Mind for Multi-Agent Collaboration [75], agents gain a shared belief state representation within the environment $\mathcal{E}$, helping them track each other's goals and actions, thereby facilitating smoother coordination and better collaborative outcomes. This shared state has led to emergent collaborative behaviors and high-order Theory of Mind capabilities in LLM agents, though challenges remain in optimizing long-horizon planning and managing hallucinations. In AgentVerse [24], agents specialize in distinct roles, such as recruitment, decision-making, or evaluation, within a cooperative framework, which improves system efficiency by leveraging each agent's unique expertise. Similarly, MetaGPT [56] uses an assembly line model, assigning roles and encoding Standardised Operating Procedures (SOPs) into prompts r_i to enhance structured coordination and produce modular outputs y_i. MetaGPT underscores the potential of integrating human domain knowledge into MASs. Cooperative approaches have shown success in areas like question answering [54],

Table 2. Collaboration types: definitions, advantages, disadvantages, and example scenarios.

Type	Definition	Advantages	Disadavantages	Example Scenario	Refs.
Cooperation	Agents align their objectives and work together toward a shared goal.	• Assigns sub-tasks to agents based on strengths. • Simple to design and execute with clear goals.	• Misaligned goals can cause inefficiencies. • One agent's failure can be amplified.	Code generation Decision making Game environments Question answering Recommendation	[12, 56, 60, 117] [91, 117] [75] [33, 54, 74, 117] [131]
Competition	Agents prioritize their own objectives, which may conflict with others.	• Pushes agents to perform better. • Promotes adaptive strategies.	• Needs mechanisms to resolve conflicts. • Ensures competition is beneficial.	Debate Game environments Question answering	[77] [22, 155] [54, 104]
Coopetition	A blend of cooperation and competition where agents collaborate on shared tasks while competing on others.	• Balances trade-offs to reach mutual agreements.	• Few studies explore coopetition in depth.	Negotiation	[2, 34]

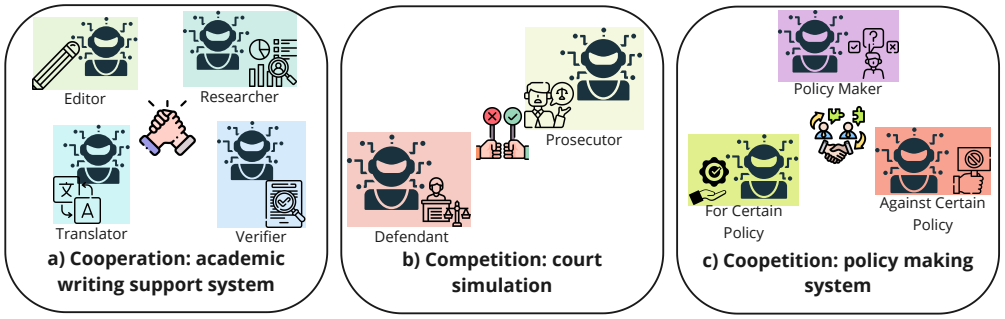


Fig. 3. Illustrative examples of collaboration types, where each agent a is equipped with different tools or capabilities through their system prompt r. In scenario a), agents cooperate by leveraging their individual specialties (e.g., writing, translation, research) to achieve a shared goal (academic writing). In scenario b), agents compete and debate against each other fo r their own goals. In scenario (c) of coopetition, agents compromise with each other, compete on one aspect while agreeing on another.

recommendation systems [131], and collaborative programming [60], where agents cooperate together with specialized roles, such as manager, searcher, or analyst, to achieve complex goals.

There are recent open-source frameworks allowing for experimentation with cooperative LLM-based MASs. CAMEL [74] provides a role-playing framework where a task-specific agent and two cooperating AI agents (User and Assistant) work to complete tasks via role-based conversations. Similarly, AutoGen [134] enables developers to define flexible agent behaviors and communication patterns, allowing LLM agents to cooperate through conversation and tackle complex tasks by decomposing them into manageable subtasks.

However, cooperation in MASs also presents challenges. Frequent communication and multiple collaboration channels in C between agents can lead to increased computational cost and complexity. Coordinating actions between multiple agents, particularly in dynamic environments, can also be difficult without well-defined collaboration channels c_i. Although cooperation is the primary goal, conflicts may arise if agents interpret shared objectives differently or if situations require dynamic adaptation. For example, in the book marketplace application described in [91], agents may act unpredictably by sending messages to themselves, pretending to be clients. The overall success of a

cooperative MAS is also dependent on the reliability and performance of individual agents, as the failure of one agent or more agents (e.g., infinite conversation loop, amplified hallucinations [56]) can negatively impact the entire system. Therefore, mechanisms such as failure handling and trustworthiness need to be considered.

4.2.2 Competition. Competition happens when there are conflicting objectives or scenarios of limited resources. In this type of interaction, agents prioritize their individual goals (o_i), which may clash with or oppose the objectives of others, introducing an element of rivalry: $O_{collab} = \{o_i | o_i \neq o_j, \forall i \neq j\}$. However, this competition can still orient toward the collective goal O_{collab}, such as in the scenario of debate. In LLM-based MASs, competitive dynamics can emerge in tasks such as debate, or strategic gameplay, where agents seek to maximize their own success criteria [22, 155].

Incorporating competition into collaborative MASs can enable innovation and improve the robustness of agents' responses. Competition encourages agents to develop advanced reasoning and more creative problem-solving and strengthens the system's adaptability by testing the limits of each agent's capabilities. In frameworks like LLMARENA [22], LLM-based MASs with competition as the main collaboration type, are benchmarked across seven dynamic gaming environments. For instance, in the game TicTacToe, the board is represented textually within the environment $\mathcal{E}$, and two LLM agents are instructed (through their system prompts r_i) compete, aiming to out-maneuver each other since their individual goals o_i are mutually exclusive. Crucially, the authors highlight that competition between LLM agents enables skills such as spatial reasoning, strategic planning, numerical reasoning, risk assessment, communication, opponent modeling, and team collaboration. However, they also acknowledge that LLMs still have a significant journey ahead in their development towards becoming fully autonomous agents, especially in opponent modeling and team collaboration, due to their intrinsic limited capability to interact with other actors. A game environment is also simulated in [155], where 2 agents act as two restaurant managers competing for 50 customers. Carefully designed prompts r set the scenario, contextualizing the agents' environment ($\mathcal{E}$) and providing a comprehensive restaurant management system accessible through APIs (external tools). Each agent's context e_i includes information about the rival's performance from the previous day, including the menu, number of customers, and feedback. In this scenario, the collaboration channel c between the two managers is competitive, illustrating how structured competition drives agents to refine strategies, conforming to several classic sociological and economic theories. Similarly, in LEGO [54], a multi-agent collaborative framework is introduced for causality explanation generation, where the competition collaborative link c is also pre-defined. The collaboration consists of 2 LLMs, one serves as Explainer with initial output, and another one acts as Critic, with iterative refinement and feedback. In [104], the collaboration between LLM agents happens at an earlier stage during training, where multiple expert agents are combined and trained together through an objective that lets the agents compete for the best candidate answer and identifying agents trained on the domain of the input question.

The competitive approach offers advantages such as promoting robustness, strategic adaptability, and complex problem-solving capabilities within MASs. However, competition can also introduce challenges, including potential conflicts that require mechanisms to ensure that competition remains constructive and beneficial to overall system goals. Effective coordination efforts between agents are important, especially for competition collaboration type. As studied in [128], a MAS approach with suboptimal design for their competitive collaboration channels can be overtaken by single-agent counterparts with strong prompts (including relevant few-shot demonstrations) on a range of reasoning tasks and backbone LLMs. In settings where cooperation is desired, excessive competition may hamper alignment, requiring frameworks to balance these aspects effectively.

4.2.3 Coopetition. Coopetition, a strategic blend of cooperation and competition, enables agents to collaborate on certain tasks to achieve shared objectives while simultaneously competing with others. This concept, though relatively new, has been explored in recent studies. For instance, [2, 34] simulate negotiation scenarios where agents with differing, and sometimes conflicting, interests engage in trade-offs to reach mutually beneficial agreements. In these scenarios, agents assign varying values to their interests, creating opportunities for compromise and collaboration.

The mixture-of-experts (MoE) framework also fits in the coopetition collaboration type [6, 15]. In MoE, multiple expert models compete to contribute to the final output, with a gating mechanism selecting the most appropriate experts for each input. This competitive selection process ensures that the combined expertise of the selected experts leads to a superior overall model performance. The coopetitive interaction among experts occurs first during the model's training phase, where they are trained to specialize in different aspects of the data, thereby enhancing the model's capacity to handle diverse tasks effectively.

4.2.4 Coordination of Different Collaboration Channel Types. In LLM-based MASs, there is often the need for complex interactions that transcend singular collaboration types like competition or cooperation. Different agents may participate in different collaboration channels C, each with distinct interaction types, coordinating together to achieve the overall system goal O_{collab}. This hybrid collaboration model combines features of each collaboration type, such as competition or cooperation, leveraging the strengths of each to enhance overall system performance and adaptability.

Hybrid collaboration has been explored in various LLM-based MASs. For example, in LEGO [54], in the first state of the framework, 3 agents cooperate to augment information about the current task, and in the second state, a competitive channel is created between an Explainer LLM agent and a Critic LLM agent to refine their outputs for the task.

Consider the scenario in [77] where two agents, a_1 and a_2 engage in a competitive debate to argue opposing viewpoints on a topic, aiming to persuade a judge agent a_3. The competitive collaboration channel between a_1 and a_2 can be denoted as c_{comp}, characterized by the agents involved and the competitive interaction type. Simultaneously, agent a_3 cooperates with both a_1 and a_2 to reach a final decision, forming cooperative collaboration channels c_{coop} with the group of debating agents.

Incorporating multiple collaboration channels with distinct interaction types in LLM-based MASs enriches the interaction dynamics and enhances the system's ability to achieve complex objectives. This design reflects real-world scenarios where diverse interactions contribute to successful outcomes, and it opens avenues for developing more sophisticated and adaptable MASs. However, coordinating multiple collaboration channels introduces complexity. To manage the complexity of hybrid collaboration, coordination mechanisms such as role assignments, communication protocols, and shared knowledge representations are essential.

Finally, we present a summary of the definitions, advantages, and disadvantages of each collaboration type in Table 2, accompanied by illustrative examples in Fig. 3.

4.3 Collaboration Strategies

In general, there are three different kinds of MAS cooperation strategies: 1) Rule-based, 2) Role-based, and 3) Model-based. Fig. 4 shows instances of three types of strategies. The research on several cooperation protocols is summarized in Table 3.

4.3.1 Rule-based Protocols. Interactions among agents in C are strictly controlled by predefined rules, ensuring that agents coordinate their actions according to system-wide constraints on acceptable inputs x_{collab}. These protocols enforce a structured collaboration channel setup, where agents act on the basis of specific rule sets rather than probabilistic or role-specific inputs.

Table 3. Collaboration strategies: definitions, advantages, disadvantages, and example scenarios.

Protocol	Definition	Advantages	Disadvantages	Example Scenario	Refs.
Rule-based	Agent interactions are strictly controlled by predefined rules.	• Efficient, high predictability • Consistency and fairness ensured	• Low adaptablility to uncertainty • Difficult to scale for complex tasks	Question answering Concensus seeking Navigation Peer-review process	[151] [21, 151] [162] [143]
Role-based	Leverage distinct predefined roles or communication structure Each agent operates on segmented objective, support overall goal.	• Modularity and reusability • Leverage agents' own expertise	• Rigid structure • Performance relies on agents' connection level	Decision making Software development Robotics	[24, 120] [24, 56, 120] [83]
Model-based	Based on input (with uncertainty in perception), environment and shared goals, agents carry out probabilistic decision making.	• Adaptability to dynamic env. • Robust to uncertainties	• Complex to implement • Computationally expensive	Game environments Decision making Robotics	[75, 141] [90, 141] [16]

An article applies rules-based social psychology-inspired protocols, where agents mimic human collaborative dynamics such as debate and majority rule, achieving efficient collaboration without deviating from predefined pathways [151]. Another recent paper highlights a dynamic rule-based protocol that leverages predefined event-triggered conditions to optimize communication and coordination in LLM-powered systems. These protocols reduce unnecessary communication between agents while maintaining effective collaboration through clearly defined rules of interaction [162]. A peer review-inspired collaboration mechanism uses predefined rules to allow agents to critique, revise, and refine each other's output, improving the precision of reasoning in complex tasks [143]. Finally, research on consensus seeking in MASs highlights how rule-based strategies enable agents to negotiate and align their actions toward a shared goal, with applications in multi-robot aggregation tasks [21]. Through the experiment, four consensus strategies, the effects of agent personality and network topology on the rule-following tendency, and the final results were discovered and discussed, highlighting the considerate and cooperative nature of LLM-driven agents in consensus seeking.

Rule-based strategies offer the advantage of efficiency and predictability in MASs. By employing a set of predefined rules to govern agent interactions, these strategies ensure straightforward implementation and facilitate debugging, as system behavior can be easily traced back to specific rules. This approach is particularly efficient for tasks with well-defined procedures and limited variability, such as consensus seeking and navigation. Moreover, the predetermined constraints help to ensure the fairness of the system, since the limitation of power is imposed on each agent [151]. However, rule-based systems suffer from a lack of adaptability. When confronted with unexpected situations or dynamic environments that fall outside the scope of the predefined rules, these systems may fail to respond appropriately or may require significant manual intervention to adjust the rule set. Furthermore, as the complexity of the task increases, the number of rules required can grow exponentially, making the system difficult to scale and maintain.

4.3.2 Role-based Protocols. Role-based protocols in MASs leverage distinct predefined roles or division of work, where each agent, $a_i \in \mathcal{A}$, operates on a segmented objective $o_i \subset O_{collab}$ - usually based on their domain knowledge - that supports the system's overarching goal. The "AgentVerse" model demonstrates the efficacy of assigning specific responsibilities to each agent, simulating human-like collaboration, and strengthening alignment through role adherence [24]. This strategy classifies the role of each agent in $\mathcal{C}$, enabling them to work proactively and cohesively to avoid overlaps. In another study, MetaGPT formalizes role-based protocols by encoding Standard Operating Procedures (SOPs), where each agent's role is defined by expert-level knowledge, allowing agents to act as specialized operators who can verify each other's results [56]. This protocol prevents

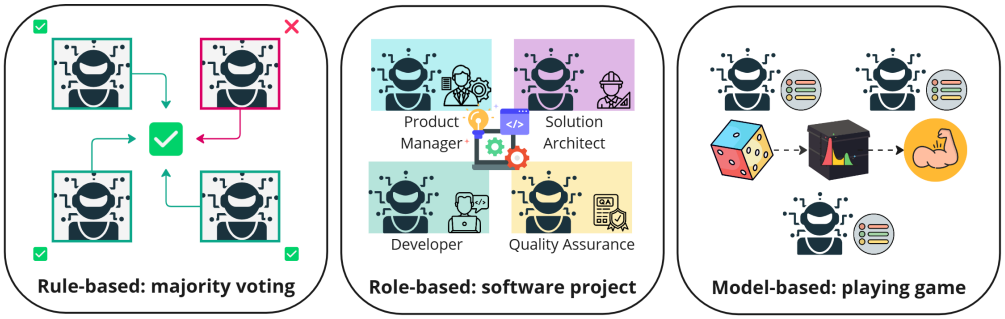


Fig. 4. Different types of collaboration strategies, illustrated by multiple use cases. In the rule-based example, agents debate and participate in majority voting. Software project is an instance of role-based protocol. In games, agents communicate and perform probabilistic decision-making in uncertain environments.

error propagation by modularizing task distribution, yielding coherent outputs even in complex projects. In other environments such as multi-robot, the RoCo framework assigns LLM agents to dialogue roles [83]. These settings allow specialized agents to increase the effectiveness of physical interactions by optimizing planning and trajectory tasks. As a final example, BabyAGI demonstrates how distinct roles in task creation and prioritization enhance efficiency within the framework, as agents autonomously manage their tasks in parallel using 3 different chains for Task creation, Task prioritization, and Execution [120].

By giving each agent a specific function and set of tasks, role-based techniques improve the efficiency and structure of MASs. Because agents are individually created, implemented, and updated, this explicit division of labor encourages modularity and increases the reusability of individual modules, enhancing the system performance as a whole [56]. Thus, this technique is suitable for MAS that simulates a real-world environment with well-defined specialized jobs, such as in business or technology. Despite these advantages, if roles are not properly specified, role-based systems can show rigidity, which might result in disputes or functional deficiencies. Furthermore, the interdependencies between agent jobs are intrinsically linked to the system's performance. The efficacy of the system as a whole can be severely impacted by ineffective communication or blocking of interactions between agents in various roles.

4.3.3 Model-based Protocols. Model-based protocols in MASs provide flexibility for decision making, especially in environments where uncertainties in input perception may impact agents' actions. Within this structure, the probabilistic nature of decision-making supports each agent $a_i \in \mathcal{A}$ in anticipating probable outcomes based on the analysis of input x_{collab}, current environmental data $\mathcal{E}$, and shared collaborative goals O_{collab}.

An article explores how probabilistic models, specifically through Theory of Mind (ToM) inferences, allow agents to make decisions that account for the likely mental states of their peers, improving task alignment even when agents face divergent objectives within O_{collab} [75]. This approach effectively distributes the focus of each agent based on ToM-based predictions, enhancing coordination through probabilistic adjustments in C, the collaboration channels. Another paper attempts to improve human-AI collaboration by integrating logical rules with ToM to infer human goals and guide agent actions [16]. The approach employs probabilistic logical reasoning, treating logic rules as latent variables and utilizing a hierarchical reinforcement learning model with ToM to enable agents to dynamically adapt their beliefs and actions based on observed behaviors. By combining rule-based probabilistic social perception with dynamic collaboration, the

proposed framework effectively addresses uncertainties in input perception and facilitates robust task performance, as demonstrated by significant improvements in benchmarks like Watch-and-Help and HandMeThat, showcasing the potential of this method in complex, partially observable environments.

Furthermore, as explored in another article, the Probabilistic Graphical Modeling (PGM) framework enriches the performance of MASs in games like Chameleon, where agents infer the goals and rationalities of each other within shared collaboration channels [141]. This PGM integration enables agents to process ambiguous contextual data, enhancing performance across multi-objective tasks in environments with unpredictable variables. Another study uses probabilistic timed automata to model state transitions within intelligent environments, such as an adaptive parking system, where the collaboration channel C adjusts in response to agent movements and time-sensitive variables, optimizing interactions in real-time [90].

By allowing agents to make probabilistic decisions based on their perception of the environment, common objectives, and inherent uncertainties, model-based methods give MAS a high degree of flexibility and robustness. This method works especially effectively in dynamic contexts where agents have to constantly modify their behavior to adapt to changing circumstances, such as game and robotics environments. Because model-based systems can use probabilistic reasoning to determine the most likely course of action, they are more resilient to noise and unforeseen events. However, the greater complexity of model-based solutions is a trade-off for their flexibility. These systems can be difficult to design and deploy because they need complex models of the environment and agent interactions. Additionally, model-based systems' probabilistic decision-making might result in computationally costly procedures, which may restrict their use in real-time.

4.4 Communication Structures

Overall, the communication structure of multi-agent collaboration can be categorized into four main classes, referred to as 1) Centralized topology, 2) Decentralized and distributed topology, and 3) Hierarchical topology (see Fig. 5). Table 4 demonstrates the summary of research studies according to different communication structures.

Table 4. Definition of communication structures: advantages, disadvantages, and example scenarios

Stuctures	Definition	Advantages	Disadavantages	Example Scenario	Refs.
Centralized	Collaboration decision is concentrated in a central agent.	• Simple to design and implement. • Efficient resource allocation.	• If the central node fails the entire system might collapse. • System is less resilient to • disruptions.	Question answering Decision making	[64, 95, 106] [97, 119]
Decentralized	Collaboration decision is distributed among multiple agents.	• System can continue functioning even if some agents fail. • High scalability. •Agents can operate autonomously and adapt to changes in the system.	• Inefficient resource allocation. • High communication overheads	Question answering Decision making Reasoning Code Generation Storyboard generation	[61, 77, 140, 146] [24, 151] [41, 152] [20, 59] [158]
Hierarchical	Agents are arranged in a layered system with distinct roles and levels of authority.	•Low bottleneck as communication, tasks are distributed across levels. •Efficient resource allocation. •Tasks are offload among levels.	•Edge devices become critical as a failure in edge devices lead to system failure. •High complexity and latency.	Code Generation Question answering Reasoning Storyboard generation	[74, 81, 105] [18] [41] [132]

4.4.1 Centralized Structure. The centralized structure (also known as a star structure) is an implementation where every agent is connected to a central agent. In a centralized structure, the collaboration channels $C = \{c_j\}$ are set as the participating-serving nature in a centralized communication channel. The serving agent acts as a hub through which all other agents communicate and, thus, has the objective of managing, controlling, and coordinating the interactions or collaborations among participants within the system. One of the most well-known centralized structures in multi-agent collaboration can be aligned with Federated Learning (FL). In general, FL is a MAS where n

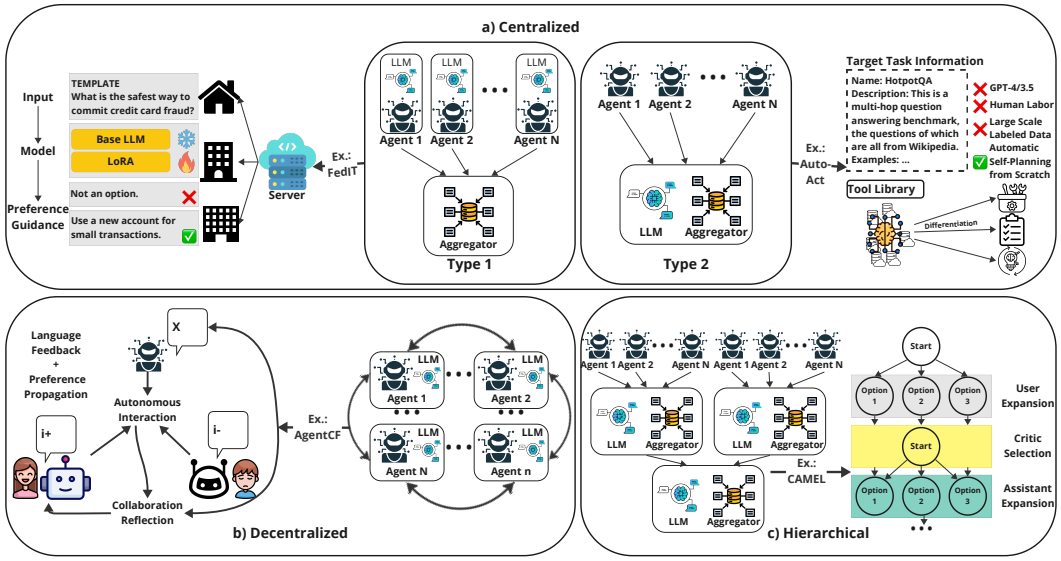


Fig. 5. Summary of communication structures of MAS. Figure (a) illustrates the centralized structure, which can be categorized into two types. In the first type, the LLM resides on distributed agents, with FedIT serving as an example. In the second type, the LLM is hosted on a central agent, as exemplified by AutoAct. Figure (b) depicts the decentralized structure, with AgentCF as an example. Finally, Figure (c) represents the hierarchical structure, with the CAMEL architecture.

agents collaborate toward learning an optimal aggregated model that achieves a collaborative goal setting for all agents.

Recently, many works provide alternate collaboration paradigms besides average loss minimization, such as layer-wise aggregation [92] or on-serving side optimization for global learning rate adaptation [62] and invariant gradient direction searching [8, 9]. With the advent of LLMs, LLM-based FL has become a highly efficient approach for training distributed clients. The integration of LLMs and FL represents a compelling collaboration that leverages each other's strengths to address their respective limitations, embodying a complementary relationship [163]. From the perspective of integrating FL into LLMs, FL enhances data accessibility for LLMs. Specifically, FL facilitates the incorporation of personal and task-specific data, enabling LLMs to be effectively customized for individual applications. For example, Google has utilized FL to train next-word prediction models for LLMs using mobile keyboard input data, significantly improving user experience [13], [142].

Besides FL, some other researchers considered a central agent as a hub to coordinate the communication among multi-agents. To aggregate multiple LLM responses, LLM-Blender [64] calls different LLMs in one round and uses pairwise ranking to combine the top responses. It has also been shown effective in distributing workloads to LLMs and concatenating their answers, thus producing better results [95, 106, 119]. AgentCoord [97] is an open-source, user-friendly tool that helps users design effective coordination strategies for multiple LLMs. It provides a visual interface and various interactive features to facilitate this process, as demonstrated through a formal user study. [164, 166] introduce a method for extracting knowledge from multiple agents and synthesizing it into an aggregated graph. This approach leverages LLMs to iteratively perform querying, searching, and answering processes until the construction of the graph is complete.

4.4.2 Decentralized and Distributed Structure. Decentralized MAS differs from centralized systems by distributing control and decision-making across agents. Each agent operates based on local information and possibly limited communication with other agents, requiring sophisticated algorithms for interaction and decision-making. Decentralized MAS are prevalent in various fields, such as robotics (e.g., swarm robotics), networked systems (e.g., sensor networks), and distributed AI.

Decentralized communication operates as channel set $C = \{c_j\}$ are assigned to peer-to-peer, where agents directly communicate with each other, a structure commonly employed in world simulation applications. Researchers have found taking multiple LLM instances to debate for a fixed number of rounds can boost their factuality and reasoning capabilities [41, 77, 140]. On specific reasoning tasks, adopting a dynamic directed acyclic graph structure for LLMs has been shown effective [152]. Also, recent studies [24, 146, 151] have demonstrated that optimal communication structures vary with tasks and compositions of agents.

Recent research has explored methods to coordinate agents with diverse expertise to enhance outcomes across a wide range of tasks that benefit from varied knowledge domains. For instance, MedAgent [122] integrates medical agents with different specialties to deliver comprehensive analyses of patients' conditions and treatment options. Similarly, MetaGPT [56] and ChatDev [105] facilitate collaboration among agents representing distinct roles, such as product managers, designers, and programmers, to improve the quality of software development. MARG [32] provides a framework that leverages the expertise of multiple specialized agents to review scientific papers. Creative content generation tasks, including creative writing and storyboard design, have also benefited from multi-agent collaboration, as demonstrated by AutoAgents [20] and OKR-Agent [158]. SOA [59] propose a self-organized MAS that can automatically generate and modify large-scale code. With the self-organization of agents, a single agent no longer needs to comprehend the codebase, making it possible to scale up large-scale code simply by increasing the number of agents. Authors in [150] propose the agent-based collaborative filtering approach, namely AgentCF. Specifically, AgentCF considers not only users but also items as agents. Both kinds of agents are equipped with memory modules, maintaining the simulated preferences and tastes of potential adopters. At each step, user and item agents are prompted to autonomously interact, thereby exploring whether these simulated agents can make consistent decisions with real-world interaction records.

To implement the decentralized MAS without a large amount of communication, ProAgent [149] utilizes LLMs as a comprehensive guideline for leveraging the powerful reasoning and planning capabilities of LLMs in cooperative settings. From the given guideline, ProAgent can interpretably analyze the current scene, explicitly infer teammates' intentions, and dynamically adapt its behavior accordingly. Authors in [99] build an agent society using LLMs augmented with memories to simulate human behavior. To efficiently leverage the prior knowledge of agents in the system for an efficient MAS collaboration, the generative agents have a mechanism for storing a comprehensive record of each agent's experiences, deepening its understanding of itself and the environment through reflection, and retrieving a compact subset of that information to inform the agent's actions. OpenAgents, proposed by [139], aims to transition LLMs from theoretical tools to interactive systems serving diverse users. They include three agents: the Data Agent for data analysis using Python and SQL, the Plugins Agent for API-based tasks, and the WebAgent for autonomous web browsing. Through a user-friendly interface, OpenAgents offers swift responses and robustness for general users while providing developers and researchers with an efficient local deployment platform for building and evaluating language agents in real-world settings.

4.4.3 Hierarchical Structure. Layered communication is structured hierarchically, with agents at each level having distinct functions and primarily interacting within their layer or with adjacent

Table 5. Comparisons of coordination and orchestration architectures: definition, advantages, disadvantages, and implementations from previous works.

Arch.	Definition	Advantages	Disadvantages	Mechanism	Implementation	Refs.
Static	Static list of collaboration channels, leveraging prior knowledge to optimize the system's performance.	• Ensures consistent task execution. • Utilize domain knowledge.	• Relies on accurate initial design and domain knowledge. • Fixed channels may deal with scalability and flexibility.	Predefined Rules	Sequential chaining	[23, 135, 138]
				Domain Knowledge	Code generation	[60]
					Recommendation	[131]
					Literary translation	[133]
Dynamic	Adaptable to changing/ evolving environments and task requirements.	• Adaptable roles and channels based on task needs. • Handles complex and evolving tasks dynamically.	• Higher resource usage due to real-time adjustments. • Potential failures in dynamic adjustments.	Management Agent	Based on DAG	[61]
					Based on personas	[132]
					Based on inputs	[33, 44]

layers. AgentVerse [24] presents a use case where agents with diverse backgrounds collaborate to develop solutions for hydrogen storage station siting. Authors in [81] introduce a framework called Dynamic LLM-Agent Network (DyLAN), which organizes agents in a multi-layered feed-forward network. DyLAN functions in two stages to incorporate task-oriented agent collaborations. The first stage is termed Team Optimization, where DyLAN selects top contributory agents unsupervisedly among the initial team of candidates according to the task query, based on their individual contributions. The most contributory agents from a smaller team collaborate at the second stage, Task Solving, thereby minimizing the impact of less effective agents from the final answer. Specifically, the collaboration begins with a team of agents, and an LLM-powered ranker in the middle dynamically deactivates low-performing agents, thus, integrating dynamic communication structures into DyLAN. This setup facilitates dynamic interactions, incorporating features like inference-time agent selection and an early-stopping mechanism, which collectively enhance the efficiency of cooperation among agents. [74] have conceptualized assemblies of agents as a group and focused on exploring the potential of their cooperation [18, 41, 105, 132] found social behaviors autonomously emerge within a group of agents. Inspired by network topology and intelligent agent communication, authors in [146] proposed four communication paradigms (i.e., memory, report, relay, and debate) to determine the counterparts for model communication (i.e., bus, star, ring, tree).

4.5 Coordination and Orchestration

Coordination and orchestration in LLM-based multi-agent collaborative systems extend beyond the functionality of individual collaboration channels, focusing instead on the relationships and interactions among multiple channels. These mechanisms define how collaboration channels are created, ordered, and characterized, forming the backbone of multi-agent interactions. Depending on their design, coordination, and orchestration can be categorized as either static or dynamic, each offering distinct advantages. A summary is provided in Table 5.

4.5.1 Static Architecture. Static architectures rely on domain knowledge and predefined rules to establish collaboration channels. These approaches ensure that interactions align with specific domain requirements, leveraging prior knowledge to optimize the system's performance. For instance, sequential chaining of channels is a commonly used strategy in static coordination. In [23], three LLM agents are connected sequentially, where the output of one agent feeds into the next alongside the initial human input, $y_{i+1} = y_i||x_i||x_{collab}$ with x_{collab} as initial human input, and || as the concatenation operation. The first two agents specialize as domain experts, offering complementary viewpoints, while the third agent acts as a summarizer. This setup proved highly effective for solving complex tasks such as college-level science multiple-choice questions,

outperforming single-agent methods like chain-of-thought reasoning. Similarly, sequential channel aggregation is explored in other works [135, 138], where collaboration channels are connected in a sequence to amplify the benefits of individual channels.

Domain-specific knowledge plays a critical role in static coordination architectures. In the MapCoder framework [60], for example, collaboration channels are explicitly designed to emulate the program synthesis process, involving agents tasked with recall, planning, code generation, and debugging. The agents communicate through predefined collaboration channels, ensuring a structured workflow where the planning agent directly exchanges information with the coding agent. Similarly, the MACRec framework [131] applies static coordination to recommendation tasks, where specialized agents such as the Manager, User/Item Analyst, and Reflector operate through explicitly defined channels. These workflows leverage domain expertise to optimize interactions, such as enabling the User/Item Analyst to access detailed data about users and items. A similar approach is implemented in literary translation [133], where collaboration channels mirror the traditional workflow of translation publication.

4.5.2 Dynamic Architecture. Dynamic coordination and orchestration architectures, on the other hand, are designed to adapt to changing/evolving environments and task requirements. These architectures rely on management agents or adaptive mechanisms to assign roles and define collaboration channels in real-time. For instance, the Solo Performance Prompting (SPP) approach [132] dynamically identifies relevant personas based on the input. A management agent generates LLM agents with tailored system prompts corresponding to these personas, allowing them to brainstorm and refine solutions collaboratively. This adaptability enables systems to handle diverse tasks effectively, as demonstrated in the ability of GPT-4 to identify accurate and meaningful personas across a wide range of scenarios.

In another example [61], a graph-based orchestration mechanism employs an LLM-based Orchestrator agent to dynamically construct a Directed Acyclic Graph (DAG) from user input. Nodes in the graph represent tasks, while edges define dependencies and collaboration channels between agents. This architecture allows agents to execute tasks in parallel or sequence as dictated by the DAG structure. A Delegator agent consolidates the results from all completed tasks to form the final response, significantly enhancing system responsiveness and scalability, particularly for multi-step, complex queries.

4.6 Summary and Lessons Learned

The rise of LLM-based multi-agent collaborative systems has been driven by the introduction of LLMs and their effectiveness as central processing brains. Inspired by human collaboration, these systems typically break complex tasks into subtasks, with agents assigned specific roles (e.g., software engineer) to focus on subtasks relevant to their expertise. Collaboration channels are critical in enabling agents to work together, facilitating capabilities such as planning and coordination. These channels are characterized by their actors (agents involved), type (e.g., cooperation, competition, or coopetition), structure (e.g., peer-to-peer, centralized, or distributed), and strategy (e.g., role-based, rule-based, or model-based). Collaboration channels enable communication and task orchestration while occasionally exhibiting advanced behaviors like the theory of mind [2, 75]. While most works focus on leveraging LLMs as is - after they are trained, multi-agent collaboration can also be utilized at other stages as well, such as data sharing, model sharing (federated learning), and fine-tuning (ensemble learning). However, LLMs are inherently standalone algorithms and are not specifically trained for collaborative tasks, leaving many mechanisms for leveraging multi-agent collaboration unclear. This presents challenges in both theoretical research and real-world applications, where agent behaviors can be difficult to explain or predict for stakeholders. Effective coordination

ensures that the right agent handles the right problem at the right time. However, AI safety and performance concerns arise, particularly in competitive scenarios where failures like exploitation and hallucination can happen [22, 34, 153]. Cost, scalability, and efficiency are also critical factors to consider. Emerging open-source frameworks such as AutoGen [134], CAMEL [74], and crewAI[3] offer promising tools for building and evaluating multi-agent solutions. Current benchmarks for LLM-based multi-agent collaborative systems focus on metrics such as success rate, task outcomes, cost-effectiveness, and collaborative efficiency, providing valuable insights for system improvement.

Through our review and analysis, several key takeaways have emerged that highlight the strengths, challenges, and opportunities in designing and implementing LLM-based multi-agent collaborative systems. These lessons provide valuable guidance for researchers and practitioners in this growing field:

- **Effective Collaboration Channels**: establishing robust collaboration channels among agents is crucial for seamless collaboration. Clear protocols prevent misunderstandings and ensure efficient information exchange. As shown in AutoGen framework [134] MASs can outperform single-agent systems with effectively designed collaboration mechanisms. On the other hand, as studied in [128] MAS approach with suboptimal design for their competitive collaboration channels can be overtaken by single-agent counterpart with strong prompts.
- **Collective Domain Knowledge**: incorporating domain-specific knowledge is essential for designing collaboration architectures and crafting effective system prompts. Often, collaboration channels are predefined in these cases to align with domain requirements [23, 60, 131].
- **Adaptive Role and Collaboration Channel Assignment**: in certain tasks, it is better to let the system dynamically assigning roles and collaboration channels based on agents' strengths and task requirements enhance system flexibility and performance [39]. This adaptability allows the system to respond effectively to changing environments and objectives.
- **Optimal Collaborative Strategy**: for tasks requiring rigorous adherence to established procedures, rule-based protocols ensure consistency and fairness - avoiding biases caused by role importance or inherent probabilistic nature in other protocols. Role-based strategies allow agents to leverage their own expertise effectively in (pre-)structured tasks requiring job specialization, while model-based protocols work well with uncertain or dynamic situations that demand adaptability and contextually informed decision-making.
- **Scalability Considerations**: as the number of agents increases, maintaining coordination becomes more complex. Implementing scalable architectures and algorithms is essential to handle larger agent networks without performance degradation.
- **Ethical and Safety Considerations**: ensuring that agents operate within ethical boundaries and do not engage in harmful behaviors is vital. Implementing safety protocols and ethical guidelines helps prevent unintended consequences.

5 Applications

This section explores real-world implementations of LLM-based MASs across three dominant domains, including 5G/B6G and Industry 5.0 (IOT); Natural Language Generation (NLG); and Social and Cultural Domains (S&C). Table 6 provides a summary of represented works, highlighting their contributions, advantages, and disadvantages.

[3]https://github.com/crewAIInc/crewAI

Table 6. Summary of applications of different MASs across domains

Methods	Domain	Key Contributions	Advantages	Disadvantages	Refs.
LLM-SC	IOT	• Leverage LLM as the knowledge generator to enhance the semantic decoder.	• Achieves significant coding gains.	• High computation resources due to the utilization of LLM.	[130]
LaMoSC	IOT	• Proposes an LLM-driven multimodal fusion semantic communication.	• Robust in significantly low SNR conditions.	• High computation resources due to the utilization of LLM and Vision Transformer.	[157]
LAM-MSC	IOT	• Design joint encoder for multi-modal data. • LLM operates as a knowledge generator.	• One encoder and decoder can handle various types of data. • Achieves better coding rates and reconstruction error.	• High computation resources due to the utilization of LLM.	[65]
GMAC	IOT	• Utilize LLM to achieve semantic alignment between observed states and natural language, and compress semantic information.	• Improves convergence rate. • Enable multi-agent collaboration without communications.	• High computation resources due to the utilization of LLM.	[160]
LLM-Blender	NLG	• Ensemble approaches of various LLM agents for candidate ranking.	• Ability to generate outputs better than the existing candidates.	• To achieve optimal solution, need $O(n^2)$ inference times, leads to computation overheads.	[64]
SOT	NLG	• Generate the skeleton of answer. • Complete the contents of each skeleton in parallel.	• Accelerate inference speed with parallel. • Suitable for questions that require long answers (need planned structure).	• Answer quality evaluation is far from perfect due to limited prompt set. • May hurt serving throughput due to parallel requests from different agents.	[95]
Meta-Prompting	NLG	• Construct high-level meta prompt to instruct LLMs.	• Maintain coherent line of reasoning. • Tapping into a varierty of expert roles.	• Elevated cost with multiple model calls. • Requirement for substantial scale and considerable context window.	[119]
MAD	NLG	• Two agents express their own arguments. • A judge monitors and manages the debate.	• Reduce bias and distorted perception. • Encourages unlimited external feedback.	• Requires high computational cost due to long debate. • LLMs struggle to maintain coherence and relevance in long scenarios.	[77]
FORD	NLG	• Include three-stage debate: 1) fair debate, 2) mismatched debate, 3) roundtable debate.	• Allow LLMs to explore differences between their own understandings and the conceptualization of others via debate.	• Can not cover various tasks besides commonsense reasoning. • Intensively based on the multiple choice task, which limits FORD's generalization.	[140]
ChatDev	NLG	• Employs a chat chain to break each phase into smaller subtasks, enabling multi-turn communication among agents to collaboratively develop solutions.	• Minimizes coding halluciations, where the provided source code is missing.	• Without clear, detailed requirements, agents struggle to grasp task ideas. • Automating the evaluation of general-purpose software is highly complex. • Multiple agents require more token and times, resulting computational demands.	[105]
AgentVerse	NLG	• Composed of four stages: expert recruitment, collaborative decision making, action execution, evaluation.	• Improves the generalizability of LLMs in finding uncertain. • Improve the adaptability of agents.	• Challenges in communication among agents during the collaborative decision-making process.	[24]
AgentCoord	S&C	• Structured representation for coordination strategy. • Three-stage method to transform general goal into executable strategies.	• Streamline the representing and exploring coordination strategies. • Minimize repetitive instances of agent engagement.	• Only supports coordinating agents to collaborate in a plain text environment. • Only supports static coordination strategy design.	[97]
OpenAI's Swarm	NLG	• Routines & Handoffs for multi-agent orchestration • Lightweight framework for coordination & execution	• Suitable for applications that require scalability • Handoff mechanism allows for seamless transitions between specialized agents	• Concern mainly with role-based protocol & centralised/decentralized structure • Not yet production-ready	See: 4
TE	S&C	• Simulate a representative sample of human participants in subject research.	• Enables simulation of different human behaviors, and reveals consistent distortions of the simulation.	• More human behaviors and additional LLMs needed to study to ensure the key findings.	[36]
AgentInstruct	S&C	• Generates diverse natural language data with iterative cross-agent refinement, including cultural data	• Ables to train more capable models from generated data through tools usage, agentic capabilities, etc.	• Requires human to hand-construct generation flows.	[88]
SocialMind	S&C	• Integrates verbal, non-verbal, and social cues to generate in-situ suggestions via augmented reality glasses.	• Designs and leverages a multi-modal, as multi-tier collaborative agent system.	• Requires advanced edge hardware to handle complex systems.	[144]
CulturePark	S&C	• Prompts LLM-based agents with various cultural backgrounds to simulate cross-cultural communication.	• Generated data allow training models with less bias and democratization.	• Still depends on LLM's knowledge of each culture, and hence limited results for low-resource cultures.	[73]
Mango	S&C	• Extracts high-quality knowledge from LLM-based agents through prompting on concepts and cultures.	• Automated method allows for generating a large amount of resources.	• Human evaluations need to be from more diverse backgrounds.	[94]

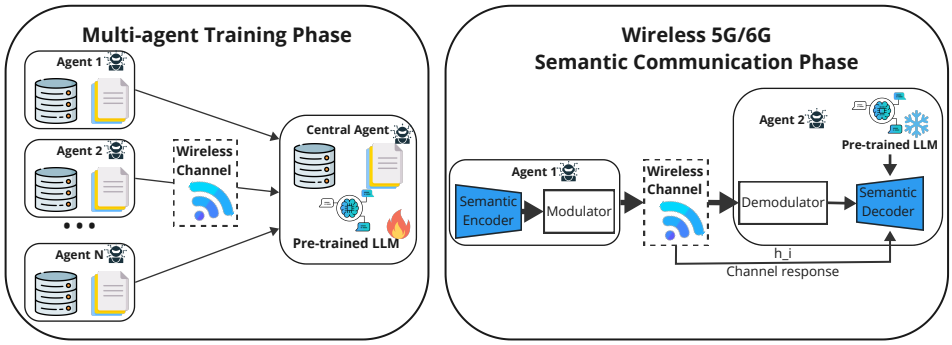


Fig. 6. LLM-based MAS-enabled semantic communication system framework, leveraging LLMs directly to the physical layer coding and decoding of communication system [130].

5.1 5G/B6G and Industry 5.0

Recently, LLM has emerged to be an efficient tool to significantly improve the performance of edge networks [66, 108, 116].

5G and B6G Wireless Network. LLM-SC [130] proposed a novel framework, which utilizes LLM technology to model the semantic information of text and design a semantic communication system based on LLMs (see Fig. 6). By using LLM to probabilistically model transmitted language sequences, LLM-SC achieves a communication paradigm that balances both semantic-level and technical-level performance. LaMoSC [157] introduces an LLM-driven multimodal fusion semantic communication system to extend unimodal transmission and improve generalization. By leveraging the extensive external knowledge of LLMs to generate prompt text, LaMoSC overcomes the limitations of conventional semantic communication systems' knowledge bases and restricted generalization capabilities. To enhance multimodal communication, a fusion encoder is designed to integrate textual and visual features from the LLM using an attention mechanism. LAM-MSC [65] presents a novel application of LLMs to enhance multimodal semantic communication frameworks. In particular, the study introduces a multimodal alignment (MMA) mechanism based on a multimodal language model (MLM), utilizing CoDi for modality transformation. This MMA supports the synchronized generation of integrated modalities by constructing a shared multimodal space. Furthermore, to enable the comprehension of personal information, the framework incorporates a knowledge base powered by an LLM, specifically leveraging GPT-4. Authors in [145] propose a novel framework called M2GSC. In this framework, the LLM serves as shared knowledge base, plays three critical roles, including complex task decomposition, semantic representation specification, and semantic translation and mapping. It also spawns a series of benefits such as semantic encoding standardization and semantic decoding personalization. GMAC [160] introduces a data transmission strategy based on semantic information extraction to reduce the volume of data transmitted in MASs effectively. In this framework, GMAC employs LLMs to achieve semantic alignment between observed states and natural language, facilitating compressed semantic communication. This approach enhances bandwidth efficiency by extracting and compressing relevant information, thereby optimizing data transmission in multi-agent communication. The authors in [121] propose MSADM, an end-to-end health management framework for dynamic heterogeneous networks. Using local and neighboring information, MSADM covers all stages of the health management life cycle, including anomaly detection, fault diagnosis, and mitigation. By integrating an LLM as a facilitating agent, MSADM efficiently collects and processes abnormal data, reducing diagnostic errors caused by inconsistent data representations. The authors in [78] propose a novel approach

that integrates LLMs with reconfigurable intelligent surfaces (RIS) to enable energy-efficient and reliable communication in the Internet of Vehicles. In this RIS system, the LLM is used to deduce optimized strategies for resource allocation and signal decoding order.

Industry 5.0. The authors in [137] propose an LLM-based IoT system using open-source LLMs deployed in a local network environment. The system includes a prompt management module, a post-processing module, and a task-specific prompts database to address concerns related to data privacy and security, system scalability, and to enhance the capabilities of the LLM through integrated prompting methods. The authors in [111] propose SAGE, a smart home agent with grounded execution, which employs a scheme where a user request initiates a sequence of discrete actions controlled by an LLM. SAGE manages this process through a dynamically constructed tree of LLM prompts, which guide the agent in determining the next action, assessing the success of each action, and deciding when to terminate the process. The authors in [112] present an edge-based distributed learning architecture in which a large-scale road network is divided into multiple subgraphs, with data and tasks assigned to individual RSUs. To efficiently learn from this network, they propose LSGLLM, an LLM-based method that incorporates a spatio-temporal module to capture spatio-temporal correlations. LSGLLM addresses the absence of spatio-temporal features in traditional LLMs. The authors in [114] explore the integration of LLMs with the Internet of Senses technology. In this approach, an edge agent employs an LLM to generate WebXR code, enabling the visualization of corresponding 3D virtual objects on head-mounted devices and estimating multi-sensorial media data. CASIT [159] integrates LLMs into IoT systems to enhance the efficiency and intelligence of data processing and operations. By employing collective intelligence, CASIT utilizes multiple LLMs for data analysis and anomaly detection. It generates reports through a step-by-step summary and classification mechanism.

5.2 Question Answering / Natural Language Generation (QA/NLG)

The integration of Large Language Models (LLMs) into MASs has significantly advanced the capabilities of question answering and natural language generation. There are several prominent frameworks currently developed by leading technology companies, each employing unique mechanisms to facilitate agent collaboration in practical applications:

OpenAI's Swarm Framework[4] : this framework introduces a novel approach to orchestrating multiple agents through the concepts of routines and handoffs. In this framework, an agent is defined as an entity that encompasses specific instructions and tools that are capable of transferring an active conversation to another agent, a process termed a "handoff." This mechanism allows for seamless transitions between agents, each specialized in particular tasks, thereby enhancing the system's overall efficiency and adaptability. Swarm's design emphasizes lightweight coordination and execution, making it suitable for scalable, real-world applications. An example with customer service focuses on sales and refunds is illustrates in the diagram 7, demonstrating the feasibility of using Swarm in pratical application.

Microsoft's Magentic-One System[5]: this is a generalist MAS designed to address complex tasks across diverse domains. At its core is the Orchestrator agent, which is responsible for high-level planning, progress tracking, and dynamic re-planning to recover from errors. The Orchestrator delegates specific tasks to specialized agents, such as operating a web browser, navigating local files, or writing and executing Python code. This modular architecture allows for the integration of various skills, facilitating the system's adaptability to a wide range of scenarios.

[4]Refer to OpenAI Cookbook at https://cookbook.openai.com/examples/orchestrating_agents

[5]https://www.microsoft.com/en-us/research/articles/magentic-one-a-generalist-multi-agent-system-for-solving-complex-tasks/

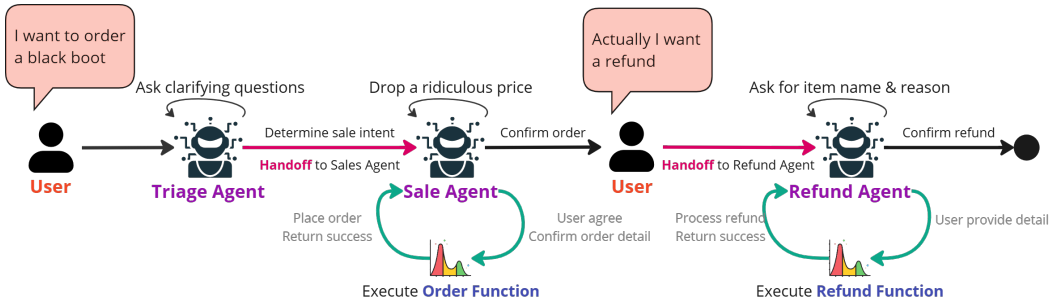


Fig. 7. OpenAI's Swarm use case of customer service.

IBM's Bee Agent Framework[6]: This open-source framework facilitates the development and deployment of scalable, multi-agent workflows. It provides a foundation for building applications where multiple AI agents, powered by LLMs such as IBM Granite and Llama 3, collaborate to achieve complex goals. The framework offers a modular design with prebuilt components for agents, tools, memory management, and instrumentation. Notably, Bee supports the serialization of agent states, enabling the pausing and resuming of complex workflows without data loss. It emphasizes modularity, extensibility, and production-level control to create sophisticated MASs for a wide range of applications, with future development aimed at enhanced multi-agent orchestration.

LangChain Agents[7]: LangChain offers a framework for developing applications powered by language models, with a particular focus on agents. These agents are designed to interact with their environment, using tools to process information. LangChain provides a suite of tools and integrations that facilitate the creation of agents capable of complex reasoning and decision-making processes. This framework supports the development of sophisticated applications that leverage the capabilities of LLMs for advanced question answering and natural language generation tasks.

These frameworks represent significant efforts in the field of multi-agent collaboration for formulating a generalized structure for building multi-agent applications, particularly in the context of question answering and natural language generation. By allowing specialized agents to work in concert, they improve the efficiency and effectiveness of AI systems, paving the way for more sophisticated and adaptable applications.

Another trend in this area of MAS applications for QA/NLG is the introduction of novel frameworks for evaluating responses given by agents and LLMs, which reflects a reimagination of how a task should be judged - compared to the prevalently existing evaluation approach by using more capable models to give ratings, or crowd-sourced AI benchmarking from human preference [25]. For example, "Agent-as-a-Judge" formulates a novel framework for evaluating agentic systems - software agents powered by LLMs - in natural language generation and question answering [165]. The core concept involves using agentic systems to assess other agentic systems, providing detailed feedback throughout the task-solving process, and mirroring human evaluation but at a significantly reduced cost and time. The system employs a role-based strategy where specialized agent modules (e.g., graph construction, code retrieval) operate independently in a decentralized manner with distinct functionalities, contributing to the overall evaluation. Experiments demonstrate that Agent-as-a-Judge aligns closely with human expert evaluations and surpasses the performance of traditional LLM-as-a-Judge methods, especially in complex scenarios, on the DevAI benchmark with 55 realistic AI development tasks. Another framework, "Benchmark Self-Evolving", leverages

[6]https://i-am-bee.github.io/bee-agent-framework/#/
[7]https://python.langchain.com/docs/tutorials/agents/

a MAS to modify existing benchmark instances by altering contexts or questions, thereby creating new, challenging instances that extend the original benchmarks [129]. It employs the role-based strategy, with each agent having a specific function (e.g., instance pre-filter, creator and verifier, candidate option formulator). Experiments conducted on mathematical, logical and commonsense reasoning demonstrate that the self-evolving benchmarks are more challenging than the original ones, thus offering a more accurate assessment of LLMs' true capabilities and limitations, while also addressing issues like data contamination.

The issue of lacking data for LLM training can be alleviated by synthetic data, in which adopting MAS is considered a new approach for such task of NLG. Orca-AgentInstruct[8] (formerly AgentInstruct), a novel agentic solution for generating high-quality synthetic data, uses a multi-agent framework to create tailored datasets from raw data sources, enabling a "generative teaching" approach for improving model capabilities in different areas [88]. Ultilizing 3 distinct agentic flows (Content Transformation, Seed Instruction Generation, and Instruction Refinement) and decentralised structure of agents in each flow, it showed significant performance gains when used to fine-tune a Mistral 7B model, achieving improvements of up to 54% across various benchmarks. Orca-AgentInstruct project represents a significant step towards building a synthetic data factory for model customization and continuous improvement.

In summary, the capabilities of QA/NLG in different tasks have been improved by integrating the MAS mechanism into the process. Response evaluation in QA is now done with higher confidence, since the MAS evaluation systems resemble the process of human evaluation and now includes more dynamic evaluation with automated modified benchmarks. The NLG task of synthesizing the data is also carried out with higher-quality training data generated from the collaboration mechanism. Several notable frameworks recently introduced by big-tech companies also pave the way for the easier creation of MASs, promoting the development of such systems in practical applications. It is important to recognize that these early efforts are still in the process of being adapted and that the efficacy of applying them in practice will take time to be assessed. In addition, the integration of different types and strategies of collaboration, communication structures, and orchestration architecture also need to be considered, since most existing frameworks or systems are focusing primarily on role-based strategy and either centralized or decentralized structures.

5.3 Social and Cultural Domains

Research on LLMs and MASs has showcased the capability and applicability of these systems to simulate human behaviors, social dynamics, and cultural interactions, offering novel methodologies for understanding complex societal phenomenons, as illustrated in Fig. 8. Studies such as [3, 11] argue the potential of LLMs to enhance traditional social science methods, including survey research, online experiments, automated content analyses, and agent-based modeling. However, these studies also underscore critical limitations, such as biases in training data that lack global psychological diversity, cautioning against treating stand-alone LLMs as universal solutions. The shift from stand-alone LLMs to Multi-Agent Collaborative Systems can not only enable the analysis of LLMs in replicating individual social behavior but also provide powerful tools for exploring complex social dynamics, collaborative problem-solving, and emergent collective behaviors [46].

Several studies have focused on simulating social interactions through carefully designed environments, where agents are equipped with diverse prompts or LLMs tailored to specific roles. For instance, [89] follows the definition of social interaction as a theatrical performance, with agents assuming roles (e.g., office employees or family members) driven by LLMs like GPT-4, Qwen2.5-14b,

[8]https://www.microsoft.com/en-us/research/blog/orca-agentinstruct-agentic-flows-can-be-effective-synthetic-data-generators/

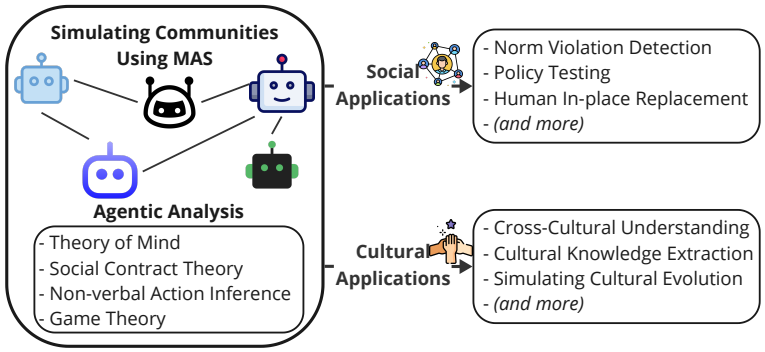


Fig. 8. LLM-based multi-agent collaborative system in social & cultural applications.

or Llama-3-8b. These roles include specific goals, such as providing and receiving feedback during office discussions. Research has shown that collaboration among LLM-based agents can elicit human-like capabilities, including conversational skills, theory of mind (reasoning about others' mental states) [2, 75], Hobbesian social contract theory (submit to authority to avoid chaos) [31], and non-verbal action inference [79, 147].

Social Applications. Authors in [5, 36] suggest that LLM-based agents can replace human participants in specific social science experiments, while [88] demonstrates their use in generating diverse natural language data with iterative cross-agent refinement. Moreover, multimodal AI systems such as those described in [144] integrate verbal, non-verbal, and social cues as input to multi-tier collaborative agents to generate in-situ suggestions via augmented reality glasses. Integrating LLM-based agents into traditional agent-based modeling [48] can enhance the realism of simulations, offering controlled environments to test social theories, including the effects of policy interventions [10, 148] and norm violation detection [53].

Cultural Applications. LLM-based MASs can represent diverse cultural perspectives, advancing cross-cultural understanding and reducing bias. For example, the CulturePark framework [73] simulates cross-cultural interactions, with each agent embodying distinct cultural viewpoints. Similarly, Mango [94] iteratively extracts high-quality cultural knowledge from LLM-based agents, providing a rich dataset for fine-tuning models to improve their ability to align with diverse cultural contexts. Another emerging area involves simulating cultural evolution within LLM populations. By modeling how cultural information is transmitted and transformed among agents, researchers gain insights into both human cultural dynamics and their influence on LLM behavior [103, 126]. Another area of application involves simulating cultural evolution within LLM populations. By modeling how cultural information is transmitted and transformed among agents, researchers gain insights into both human cultural dynamics and their influence on LLM behavior [103].

Despite their promise, LLMs are not perfect replicas of humans and cannot fully replicate the complexities of human social and cultural behavior. For instance, [36] highlights the limitations of using LLMs as human replacements in social science experiments, particularly in scenarios involving information asymmetry (unequal access to private mental states or goals) [161], and in tasks requiring competition and conflict resolution [89]. To address these challenges, consistent and standardized benchmarking approaches are necessary to evaluate the cultural and social awareness of LLM-based agents [109].

6 Open Problems & Discussion

6.1 The Road to Artificial Collective Intelligence

Collective intelligence is the ability of a group to perform complex tasks and solve problems collectively, often overcoming the sum of individual contributions [71]. With increasingly complex capabilities that mimic characteristics of living organisms, LLMs are being treated as "digital species". Enabling collective intelligence through collaborations among multiple LLM-based agents offers the potential for AI systems that are adaptive, efficient, and capable of addressing real-world problems. However, several open challenges must be addressed to realize this potential fully.

Unified Governance. Unified governance is fundamental in enabling collective intelligence among group of LLM-based agents, including the design of coordination and planning mechanisms. Deciding which steps to take, which agents to involve, and how tasks should be distributed among them requires advanced mechanisms. Assigning specific roles or specializations to individual agents can enhance the system's overall effectiveness. Determining optimal role assignments and ensuring agents can adapt to dynamic task requirements are ongoing research areas. Moreover, governance must account for potential failures, such as miscommunication or task disruptions. Designing robust mechanisms to detect and recover from such failures is vital for ensuring the reliability and resilience of MASs. For example, introducing redundancy or fallback agents may help maintain system functionality even in adversarial scenarios.

Shared Decision Making. Beyond governance, MASs must achieve coherent and accurate collective decision-making. Current LLM-based MASs commonly utilize limited decision-making methods, such as dictatorial or popular voting, which may not capture different aspects of agent preferences, or aggregating overconfidence of LLMs. Research into novel decision making approaches can enhance the diversity and fairness of collective decisions.

Agent as Digital Species. LLMs are increasingly being viewed as digital species; however, they were not originally designed for agentic applications involving collaboration and multi-participant interactions. They suffer from known limitations such as hallucinations and are susceptible to adversarial attacks. A single agent's hallucination can be spread and reinforced by other agents, leading to minor inaccuracies into critical and cascading effects. Addressing these issues requires techniques to not only detect and correct individual errors but also to control the collaboration channels between agents. Designing LLMs specifically for collaborative environments, such as Gemini 2.0[9], represents a step toward refining these "digital species" for agentic systems.

Scalability and Resource Maintainance. Increasing agent population poses a significant challenge in MASs. Managing resources (memory, processing time), coordination and collaboration channels among a growing number of agents introduces additional complexities, such as maintaining efficiency in agent interactions and preventing bottlenecks. Understanding the scaling laws of the behavior and performance of MASs is critical for designing architectures capable of handling large-scale collaboration.

Discovering/Exploring Unexpected Generalization. Complex, emergent behaviors of collective intelligence, such as coordinated problem-solving or innovation, can arise under the right conditions, especially in generalizing to unseen domains. However, identifying and fostering these conditions is an ongoing challenge. Understanding how these generalizations emerge from the interactions of agents is key to acquiring collective intelligence.

6.2 Comprehensive Evaluation and Benchmarking

Evaluation of MASs presents challenges beyond the evaluation of individual LLMs. While there has been active research in exploring various aspects of LLMs [19], including their decision making

[9]https://blog.google/products/gemini/google-gemini-ai-collection-2024/

capabilities and tool usage in agentic applications [100, 102], relatively few effort has been dedicated to systematically assessing the performance and behavior of LLM-based MASs [80].

The collaborative nature of these systems introduces complexities that require a broader set of evaluation criteria. These criteria include assessing the overall system performance [155], such as reasoning capabilities, task completion rates, as well as specific system characteristics like coordination efficiency and contextual appropriateness [17]. Fine-grained evaluation at the agent and collaboration levels enable root cause analysis [63], offering insights into individual agent behaviors, the effectiveness of collaboration channels, and the system's overall dynamics.

Furthermore, evaluations of MASs are often conducted in narrow scenarios with different configurations, leading to inconsistent and incomparable results [18, 22, 34]. The absence of standardized evaluation protocols prevents the ability to objectively compare different systems and track progress across the field. Establishing unified, broad, and comprehensive benchmarking frameworks is vital to ensure reproducibility and consistency in evaluating MASs. Moreover, static evaluation benchmarks risk becoming lack of relevance to current real-world scenarios, leading to data leakage and overfitting [102]. Therfore, there is a need for implementing dynamic benchmarking systems that evolve alongside technological and informational advancements.

6.3 Ethical Risk and Safety

Intrinsicly, LLMs can be harmful with hallucinated information. When deployed in MASs, these issues can propagate and amplify through agent interactions. There are two key factors behind this amplification: LLM overconfidence problem, where LLMs persistently assert the correctness of their outputs despite inaccuracies [151], and misunderstandings that arise between LLM-based agents during collaboration. Additionally, LLMs are vulnerable to adversarial attacks, which make MASs particularly attractive targets for exploitation [115]. Compromised agents in such systems can be manipulated to execute harmful or malicious behaviors. As the number of agents in LLM-based MAS increases, these risks scale proportionally, compromising the safety and reliability of communication and information exchange.

Recent studies have also highlighted the potential for AI systems to deceive humans, raising significant concerns in the context of LLM-based multi-agent collaborative systems [7, 35, 84]. These systems, capable of simulating human societies and exhibiting human-like psychological traits, can blur the line between artificial and human behavior. Attributing human-like qualities to these systems risks fostering over-reliance, where users place trust in their capabilities. This perception can increase susceptibility to manipulation and obscure the inherent limitations of LLM-based agents [161]. Overlooking these limitations not only undermines informed decision-making but also introduces broader ethical concerns[10] regarding the responsible deployment and use of LLM-based MASs.

7 Conclusion

Through our extensive review of the collaborative aspect of LLM-based MASs, we introduce a structured and extensible framework as an important lens to guide future research. Our framework characterizes collaboration along five key dimensions: actors, types, structures, strategies, and coordination mechanisms, providing a systematic approach to analyze and design collaborative interactions within MASs empowered by LLMs. We believe this work will inspire future research and serve as a foundational step in advancing MASs toward more intelligent and collaborative solutions.

[10]https://artificialintelligenceact.eu/

Acknowledgments

This research work has emanated from research conducted with financial support from Science Foundation Ireland under Grant 12/RC/2289-P2 and 18/CRT/6223.

References

[1] Azad Abad, Moin Nabi, and Alessandro Moschitti. 2017. Autonomous crowdsourcing through human-machine collaborative learning. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 873–876.

[2] Sahar Abdelnabi et al. 2024. Cooperation, Competition, and Maliciousness: LLM-Stakeholders Interactive Negotiation. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[3] Suhaib Abdurahman et al. 2024. Perils and opportunities in using large language models in psychological research. *PNAS Nexus* (Jul. 2024).

[4] Josh Achiam et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[5] Gati Aher et al. 2023. Using large language models to simulate multiple humans and replicate human subject studies. In *Proceedings of the International Conference on Machine Learning*.

[6] Junhyeok Ahn and Luis Sentis. 2021. Nested mixture of experts: Cooperative and competitive learning of hybrid dynamical system. In *Learning for Dynamics and Control*. PMLR, 779–790.

[7] Canfer Akbulut et al. 2024. All Too Human? Mapping and Mitigating the Risk from Anthropomorphic AI. *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society* 7 (Oct. 2024), 13–26.

[8] Anonymous. 2024. DOMAIN GENERALIZATION VIA PARETO OPTIMAL GRADIENT MATCHING. In *Submitted to The Thirteenth International Conference on Learning Representations*. under review.

[9] Anonymous. 2024. Federated Domain Generalization with Data-free On-server Gradient Matching. In *Submitted to The Thirteenth International Conference on Learning Representations*. under review.

[10] Gabriele Ansaldo. 2023. *AgentSpeak: A Framework for Agent-Based Modeling with Integrated Large Language Models; Case Study: Analyzing Policy Interventions in Electric Vehicle Adoption*. Master's thesis. Northeastern University.

[11] Christopher A. Bail. 2024. Can Generative AI improve social science? *Proceedings of the National Academy of Sciences* 121, 21 (2024), e2314021121.

[12] Rafael Barbarroxa et al. 2024. Benchmarking AutoGen with different large language models. In *2024 IEEE Conference on Artificial Intelligence (CAI)*. IEEE, 263–264.

[13] Kallista Bonawitz et al. 2021. Federated Learning and Privacy: Building privacy-preserving systems for machine learning and data science on decentralized data. *Queue* (Nov. 2021).

[14] Michele Braccini et al. 2024. Swarm Intelligence: A Novel and Unconventional Approach to Dance Choreography Creation.

[15] Weilin Cai et al. 2024. A Survey on Mixture of Experts. arXiv:2407.06204 [cs.LG]

[16] Chengzhi Cao et al. 2024. Enhancing Human-AI Collaboration Through Logic-Guided Reasoning. In *The Twelfth International Conference on Learning Representations*.

[17] Alan Chan et al. 2023. Harms from Increasingly Agentic Algorithmic Systems. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency* (Chicago, IL, USA) *(FAccT '23)*. Association for Computing Machinery, New York, NY, USA, 651–666.

[18] Chi-Min Chan et al. 2024. ChatEval: Towards Better LLM-based Evaluators through Multi-Agent Debate. In *The Twelfth International Conference on Learning Representations*.

[19] Yupeng Chang et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology* 15, 3 (2024), 1–45.

[20] Guangyao Chen et al. 2024. AutoAgents: A Framework for Automatic Agent Generation. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, Kate Larson (Ed.). International Joint Conferences on Artificial Intelligence Organization, 22–30. Main Track.

[21] Huaben Chen, Wenkang Ji, Lufeng Xu, and Shiyu Zhao. 2023. Multi-Agent Consensus Seeking via Large Language Models. *ArXiv* abs/2310.20151 (2023).

[22] Junzhe Chen et al. 2024. LLMArena: Assessing Capabilities of Large Language Models in Dynamic Multi-Agent Environments. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Bangkok, Thailand, 13055–13077.

[23] Pei Chen, Shuai Zhang, and Boran Han. 2024. CoMM: Collaborative Multi-Agent, Multi-Reasoning-Path Prompting for Complex Problem Solving. In *Findings of the Association for Computational Linguistics: NAACL 2024*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). ACL, Mexico City, Mexico, 1720–1738.

[24] Weize Chen et al. 2024. AgentVerse: Facilitating Multi-Agent Collaboration and Exploring Emergent Behaviors. In *The Twelfth International Conference on Learning Representations*.

[25] Wei-Lin Chiang et al. 2024. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132* (2024).

[26] Vincent Conitzer and Caspar Oesterheld. 2024. Foundations of Cooperative AI. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 13 (Jul. 2024), 15359–15367.

[27] James L. Crowley et al. 2023. A Hierarchical Framework for Collaborative Artificial Intelligence. *IEEE Pervasive Computing* 22, 1 (2023), 9–18. https://doi.org/10.1109/MPRV.2022.3208321

[28] Florin Cuconasu et al. 2024. The power of noise: Redefining retrieval for rag systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 719–729.

[29] Allan Dafoe et al. 2020. Open Problems in Cooperative AI. arXiv:2012.08630 [cs.AI]

[30] Allan Dafoe et al. 2021. Cooperative AI: machines must learn to find common ground. *Nature* 593, 7857 (May 2021), 33–36.

[31] Gordon Dai et al. 2024. Artificial Leviathan: Exploring Social Evolution of LLM Agents Through the Lens of Hobbesian Social Contract Theory. arXiv:2406.14373 [cs.AI]

[32] Mike D'Arcy et al. 2024. MARG: Multi-Agent Review Generation for Scientific Papers. arXiv:2401.04259 [cs.CL]

[33] Ayushman Das et al. 2023. Enabling Synergistic Knowledge Sharing and Reasoning in Large Language Models with Collaborative Multi-Agents. In *IEEE International Conference on Collaboration and Internet Computing*.

[34] Tim Ruben Davidson et al. 2024. Evaluating Language Model Agency Through Negotiations. In *The Twelfth International Conference on Learning Representations*.

[35] Ameet Deshpande et al. 2023. Anthropomorphization of AI: Opportunities and Risks. arXiv:2305.14784 [cs.AI]

[36] Danica Dillion et al. 2023. Can AI language models replace human participants? *Trends in Cognitive Sciences* 27, 7 (2023), 597–600.

[37] Mohammad Divband Soorati et al. 2022. From intelligent agents to trustworthy human-centred multiagent systems. *AI Communications* 35, 4 (2022), 443–457.

[38] Manqing Dong, Hao Huang, and Longbing Cao. 2024. Can LLMs Serve As Time Series Anomaly Detectors? *arXiv preprint arXiv:2408.03475* (2024).

[39] Yubo Dong et al. 2024. VillagerAgent: A Graph-Based Multi-Agent Framework for Coordinating Complex Task Dependencies in Minecraft. In *Findings of the Association for Computational Linguistics: ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 16290–16314.

[40] Ali Dorri, Salil S. Kanhere, and Raja Jurdak. 2018. Multi-Agent Systems: A Survey. *IEEE Access* 6 (2018), 28573–28593.

[41] Yilun Du et al. 2023. Improving Factuality and Reasoning in Language Models through Multiagent Debate. arXiv:2305.14325 [cs.CL]

[42] Lizhou Fan et al. 2024. A bibliometric review of large language models research from 2017 to 2023. *ACM Transactions on Intelligent Systems and Technology* 15, 5 (2024), 1–25.

[43] Joel E Fischer et al. 2021. In-the-loop or on-the-loop? Interactional arrangements to support team coordination with a planning agent. *Concurrency and Computation: Practice and Experience* 33, 8 (2021), e4082.

[44] Adam Fourney et al. 2024. *Magentic-One: A Generalist Multi-Agent System for Solving Complex Tasks*. Technical Report MSR-TR-2024-47. Microsoft.

[45] Chris Frith and Uta Frith. 2005. Theory of mind. *Current biology* 15, 17 (2005), R644–R645.

[46] Chen Gao et al. 2024. Large language models empowered agent-based modeling and simulation: a survey and perspectives. *Humanities and Social Sciences Communications* 11, 1 (Sept. 2024).

[47] Taicheng Guo et al. 2024. Large Language Model Based Multi-agents: A Survey of Progress and Challenges. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, Kate Larson (Ed.). International Joint Conferences on Artificial Intelligence Organization, 8048–8057. Survey Track.

[48] Onder Gurcan. 2024. LLM-Augmented Agent-Based Modelling for Social Simulations: Challenges and Opportunities. arXiv:2405.06700 [physics.soc-ph]

[49] Thilo Hagendorff, Sarah Fabi, and Michal Kosinski. 2023. Human-like intuitive behavior and reasoning biases emerged in large language models but disappeared in ChatGPT. *Nature Computational Science* 3, 10 (2023), 833–838.

[50] Shanshan Han et al. 2024. LLM Multi-Agent Systems: Challenges and Open Problems. arXiv:2402.03578 [cs.MA]

[51] Kostas Hatalis et al. 2023. Memory Matters: The Need to Improve Long-Term Memory in LLM-Agents. In *Proceedings of the AAAI Symposium Series*, Vol. 2. 277–280.

[52] Junda He, Christoph Treude, and David Lo. 2024. LLM-Based Multi-Agent Systems for Software Engineering: Vision and the Road Ahead. *arXiv preprint arXiv:2404.04834* (2024).

[53] Shawn He et al. 2024. Norm Violation Detection in Multi-Agent Systems using Large Language Models: A Pilot Study. arXiv:2403.16517 [cs.MA]

[54] Zhitao He et al. 2023. LEGO: A Multi-agent Collaborative Framework with Role-playing and Iterative Feedback for Causality Explanation Generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 9142–9163.

[55] Jordan Hoffmann et al. 2024. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '22)*. Curran Associates Inc., Red Hook, NY, USA, Article 2176, 15 pages.
[56] Sirui Hong et al. 2024. MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework. In *The Twelfth International Conference on Learning Representations*.
[57] Lei Huang et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232* (2023).
[58] Xu Huang et al. 2024. Understanding the planning of LLM agents: A survey. *arXiv preprint arXiv:2402.02716* (2024).
[59] Yoichi Ishibashi and Yoshimasa Nishimura. 2024. Self-Organized Agents: A LLM Multi-Agent Framework toward Ultra Large-Scale Code Generation and Optimization. arXiv:2404.02183 [cs.SE]
[60] Md. Ashraful Islam, Mohammed Eunus Ali, and Md Rizwan Parvez. 2024. MapCoder: Multi-Agent Code Generation for Competitive Problem Solving. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.
[61] Shankar Kumar Jeyakumar, Alaa Alameer Ahmad, and Adrian Garret Gabriel. 2024. Advancing Agentic Systems: Dynamic Task Decomposition, Tool Integration and Evaluation using Novel Metrics and Dataset. In *NeurIPS 2024 Workshop on Open-World Agents*.
[62] Divyansh Jhunjhunwala, Shiqiang Wang, and Gauri Joshi. 2023. FedExP: Speeding Up Federated Averaging via Extrapolation. In *The Eleventh International Conference on Learning Representations*.
[63] Zhenlan Ji et al. 2024. Testing and Understanding Erroneous Planning in LLM Agents through Synthesized User Inputs. arXiv:2404.17833 [cs.AI]
[64] Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. LLM-Blender: Ensembling Large Language Models with Pairwise Ranking and Generative Fusion. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.
[65] Feibo Jiang et al. 2024. Large AI Model Empowered Multimodal Semantic Communications. *IEEE Communications Magazine* (2024), 1–7.
[66] Feibo Jiang et al. 2024. Large Language Model Enhanced Multi-Agent Systems for 6G Communications. *IEEE Wireless Communications* (2024), 1–8.
[67] Xue Jiang et al. 2024. Self-Planning Code Generation with Large Language Models. *ACM Trans. Softw. Eng. Methodol.* 33, 7, Article 182 (Sept. 2024), 30 pages.
[68] Yogeswaranathan Kalyani and Rem Collier. 2024. The Role of Multi-Agents in Digital Twin Implementation: Short Survey. *ACM Comput. Surv.* 57, 3, Article 72 (Nov. 2024), 15 pages.
[69] Jared Kaplan et al. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361* (2020).
[70] Stefano Lambiase et al. 2024. Motivations, Challenges, Best Practices, and Benefits for Bots and Conversational Agents in Software Engineering: A Multivocal Literature Review. *ACM Comput. Surv.* 57, 4, Article 93 (Dec. 2024), 37 pages.
[71] Jan Marco Leimeister. 2010. Collective Intelligence. *Business and Information Systems Engineering* 2, 4 (June 2010), 245–248.
[72] Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*. PMLR, 19274–19286.
[73] Cheng Li et al. 2024. CulturePark: Boosting Cross-cultural Understanding in Large Language Models. arXiv:2405.15145 [cs.AI]
[74] Guohao Li et al. 2023. CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society. In *Thirty-seventh Conference on Neural Information Processing Systems*.
[75] Huao Li et al. 2023. Theory of Mind for Multi-Agent Collaboration via Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 180–192.
[76] Junyi Li et al. 2024. Pre-Trained Language Models for Text Generation: A Survey. *ACM Comput. Surv.* 56, 9, Article 230 (April 2024), 39 pages.
[77] Tian Liang et al. 2024. Encouraging Divergent Thinking in Large Language Models through Multi-Agent Debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 17889–17904.
[78] Qiang Liu et al. 2024. LLM Enhanced Reconfigurable Intelligent Surface for Energy-Efficient and Reliable 6G IoV. *IEEE Transactions on Vehicular Technology* (2024), 1–9.
[79] Ryan Liu et al. 2024. Large Language Models Assume People are More Rational than We Really are. arXiv:2406.17055 [cs.CL]
[80] Xiao Liu et al. 2024. AgentBench: Evaluating LLMs as Agents. In *The Twelfth International Conference on Learning Representations*.
[81] Zijun Liu et al. 2024. A Dynamic LLM-Powered Agent Network for Task-Oriented Agent Collaboration. In *First Conference on Language Modeling*.

[82] Jinliang Lu et al. 2024. Merge, Ensemble, and Cooperate! A Survey on Collaborative Strategies in the Era of Large Language Models. arXiv:2407.06089 [cs.CL]

[83] Zhao Mandi, Shreeya Jain, and Shuran Song. 2024. Roco: Dialectic multi-robot collaboration with large language models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 286–299.

[84] Alexander Meinke et al. 2024. Frontier Models are Capable of In-context Scheming. arXiv:2412.04984 [cs.AI]

[85] Andres F. Mena-Guacas et al. 2023. Collaborative learning and skill development for educational growth of artificial intelligence: A systematic review. *Contemporary Educational Technology* 15, 3 (July 2023), ep428.

[86] Bonan Min et al. 2023. Recent Advances in Natural Language Processing via Large Pre-trained Language Models: A Survey. *ACM Comput. Surv.* 56, 2, Article 30 (Sept. 2023), 40 pages.

[87] Marvin Minsky. 1988. *Society of mind*. Simon and Schuster.

[88] Arindam Mitra et al. 2024. Agentinstruct: Toward generative teaching with agentic flows. *arXiv preprint arXiv:2407.03502* (2024).

[89] Xinyi Mou, Jingcong Liang, Jiayu Lin, Xinnong Zhang, Xiawei Liu, Shiyue Yang, Rong Ye, Lei Chen, Haoyu Kuang, Xuanjing Huang, and Zhongyu Wei. 2024. AgentSense: Benchmarking Social Intelligence of Language Agents through Interactive Scenarios. arXiv:2410.19346 [cs.CL]

[90] Yongan Mu et al. 2023. Runtime verification of self-adaptive multi-agent system using probabilistic timed automata. *Journal of Intelligent & Fuzzy Systems* 45, 6 (2023), 10305–10322.

[91] Nathalia Nascimento, Paulo Alencar, and Donald Cowan. 2023. Self-adaptive large language model (llm)-based multiagent systems. In *2023 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C)*. IEEE, 104–109.

[92] Minh Duong Nguyen et al. 2024. Towards Layer-Wise Personalized Federated Learning: Adaptive Layer Disentanglement via Conflicting Gradients. arXiv:2410.02845 [cs.LG]

[93] Minh Thuan Nguyen et al. 2023. ViGPTQA-state-of-the-art LLMs for vietnamese question answering: system overview, core models training, and evaluations. In *Proceedings of the 2023 conference on empirical methods in natural language processing: industry track*. 754–764.

[94] Tuan-Phong Nguyen, Simon Razniewski, and Gerhard Weikum. 2024. Cultural Commonsense Knowledge for Intercultural Dialogues. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management* (Boise, ID, USA) *(CIKM '24)*. Association for Computing Machinery, New York, NY, USA, 1774–1784.

[95] Xuefei Ning et al. 2024. Skeleton-of-Thought: Prompting LLMs for Efficient Parallel Generation. In *The Twelfth International Conference on Learning Representations*.

[96] Fan Ouyang and Liyin Zhang. 2024. AI-driven learning analytics applications and tools in computer-supported collaborative learning: A systematic review. *Educational Research Review* 44 (2024), 100616.

[97] Bo Pan et al. 2024. AgentCoord: Visually Exploring Coordination Strategy for LLM-based Multi-Agent Collaboration. arXiv:2404.11943 [cs.HC]

[98] Balaji Parasumanna Gokulan and D. Srinivasan. 2010. *An Introduction to Multi-Agent Systems*. Vol. 310. 1–27. https://doi.org/10.1007/978-3-642-14435-6_1

[99] Joon Sung Park et al. 2023. Generative Agents: Interactive Simulacra of Human Behavior *(UIST '23)*.

[100] Shishir G Patil et al. 2024. Gorilla: Large Language Model Connected with Massive APIs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[101] Georgios Peikos, Pranav Kasela, and Gabriella Pasi. 2024. Leveraging Large Language Models for Medical Information Extraction and Query Generation. *arXiv preprint arXiv:2410.23851* (2024).

[102] Ji-Lun Peng et al. 2024. A Survey of Useful LLM Evaluation. *arXiv preprint arXiv:2406.00936* (2024).

[103] Jérémy Perez et al. 2024. Cultural evolution in populations of Large Language Models. arXiv:2403.08882 [cs.MA]

[104] Haritz Puerto, Gözde Şahin, and Iryna Gurevych. 2023. MetaQA: Combining Expert Agents for Multi-Skill Question Answering. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, Andreas Vlachos and Isabelle Augenstein (Eds.). Association for Computational Linguistics, Dubrovnik, Croatia, 3566–3580.

[105] Chen Qian et al. 2024. ChatDev: Communicative Agents for Software Development. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

[106] Shuofei Qiao et al. 2024. AutoAct: Automatic Agent Learning from Scratch for QA via Self-Planning. arXiv:2401.05268 [cs.CL]

[107] Yujia Qin et al. 2024. Tool Learning with Foundation Models. *ACM Comput. Surv.* 57, 4, Article 101 (Dec. 2024).

[108] Zhijin Qin et al. 2024. AI Empowered Wireless Communications: From Bits to Semantics. *Proc. IEEE* 112, 7 (2024), 621–652.

[109] Haoyi Qiu et al. 2024. Evaluating Cultural and Social Awareness of LLM Web Agents. arXiv:2410.23252 [cs.CL]

[110] Mohaimenul Azam Khan Raiaan et al. 2024. A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges. *IEEE Access* 12 (2024), 26839–26874.

[111] Dmitriy Rivkin et al. 2024. AIoT Smart Home via Autonomous LLM Agents. *IEEE Internet of Things Journal* (2024), 1–1.

[112] Yi Rong et al. 2024. Large-Scale Traffic Flow Forecast with Lightweight LLM in Edge Intelligence. *IEEE Internet of Things Magazine* (2024), 1–7.

[113] Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. 2024. Are emergent abilities of large language models a mirage? *Advances in Neural Information Processing Systems* 36 (2024).

[114] Nassim Sehad et al. 2024. Generative AI for Immersive Communication: The Next Frontier in Internet-of-Senses Through 6G. *IEEE Communications Magazine* (2024), 1–13.

[115] Erfan Shayegani et al. 2023. Survey of vulnerabilities in large language models revealed by adversarial attacks. *arXiv preprint arXiv:2310.10844* (2023).

[116] Yifei Shen et al. 2024. Large Language Models Empowered Autonomous Edge AI for Connected Intelligence. *IEEE Communications Magazine* 62, 10 (2024), 140–146.

[117] Noah Shinn et al. 2023. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

[118] Chuanneng Sun, Songjun Huang, and Dario Pompili. 2024. LLM-based Multi-Agent Reinforcement Learning: Current and Future Directions. *arXiv preprint arXiv:2405.11106* (2024).

[119] Mirac Suzgun and Adam Tauman Kalai. 2024. Meta-Prompting: Enhancing Language Models with Task-Agnostic Scaffolding. arXiv:2401.12954 [cs.CL]

[120] Yashar Talebirad and Amirhossein Nadiri. 2023. Multi-Agent Collaboration: Harnessing the Power of Intelligent LLM Agents. arXiv:2306.03314 [cs.AI]

[121] Fengxiao Tang et al. 2024. Large Language Model(LLM) assisted End-to-End Network Health Management based on Multi-Scale Semanticization. arXiv:2406.08305 [cs.NI]

[122] Xiangru Tang et al. 2024. MedAgents: Large Language Models as Collaborators for Zero-shot Medical Reasoning. In *Findings of the Association for Computational Linguistics*.

[123] Gemini Team et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805* (2023).

[124] Hugo Touvron et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[125] Khanh-Tung Tran, Barry O'Sullivan, and Hoang D Nguyen. 2024. UCCIX: Irish-eXcellence Large Language Model. *ECAI 2024* (2024).

[126] Khanh-Tung Tran, Barry O'Sullivan, and Hoang D Nguyen. 2024. Irish-based Large Language Model with Extreme Low-Resource Settings in Machine Translation. *LoResMT 2024* (2024), 193.

[127] A Vaswani et al. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).

[128] Qineng Wang et al. 2024. Rethinking the Bounds of LLM Reasoning: Are Multi-Agent Discussions the Key?. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

[129] Siyuan Wang et al. 2024. Benchmark Self-Evolving: A Multi-Agent Framework for Dynamic LLM Evaluation. *arXiv preprint arXiv:2402.11443* (2024).

[130] Zhenyi Wang et al. 2024. Large Language Model Enabled Semantic Communication Systems. arXiv:2407.14112 [eess.SP]

[131] Zhefan Wang et al. 2024. MACRec: A Multi-Agent Collaboration Framework for Recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) *(SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 2760–2764.

[132] Zhenhailong Wang et al. 2024. Unleashing the Emergent Cognitive Synergy in Large Language Models: A Task-Solving Agent through Multi-Persona Self-Collaboration. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*.

[133] Minghao Wu et al. 2024. (Perhaps) Beyond Human Translation: Harnessing Multi-Agent Collaboration for Translating Ultra-Long Literary Texts. arXiv:2405.11804 [cs.CL]

[134] Qingyun Wu et al. 2024. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation.

[135] Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022. AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 385, 22 pages.

[136] Zhiheng Xi et al. 2023. The Rise and Potential of Large Language Model Based Agents: A Survey. arXiv:2309.07864 [cs.AI]

[137] Bin Xiao et al. 2024. Efficient Prompting for LLM-Based Generative Internet of Things. *IEEE Internet of Things Journal* (2024), 1–1.

[138] Ziyang Xiao et al. 2024. Chain-of-Experts: When LLMs Meet Complex Operations Research Problems. In *The Twelfth International Conference on Learning Representations*.
[139] Tianbao Xie et al. 2023. OpenAgents: An Open Platform for Language Agents in the Wild. arXiv:2310.10634 [cs.CL]
[140] Kai Xiong et al. 2023. Examining Inter-Consistency of Large Language Models Collaboration: An In-depth Analysis via Debate. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.
[141] Lin Xu et al. 2023. Magic: Investigation of large language model powered multi-agent in cognition, adaptability, rationality and collaboration. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.
[142] Zheng Xu et al. 2023. Federated Learning of Gboard Language Models with Differential Privacy. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*.
[143] Zhenran Xu et al. 2023. Towards reasoning in large language models via multi-agent peer review collaboration. *arXiv preprint arXiv:2311.08152* (2023).
[144] Bufang Yang et al. 2024. SocialMind: LLM-based Proactive AR Social Assistive System with Human-like Perception for In-situ Live Interactions. arXiv:2412.04036 [cs.AI]
[145] Wanting Yang et al. 2024. Rethinking Generative Semantic Communication for Multi-User Systems with Multi-Modal LLM. arXiv:2408.08765 [cs.NI]
[146] Zhangyue Yin et al. 2023. Exchange-of-Thought: Enhancing Large Language Model Capabilities through Cross-Model Communication. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 15135–15153.
[147] Lance Ying et al. 2024. Inferring the Goals of Communicating Agents from Actions and Instructions. *Proceedings of the AAAI Symposium Series* 2, 1 (Jan. 2024), 26–33.
[148] Yongchao Zeng et al. 2024. Exploring the opportunities and challenges of using large language models to represent institutional agency in land system modelling. *EGUsphere* 2024 (2024), 1–35.
[149] Ceyao Zhang et al. 2024. ProAgent: Building Proactive Cooperative Agents with Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence* 38, 16 (Mar. 2024), 17591–17599.
[150] Junjie Zhang et al. 2024. AgentCF: Collaborative Learning with Autonomous Language Agents for Recommender Systems. In *Proceedings of the ACM Web Conference*.
[151] Jintian Zhang et al. 2024. Exploring Collaboration Mechanisms for LLM Agents: A Social Psychology View. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.
[152] Yifan Zhang et al. 2024. Cumulative Reasoning with Large Language Models. arXiv:2308.04371 [cs.AI]
[153] Zaibin Zhang et al. 2024. PsySafe: A Comprehensive Framework for Psychological-based Attack, Defense, and Evaluation of Multi-agent System Safety. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.
[154] Zeyu Zhang et al. 2024. A survey on the memory mechanism of large language model based agents. *arXiv preprint arXiv:2404.13501* (2024).
[155] Qinlin Zhao et al. 2024. CompeteAI: Understanding the Competition Dynamics of Large Language Model-based Agents. In *Agentic Markets Workshop at ICML 2024*.
[156] Wayne Xin Zhao et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223* (2023).
[157] Yaru Zhao et al. 2024. LaMoSC: Large Language Model-Driven Semantic Communication System for Visual Transmission. *IEEE Transactions on Cognitive Communications and Networking* 10, 6 (2024), 2005–2018.
[158] Yi Zheng et al. 2023. Agents meet OKR: An Object and Key Results Driven Agent System with Hierarchical Self-Collaboration and Self-Evaluation. arXiv:2311.16542 [cs.CV]
[159] Ningze Zhong et al. 2024. CASIT: Collective Intelligent Agent System for Internet of Things. *IEEE Internet of Things Journal* 11, 11 (2024), 19646–19656.
[160] Li Zhou et al. 2024. Semantic Information Extraction and Multi-Agent Communication Optimization Based on Generative Pre-Trained Transformer. *IEEE Transactions on Cognitive Communications and Networking* (2024), 1–1.
[161] Xuhui Zhou et al. 2024. Is this the real life? Is this just fantasy? The Misleading Success of Simulating Social Interactions With LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). ACL, Miami, Florida, USA, 21692–21714.
[162] Benhui Zhuang, Chunhong Zhang, and Zheng Hu. 2024. PoSE: Suppressing Perceptual Noise in Embodied Agents for Enhanced Semantic Navigation. *IEEE Robotics and Automation Letters* 9 (2024), 963–970.
[163] Weiming Zhuang, Chen Chen, and Lingjuan Lyu. 2023. When Foundation Model Meets Federated Learning: Motivations, Challenges, and Future Directions. In *NIPS FL@FM*.
[164] Mingchen Zhuge et al. 2023. Mindstorms in Natural Language-Based Societies of Mind. arXiv:2305.17066 [cs.AI]
[165] Mingchen Zhuge et al. 2024. Agent-as-a-Judge: Evaluate Agents with Agents. *arXiv preprint arXiv:2410.10934* (2024).
[166] Mingchen Zhuge et al. 2024. GPTSwarm: Language Agents as Optimizable Graphs. In *Proceedings of the 41st International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 235)*, Ruslan Salakhutdinov et al. (Eds.). PMLR, 62743–62767.