Li et al. Vicinagearth (2024) 1:9
https://doi.org/10.1007/s44336-024-00009-2

REVIEW

Open Access

A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges

Xinyi Li[1], Sai Wang[1], Siqi Zeng[1], Yu Wu[1*] and Yi Yang[2]

Abstract

The pursuit of more intelligent and credible autonomous systems, akin to human society, has been a long-standing endeavor for humans. Leveraging the exceptional reasoning and planning capabilities of large language models (LLMs), LLM-based agents have been proposed and have achieved remarkable success across a wide array of tasks. Notably, LLM-based multi-agent systems (MAS) are considered a promising pathway towards realizing general artificial intelligence that is equivalent to or surpasses human-level intelligence. In this paper, we present a comprehensive survey of these studies, offering a systematic review of LLM-based MAS. Adhering to the workflow of LLM-based multi-agent systems, we synthesize a general structure encompassing five key components: profile, perception, self-action, mutual interaction, and evolution. This unified framework encapsulates much of the previous work in the field. Furthermore, we illuminate the extensive applications of LLM-based MAS in two principal areas: problem-solving and world simulation. Finally, we discuss in detail several contemporary challenges and provide insights into potential future directions in this domain.

Keywords Large language model, Multi-agent system, Systematic workflow

1 Introduction

Enhancing the reliability and intelligence of autonomous intelligent systems has long been regarded as a highly promising research avenue. With the advent of the agent concept, which refers to an entity capable of perceiving its environment and taking action, agent-based intelligent systems have garnered considerable attention in recent years. Historically, RL-based intelligent systems have dominated this field, wherein agents are typically assigned to perform simple, well-defined actions or tasks with constraint interaction with their environment. However, this approach has inherent limitations in terms of adaptability and complexity, prompting the exploration of more advanced and interactive agent-based systems.

*Correspondence:
Yu Wu
wuyucs@whu.edu.cn
[1] School of Computer Science, Wuhan University, Wuhan 430072, China
[2] ReLER, CCAI, Zhejiang University, Hangzhou 310027, China

Large language models (LLMs) have demonstrated exceptional potential in reasoning and planning, aligning precisely with the human expectations for LLM-based agents capable of perceiving their surroundings, making decisions, and taking actions within an interactive environment. Motivated by this, LLM-based agents have made significant strides in interacting with complex environments and solving intricate tasks across a wide range of applications [1], akin to human life in society. Notably, LLM-based multi-agent systems have been proposed as a pivotal pathway to harness collective intelligence while preserving the specialized characteristics of individual agents, thereby advancing toward more sophisticated autonomous intelligent systems. Specifically, multiple specialized agents, endowed with distinct identities, engage in communication and collaboration to achieve task objectives. This process underscores the importance of inter-agent communication, reasoning with knowledge and experience to generate decisions, and evolution (reflecting on its actions and behaviors for achieving personal growth) within the interactive environment.

Consequently, an increasing number of research studies employ LLM-based multi-agent systems to tackle a variety of complex tasks, such as industrial engineering [2–4], scientific experimentation [5–7], embodied agents [8–10], gaming [11–13], and societal simulation [14–16]. However, previous works have been independently executed, lacking a systematic and comprehensive synthesis of the framework structure of LLM-based multi-agent systems. There is a need to clarify the construction of the system, collate application methods for each module, summarize the diverse application scenarios, and identify the existing challenges and opportunities in this field. This forms the core of our paper, where we elucidate our work clearly based on the workflow of LLM-based multi-agent systems.

In this paper, we conduct a comprehensive and systematic survey of the field of LLM-based multi-agent systems. Specifically, following the workflow of LLM-based multi-agent systems, we organize our survey around three key aspects: construction, application, and discussion of this field. For system construction, we introduce a unified agent framework comprising five essential modules: (1) Profile: how agents are created and endowed with personalized characteristics in Sect. 3.1; (2) Perception: how agents perceive environmental information to acquire knowledge and experience in Sect. 3.2; (3) Self-Action: how agents utilize memory mechanisms to store information, and how they perform reasoning and planning to undertake complex tasks in Sect. 3.3; (4) Mutual interaction: how agents communicate with each other in Sect. 3.4; (5) Evolution: how agents achieve self-reflection to progressively enhance their intelligence and experience in Sect. 3.5. Additionally, we systematically overview the various applications of LLM-based multi-agent systems in two main areas: problem-solving and world simulation. Finally, we address several challenges faced by LLM-based multi-agent systems and provide insights into potential future directions in this field. The overall framework is displayed in Fig. 1.

In summary, this paper establishes a holistic yet detailed cognitive framework for existing studies within the burgeoning field of LLM-based multi-agent systems. Our focus centers on the workflow of LLM-based multi-agent systems, encompassing the sequential steps of agent creation, perception, self-action, mutual interaction, and evolution. Drawing from an extensive body of

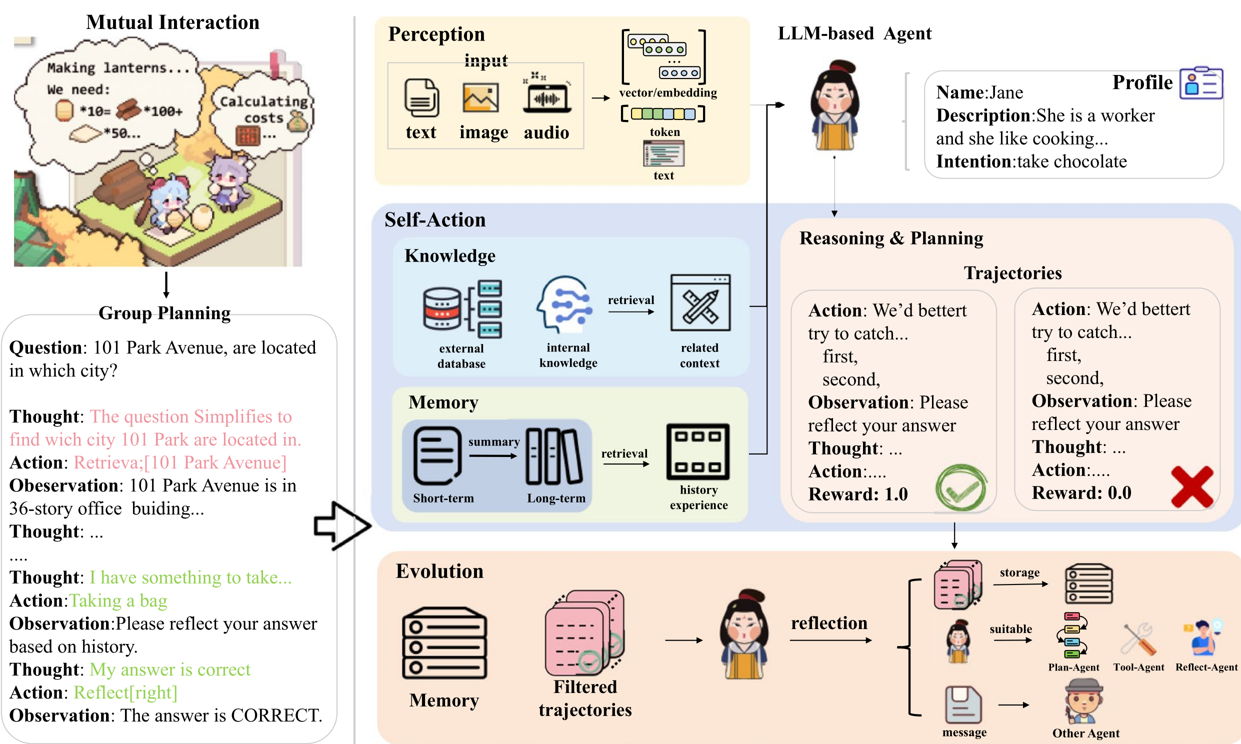

Fig. 1 Overview of the general multi-agent system. Typically, in a multi-agent system, the initial step involves the creation of profiles that endow each agent with personalized characteristics and subtask allocations. Based on the task planning, the agent formulates specific plans to perceive multi-modal information from the interactive environment, accesses external knowledge, and retrieves their historical experiences and knowledge from memory. Utilizing the profound abilities of LLMs, agents are able to devise concrete action plans. Simultaneously, agents engage in evolution, which involves the ongoing reflection on their decisions and actions. Throughout this process, the execution of tasks relies on the interactions among agents, which collectively contribute to the planning and implementation of the overall mission

prior research, we systematically categorize the diverse applications and challenges faced by LLM-based multi-agent systems. We anticipate that our survey will serve as a foundational yet comprehensive guide for beginners in this field, providing readers with a thorough understanding of LLM-based multi-agent systems (LLM-MAS). Readers will gain insights into the fundamental modules essential for establishing multi-agent systems based on LLMs and become acquainted with the latest research trends and applications in this dynamic domain. Acknowledging that this field is in its nascent stages and rapidly evolving with innovative methodologies and applications, we expect that our survey will inspire further exploration and innovation within this domain, as well as novel investigations across interdisciplinary fields.

2 Background

2.1 Single agent

A single-agent system consists of a single LLM-based intelligent agent capable of independently perceiving its environment and making decisions. The design of single-agent systems aims to perform specific tasks, ranging from simple automation to complex decision-making. The core of a single-agent system lies in the individual characteristics, perception abilities, and self-action capabilities of the agent [17–20]. From the perspective of individual characteristics, a single agent is endowed with a set of unique attributes and capabilities that define its behavior patterns and role within the environment. These attributes may include the agent's goals, knowledge, skills, and modes of interaction with other agents. The perception aspect involves how the agent understands and interprets the external world through its sensory system, which typically includes receiving and processing information from sensors or other data sources to form an understanding of the environment. Finally, self-action refers to the agent's ability to make decisions and execute actions based on its perception and internal state, with these actions aimed at achieving its goals or responding to environmental changes. Together, these three aspects constitute the basic framework of a single agent, enabling it to operate independently in specific tasks or environments and interact effectively with the external world.

A notable advantage of single-agent systems lies in their focus and efficiency. Due to the concentration of system resources and computational capabilities on a single agent, these systems can quickly respond to and execute specific tasks. This centralized processing reduces the need for resource allocation among multiple agents, thereby improving overall efficiency. Furthermore, compared to multi-agent systems, the design and maintenance of single-agent systems are simpler and more straightforward. They do not require complex communication and coordination mechanisms, reducing system complexity and simplifying the process of troubleshooting and performance optimization.

2.2 Multi agents

Although single-agent systems excel in specific tasks, they may encounter limitations when dealing with complex problems that require extensive collaboration and collective intelligence. This is where multi-agent systems (MAS) come into play. MAS is a complex system composed of multiple interacting intelligent agents [21], capable of simulating social interactions and teamwork in the real world, enhancing overall adaptability and efficiency through decentralized decision-making processes and information sharing.

The core advantage of MAS lies in its distributed decision-making and problem-solving capabilities. As pointed out by [22], MAS provides a modern approach to address distributed artificial intelligence problems. In MAS, each agent possesses a degree of autonomy, capable of independently perceiving the environment and making decisions. They can also interact with other agents by simulating real-world collaboration patterns such as cooperation, competition, and hierarchical organization [19, 23], thereby enhancing overall collaborative efficiency. This is typically achieved through the integration of control theory and reinforcement learning methods. Reinforcement learning (RL), as a core MAS technology, allows agents to learn optimal behavioral strategies through interaction with the environment. Building on this foundation, Marllib [24] distinguish MARL algorithms based on four dimensions: task patterns, agent types, learning styles, and knowledge sharing.

The structure of MAS determines how agents organize and interact. MAS is divided into agent-level and system-level characteristics in [25]. Similarly, In [26], four different MAS prototypes were summarized based on the dimensions of agent heterogeneity and communication level. For agent systems, MAS employs diverse agent architectures, including combinations of homogeneous and heterogeneous agents; homogeneous agents perform similar tasks in the system, while heterogeneous agents collaborate based on their specific abilities and expertise. This structural design enables MAS to adapt flexibly to different task requirements and environmental changes, while also promoting complementarity and synergy among agents. Regarding system architecture, communication is the most crucial part. The four-dimensional framework proposed by [27] emphasizes the diversity of communication protocols, distinguishing between blackboard and message-based systems, and setting different gradients from low to high-level content. In [23], the work proposes that communication mechanisms can be divided into three parts: communication paradigms, communication structures, and communication content. Among them, communication paradigms refer to interaction styles, while communication structures are categorized into four types, including decentralized, centralized, layered, or nested, to adapt to different task requirements and environmental conditions.

In recent years, with the rapid development of LLMs, LLM-based MAS has begun to emerge. In such systems, agents leverage the powerful capabilities of LLMs for natural language understanding and generation, enabling more complex and flexible interactions. LLM-MA systems, through diverse agent configurations, agent interactions, and collective decision-making processes, can address more dynamic and complex tasks. In [19], the work uses a graph $G(V, E)$ to represent relationships among multiple LLM-based agents. Here, V is the node set, V_i represents an LLM-based agent, E is the edge set, and E_{ij} represents message transmission and relationships between LLM-based agents V_i and V_j. It is also proposed to classify LLM-based MAS based on two factors: multi-role coordination and planning types.

3 LLM-based multi-agent work

The LLM-based multi-agent system has been applied to execute a variety of complex tasks and downstream scenarios. From the perspective of the system's workflow, we meticulously explore the lifecycle of agents, including their creation, perception, reasoning, action, and self-learning processes. Motivated by this exploration, this section constructs a comprehensive unified framework for LLM-based multi-agent systems, which comprises five critical functional modules: the profile, perception, self-action, mutual interaction, and evolution. A detailed analysis of current works from various perspectives is presented in Table 1.

For each sub-task execution, the profile module in Sect. 3.1 initially generates LLM-based agents, each with specific characteristics according to the task objectives. The perception module in Sect. 3.2 captures essential information to understand the current interactive environment. Specifically, the self-action module in Sect. 3.3 integrates historical knowledge and experiences stored in memory, supplemented by external knowledge, and perceived information, to make decisions and generate plans using reasoning, planning, and generalization abilities. The mutual interaction module in Sect. 3.4 facilitates communication and collaborative coordination among agents. The evolution module in Sect. 3.5 enhances the agents' cognitive and task-handling capabilities through self-reflection during environmental interactions. For each module, we systematically organize the execution strategies from the workflow perspective of task execution. The following sections will discuss these five key modules in detail.

3.1 Agent profile

LLM-based multi-agent systems typically perform complex task execution or simulate intricate scenarios by assuming various roles [32, 40, 55]. The definition of these roles involves the meticulous crafting of agent profiles, ensuring that each agent is well-suited to its designated function. The Agent's profile is designed to instantiate independent intelligent entities with personalized styles, akin to a person, thereby enabling them to accomplish specific sub-tasks [30, 56]. For example, in the context of simulating the operations of a school, appropriate roles would include teachers, students, and the principal. Corresponding agent profiles should be meticulously designed to create intelligent agents that accurately represent these roles, which get involved in the simulated school environment.

3.1.1 Profile context

In accordance with specific contexts or user specifications, agent profiles may encompass varying types and contents of information. Serving as the fundamental intrinsic traits of the agent, the profile typically encompasses basic information such as name, age, gender, and career [28, 43, 57]. Additionally, the profile may include psychological attributes like current emotions, personality traits, and life goals, thereby reflecting the distinct personalities of the agents [58, 59]. Moreover, the profile may summarize social relationships and environmental contexts relevant to the agents' interactions [40, 60]. Furthermore, restrictive information may be incorporated to delineate behaviors that the agent is not permitted to engage in.

3.1.2 Generation strategy

The selection of information to profile the agent is predominantly dictated by the specific application scenarios, thereby guiding the trajectory of profile generation. In light of the intricate relationship between scenario modeling and agent generation, existing literature commonly adopts the following three strategies.

Contextuliazed Generation Method. In this method, the analysis and decomposition of complex scenarios lead to the concretization of agents tasked with executing various sub-tasks. For instance, within a corporate setting, the workflow of a task-comprising encompasses decision-making, distribution, execution, and feedback-necessitates the collaboration of four agents: a manager, a secretary, regular employees, and consultants. This method has historically been predominant in prior works. For example, Generative

Table 1 A review of representative works on llm-based multi-agent systems

Work	Object	Modality	Base model	Train	Feedback	Evaluation	Interaction
Generative Agent [28]	Sociology (25 agents)	Text	GPT3.5-turbo	No	Environment, Agent Interaction	-	-
Planner-Actor-Reporter [29]	Embodied environments	Vision, Text	-	Yes	Environment	win rate	Cooperative
ChatDev [30]	Software Development	Text	Domain-specific Model	No	Environment, Agent Interaction, Human	on dataset, with models	Cooperative
MetaGPT [31]	Software Development	Vision, Text	Domain-specific Models	Yes	Environment, Agent Interaction, Human	on dataset, with models	Cooperative
Dong et al. [32]	Software Development	Text	GPT-3.5	No	Environment, Agent Interaction	on benchmark, with models	Cooperative
Chen et al. [33]	Multi-robot Planning	Vision, Text	GPT-4-0613, GPT-3.5-turbo-0613	No	Environment, Agent Interaction	with frameworks	Cooperative
Roco [34]	Multi-robot collaboration	Vision, Text	GPT-4	No	Environment, Agent Interaction	on dataset	Cooperative
Zhang et al. [35]	Multi-Agents Cooperation	Vision, Text	GPT-4	Yes	Environment, Agent Interaction	with models	Cooperative
Du et al. [36]	Improving Factuality	Text	GPT-based model	No	Agent Interaction	on dataset	Debate
Xiong et al. [37]	Examining Inter-Consistency	Text	6 LLMs	No	Agent Interaction	on dataset, with models	Debate
ChatEval [38]	Evaluators for debates	Text	GPT-4, GPT-3.5-turbo	No	Agent Interaction	on dataset	Debate
Medagents [39]	Medication	Text	GPT-4, GPT-3.5-Turbo	No	Agent Interaction	on dataset	Debate, Cooperative
Social Simulacra [40]	Sociology (1000 agents)	Text	GPT-3	No	Agent Interaction	on dataset, Human	-
S3 [14]	Emotion propagation	Text	GPT-3.5, ChatGLM	Yes	Agent Interaction	on dataset	-
Lyfe Agents [41]	Real-time social interaction	Vision, Text	GPT-3.5	No	Environment, Agent Interaction	experimental scenarios	-
Li et al. [42]	Opinion dynamics	Text	-	No	Agent Interaction	on benchmark	-
Xu et al. [43]	WereWolf	Text	GPT-3.5-turbo-0301	No	Environment, Agent Interaction	win rate	Mixed
Avalonbench [44]	Avalon	Text	GPT-3.5, Llama2	No	Environment, Agent Interaction	win rate, with models	Mixed
Welfare diplomacy [45]	Welfare Diplomacy	Text	-	No	Environment, Agent Interaction	with models	Mixed
Aher et al. [46]	Human behavior Simulation	Text	GPT models	No	Agent Interaction	on dataset, Human	-
Zhang at.all [47]	Exploring Collaboration	Text	GPT-3.5-turbo-1106	No	Agent Interaction	on dataset, with models	Mixed
Agent4Rec [48]	Recommender Systems (1000 agents)	Text	GPT-3.5-turbo	No	Environment	on dataset, Human	-
AgentCF [49]	simulating user-item interaction	Text	-	No	Environment, Agent Interaction	on dataset, with models	Cooperative
EconAgent [50]	Macro-economic simulation	Text	GPT-3.5-turbo-0613	No	Agent Interaction	with models	Cooperative
Weiss et al. (Weiss et al.: Rethinking the buyer's inspection paradox in information markets with language agents, under review)	simulated Marketplaces	Text	Llama 2	No	Environment, Agent Interaction	on dataset, with models	Mixed

Table 1 continued

Work	Object	Modality	Base model	Train	Feedback	Evaluation	Interaction
Tradinggpt [51]	Improving financial trading	Multi	GPT-3.5 turbo	Yes	Environment, Agent Interaction	on dataset, with models	Adversarial
Williams et al. [52]	Epidemiology research	Text	GPT-3.5-turbo-0301	No	Environment, Agent Interaction	-	Cooperative
Boiko et al. [6]	Chemistry	Multi	GPT-3.5, GPT-4	No	Environment, Agent Interaction	-	Mixed
GPT4IA [3]	Industrial environment	Multi	GPT-models	Yes	Environment, Agent Interaction	-	Cooperative
ProAgent [53]	Team cooperation	Multi	-	No	Environment, Agent Interaction, Human	with models	Mixed
SAMA [54]	Game	Text	GPT-3.5, GPT-4	No	Environment, Agent Interaction	with models	Cooperative

We present current representative works, providing a detailed analysis of each work from different perspectives, including object, modality, base model, training, feedback, evaluation, and interaction. "-" denotes that a particular element is not specifically mentioned in this work

Agent [28], immersed in the context of a software company, utilizes authored natural language descriptions to define each agent's identity, encompassing their occupation and inter-agent relationships, serving as seed memories. MetaGPT [31] specify the agent's profile, which includes their name, profile, goal, and constraints for each role, and then initialize the specific context and skills for each role in the context of computer game software engineer task. ChatDev [30] incorporates essential details pertaining to the assigned task and roles, communication protocols, termination criteria, and constraints designed to avert undesirable behaviors. In general, the contextualized generation method flexibly determines the types and contents of agent profiles based on the context of the complex task, ensuring optimal alignment with task requirements. However, this approach is both one-time and labor-intensive, as it necessitates the regeneration of agent profiles for each new scenario.

Pre-defined Method. In this method, Large Language Models (LLMs) are broadly employed to define multiple agents, collectively forming an agent pool. When faced with specific scenarios, suitable agents are selected from this pool to execute relevant sub-tasks. Typically, the process commences with the delineation of profile generation rules, clarifying the composition and attributes of agent profiles within the prompt, which the LLM utilizes to generate agents with distinct characteristics. Subsequently, appropriate agents are either manually designated or selected by the LLM to assume various roles and immerse themselves in the complex task. Finally, LLMs are responsible for updating the agents' state information to facilitate their recovery or subsequent actions. For instance, SpeechAgents [58] initially generates seed profiles for a limited number of agents by meticulously crafting their backgrounds, encompassing aspects such as age, personal traits, and movie preferences. This structured methodology ensures that agents are well-defined and adequately equipped to perform their designated roles effectively within the task environment. Similarly, In [61], it is focused on assigning roles to GPT-3 based on the demographic backgrounds-such as race and ethnicity. While the pre-defined method significantly reduces the time required when the number of agents is large, it may lack precise control over the generated profiles, potentially limiting the customization and accuracy of agent behaviors.

Learning-based Method. In this method, a few agents are initially defined in broad terms. When specific scenarios arise, these pre-defined agents execute sub-tasks while new agents are subsequently generated to handle brand-new tasks, thereby adapting to new circumstances. Typically, the creation of new agents during task execution leverages LLMs, which automatically generate agents by combining previous agent profiles with profile generation rules. The LLM then assigns sub-tasks to these newly generated agents to accommodate evolving situational demands. In self-collaboration [32], different roles with their associated responsibilities are predefined within the context of software development, with distinct profiles meticulously assigned to each agent to facilitate enhanced collaboration. For instance, RecAgent [60] initially constructs profiles for a limited number of agents by manually detailing attributes such as age, gender, personal traits, and movie preferences. Following this, ChatGPT is employed to generate additional agent profiles based on the initial seed information, thereby ensuring the creation of a comprehensive and adaptable agent pool. This method integrates the advantages of the aforementioned approaches, providing increased flexibility in defining agent profiles for specific scenarios while saving much time. However, there are potential pitfalls in the generation of new agents, such as large model hallucinations and mismatches between generated profiles and corresponding tasks.

3.2 Perception

Most humans and animals acquire information through sensory organs such as the eyes, ears, and hands, which serve as crucial determinants of individual cognition and behavior. Similarly, information acquisition is vital for agents as independent intelligent entities, enabling them to perceive external environmental conditions and their internal states. This information is then converted into intermediate representations through perception modules, which then determine the agent's autonomous decision-making outcomes and behavioral responses [62, 63].

Owing to the exceptional text processing capabilities of LLMs [64–66], previous work has predominantly utilized textual messages as the medium for information perception and dissemination. In these studies, extracting textual information from the external environment requires specialized models to convert information into text [67–69], while the internal state information of LLMs relies on the models themselves to extract and summarize knowledge in textual form [70]. The advent of multi-modal Large Language Models (MLLMs) has shifted this paradigm, facilitating the transition from unimodal to multi-modal information perception and unifying the modalities [71] in a manner more akin to human perception. For LLM-based agents, it is crucial to receive information from diverse sources and modalities. This expanded perceptual framework enhances the agents' understanding of their environment and internal states, enabling them to make more intelligent decisions and exhibit more sophisticated behaviors. Consequently, this capability broadens their proficiency across a wider array of tasks, establishing it as a critical direction for future development.

In this section, we discuss the sources from which LLM-based agents perceive multi-modal information in Sect. 3.2.1, and the methodologies that endow LLM-based agents with multi-modal perceptual abilities in Sect. 3.2.2.

3.2.1 Message source

When immersed in specific scenarios, LLM-based agents perceive, process, and generate messages during interactions and communications, which serve as crucial conduits for the agents to collaboratively accomplish complex tasks. According to the nature of the agents' interactions or communication counterparts, the existing literature categorizes the sources of perceptual information for agents into the following three types:

Entire Environment Message. This type of message conveys basic information about the agents' surrounding environment, such as scene location, layout, and furnishings, as well as time-sensitive information like scene transitions and facility changes [72]. Additionally, it considers emotionally nuanced information such as ambiance and atmosphere [73, 74]. Such information is intimately linked to the task scenarios and has consistently held significance in previous works, whether in single-agent or multi-agent settings. Initially, this information is initially determined by the user-defined task scenario. However, it can be automatically generated by the agents themselves or by additional LLMs, especially during agents interaction. Typically, this message arises from interactions between agents and the inherent elements of their environment, leading to changes in agent behavior and updates to the surrounding environment. Occasionally, it serves as supplementary background information [35, 75], influencing both the agents' self-interactions and their interactions with one another.

Interaction Message. This category of message encompasses information exchanged during interactions between agents, with content that is flexibly determined based on task requirements or simulated scenarios. For example, in a communication-based scenario, the message content pertains to dialogue information between agents on a specific topic. Each interaction message usually signifies an independent information exchange between two specific agents, characterized by individual specificity and temporal relevance, given the multi-round nature of inter-agent communication [76, 77]. Such messages are generally autonomously generated by the interacting agents, though occasionally they may be produced by additional LLMs as control signals directed to specific agents. Serving as the primary medium for inter-agent communication and interaction, these messages predominantly influence each agent's decision-making outcomes and behavioral responses.

Self-Reflection Message. This message typically hints at the self-reflection and self-updating processes of agents, containing a blend of historical messages generated by the agent itself, interaction messages resulting from interactions with other agents, and comprehensive environment background information [78, 79]. These pieces of information serve as guiding indicators for the agent to engage in introspection and generate updating signals, corresponding to the agents' own changes and innovations.

The behavior of the agent's self-reflection are influenced by various factors, which may stem from user-defined task scenarios, be guided by additional LLMs through control signals, or autonomously generated by the agent itself within their designated settings [80, 81]. Rooted in perceived information, this message is typically generated internally within the agent, leading to changes in the agent's decision outcomes and behavioral transformations, thereby better aligning with the requirements of the task.

3.2.2 Message type

After taking into account various sources of information messages, we also emphasize the importance of enabling agents to receive and comprehend multiple modalities of information through perception modules [82]. This section delves into the methods by which LLM-based agents can attain multi-modal perception capabilities, encompassing textual, visual, and auditory inputs, aiming to enrich the agent's perception domain and bolster its adaptability and versatility.

Textual Message. Text serves as a fundamental and intuitive representation of human perceptions towards the surrounding environment and their subjective experiences, making text communication is the primary approach for humans to interact with the world. Considering the preferences of LLMs for text-based inputs and outputs, LLM-based agents also utilize textual messages as the principal information medium for interaction and dissemination. Textual messages encompass raw textual information, such as environmental descriptions, textual outputs from other agents, and the agent's own textual data. They also include converted data derived from other modalities, for example, caption information extracted from images via visual models [83, 84]. These messages cover a broad spectrum of information, including dialogues, task planning, feedback, etc.

For LLM-based agents, the primary task is to comprehend, analyze, and synthesize complex and lengthy texts, akin to the capabilities of human experts. This leverages the core functionalities of LLMs: understanding, reasoning, and generation. Some research efforts have enhanced the understanding and reasoning capabilities of LLMs through in-context learning [85] and chain-of-thought (CoT) reasoning [86], aiming to produce outputs that align more closely with human cognitive preferences and real-world situation. Similarly, prompt engineering and fine-tuning techniques have been employed for more accurate outputs [87, 88]. Furthermore, some studies focus on analyzing and understanding the implicit meanings and emotional content within texts. For instance, certain approaches employ reinforcement learning to interpret implied meanings and model feedback to derive rewards [89–91]. Some other methods rely on expert models for fine-grained linguistic analysis to achieve a deeper understanding of textual nuances (Ye et al.: Tooleyes: Fine-grained evaluation for tool learning capabilities of large language models in real-world scenarios, under review) [92], which helps deduce the speaker's preferences and leads to more personalized and accurate agent responses. Additionally, LLM-based agents must be capable of responding promptly to novel situations encountered in complex real-world scenarios. This underscores the importance of enhancing the agents' abilities to perceive and understand new tasks through text. In certain works, LLMs that have undergone instruction tuning demonstrate impressive zero-shot instruction understanding and generalization abilities [93], obviating the need for task-specific fine-tuning. While some approaches introduce an additional module to incorporate external knowledge, thereby endowing the LLM with a more comprehensive understanding of new tasks.

Visual Message. Concurrently, textual messages have predominantly served as the primary medium for information exchange and dissemination when LLM-based agents interact with the world. However, textual messages fall short in capturing and conveying the nuanced characteristics that visual information can adeptly represent. For instance, they cannot as effectively communicate the detailed properties of objects, the subtle spatial relationships between agents, or the intricate atmospheric conditions [71]. Consequently, integrating visual information can provide the agent with a richer context and a more precise understanding, thereby deepening the agent's perception of the scene within interactions and communications.

To equip agents with the ability of comprehending visual information, previous work has either employed visual language models(VLMs) as adapters to extract visual features and integrate them into the LLM's knowledge base, or added parallel network layers integrated with the LLM to function as visual feature perception modules without requiring additional processing of visual information. Regarding the extraction of visual features, a straightforward approach involves generating corresponding textual descriptions for visual messages through visual caption models [94–96] or user definitions [83, 84, 97]. These textual captions can then be directly fed into the LLM-based agent alongside other textual information for comprehension and analysis. This method is simple and direct, requiring minimal modifications to the LLM. However, the accuracy and detail of the textual captions in conveying visual perception are heavily dependent on the specific VLM employed, which generally only produces broad and coarse descriptions of visual images. Consequently, this captioning approach often loses much of the implicit visual information, potentially leading to deviations in the agents' decision-making processes.

Building on the impressive performance of GAN models in visual information processing, a significant part of prior work has utilized GAN architectures to encode visual information into visual vectors within the generator's latent space [81, 98]. During the training process, reconstruction loss is employed to ensure that the images generated by generator G from the input visual vectors closely resemble the original images, thereby aiming to deceive the discriminator. However, this method often results in latent vectors that lack interpretability, making it challenging to directly understand the extracted visual features and their relevance to the corresponding tasks.

Another representative approach, exemplified by works like ViT and VQVAE, encodes visual information into visual tokens typically based on transformers, similar to how LLMs process textual information by converting text into discrete tokens [99–102]. This method begins by segmenting the image into fixed-size patches, which are then flattened and mapped to a high-dimensional vector space through linear layers. Positional encoding is subsequently added to retain the spatial information of the image patches. The position-encoded patches are then embedded into a sequence and fed into a Transformer encoder. For each layer, based on the self-attention mechanism, the similarity between all image patches in the input sequence is computed using query, key, and value vectors. Through these steps, the visual encoder output finely represents both the global and local features of the image, resulting in a highly effective means of visual content perception. Consequently, current works typically integrate the visual encoder as an additional module within the LLM to achieve end-to-end processing of images and text, or employ it as an adapter to provide pre-converted visual tokens to the LLM. Although this approach significantly enhances the granularity and accuracy of visual perception, it imposes substantial demands on computational resources and exhibits suboptimal performance on small-sample tasks.

Furthermore, to directly align image encodings with the intermediate data representations within the LLM, concurrent research has concentrated on transforming image encodings into visual embeddings, which are subsequently integrated with other modality information. Typically, after obtaining a feature vector from the visual encoder, an additional learnable interface layer is employed to align the visual feature vector with the LLM's textual embeddings. When integrating visual information with other modalities, some prior work has adopted the approach of directly concatenating embedding vectors from different modalities to form a joint embedding. However, this method may overlook the complex inter-modal relationships. While some studies have used weighted summation of embedding vectors from different modalities, with weights being either fixed or dynamically learned, but this approach requires meticulous design for weight adjustments and may struggle to handle intricate relationships flexibly. Additionally, certain research leverages attention mechanisms to facilitate information exchange and fusion between different modal embeddings. For instance, BLIP-2 [103] and InstructBLIP [104] employ the Querying Transformer (Q-Former) module as an intermediate layer between the visual encoder and the LLM, while in [105] and [81], they compute cross-modal attention maps to combine textual embeddings and visual features by using GPT-4V. These methods significantly enhance the LLM's capability to extract language-informative visual representations, thereby deepening its perception of critical aligned information across multi-modal data. Simultaneously, some researchers have adopted a single projection layer to achieve visual-text alignment, which is efficient method by reducing the need for training additional parameters [106–108]. The projection layer can dynamically adjust to the dimensions of LLM's textual embeddings, providing flexibility while ensuring stability in multi-modal data integration.

For video perception, compared to images, there is a greater emphasis on the continuity and variability of the temporal dimension. Typically, this method involves converting videos into a series of image frames extracted at specific intervals. Consequently, agents can leverage their image perception capabilities to understand and interpret video content, which necessitates additional attention to the transitions and changes across the sequence of frames [109–111]. For instance, previous research efforts like Flamingo [112] and VideoAgent [113] extract video frames at certain frequencies and rigorously follow the chronological order to perform visual understanding on each frame. However, some approaches focus on end-to-end video comprehension, employing an interface layer to input the entire video as a unified entity. Video perception aligns more closely with real-world complex environments, broadening the LLM-based agent's perceptual dimension and enhancing its sensitivity to interactive settings.

Auditory Message. Audio information encompasses a diverse array of types and content, broadly classified into environmental sounds (such as birds chirping and wind rustling through trees), music, and speech, with speech specifically referring to sounds produced by humans.

Due to its unique time-frequency characteristics, audio messages convey perceptual information that text and visual data cannot replicate. Generally, audio information includes not only linguistic text but also various linguistic elements such as tone, intonation, rhythm, and emotional nuance. Additionally, it often implicitly indicates occurrences or positional changes of objects within the environment.

Similarly, as LLM-based agents perceive visual information, their perception of audio information can be broadly categorized into three types. Previous work, such as WavJourney [114] and AudioLM [115], has often employed audio encoders to convert audio into discrete tokens, which are then integrated into the LLM's knowledge base. However, this approach neglects the temporal continuity characteristics of audio. Another representative method encodes audio information as latent vectors within diffusion models' latent space [116, 117], enhancing feature computation efficiency, but the extracted low-dimensional audio features might be overly simplistic. A prevalent approach is to represent audio information as embeddings [118, 119], which usually requires integration with data from other modalities. This involves extracting aligned features using methods such as fully connected layers, multi-head cross-attention, and Q-Former as connectors for integrating ASR models with LLMs. This approach not only captures fine-grained audio features but also reduces training time and computational costs by freezing encoders. An interesting alternative method involves converting the perception of audio information into the encoding of visual information [120–122], as audio spectrograms can be visualized as flat images. For instance, AST (Audio Spectrogram Transformer) [120] employs a Transformer architecture to process audio spectrogram images, effectively encoding audio information by segmenting the spectrogram into patches.

After considering the in-depth perception and understanding of audio information by LLM-based agents, another significant focus of previous work has been leveraging the excellent tool-using capabilities of LLMs. These agents function as control hubs, enabling the flexible invocation of existing toolsets or model repositories in a cascading manner to achieve downstream audio applications, such as audio understanding and audio editing. AudioGPT [123] and HuggingGPT [124] exemplify the use of LLMs for audio understaning by orchestrating tools through LLM-driven interfaces. Specifically, AudioGPT employs ChatGPT as a central node for audio and speech applications, relying on external audio systems for various functionalities. Similarly, HuggingGPT operates as an agent that synergizes ChatGPT's linguistic capabilities with a diverse array of AI models from the Hugging Face community, thereby enhancing its proficiency in understanding audio content.

As previously discussed, numerous studies have investigated perception units for text, visual, and audio inputs. Nevertheless, LLM-based agents possess the potential to be endowed with more extensive perceptual capabilities, enabling them to process a diverse array of modalities in the real world. These modalities encompass 3D point cloud maps, GPS location data, human pose information, among others [125–127]. Furthermore, these advanced perception abilities can be seamlessly integrated with traditional data modalities, creating a richer and more comprehensive sensory framework. This expanded perceptual spectrum allows agents to engage with their environments in a more nuanced and comprehensive manner.

3.3 Self-action

In social contexts, the human, as an autonomous entity, processes perceived information to form memory units, construct their cognitive awareness, develop individual thoughts, and undertake actions [128]. Similarly, self-action represents a pivotal mechanism for the agent, functioning as an independent entity, to make autonomous decisions and perform actions necessary for their survival and evolution in the interaction environment. This section delves into the detailed processes by which individual agents autonomously learn and reason within their environments. Upon receiving perceived information, the self-action module initially invokes memory in Sect. 3.3.1 to extract relevant historical experiences, possibly supplemented by additional knowledge in retrieval from external knowledge bases in Sect. 3.3.2. This amalgamation of information serves as the context, aiding the agent in reasoning, planning, and generalization in Sect. 3.3.3, ultimately culminating in decision-making. Based on these decisions, the agent executes corresponding actions to achieve real-world interactions in Sect. 3.3.4. Concurrently, during the processes of thinking and action, agents engage in self-updating and evolution of memory by comparing historical experiences, current knowledge, and newly generated insights. In the subsequent sections, we will delineate the components of the self-action module in detail.

3.3.1 Memory

The memory module, serving as the storage and recall unit for the agent, is instrumental in allowing it to leverage existing cognitive and experiential knowledge to adapt to dynamic interactions with the environment or other agents [30, 58]. Through this process, agents accumulate new insights and experiences, which can further enhance their cognitive abilities and intelligence by

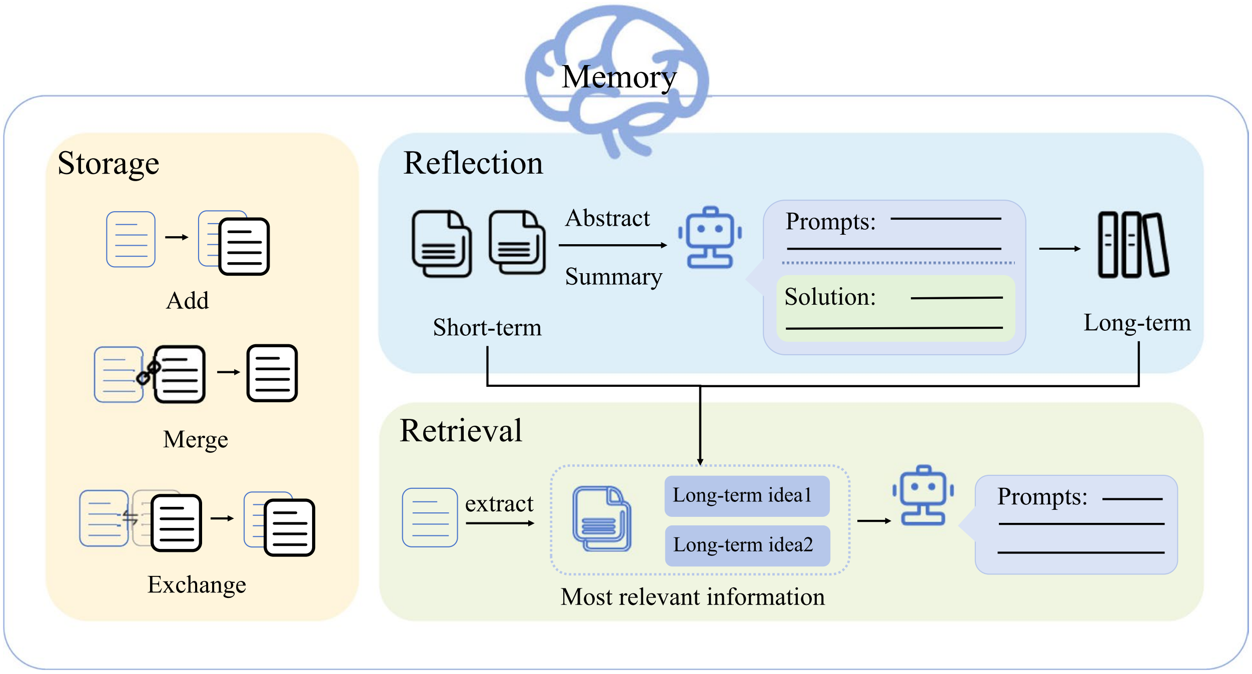

Fig. 2 The operational mechanism of the memory module

updating the memory [129, 130]. The core functionalities of the memory visualization are illustrated in Fig. 2. This capability is crucial for the agent as an independent intelligent individual to flexibly navigate complex environments and tackle novel tasks. The realization of this adaptive functionality is primarily achieved through three critical memory operations [17]: memory retrieval, memory storage, and memory reflection.

Memory Retrieval In the realm of intelligent agents, effective information retrieval is paramount for facilitating dynamic interactions within complex environments or other agents, and retrieval information is always treated as substantial experiential references. Memory retrieval aims to enhance decision-making accuracy by extracting valuable information pertinent to the current situation from an agent's memory. This information encompasses various elements such as environmental perception, records of historical interactions, experiential data, and external knowledge. In scenarios involving short-term memory [30, 131], the retrieval module typically extracts the entire body of information as content. However, when dealing with long-term memory, the retrieval module generally employs filtering mechanisms to discern and present only the most relevant memories to the agent [130, 132, 133]. This distinction underscores the necessity of tailored retrieval strategies to optimize the utility and relevance of accessed information, thereby bolstering the agent's operational efficacy in diverse contexts.

Retrieval Methods. To maintain the flexibility and dynamic adaptability of agents, memories are retrieved in an automated manner [134, 135]. a pivotal methodology in previous research emphasizes that serving as the context of prompt, memory information is evaluated based on predefined metrics: Recency, Relevance, and Importance [28]. These metrics are used to calculate a weighted score for each memory, with those scoring the highest being prioritized for contextual use, while the model's parameters remain fixed. Another notable approach considers the retrieved information as a learnable representation, such as embeddings and vectors [30, 129, 135], which serve as soft guidance for fine-tuning the model to accommodate various tasks. Techniques such as online reinforcement learning [136, 137], multitask learning [138, 139], and attention mechanisms [140–142] facilitate real-time updates and adjustments to the model parameters, thereby enhancing the agent's responsiveness to evolving tasks and environments.

Retrieval Extension. Several studies have focused on employing LLM-based agents as a central control interface to facilitate downstream applications involving memory management. For instance, some research has designed interactive memory mechanisms for LLM-based agents with the objective of enhancing the operability of memory to allow for more human-like intervention and control [143, 144]. In such systems, information representations can be manipulated, edited, deleted, or amalgamated through summarization. In certain studies, users are empowered to view and manipulate the dialogue history, thereby modifying the agents' history memory. Specifically, in [143], it enabled memory operations, such as deletion, based on user commands to adjust the memory information accordingly. These approaches aim to provide more intuitive and flexible control over the memory systems within LLM-based agents.

Memory Storage Storing critical information in memory constitutes the foundational knowledge base upon which agents rely to perceive and act within complex environments, thereby enhancing their efficiency and rationality. The purpose of memory storage is to archive the information perceived and the experiences learned by agents during interactions. Typically, this process involves writing natural language text into memory, a task that encompasses selecting appropriate storage locations within the memory and managing the replacement of information [64, 145, 146]. This systematic approach to memory storage ensures that the most pertinent data is readily accessible, facilitating informed decision-making and adaptive responses by the agents.

Storage Format. Memory storage is typically realized through the use of natural language formed text [147–149], although it also encompasses multi-modal information such as visual and audio data [58]. The storage format is determined by the specific nature of the task and the attributes of the data modality. By tailoring the storage format to the modality and task requirements, agents can more effectively utilize stored information, thereby enhancing their performance in diverse and complex environments.

Using improved data storage structures, existing representative methodologies have achieved more efficient and flexible information storage within memory. Notably, some studies emphasize on generating condensed memory representations in the reflective processes [28]. For example, several methods adopt embedding vectors to represent memory sections and history dialogues [129, 134, 135]. In [145], it involves translating sentences into triplet configurations, while others perceive memory as a unique data object, facilitating diverse interactions [144]. These varied techniques underscore the ongoing efforts to enhance the functionality and accessibility of memory storage in complex computational environments.

Another effective approach involves adopting more intuitive data interaction methods to achieve effective memory storage. For instance, ChatDB [143]and DB-GPT [150] encompass data manipulation through SQL commands by integrating the LLM with databases. This integration enables a seamless and efficient interface for managing and querying stored data, thereby enhancing the overall efficiency and usability of the memory system.

Storage Methods. When considering the memory writing process, two predominant challenges must be meticulously addressed: the relationship between the new information and the existing memories, and effective information exchange strategy when the memory storage capacity is reached. (1) Memory Modification. When considering the similarity between new information and existing memories, it is crucial to determine the appropriate method of incorporation: whether to add new information, merge it with existing data, or substitute erroneous existing information. For instance, one approach stores successful action sequences with the same subgoal into a single list [64]. When the length of this list exceeds the predefined limit, all entries are compressed into a unified solution by LLMs, which subsequently replaces the original entries in the list. (2) Memory Exchange. Given that memory storage is typically limited, designing an effective information exchange strategy is significant for ensuring that the memory retains the most beneficial information for agents. When considering the writing of new information into a full memory, existing methods employ strategic information exchange mechanisms to maximize the retention of the most proximate and relevant information. For example, RET-LLM [145] utilizes a first-in-first-out (FIFO) strategy to overwrite the oldest entries in a fixed-size memory, while ChatDB [143] deletes irrelevant information to free up memory space. These approaches are critical for ensuring that the memory system remains both coherent and efficient, allowing for optimal information retrieval and utilization in complex environments.

Memory Reflection Memory Reflection is the process through which agents engage in self-improvement based on the perceived information and learned experience from historical interactions stored in memory. This process emulates the human practice of summarizing, refining, and reflecting upon existing knowledge, with the objective of enhancing the agent's adaptability to new environments and tasks.

The memory reflection process typically occurs automatically, with agents independently updating their memory based on newly acquired knowledge, thereby achieving self-recognition updates [129, 130]. In a multi-agent environment, a central LLM-based agent exerts control over the memory reflection of individual agents.

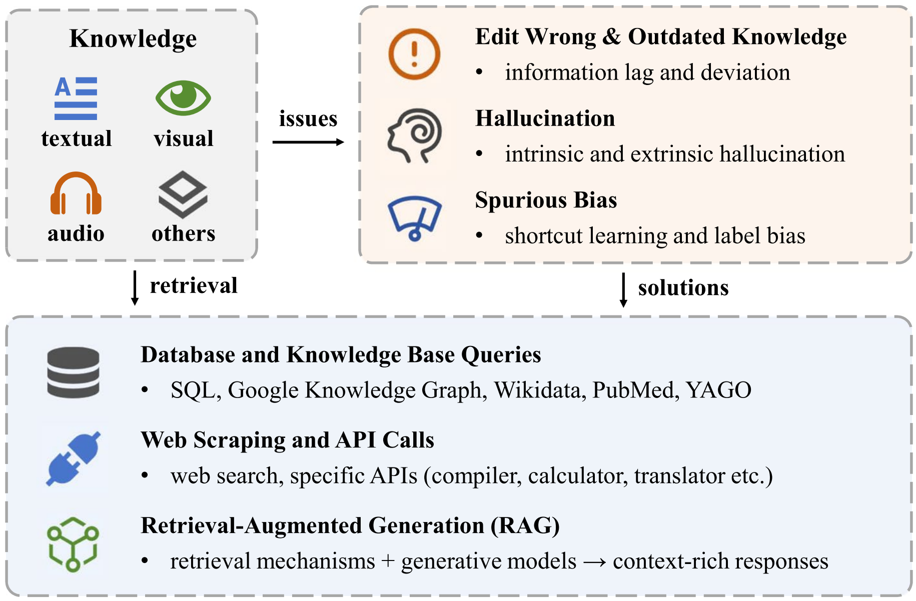

Fig. 3 Knowledge utilization

This central agent sends specific control signals to guide the reflection process, ensuring coherence and coordination across the network of agents. This method facilitates the systematic updating of memory, enabling agents to refine their cognitive models and enhance their adaptability to dynamic tasks and environments.

After establishing the mechanisms underlying memory reflection, it is crucial to carefully consider the content of memory reflection. A significant portion of previous work has focused on hierarchical information storage, emphasizing the abstraction, summarization, and distillation of acquired knowledge and experiences. For instance, in Generative Agent [28], the agent is capable of summarizing its past experiences stored in memory into broader and more abstract insights. This process begins with the agent generating three key questions based on its recent memories. These questions are then used to query the memory to retrieve relevant information. Based on the acquired information, the agent generates high-level ideas. In the ExpeL [130] framework, during task execution, agents learn from the experiences of correct trajectories and derive lessons from incorrect ones. Another significant approach focuses on the generalization of existing knowledge. Notably, in GITM [129], when encountering a new task, the actions of agents that successfully accomplish the sub-goals are stored in a list. This hierarchical and reflective process of memory utilization enables agents to refine their strategies and improve performance across varying tasks and environments.

3.3.2 Knowledge utilization

Knowledge utilization focuses on integrating external knowledge (excluding memory information) into LLM-based planning. By leveraging up-to-date textual, visual, and audio data, LLMs enhance their ability to perform complex tasks accurately and contextually. Techniques such as retrieval-augmented generation and real-time web scraping allow these models to combine internal capabilities with external information, thereby improving planning and decision-making processes. The overall flowchart illustrating the operational mechanism of memory is presented in Fig. 3.

Knowledge for LLM-based Agents The diverse nature of tasks requires varying forms of knowledge. In this section, we examine how LLM-based agents utilize textual, visual, audio, and other domain-specific knowledge. By understanding these mechanisms, we can appreciate the versatility and effectiveness of LLMs in handling a wide range of tasks.

Textual Knowledge. Textual knowledge is the backbone of LLMs, given their training on extensive text corpora. This knowledge is vital for tasks such as natural language understanding, text generation, translation, and more. The formats of textual knowledge include natural language, embeddings, tokens, and tree structures. Natural language is the primary input and output format, embeddings capture semantic meaning, tokens segment text into processable units, and tree structures enable complex reasoning tasks.

LLMs utilize both internal and external textual knowledge to perform these tasks [17]. Pretrained on vast datasets, LLMs can understand and generate text based on internalized knowledge, including language syntax, semantics, and general world knowledge. This allows LLMs to perform tasks like text generation, summarization, translation, and even planning with minimal additional context. For example, in [151], it demonstrates how LLMs can use embeddings derived from textual data to generate planning actions in PDDL format. The model processes natural language inputs to understand the context and objectives, converting this understanding into actionable plans by leveraging its pretrained knowledge base. Additionally, LLMs often access external data to provide accurate and up-to-date information. Techniques such as few-shot learning enhance their performance, as shown by [152], who illustrate how fine-tuning LLMs with specific examples improves their ability to translate natural language instructions into planning goals.

Visual Knowledge. In LLM agents, visual knowledge is primarily represented through continuous embeddings generated by visual encoders, which are then integrated with textual information to facilitate multi-modal data understanding and reasoning. The representation of visual knowledge typically includes latent vector representations of images (e.g., visual Transformer encodings), object-centric encodings, and other forms, all processed alongside language information through standard self-attention mechanisms. LLM agents leverage these visual embeddings to achieve strong performance across various tasks, such as VQA, image captioning, and embodied reasoning. In practical applications, [153] proposes freezing the parameters of the LLM while optimizing the visual encoder to process visual inputs, converting visual features into embeddings interpretable by the language model, thereby enabling the integration of visual and linguistic information. Building on this foundation, PaLM-E [71] further incorporates continuous inputs like visual data and state estimations into the LLM, enabling embodied reasoning and decision-making through a unified multi-modal processing framework, demonstrating cross-task transfer learning capabilities. Models like LLaVA [154] integrate CLIP visual encoders with language models and apply visual instruction fine-tuning, enabling joint reasoning over visual and textual information in complex tasks.

Audio Knowledge. Audio knowledge encompasses speech and audio events, which can be represented through forms such as speech encoders and spectrogram images. When processing speech, LLM agents can discretize speech input via connection modules and embed it into a vector space shared with text. For instance, in SpeechGPT [155], speech tokens generated by the HuBERT [121] encoder are embedded into the LLaMA [156] vocabulary, enabling the LLM to process speech input. Another approach involves aligning speech encoders with the LLM using connectors such as fully connected layers, multi-head cross-attention [157], or Q-Former [103], which preserves more speech information and achieves efficient compression, thereby supporting the processing of long speech segments [158]. Audio events are typically treated as fixed-size spectrogram images and processed using methods from visual language models. Additionally, end-to-end audio LLMs, such as AudioPaLM [159], can simultaneously handle speech and other audio signals to meet broader auditory requirements. For example, in AudioGPT [123], the LLM is integrated with various foundational audio models to process complex audio information, enabling automatic speech recognition (ASR) and text-to-speech (TTS) conversion. These examples demonstrate the robust capabilities and extensive adaptability of LLMs in processing and generating audio.

Other Knowledge. Beyond text, visual, and audio data, LLMs often need to utilize specialized knowledge from specific domains such as scientific research [160, 161], medical information [162–164], or technical specifications [165–167]. This enhances their ability to handle tasks that require deep domain expertise. Formats of domain-specific knowledge include natural language descriptions, embeddings, tokens, and tree structures, which enable LLMs to process and understand complex information from various fields.

In scientific domains, LLMs can assist in data analysis, hypothesis generation, and literature review. For instance, in [160], it highlights how integrating domain-specific knowledge enhances the performance of LLMs in specialized tasks. While in [161], it enhances the capability of large language models to perform multi-step mathematical reasoning by training verifiers on a diverse dataset of elementary math word problems, which evaluate the correctness of model-generated solutions and select the most accurate answer.

In the medical field, LLMs can support professionals by retrieving and synthesizing medical information from databases like PubMed [168]. This capability is crucial for applications such as clinical decision support, where accurate and up-to-date information is essential. For example, MedPaLM [162], an LLM fine-tuned for medical dialogue, leverages domain-specific knowledge to provide accurate and reliable responses to medical queries. This model integrates medical literature and clinical guidelines into its knowledge base, enhancing its ability to perform tasks such as diagnosis assistance and patient education.

Knowledge Retrieval Knowledge retrieval is a critical aspect of utilizing LLMs effectively, as it involves accessing external information to supplement the model's inherent knowledge base. This ensures that LLMs can provide accurate and contextually relevant responses, enhancing their performance across various domains. Several methods are employed for efficient knowledge retrieval, each with its own approach and applications.

Database and Knowledge Base Queries. Database and knowledge base queries involve accessing structured data from repositories like Google Knowledge Graph, PubMed [168], and other domain-specific databases. These sources offer reliable and organized information that can be integrated with LLM outputs to enhance the accuracy and relevance of generated responses. A notable example of integrating external databases is the ChatDB [143] system, which uses SQL queries to fetch relevant data logically, making it easier for agents to operate. Similarly, SQL-PALM [169] employs a Text-to-SQL model based on LLMs, significantly enhancing query accuracy and database interactions. Another example, KnowledGPT [170], enables LLMs to access and retrieve knowledge from external knowledge bases through "Program of Thoughts" prompting, thereby enhancing their ability to answer questions.

Web Scraping and API Calls. Web scraping and API calls allow LLM-based agents to collect real-time information from the internet. This method is particularly useful for tasks requiring up-to-date data, such as news summarization or market analysis. Web scraping involves using automated tools to extract data from web pages, providing large amounts of data from diverse sources. API calls, on the other hand, involve querying APIs to fetch specific information, such as news articles, weather updates, or financial data. Several studies have integrated LLMs with specific tools like web search [171], compiler [172], and calculator [161]. Talm [173] created a dataset for instruction API and fine-tuned LLMs to help them use tools and retrievers effectively. Gorilla [174] is a fine-tuning LLM that surpasses the performance of GPT-4 [64] in writing API calls, aiming to generate precise input parameters for API calls and alleviate hallucinations during external API calls.

Retrieval-Augmented Generation (RAG). RAG models combine retrieval mechanisms with generative models to produce context-rich responses [175]. This approach is effective for open-domain question answering and conversational agents. In the retrieval stage, the system extracts document fragments relevant to the query from external knowledge sources. The primary retrieval source is textual data, but it can be extended to semi-structured data (e.g., PDFs) [176, 177], structured data (e.g., knowledge graphs) [178], and content generated by LLMs themselves [179, 180]. Beyond the commonly used single-step retrieval, RAG incorporates three types of retrieval enhancement processes: iterative retrieval [181], recursive retrieval [182], and adaptive retrieval [183, 184], which are designed to improve efficiency and accuracy in solving complex queries [185]. In the generation stage, the model improves the quality of responses from LLMs by re-ranking document segments to highlight the most relevant results [186] or by selecting or compressing contexts to reduce redundant information and manage overly long inputs [187, 188]. Additionally, LLMs can be fine-tuned for specific scenarios and data characteristics, enhancing the relevance and accuracy of the generated responses [189, 190].

Extraction Issues In the development and application of LLMs, a range of extraction issues are encountered, directly impacting the accuracy, applicability, and bias of the models. These issues encompass challenges related to knowledge update, hallucination, and bias. Addressing these challenges necessitates a comprehensive approach integrating strategies such as leveraging external knowledge sources, enhancing transparency, and employing debiasing techniques.

Edit Wrong and Outdated Knowledge. One of the primary challenges for LLM agents in knowledge extraction is ensuring the timeliness and accuracy of information. Since LLMs are typically trained on historical data, this can lead to a lag in processing the latest information. When tasks require knowledge that is more recent than the training data, LLMs often struggle to cope. A direct approach is to regularly update LLMs with new data, but fine-tuning LLMs incurs high costs, and incremental training may result in catastrophic forgetting [191], where the model loses the broad knowledge it acquired during pretraining. Therefore, developing efficient methods to incorporate new knowledge into existing LLMs to keep them up-to-date becomes paramount.

Current approaches include leveraging external knowledge sources to supplement the knowledge base of LLMs [170, 192]. By integrating retrieved relevant information into the context, LLMs can acquire new factual knowledge and perform better on relevant tasks. However, these methods still fall short when dealing with more profound knowledge updates. Model editing techniques [193–195] are also employed to alter model behavior, either by modifying model parameters or using external post-editing mechanisms to achieve knowledge updates, but they still face limitations in real-world applications due to their low specificity.

Hallucination. Hallucination refers to the phenomenon where LLM agents generate text that deviates from reality [196–198]. Hallucinations can occur due to overgeneralization of the model from training data or erroneous interpretations of incomplete or misleading information. Hallucinations generated by LLMs can be categorized into two types: intrinsic hallucinations and extrinsic hallucinations [199]. Intrinsic hallucinations involve text generation that contradicts input logic, while extrinsic hallucinations involve text generation containing information that cannot be verified with existing information.

To address the hallucination problem, researchers have proposed various methods. One approach is to integrate external knowledge bases and fact-checking systems to verify the accuracy of generated content [143, 200, 201]. Another approach is to enhance the transparency and interpretability of the model to improve the credibility of outputs [86, 202, 203]. These methods include fine-tuning with high-quality data or fine-tuning based on human feedback [204–206]. For example, the TruthfulQA [207] task aims to detect whether the model mimics human false statements. Additionally, some techniques such as retrieval-augmented generation and decoding strategies are being explored to reduce hallucinations. Retrieval-augmented generation methods [175, 183, 208] enhances the accuracy of language generation by introducing additional source material and providing mechanisms to check for inconsistencies between the generated response and the source material. Decoding strategies [202, 209, 210] optimize the way language models select output tokens during text generation, balancing diversity and factual accuracy, thereby mitigating the occurrence of hallucinations.

Spurious Bias. In the realm of artificial intelligence, the fairness and accuracy of models are frequently compromised by serendipitous biases and class imbalances present within the training data, collectively referred to as spurious bias. One major concern is shortcut learning, where models rely on spurious, non-generalizable cues in the training data rather than learning robust features. For instance, a language model might incorrectly predict due to the frequent co-occurrence of function words with specific labels in the training set [211]. Additionally, models might develop preferences based on the order of training samples, such as the position of answers in question-answering tasks potentially influencing model judgments [212]. Shortcut learning can be mitigated through methods such as data debiasing, adversarial training, interpretive regularization, and confidence regularization [213].

Label bias represents another form of pseudo-bias, often stemming from class imbalance within the training dataset. This imbalance can cause models to be overly sensitive to the majority class while being insufficiently sensitive to minority classes. For example, in sentiment analysis tasks, a model might skew towards predicting positive sentiment due to an overabundance of positive samples, even when the text actually conveys negative sentiment [214]. To reduce such bias, researchers propose rebalancing datasets, employing advanced sampling techniques, and developing new evaluation metrics to enhance model fairness and robustness [215].

3.3.3 Agent's ability utilization

The capabilities of LLM-based agents are a manifestation of their cognitive intelligence. Leveraging these abilities allows agents to analyze, synthesize perceived information, and engage in creative thinking. Given the exceptional proficiency of LLMs in handling long contextual information, we categorize agents' abilities into three primary types: reasoning, planning, and generalization. Reasoning involves logical inference based on historical experiences and current knowledge, extracting universal paradigms. Planning entails the application of high-level general rules to new scenarios, resulting in concrete, actionable plans. Generalization seeks to apply existing experiences to tackle novel situations and problems. In the following sections, we will provide an in-depth exploration of the utilization of these three capabilities.

Reasoning and Planning Reasoning and planning involve the systematic process of leveraging historical experience, common knowledge, and current state information to perform logical analysis, thereby deriving high-level, more profound insights. Subsequently, these insights are applied to the present situation to generate updated inferential outcomes. Furthermore, as tasks advance, agents can utilize introspection to adjust their plans, ensuring a closer alignment with real-world conditions, ultimately facilitating adaptive and successful task execution. These abilities are fundamental human capabilities that underpin problem-solving, decision-making, and critical analysis, forming the bedrock of human perception and social interaction [216–218]. These cognitive processes encompass three key components: deductive, inductive, and abductive reasoning [219]. Leveraging the robust text-based reasoning and planning capacity inherent in large language models (LLMs) [65, 66], it becomes essential for LLM-based agents to effectively perceive complex environments, execute intricate tasks, and engage in human-like interactions. This sophisticated reasoning and planning framework enables agents to navigate and adapt to dynamic scenarios, thereby enhancing

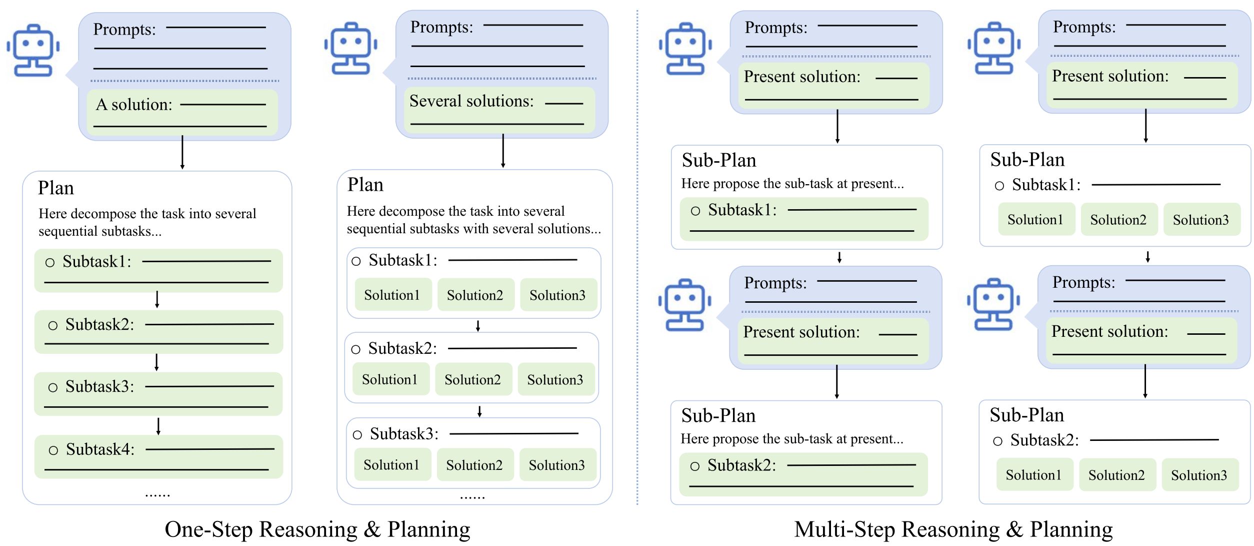

Fig. 4 Approaches for reasoning and planning with large language models

their ability to perform and interact in a manner akin to human cognitive processes. In accordance with the steps and decision outcomes associated with agents' reasoning and planning, we will delineate the relevant methodologies and inferential processes from two perspectives: (1) One-step Reasoning and (2) Multi-step Reasoning. These approaches provide a structured framework for understanding how agents utilize reasoning and planning capabilities to derive decisions, adapt to new information, and effectively plan their actions within varying contexts. The visualization of these approaches is presented in Fig. 4.

Plan Structure. During the process of plan formulation, agents generally decompose an overarching task into numerous sub-tasks, and various approaches have been proposed in this phase. Notably, some works advocate for LLM-based agents to decompose problems comprehensively in one go, formulating a complete plan at once and then executing it sequentially [220–223]. In contrast, other studies like the CoT-series employ an adaptive strategy, where they plan and address sub-tasks one at a time, allowing for more fluidity in handling intricate tasks in their entirety [86, 224, 225]. Additionally, some methods emphasize hierarchical planning, while others underscore a strategy in which final plans are derived from reasoning steps structured in a tree-like format [226, 227]. The latter approach argues that agents should assess all possible paths before finalizing a plan. While LLM-based agents demonstrate a broad scope of general knowledge, they can occasionally face challenges when tasked with situations that require expertise knowledge. Enhancing these agents by integrating them with planners of specific domains has been shown to yield better performance.

One-Step Method. In this strategy, agents decompose a complex task into several sub-tasks through a single reasoning & planning process based on the current task directives. These sub-tasks are sequentially ordered, with each sub-task logically following the preceding one. LLM-based agents adhere to these steps to achieve the final objective. Typically, agents perform the reasoning process through prompt-based elicitation, where the context includes historical records from memory, the state of the surrounding environment, and the agents' current status as auxiliary decision-making information. Based on the current task directives, agents integrate their inherent intelligence with external knowledge to deduce a series of rational and feasible steps for solving complex tasks.

In specific, in-context learning introduces a methodology where LLMs are provided with a few reasoning and planning examples, enabling them to infer solutions for new situations through analogous reasoning and planning. For instance, the Chain of Thought (CoT) [86] technique prompts LLMs to think through problems step-by-step, systematically deconstructing intricate tasks into manageable components, thereby facilitating long-term planning and deliberation. The Zero-shot-CoT [224] approach empowers LLMs to autonomously generate reasoning processes for tasks by prompting them with trigger sentences such as "think step by step". Moreover, Auto-CoT [228] methods further refine this process, enhancing the agents' ability to tackle complex tasks efficiently by leveraging structured and context-aware reasoning paradigms.

To enhance the decision-making rationality and accuracy of LLMs, mitigating the hallucination problem that can occur during single-step reasoning, several approaches employ multi-path reasoning to select the optimal outcome. Each intermediate step may lead to multiple subsequent steps. Specifically, Self-consistency CoT [229] employs CoT to generate multiple reasoning paths, seeking diverse answers and filtering out the answer with the highest frequency as the final result. The Tree of Thought (ToT) [230] approach decomposes problems into a tree structure, creating multiple solution paths with each node representing a different "thinking" stage. Algorithm of Thought [231] introduces a novel method to enhance LLM reasoning by incorporating algorithmic examples into the prompts, remarkably requiring only one or a few queries to the LLM. In RecMind [232], a self-inspiring mechanism is designed where discarded historical information in the planning process is leveraged to derive new reasoning steps. The Graph of Thought [233] expands the tree-like reasoning structure in ToT to graph structures, resulting in more robust prompting strategies. Furthermore, in [234], LLMs are utilized as zero-shot planners. At each planning step, they generate multiple potential next steps and determine the final one based on their proximity to admissible actions. The RAP [235] constructs a world model to simulate the potential benefits of various plans, ultimately generating the final plan by aggregating multiple iterations. These methods collectively contribute to a more robust and reliable decision-making framework for LLM-based agents.

While some work focuses on employing feedback mechanisms to correct errors in the reasoning and planning processes of agents, guiding them to execute accurate reasoning chains, previous work can be categorized into three primary sources of feedback: (1) LLM's internal reflection based on memory; (2) human feedback; (3) environmental feedback. Regarding the first category, LLM-based agents derive insights from historical experiences to update or optimize strategies and planning methods. For instance, the Re-Prompting [223] approach involves verifying if each step fulfills the necessary prerequisites before progressing with the plan. If a step fails to meet these prerequisites, a prerequisite error message is generated, prompting the LLM to revise the plan accordingly. Similarly, ReWOO [222] introduces a paradigm where plans and external observations are generated independently by the agents. These independently derived plans and observations are then integrated to produce the final outcomes. These methodologies collectively enhance the decision-making capabilities of LLM-based agents by leveraging structured, multi-path, and context-aware reasoning paradigms, thereby enabling them to tackle complex tasks more effectively. The integration of feedback mechanisms ensures a dynamic and iterative refinement process, crucial for achieving accurate and reliable autonomous reasoning in LLM-based systems.

Multi-Step Method. Unlike one-step reasoning, multi-step reasoning requires iterative invocation of LLMs for multiple reasoning cycles, where each cycle generates one or several incremental steps based on the current context while maintaining consistency with the overall objective. Multi-step reasoning aims to enhance the LLM's capability to solve complex problems and understand long-term tasks through structured reasoning processes. This approach ensures that the reasoning and planning remain adaptive and responsive to evolving task requirements and environmental dynamics, thereby facilitating robust decision-making and problem-solving capabilities in LLM-based systems.

Multi-stage methods dissect the planning process into distinct stages, aiming to improve LLM's performance in complex reasoning and problem-solving tasks. SwiftSage [227] is a framework inspired by the dual-process theory that combines the advantages of behavior cloning and guided LLMs to enhance task completion performance and efficiency. It consists of two primary modules: the SWIFT module, responsible for rapid, intuitive thinking, and the SAGE module, handling deliberative thinking. The exploration process of DECKARD [236] is divided into the Dreaming and Awake stages. During the Dreaming stage, the agent utilizes an LLM to decompose the task into sub-goals. In the Awake stage, the agent learns a modular strategy for each sub-goal, verifying or rectifying assumptions based on the agent's experience.

External Reasoner and Planner. While LLMs exhibit powerful reasoning and planning capabilities across diverse applications, generating precise and efficient plans for domain-specific problems poses significant challenges. Consequently, several research studies have integrated LLMs with external tools to collaboratively address specialized challenges. These external tools encompass domain-specific skills such as APIs, expert models, and techniques involving external databases [174, 237, 238], renowned for their proficiency and high accuracy in specific domains. Leveraging these specialized capabilities, LLM-based agents equipped with external planners can generate more efficient, and in some cases optimal plans. Specifically, CO-LLM [35] utilizes LLMs to generate high-level plans for current tasks, complemented by an external model that refines these plans into finer-grained strategies. On the other hand, LLM+P [239] transforms prompt contexts containing the agent's current state, environmental observations, and historical experiences into formal Planning Domain Definition Languages (PDDL). Subsequently, this textual information is fed to an external reasoner for inference and the generation of detailed planning arrangements. This integrated approach enhances the planning capabilities of LLMs by leveraging both their text-based reasoning prowess and the precision of external reasoning models tailored to specific domains.

These methods significantly enhance the adaptive and perceptual capabilities of LLM-based agents in navigating complex environments, thereby improving their ability to plan for and engage in sophisticated problem-solving and collaborative interactions. By employing these methodologies, agents can be guided toward more efficient, rational, and effective processes of reasoning, planning, and execution.

Generalization The generalization capabilities of LLM agents are critical for their effectiveness across a wide array of dynamic and unpredictable environments. Generalization specifically manifests in the form of transferability, allowing agents to apply knowledge learned in one domain to another, and robustness, supporting adaptation to diverse input variations. This generalization capability ensures that LLM agents can maintain high performance across different contexts without extensive retraining or human intervention. The utilization of LLM agents' generalization abilities is prominently reflected in areas such as zero-shot learning, few-shot learning, and many-shot learning.

Unseen tasks refer to those that the agent did not encounter during the training phase. The dynamic nature of most application environments necessitates that models possess the capability to effectively respond to unforeseen situations. LLM agents can leverage their large-scale training on diverse datasets to infer and apply relevant knowledge, enabling them to adapt to new tasks more quickly and robustly than traditional models. The generalization approaches for LLM agents to unseen tasks can broadly be categorized based on whether the model has undergone fine-tuning.

In-Context Learning (ICL). In-context learning involves providing examples of the current task within the input prompt, allowing the model to use these examples to infer the task requirements and generate appropriate responses [85]. This method was highlighted in the work of [240], demonstrating GPT-3's ability to learn to perform complex tasks through examples in the context. The advantage of ICL is that it does not require parameter updates, making it computationally efficient and easy to implement. Nonetheless, the model's performance is sensitive to specific settings, including the selection of prompt templates, the choice of contextual exemplars, and the sequence of examples, and it exhibits a propensity to predict answers that frequently occur at the conclusion of prompts or are prevalent in the pre-training dataset [241].

Zero-Shot Learning. Zero-shot learning requires the model to perform new tasks without any specific task examples or fine-tuning, relying entirely on its pre-trained knowledge. In [242], it demonstrated zero-shot learning with GPT-2, where the model showed the ability to handle various tasks without prior specific task training. This method highlights the model's inherent generalization capabilities and does not require additional data or training. However, the performance of zero-shot learning may be limited for highly specialized or complex tasks, as the model may lack the specific knowledge required to execute them effectively.

3.3.4 Action

Actions represent the tangible behavioral outcomes of agents within an interactive environment, thereby effectuating real changes in the environment and significantly impacting the interactions among agents. These actions are typically determined by a combination of profiles, memory, and the interactive context (including agent-to-agent, agent-to-environment and agent-to-human interactions). Situated at the most downstream position, actions vary widely depending on the application scenario. The action mechanism can be elucidated from two perspectives: the process of action creation and the application of actions: (1)Action Creation: This involves the processes and steps through which actions are generated. It encompasses the decision-making frameworks, algorithms, and procedures that lead to the formulation of specific actions based on the agent's internal state and external stimuli. (2)Action Application: This refers to the contexts in which actions are applied and the subsequent effects of these actions on the application scenarios.

Action Creation Action creation represents the final stage where agents manifest their intelligence within multi-agent systems' interactive environments. As the environment dynamics fluctuate and task directives vary, agents employ diverse strategies and information sources to enact actions aligned with the system's overarching objectives. Based on the temporal nexus between decision-making and action-taking in interactive environments, we will delineate three prevalent strategies for action creation.

One-Step Decision. Firstly, instant decision-making involves agents extracting recent, pertinent, and significant information from their memory banks. When necessary, agents supplement this information by accessing external knowledge bases. Guided by prompts derived from the amalgamation of current task requirements, memory recollections, and external knowledge, agents promptly formulate plans and execute corresponding actions. For instance, Generative agents [28] maintain a continuous memory stream, using recent and relevant information to guide their actions. Similarly, in GITM [129], agents query their memory to identify successful experiences relevant to achieving low-level sub-goals, replicating effective actions from previous tasks. Collaborative agents like ChatDev [30] and MetaGPT [31] engage in dialogue interactions where conversational histories stored in memory influence each agent's utterances. These strategies underscore the adaptive capacity of agents to dynamically integrate internal and external information, facilitating effective decision-making and responsive action execution in complex interactive environments.

Pre-defined Planning. In this strategy, each action undertaken by LLM-based agents strictly adheres to predefined planning, which can either be autonomously generated by the agent or predefined by users. For example, in DEPS [243], agents initiate action planning for a specific task and proceed with execution unless indications of plan failure emerge during the process. This method ensures agents maintain consistency and adherence to planned courses of action throughout their operational sequences.

Dynamic Creation. This strategy represents a synthesis of the preceding two approaches, effectively balancing the pre-defined nature of task planning with adaptability to dynamic environments. Initially configured with a comprehensive goal plan, agents generate an overarching objective plan. Subsequently, during interactions, agents adhere to these overarching goals while retaining the flexibility to make instant decisions based on the interactive environment. In GITM [129], for instance, agents formulate high-level plans by decomposing tasks into multiple sub-goals. These plans guide the sequential execution of actions aimed at addressing each sub-goal, ultimately achieving the completion of the overall task.

Action Application The context of actions typically undergoes dynamic changes based on specific application scenarios. Action application refers to the direct interaction and influence between agents and their environment, where the outcomes of their behaviors directly impact the realization of current tasks and the overall progression of multi-agent systems. Depending on the diverse interaction scenarios encountered by agents, we will delineate these aspects across three dimensions:

Task-Driven. In this scenario, the actions of LLM-based agents are aimed at accomplishing specific subtasks, which collectively contribute to the completion of larger overarching tasks through collaborative division of labor among agents. Leveraging the planning capabilities inherent in LLMs, DEPS [243] has developed a Minecraft agent capable of solving complex tasks by breaking them down into manageable sub-goals. Similar systems such as GITM [129] and Voyager [244] also rely extensively on LLMs' planning abilities to successfully navigate and accomplish diverse tasks. TaskMatrix.AI [245] integrates LLMs with millions of APIs to facilitate task execution. At its core is a multi-modal conversational foundational model that engages with users, comprehends their objectives and context, and subsequently generates executable code tailored to specific tasks.

Communication Interaction. The primary task of agent interaction revolves around engaging in discussions on a specific topic to exchange ideas or foster innovation. For instance, agents in ChatDev [30] collaborate through communication to collectively accomplish software development tasks. Similarly, in Inner Monologue [9], the agent actively engages in dialogue with humans and dynamically adjusts its action strategies based on the feedback received from these interactions.

Environment Exploration. Environment exploration primarily entails agents collaborating to explore and adapt to dynamically changing environments, thereby expanding their perceptual capabilities and skillsets. For example, the agent in Voyager [244] engages in the exploration of unknown skills during task completion, continually refining the execution of these skills based on environmental feedback through iterative trial and error.

Upon the agents' execution of actions, some studies consider the direct impact on the interactive environment and attempt to seamlessly integrate downstream applications with the agents' actions. This integration primarily involves incorporating LLMs with the utilization of external tools or knowledge. Specifically, it includes APIs, calculators, code interpreters, expert-designed models, and external knowledge bases [238, 246, 247]. By leveraging these external resources, agents can enhance their decision-making processes and improve the efficiency and accuracy of task execution.

This approach further expands the application scope and capabilities of the action module, facilitating more direct strategic planning and tool utilization in downstream applications. It enhances the agent system's ability to adapt to new situations and leverage novel tools, thereby broadening the potential for effective and efficient task execution.

3.4 Mutual-interaction

Mutual interaction encompasses the exchange of information and coordination of actions among agents, which is crucial for enhancing the collective intelligence within a multi-agent system. This interaction can be decomposed into three fundamental components: (1)**Message Delivery**: This pertains to the content and transmission methods of communication between agents, focusing on the specifics of the information exchanged. (2)**Interaction Structure**: This involves the organization and architecture of communication networks within the multi-agent system, detailing the modes and structures of interaction among agents. (3)**Interaction Scene**: This relates to the modes of collaboration among agents and the surrounding environment in which this cooperation takes place.

3.4.1 Message delivery

Message delivery, an essential component for enabling communication and collaboration among agents, involves the exchange of information between agents. Messages are typically recorded and transmitted in textual form, though some work also incorporates multi-modal information such as visual and audio data. The content of messages dynamically varies based on task assignments and interaction communication scene, generally encompassing historical and current state information as well as communication messages from other agents.

In general, message delivery is triggered by task assignments, interaction with other agents, or external control signals. Depending on how agents access messages, delivery methods can be direct, such as broadcasting and point-to-point communication [248–250], or indirect, where agents first store messages in a shared memory pool that other agents can access to retrieve information [31]. Additionally, message delivery must account for supplementary overhead, including transmission efficiency, bandwidth, and the timeliness of message delivery [136, 251]. These considerations are crucial to ensure that communication and collaboration among agents remain advanced and synchronized, facilitating effective coordination and operational coherence within the multi-agent system.

3.4.2 Interaction structure

The interaction structure delineates the communication framework within a multi-agent system, typically organized and arranged based on the content of messages, thereby assigning different roles and responsibilities to the agents. This structure inherently reflects the relationships among agents and the potential methods and pathways for message delivery. Based on the modes of message delivery and the relationship of inter-agent communication, interaction structures can be categorized into four types: hierarchical, decentralized, centralized, and shared memory. Each type of structure defines specific dynamics and protocols for information exchange, influencing the overall efficiency and coherence of the multi-agent system.

Hierarchy. In hierarchical interaction structures, agents at different levels assume distinct roles, with a clear distinction between higher-level and lower-level agents. Higher-level agents typically perform supervisory roles, making critical decisions and issuing directives to subordinate agents. This interaction model mimics traditional organizational structures and enhances efficiency by clearly delineating authority and responsibility boundaries. For instance, DyLAN [252] constructs a dynamic hierarchical agent architecture, enabling LLM agents to engage in multi-turn dynamic interactions for complex tasks. DyLAN leverages mechanisms such as agent selection during inference and early termination to enhance inter-agent collaboration efficiency and performance. Furthermore, it employs an unsupervised agent importance scoring algorithm to automatically optimize the agent team, thereby improving task execution accuracy.

Decentralized. Decentralized communication operates within peer-to-peer networks where agents communicate directly with each other without relying on central authority. This structure not only promotes equality among agents, allowing for more flexible and dynamic interactions, but also reduces the computational burden on individual LLMs, enhancing system robustness. However, when applied to large-scale systems, coordination and communication overhead can become significant, potentially impacting overall performance. For example, in the decentralized multi-agent communication framework (DMAS) [253], each robot's LLM agent engages in task planning through turn-taking dialogues. This decentralized strategy enables each agent to independently express its opinions and consider feedback from other agents during the conversation, collectively advancing task completion.

Centralized. In centralized interaction structures, a central agent or a group of central agents coordinate the system, managing and orchestrating interactions among all agents. This structure centralizes control and coordination, simplifying the decision-making process, avoiding potential conflicts, and improving overall system efficiency. However, due to the system's reliance on the central agent, it is susceptible to single points of failure and communication delays, making it challenging to respond

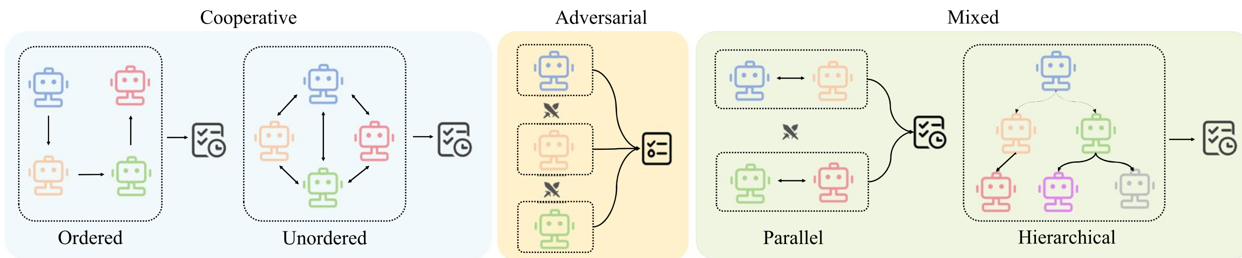

Fig. 5 The agent interaction scene

swiftly to environmental changes. For instance, ACORM [254] introduces a centralized architecture by using a single LLM as the central planner, which generates actions for each agent based on global state information, thereby achieving centralized training and decentralized execution. The introduction of a centralized architecture simplifies the MARL planning process, reduces the need for extensive context, and enhances the scalability and inference efficiency of large language models in multi-agent tasks.

Shared Message Pool. The Shared Message Pool [31] is a mechanism for information exchange among LLM agents, where agents publish and subscribe to information via a shared message pool. This structure allows agents to subscribe to relevant messages based on their needs and profiles without requiring direct point-to-point communication, thereby improving communication efficiency. Advantages include simplified communication processes, reduced complexity of information transmission, and a unified message management approach. However, simultaneous access to the shared message pool by multiple agents may lead to contention and synchronization issues.

Shared messages can be divided into central knowledge repositories and shared parameters [19]. A typical example of the former is MetaGPT [31], which maintains a shared message pool, allowing each agent to dynamically observe and extract the necessary information, thereby optimizing collaboration and communication efficiency among agents. Shared parameters refer to the partial or complete sharing of model parameters among agents, allowing an agent to update its weights based on new knowledge and synchronize these parameters with other agents.

3.4.3 Interaction scene

In multi-agent systems, the interaction scenarios among agents are crucial as they not only determine the behavior patterns of the agents but also affect the overall efficiency and effectiveness of the system. Interaction scenarios in MAS based on LLMs can be classified into three major categories: communication, task execution, and environment exploration.

The communication scenario is one of the most fundamental forms of interaction in MAS. Agents coordinate and make decisions by exchanging information. This information exchange can take a direct form, such as transmitting each agent's status, plans, and suggestions through specific communication channels [31, 255], or an indirect form, such as sharing knowledge about the environment, tasks, or other agents. The task execution scenario focuses on how agents execute specific actions based on predefined task allocations, which may include role-playing games [12, 256, 257], distributed task assignments [258–260], and more. The environment exploration scenario requires agents to utilize perception and learning mechanisms to continuously adapt and optimize their behavior in unknown environments, which can include both simulated [2, 5, 261] and real physical [8–10] environments.

Analyzing the interrelationships among agents in these interaction scenarios is particularly critical as these relationships dictate how agents interact and collaborate. Currently, the interaction scenarios in LLM-based MAS can be summarized into three basic types: cooperative, adversarial, and mixed. These types provide MAS with a rich array of interaction patterns, enabling the system to adapt to diverse application scenarios and challenges, which are visualized in Fig. 5.

Cooperative. In cooperative interaction scenarios, agents work together to achieve a common goal. The basic process of cooperative MAS includes goal setting, task decomposition, information sharing, collaborative decision-making, and execution feedback. Agents first set common goals based on task requirements, then decompose complex tasks into multiple subtasks assigned to different agents. The agents share information and jointly make decisions through communication and negotiation to reach a consensus. During task execution, agents perform tasks based on their respective roles and provide feedback to adjust strategies and optimize the execution process.

Existing multi-agent cooperation models are mainly divided into unordered cooperation and ordered cooperation [20, 262]. A typical example of unordered cooperation is ChatLLM [258], which facilitates natural collaboration among agents by constructing a network that allows multiple ChatGPT instances to communicate, provide feedback, and think collectively without fixed role assignments. In contrast, METAGPT [31] achieves ordered cooperation by encoding standardized operating procedures (SOPs) into prompt sequences, enabling agents to perform specific tasks based on assigned roles and expertise. SPP [263] transforms an LLM into a cognitive collaborative entity capable of solving complex tasks by simulating multi-role self-cooperation within a single LLM, effectively enhancing the LLM's knowledge acquisition, hallucination reduction, and strong reasoning capabilities.

Adversarial. In adversarial interaction scenarios, agents are in a competitive relationship, each pursuing the maximization of their own interests. The basic process includes goal setting, strategy formulation, interaction games, and result evaluation. Agents first set goals to maximize their own interests and then formulate competitive strategies based on the behavior of their opponents. In the interaction game stage, agents implement strategies through interactions to strive for maximum benefits. Finally, agents evaluate the game results and adjust strategies to cope with future competition. For instance, ChatEval [38] simulates the collective wisdom and cognitive collaboration of human evaluators by constructing a multi-agent debate system, utilizing LLMs with different roles and communication strategies to improve the accuracy and consistency of text evaluation with human judgments. MAD [264] addresses the "thought decay" problem in LLMs' self-reflection by promoting divergent thinking through multi-agent debates, significantly enhancing performance in complex reasoning tasks.

Mixed. Mixed interaction scenarios combine features of both cooperative and adversarial interactions, requiring agents to find a balance between cooperation and competition. This type of interaction can be further subdivided into parallel and hierarchical forms. (1)**Parallel**: In parallel interactions, agents collaborate independently on separate tasks, sharing some information without interfering with each other. Agents set independent goals, execute their tasks in parallel, share some information to improve overall efficiency, and finally evaluate task completion and adjust information-sharing strategies. In the workflow of SoT [259], the model first creates an answer outline and then expands each outline point in parallel. This parallel processing strategy allows multiple LLM agents to collaborate, with each agent responsible for generating an independent part of the answer, ultimately aggregating into a complete response, thereby achieving rapid and efficient response generation. (2)**Hierarchical**: In hierarchical interactions, the relationships among agents typically manifest as a tree structure. The parent node agents set global goals, decompose tasks, and assign them to child node agents. The child node agents execute specific tasks and provide feedback on the execution. The parent node agents adjust the global strategy based on the feedback to optimize the overall task execution. AutoGen [260] is a multi-agent dialogue framework that constructs a hierarchical structure dominated by managing agents, enabling parent agents to decompose complex tasks and dynamically assign them to child agents, achieving hierarchical interaction. AgentLite [265] builds on this by providing a lightweight platform that allows developers to easily implement and extend complex interactions and collaborative tasks among LLM agents based on this hierarchical concept.

Overall, different interaction scenarios adopt different agent interaction strategies, greatly expanding the capabilities of agents. In cooperative scenarios, agents achieve efficient task completion through coordination and information sharing; in adversarial scenarios, agents optimize their strategies through games and competition; in mixed scenarios, agents balance cooperation and competition to achieve optimal solutions for complex systems. These models strive to enhance the realism, fidelity, and reliability of the reasoning process, driving the development and application of LLM-based multi-agent systems.

3.5 Evolution

Similar to how humans continuously refine their cognitive abilities and acquire knowledge through interactions with their environment and others, evolution in agents involves the ongoing reflection on their decisions and actions to dynamically update their knowledge and experiences, based on existing experiences and the feedback received during interactions, which is visualized in Fig. 6. By adopting evolution mechanisms, agents can continuously refine or revise their current understanding, thereby deepening their proficiency in known tasks and expanding their successful exploration of unknown tasks. Considering the sources of external feedback obtained during interactions, existing work can be categorized into three main types: information perceived from the surrounding environment, exchanged with other agents, or conveyed by humans. To equip agents with these diverse sources of information, various methods have been employed to enhance their evolution capabilities. In the following sections, we provide a detailed introduction

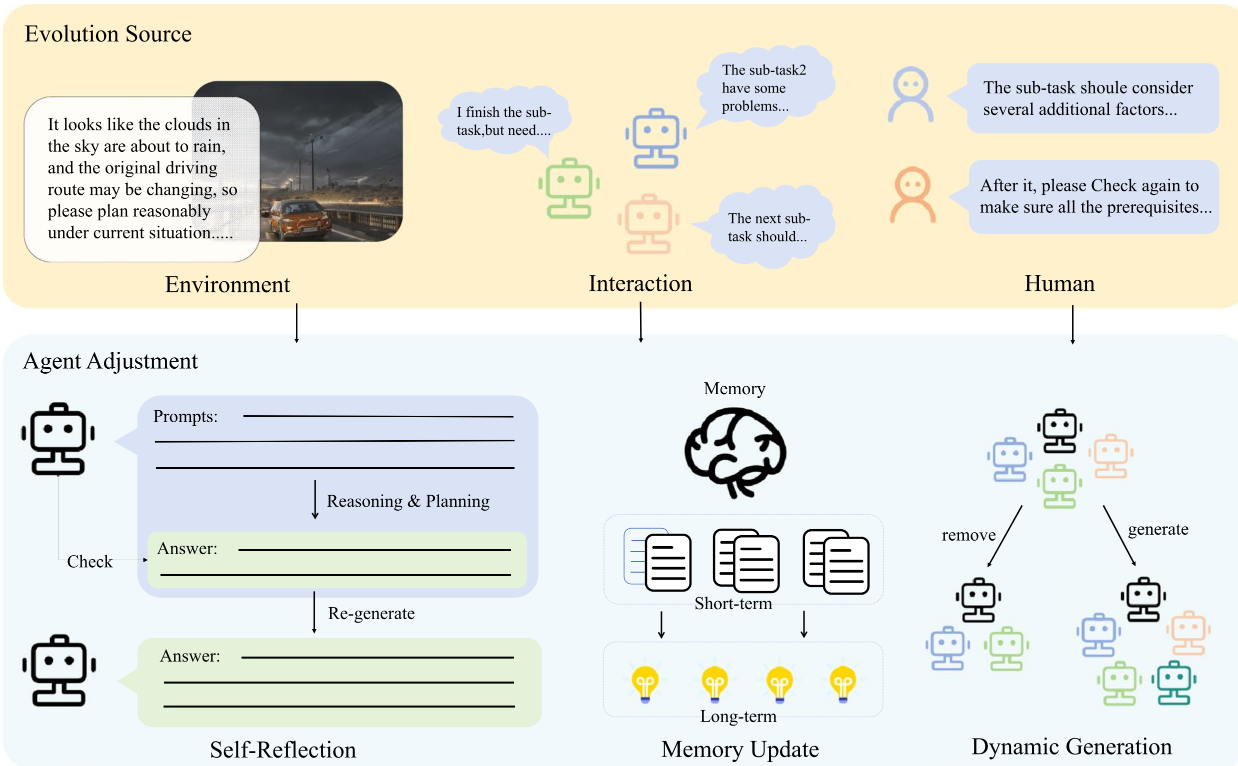

Fig. 6 The overall workflow of the evolution module

to each of these approaches, elucidating the techniques used to bolster the evolution process in agents.

3.5.1 Evolution source

Feedback received during interactions serves as indispensable reference information for agents to achieve evolution. This feedback encompasses the outcomes and impacts of the agents' decisions and actions, guiding them to introspect and thereby dynamically improve their adaptation to complex environments or tasks. Previous work has predominantly captured and conveyed feedback in textual form [243, 266, 267]. Based on the sources from which agents receive this feedback, it can be categorized into three distinct types. Each source provides unique insights that contribute to the agents' self-reflection and continuous improvement processes.

Environment Feedback. Environment feedback refers to the information perceived by agents within either real-world or virtual environments. This type of feedback generally pertains to the changing information in the environment resulting from the agents' decisions and actions during their interactions with the external surroundings. Such feedback acts as a reward signal, informing agents about the consequences of their actions. This mechanism is vividly demonstrated in complex task planning and robotic simulations within dynamic environmental scenarios [17, 132, 268]. By incorporating these environmental changes as feedback, agents can refine their strategies and actions, thereby improving their adaptability and performance in real-time and simulated environments.

Agents Interaction. In multi-agent systems, agent interaction information involves the exchange of collaborative information between agents. This information typically includes evaluations or status updates from other agents regarding a particular agent's decisions or actions, as well as contextual communication between agents. Serving as internal signals, this interaction information facilitates coherence and integration among agents, thereby continuously enhancing and expanding the collaborative capabilities of the multi-agent system. This is particularly evident in the hierarchical execution of tasks and agent communication within world simulations [49, 269, 270]. Through such exchanges, agents can refine their coordination and improve overall system performance in complex, dynamic scenarios.

Human Feedback. Apart from the aforementioned environmental and agent interaction feedback, human feedback constitutes a guiding signal provided by humans to direct agents toward making better decisions and actions, thereby enhancing their cognitive capabilities. As a subjective signal, human feedback effectively aligns agents with human values and preferences and helps mitigate issues such as hallucination. This type of feedback is extensively utilized in systems where agents collaborate and communicate with humans [271, 272], ensuring that the agents' actions and decisions are in harmony with human expectations and standards.

3.5.2 Evolution methods

Evolution methods encompass a variety of techniques designed to enhance the capabilities and adaptability of agents through self-improvement and learning from interactions with their environment. These methods are crucial for developing intelligent systems that can autonomously refine their strategies and behaviors to achieve better performance across diverse tasks and scenarios. The section below delves into several key approaches, including feedback learning, supervised fine-tuning, prompt engineering, and reinforcement learning, each contributing distinctively to the evolutionary trajectory of intelligent agents.

Fine-tuning. Fine-tuning involves updating the parameters of a pre-trained model to adapt it to new tasks or domains. This method ensures that the model is specifically tailored for new challenges. There are three main categories of fine-tuning methods: full model fine-tuning, partial pre-trained parameter fine-tuning, and additional parameter fine-tuning:

(1) **Full Fine-tuning**: Full fine-tuning involves updating all parameters of the pre-trained model to adapt it to specific new tasks. As noted in FireAct [273], full model fine-tuning can be more optimal, particularly when deep learning of the model for specific tasks is required, provided resources allow. However, it is computationally expensive and time-consuming, and when new task data is limited, there is a risk of overfitting.

(2) **Repurposing**: Repurposing typically focuses on fine-tuning specific layers of a pre-trained model, usually the higher layers, while keeping the lower layers unchanged [274–276]. Additionally, BitFit [277] demonstrates that by adjusting only the bias terms of the model or a subset thereof, performance comparable to or even better than full-model fine-tuning can be achieved on small to medium-sized training datasets. Similarly, SIFT [278] proposes leveraging the gradient sparsity of the model in downstream tasks by updating only the key parameters that contribute most significantly to the gradient norm. Although repurposing enhances efficiency, it may not match the performance of full-parameter fine-tuning when delving deeply into specific tasks [277]. Furthermore, the selection of parameters or layers to update is often based on heuristic rules, which may require further research to optimize the selection process.

(3) **Additional Parameter Fine-tuning**: Additional parameter fine-tuning introduces an extra set of parameters to the original model, allowing efficient fine-tuning without altering the pre-trained parameters. (1)**Adapter**: Adapter training introduces small neural network structures, known as adapters, between the layers of the pre-trained model. During fine-tuning, only these adapters are trained while the original model parameters remain unchanged. Specifically, adapters can be integrated into various layers of the model in a serial, parallel, or reparameterized manner [279–282], and by adjusting the parameters of these adapters, the performance of the model on specific tasks is enhanced while maintaining the model's generalization capability. However, its performance is limited by the capacity of the adapters and may not fully capture the complexity of highly specialized tasks. (2)**Low-Rank Adaptation (LoRA)**: LoRA [283] involves adding low-rank matrices to the model's parameters and then fine-tuning these matrices to adapt to new tasks. QLORA [284] reduces the memory required for fine-tuning large language models without sacrificing performance by introducing LoRA in frozen, quantized pre-trained language models. This exemplifies LoRA's efficiency in computational resources and memory. However, its performance may be slightly inferior to full model fine-tuning for tasks requiring extensive modifications. (3)**Prefix Tuning**: Prefix tuning adapts to various tasks by adding task-specific prefix vectors to the model's input. For instance, in [285], it demonstrates that by optimizing these prefixes, it is possible to achieve performance comparable to full-parameter fine-tuning with significantly fewer parameters. However, fixed-length prefixes may be insufficient to address the diversity of tasks. To address this, APT [286] employs a gating mechanism to dynamically adjust the prefixes, enhancing the efficiency and effectiveness of fine-tuning, though its applicability to non-Transformer architectures is limited. The advantage of prefix tuning lies in reducing the number of parameters, but it may require task-specific adjustments to the prefixes, and its performance may still be limited for certain tasks. (4)**Prompt Tuning**: Prompt tuning adapts pretrained LLMs to specific tasks by introducing trainable "soft prompts" [287]. This method leverages backpropagation to optimize the prompts while keeping the rest of the model frozen. For example, P-Tuning

[288] stabilizes the training process by combining continuous prompt embeddings with discrete prompts and has achieved significant performance improvements in natural language understanding tasks such as LAMA [289] and SuperGLUE [290]. Although prompt tuning is favored for its parameter efficiency and model reusability, it may require carefully designed prompts and a deep understanding of the task, and it might not fully match the effectiveness of full-parameter fine-tuning for some complex tasks.

Feedback Learning. Feedback learning is an approach that employs feedback information as context, enabling an agent to "reinforce" policy generation iteratively without the need to update weights. Feedback information can take on multiple forms, such as prompt contexts [240], embeddings [287], tokens [288, 291–293]. Reflexion [293] is an innovative feedback learning mechanism that enables language agents to reinforce learning by receiving verbal feedback, rather than through weight updates. The agent reflects on task feedback signals and stores the results of reflection as text in episodic memory, guiding future decision-making processes and thereby improving performance in successive attempts. InstructGPT [205] learns by collecting evaluations from human annotators on the model's output, which include preference rankings for the text generated by the model, serving as a feedback signal. Similarly, DPO [294] directly adjusts model behavior based on user preference rankings, offering a more targeted optimization by aligning outputs with human feedback in a computationally efficient manner.

Prompt Engineering. Prompt engineering is a method that utilizes well-designed prompts and feedback as contextual cues. For example, Retroformer [266] enables an agent to reflect on its past failures, integrating these reflections into prompts to guide future actions.

Prompt engineering has a wide range of applications in large language models. For instance, AutoPrompt [295] enhances GPT-3's performance on specific tasks by generating custom prompts, thus improving its output quality. The AutoPrompt approach demonstrates that by automating the generation and optimization of prompts, the performance of language models on specific tasks can be significantly improved. The core of this method lies in the automatic generation of prompts, which through continuous adjustment and optimization, enables the model to better understand task requirements and produce high-quality outputs.

Prefix-tuning [285] is another prompt engineering technique that involves adding prefixes to prompts, allowing the language model to better understand and execute specific tasks. Prefix-tuning shows that by optimizing prompts without changing the model weights, the performance of the model can be significantly enhanced. This method adds specific prefixes to input prompts, enabling the model to reference more contextual information during generation, thus improving the relevance and accuracy of the output.

Reinforcement Learning. In reinforcement learning, an agent learns the optimal strategy through interaction with the environment. Each action produces corresponding feedback (such as rewards or penalties), and the agent continually adjusts its strategy based on this feedback to maximize cumulative rewards. The core of reinforcement learning lies in trial and error and optimization, where the agent gradually learns to make optimal decisions in different contexts through multiple trials and errors. For example, ICPI [296] learns in context by using large language models to perform policy iteration without expert demonstrations or gradient updates, improving strategies through trial-and-error interaction. GLAM [297] employs online reinforcement learning, allowing the LLM Agent to gradually adjust its strategy through interaction with the environment, thereby enhancing performance in achieving specific goals. InstructGPT [205], on the other hand, fine-tunes GPT-3 through reinforcement learning with human feedback, making it better at following user instructions and improving its alignment and performance across various tasks.

3.5.3 Agents adjustment

A key aspect of the evolution mechanism is the continual updating of agents' existing knowledge and experiences, or the refinement of current decisions and behaviors before execution. This process aims to deepen the agents' cognitive capabilities and enhance their responsiveness to complex and dynamic environments. Through iterative learning and adaptation, agents can improve their performance and maintain relevance in ever-changing contexts.

Memory Update. A significant approach emphasizes the expansion and deepening of agents' self-awareness and learning experiences. This method generally involves agents utilizing memory mechanisms to engage in self-reflection based on collected feedback, through processes of abstraction, summarization, and synthesis. The newly acquired knowledge and experiences are then stored in memory or an external database. For instance, in GITM [129], the agent initially makes explorations in the interaction environment. Upon successfully accomplishing a task, the agent stores the actions used in its memory. Similarly, in AppAgent [298], the agent learns through a dual approach of autonomous exploration and the

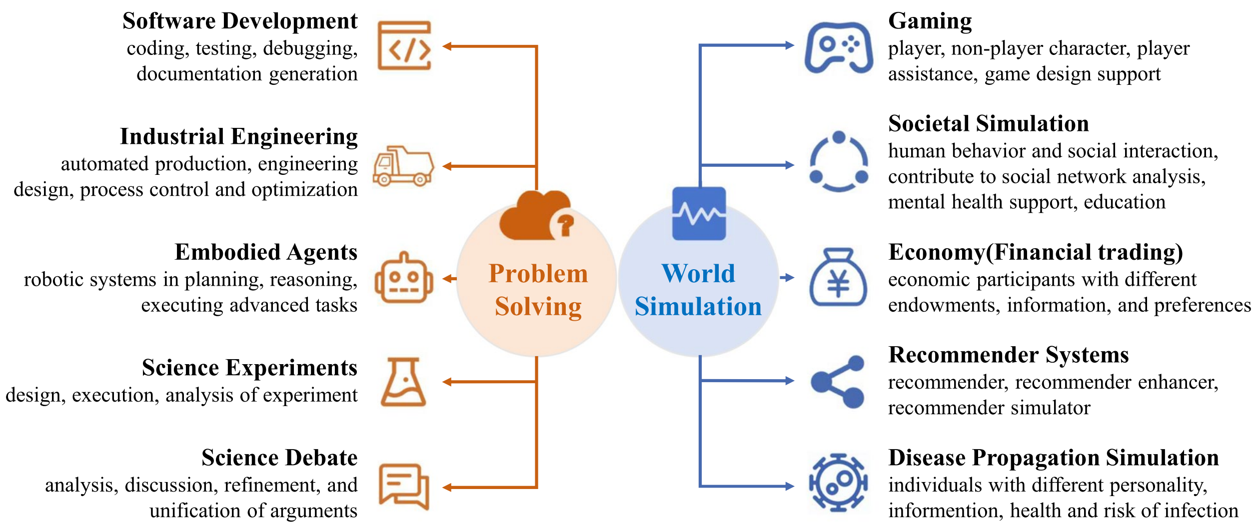

Fig. 7 Diverse application of the llm-based multi-agent system

Table 2 Representative applications of the LLM-based multi-agent system

Application	Domain	Work
Problem Solving	Software Development	Dong et al. [32], ChatEDA [303], LIBRO [304], PENTESTGPT [305]
	Industrial Engineering	Mehta et al. [2], Xia et al. [3], Li et al. [4]
	Embodied Agents	SayCan [8], Inner Monologue [9], TidyBot [10], RoCo [34], CoELA [35]
	Science Experiments	Ghafarollahi et al. [5], Boiko et al. [6], ChemCrow [7]
	Science Debate	Du et al. [36], Liang et al. [264], ChatEval [38]
World Simulation	Gaming	Li et al. [12], Renella et al. [13], MarioGPT [306]
	Societal Simulation	Gao et al. [14], Ma et al. [15], CGMI [16]
	Economy (Financial trading)	Horton et al. [307], Akata et al. [308], Guo et al. [256], CompeteAI [257]
	Recommender Systems	Zhang et al. [309], TALLRec [310], Hou et al. [311], Liu et al. [312], Chat-Rec [313], Dai et al. [314], KAR [315], GENRE [316], LLMRec [317], RecAgent [60], RecSim [318], Agent4Rec [59], AgentCF [49]
	Disease Propagation Simulation	Williams et al. [52], Ghaffarzadegan et al. [319]

observation of human demonstrations. This iterative process facilitates the construction of a comprehensive knowledge base, which subsequently serves as a reference for executing complex tasks across diverse mobile applications. In MemPrompt [299], natural language feedback from users concerning the agent's problem-solving intentions is captured and stored in memory. Subsequently, when the agent confronts analogous tasks, it accesses these stored memories to formulate more appropriate responses.

Self-Reflection. While previous research has predominantly focused on enhancing agents' capabilities for zero-shot task decision-making and efficient execution, a general approach involves agents dynamically evolving by adapting their initial goals and planning strategies based on feedback and communication records. In LMA3 [268], for instance, autonomous goal-setting is a key feature where agents progressively enhance their capabilities through environmental exploration and feedback from a reward mechanism. Through this process, agents accumulate knowledge and develop skills according to their individual preferences. RoCo [34] introduces a method for multi-robot collaboration tasks where agents initiate sub-task plans and plot 3D waypoint paths for each robot. LLMs are employed by agents to refine plans and waypoints until they meet validation criteria. While ReAd [300] takes the advantage function evaluated by a critic as feedback, and revises the plan for more efficient interaction. MemoryBank [134] undertakes conversation processing to distill daily events into concise summaries akin to human memory consolidation of significant experiences. Through ongoing interactions, agents continuously assess and enhance their knowledge base, generating daily insights into evolving personality traits.

Dynamic Generation. In certain contexts, the focus is on the autonomous maintenance of multi-agent systems to ensure their continuous operation. Given the complexity of the environment, the system can dynamically adjust its scale by generating or removing task-specific agents. For example, in [301, 302], they allow the system to effectively scale and tailor its resources, deploying agents specifically crafted to address current operational demands and challenges.

4 Application

The application of LLMs spans a wide array of fields, revolutionizing how tasks are performed and how virtual environments are simulated (Fig. 7). This section delves into the diverse applications of LLMs, focusing on their roles in problem-solving and world simulation, illustrating their transformative impact on software development, industrial engineering, science experiments, societal simulation, gaming, and more. We list representative applications in Table 2.

4.1 Problem solving

LLMs are transforming problem-solving across various domains by leveraging their advanced capabilities in natural language understanding and other application [320–323]. These models excel in breaking down complex tasks, offering solutions, and facilitating efficient collaborations among virtual agents. Below, we explore the specific applications of LLM agents in software development, embodied agents, science experiments, and debates.

Software Development. In the fields of computer science and software engineering, LLM-based agents can understand, infer, and generate human-like text by leveraging a training corpus encompassing various domains such as computer science and cybersecurity. This capability enables automation in coding, testing, debugging, and documentation generation. In [32], it employs multiple LLM agents, each acting as a different "expert" to collaboratively handle complex coding tasks through a virtual team approach, thereby enhancing code quality and efficiency. ChatEDA [303] introduces an autonomous agent for Electronic Design Automation (EDA) powered by a fine-tuned LLM, AutoMage, which manages task planning, script generation, and execution, thereby improving the design flow from Register-Transfer Level (RTL) to Graphic Data System Version II (GDSII). LIBRO [304] utilizes a pre-trained LLM to analyze defect reports and generate prospective tests, effectively reproducing a large number of errors from the Defects4J benchmark. PENTESTGPT [305] is a specialized tool that simulates human-like behavior in penetration testing, equipped with reasoning, generation, and parsing modules, allowing it to adopt a divide-and-conquer approach when encountering problems.

Industrial Engineering. The application of LLM Agents in the industrial sector encompasses various areas such as automated production, engineering design, process control, and optimization. With their robust natural language processing capabilities, LLM Agents can comprehend and generate complex instructions and information, thereby automating task execution and data analysis processes. Additionally, these agents can continuously improve their performance through learning and adaptation, offering more accurate and efficient decision-making. For instance, in civil engineering [2], it proposes a 3D interactive framework where an interactive agent can understand natural language instructions to place building blocks and detect confusion, seeking clarification based on human feedback. In automated production [3], it integrates LLM Agents with digital twin systems, enabling intelligent planning and control of production processes, thereby enhancing production efficiency and flexibility. In the field of chip design [4], it uses GPT-3.5 and GPT-4 agents to assist in developing finite-difference time-domain (FDTD) simulation code and deep reinforcement learning code, optimizing the structure of photonic crystal surface-emitting lasers (PCSEL).

Embodied Agents. With strong commonsense reasoning and language understanding capabilities, LLMs can assist robotic systems in planning, reasoning, and executing advanced tasks through interactions with physical and virtual environments. For instance, SayCan [8] utilizes LLM-generated high-level plans, combining them with environmental states and value functions to create feasible plans for robots, thereby improving task efficiency. Inner Monologue [9] introduces feedback mechanisms, enabling the LLM to continually learn and optimize during the planning process to adapt to complex environments. TidyBot [10] generates personalized household cleaning task plans by learning user preferences, catering to diverse user needs. In multi-robot collaboration, projects such as RoCo [34] employ LLMs for high-level communication and low-level path planning, achieving effective coordination among robotic arms. CoELA [35] demonstrates the coordination and management capabilities of LLMs in decentralized control and complex task planning within multi-robot environments. These applications not only advance the development of embodied intelligence technologies but also provide new possibilities for the intelligent and personalized future of robotic systems.

Science Experiments. The integration of LLMs into scientific disciplines has facilitated the creation of intelligent agents capable of autonomously conducting chemical experiments. These agents, harnessing the capabilities of LLMs, have automated the entire experimental process, from design to execution, representing a significant advancement in laboratory automation. The ProtAgents platform [5], which employs multi-agent collaborations and LLMs for de novo protein design, integrating physical simulations with machine learning. In [6], it presents an intelligent agent system that amalgamates multiple LLMs to tackle intricate scientific tasks, such as catalyzed cross-coupling reactions, thereby showcasing the scientific research proficiency of LLM Agents. Furthermore, the introduction of ChemCrow [7], an LLM-based agent equipped with 17 meticulously developed tools, has streamlined the planning and execution of chemical synthesis.

Science Debate. LLM Agents excel in scientific debates, drawing from their broad training and ability to produce coherent, contextually fitting responses. Debates are typically structured into rounds, where multiple instances of LLM offer analyses, engage in collaborative discussions, and refine arguments until consensus or a reasoned conclusion is achieved. In [36], it deploys multiple instances of LLMs in debates to achieve consensus, thereby enhancing reasoning and factual accuracy. The Multi-Agent Debate (MAD) framework [264] encourages divergent thinking in LLMs, addressing the issue of Domain of Thought (DoT). Additionally, ChatEval [38] employs multiple agents in a structured debate format to critique and evaluate the outcomes produced by various candidate models, aiding in improving the evaluative performance concerning text quality to better align with human preferences.

4.2 World simulation

Another primary application scenario of LLM-MA is world simulation. LLM agents can comprehend and generate coherent, semantically rich text, thereby simulating human behavior and interaction. This capability enables LLM agents to play various roles in simulating the world and interacting with the environment and other agents, thereby constructing a virtual world with a certain degree of realism. In world simulation, LLM agents can be endowed with different tasks and attributes, such as playing roles in games, simulating human behavior in society, and conducting decision analysis in economics, thus facilitating simulation and research in various domains.

Gaming. The application of LLM agents in the gaming domain encompasses various roles, ranging from acting as players participating in games, simulating non-player character (NPC) dialogues and behaviors, to providing player assistance and game design support [11]. These agents are capable of generating coherent text to enhance the interactivity of in-game characters and the quality of storytelling, while also supporting game design processes such as level generation and concept design. In [12], it employs a variant of the GPT model to simulate players in predicting legal moves in the board game "Othello", revealing evidence of an emergent nonlinear internal representation of the board state despite lacking prior knowledge about the game or its rules. A method based on the game "League of Legends" was proposed to automatically generate live commentary during gameplay, supporting automatic identification of key events and utilizing ChatGPT to generate speech output in [13]. MarioGPT [306] is a fine-tuned GPT-2 model specifically designed to generate tile-based Super Mario game levels from textual prompts, and when combined with novelty search, it produces diverse and playable game content in an open-ended manner.

Societal Simulation. Within the sphere of social sciences, the utilization of LLM Agents primarily revolves around the emulation of human behavior and social interactions. They are capable of engaging in conversations with humans through natural language processing techniques, participating in multi-turn dialogues, and learning social interactions within simulated environments. These agents contribute to areas such as social network analysis, mental health support, and education by analyzing language data, identifying patterns of social behavior, and making decisions or predictions based on this information. In [14], it creates an LLM-based Multi-Agent System using prompt engineering and fine-tuning techniques, encompassing information on emotions, attitudes, and interaction behaviors to support individual and group-level simulations. While in [15], it conducts a qualitative analysis of 2917 user comments based on Replika, a popular and leading LLM-based Conversational Agent, finding that it facilitates on-demand, nonjudgmental support, enhances user confidence, and aids in self-discovery, but has limitations in preventing harmful or false information. Additionally, CGMI [16], as a configurable general Multi-Agent Interaction framework, can be utilized to simulate classroom interactions between teachers and students, indicating a close correlation with real classroom environments concerning teaching methods, curriculum, and student performance.

Economy (Financial Trading). Given the enhanced text comprehension and complex decision-making capabilities of LLM-based agents, researchers endow these agents with attributes such as endowments, information, and preferences to simulate the decision-making of humans or economic participants, conducting in-depth economic and financial research. In [307], it compares the economic behavior of LLMs with actual human behavior by placing LLMs in multiple economic scenarios, such as the dictator game and minimum wage issues, to gain new insights into economics. The studies conducted by [308] and [256] both focus on planning and cooperation in interactive behavior. In [308], it employs behavioral game theory to study cooperation and coordination in LLMs through repeated games, revealing persistent behavioral signatures and the ability of LLMs to adapt strategies based on social preferences. On the other hand, in [256], it investigates the strategic decision-making of GPT in the ultimatum game and the prisoner's dilemma, demonstrating that GPT exhibits human-like responses and can be influenced by traits of fairness concern or selfishness. CompeteAI [257] introduces a versatile competition framework applicable to various competitive situations, simulating a virtual town with two types of agents: restaurants and customers, thereby validating existing classical theories such as social learning and the Matthew effect.

Recommender Systems. In the field of recommender systems, LLMs, due to their powerful domain generalization and language generation capabilities, are often used as recommender and for enhancing or simulating recommender. When used as recommendation models, LLMs can be specialized for personalized recommendations after parameter fine-tuning [309, 310] and can also perform recommendation tasks under a zero-shot paradigm [311, 312]. The introduction of prompt engineering methods [313] can trigger LLMs to perceive the sequence of behavioral order and alleviate potential position bias and popularity bias issues. Additionally, the general knowledge encoded in LLMs can be used to improve traditional recommender systems [315–317] such as by encoding and inferring user information and feeding the resulting informative representations into traditional recommender systems. When LLMs are used as recommendation simulators, most are user-oriented [59, 60, 318], simulating real user behaviors in personalized recommendation systems but failing to understand the essence of user-item relationships. To address this, AgentCF [49] creatively treats both users and items as agents and develops a collaborative learning approach to capture the bidirectional relationship between users and items.

Disease Propagation Simulation. In disease transmission and epidemiological modeling, LLMs can simulate the behaviors and interactions of various agents in disease transmission, aiding researchers in gaining a deeper understanding of disease transmission dynamics and developing effective control strategies. For instance, in [52], it develops generative agents using ChatGPT to mimic behaviors like self-quarantining, which contributed to a more realistic flattening of the epidemic curve. While in [319], it creates a simulated environment with LLM-powered agents that exhibited human-like behaviors, such as changing attitudes and emotions in response to social events.

5 Discussion

While previous work on LLM-based autonomous agents has obtained many remarkable successes, this field is still at its initial stage, and there are several significant challenges that need to be addressed in its development. In the following, we present many representative challenges. Despite the robust capabilities and extensive applications of LLM-based agents, numerous concealed risks persist. In this section, we delve into some of these risks and offer potential solutions or strategies for mitigation.

5.1 Open problem

In the rapidly evolving field of AI, MAS have garnered significant attention due to their potential to tackle complex tasks through collaboration and coordination. However, the implementation and deployment of these systems present numerous challenges. This paper delves into some key open issues encountered in MAS development, with a particular focus on the intrinsic constraints of LLMs, misuse of these systems, challenges in scaling MAS, and the necessity for adaptation to dynamic environments.

LLM's Intrinsic Constraints. This section introduces the inherent limitations of LLMs, covering key issues such as the opacity of their decision-making processes, the tendency to produce hallucinations, and the presence of biases in their outputs. (1)**Black Box Effect and Decision Accuracy Assessment**: LLMs operate as black-box systems, rendering their decision-making processes opaque. This opacity poses significant challenges in evaluating the accuracy and reliability of their decisions, which is crucial in high-stakes applications. To mitigate this issue, existing work often employs methods to explain model decisions [203, 324], guide the generation of reasoning processes [86], and uncover the models' inherent reasoning abilities [224]. In [324], it developed model interpretability techniques that provide insights into LLMs' decision-making by highlighting the importance of individual input features. SHAP employs game theory to assess feature contributions, offering both local and global explanations, while LIME approximates predictions using local linear models, enhancing trust and understanding of model outputs. These techniques render the inner workings of complex models more transparent. (2)**Hallucination**: LLMs can produce information that sounds plausible but is factually incorrect or nonsensical, known as hallucination [199]. This phenomenon may arise from the model's overgeneralization of training data or misinterpretation of incomplete or misleading information. Researchers have adopted various strategies to address this challenge, including integrating external knowledge bases to enhance information accuracy [143, 238], increasing model transparency to foster understanding of decision processes [86, 202, 203], developing fact-checking systems to verify outputs [200, 201], and designing hallucination detection tasks [325, 326] to evaluate and improve model performance. For example, CoVe [202] encourages models to generate initial responses, followed by verification queries to check the draft's factual accuracy before producing a refined response, thereby enhancing output accuracy. (3)**Bias**: Bias in LLMs manifests as the propagation and amplification of discriminatory tendencies present in training data, such as racial and gender biases, leading to unfair or harmful outputs [327–330]. Detecting and mitigating these biases is crucial for developing fair and ethical AI systems. Techniques such as rebalancing training datasets [331–333], applying bias mitigation algorithms [334–336], and regularly auditing model outputs [337, 338] are essential in this regard. For instance, BERT [339] has been enhanced for bias robustness through adversarial training, while the GPT series [64, 240] incorporates human feedback to optimize models and reduce inappropriate behavior. These methods collectively advance the construction of more just and non-discriminatory AI systems.

Misuse. Despite the powerful capabilities of MAS and LLMs, they can be maliciously exploited for large-scale disinformation generation [340–342], cyber-attacks [343–346], and other inappropriate behaviors [347, 348]. Such misuse can pose threats to individual, societal, and national security. To prevent these threats, researchers have implemented various measures. For instance, some studies employ methods such as instruction processing and malicious detection to eliminate potential adversarial contexts or malicious intents [349–351]. Adversarial training and prompting [352, 353] enhance the robustness of agents, enabling them to withstand malicious inputs and attacks. Additionally, establishing AI ethics and policies guides the development and deployment of agent systems, ensuring they operate within ethical and legal frameworks, thereby reducing the risk of misuse [354, 355]. These comprehensive measures contribute to the enhanced security of multi-agent systems, preventing their exploitation for improper purposes.

Scaling Up the Multi-Agent System. Scaling up multi-agent systems involves increasing the number of agents to achieve larger-scale social simulations and more complex task processing. While this process can enhance system performance and realism, it also introduces challenges related to computational resources, communication efficiency, and system coordination. To address these difficulties, researchers have adopted various strategies:

Firstly, static adjustment and dynamic scaling methods are widely applied [356]. Static adjustment methods [37, 255] design systems by pre-determining the number and roles of agents, which is effective for fixed tasks or goals but lacks flexibility in response to task changes. Dynamic scaling methods allow systems to adjust the number of agents during operation based on demand, providing greater adaptability and flexibility. For example, AGENTVERSE [302] optimizes task execution efficiency and quality by dynamically adjusting team composition and role allocation by simulating human team collaboration.

Secondly, optimization of communication and coordination mechanisms helps reduce biases and redundancies in the information dissemination process. Existing work often improves cooperation efficiency among agents through role specialization and standardized operating procedures. For instance, MetaGPT [31] employs a structured communication mechanism by defining message formats and sharing message pools, reducing ambiguities in agent communication, while introducing a publish-subscribe mechanism to effectively manage information flow and avoid information overload.

Lastly, innovations in system architecture and design are crucial for the stable operation of large-scale multi-agent systems. This involves building system architectures that support distributed computing and efficient data management, and designing agents that can flexibly adapt to different environments and tasks. In [357], it constructed a cascading architecture of large language models that intelligently allocate tasks to either cost-effective models or more powerful but costlier models based on answer consistency, effectively reducing the cost of scaling multi-agent systems.

Dynamics Environment Adaptation. Dynamic environment adaptation refers to the capability of AI agents to operate effectively in constantly changing environments. This capability requires agents to not only understand the state of the environment but also predict and adapt to changes to achieve continuous task execution and goal attainment. The dynamic nature of the environment arises partly from the heterogeneity of multi-modal data streams and partly from the continual iteration of external conditions and task demands.

Regarding multi-modal data streams, existing work enhances LLM agents' data processing and comprehension abilities through external integration and internal processing methods. Firstly, by integrating multi-modal models, LLM agents can process and understand various data types such as images, videos, and speech by converting multi-modal inputs into text. For example, MMReact [358] completes multi-modal reasoning tasks by combining a library of visual experts with language models. Additionally, some models like LLaVA [154] and PALM-E [71] improve their understanding and generation capabilities of visual information by training on large-scale text-image paired datasets during the pre-training stage, supporting agents to directly handle multi-modal inputs and improve performance in multi-modal tasks.

Furthermore, to address the continual iteration of external conditions and task demands, researchers have designed flexible task execution frameworks and continual learning mechanisms. For instance, through instruction tuning [359, 360] and alignment tuning [205, 361], LLMs can better adapt to specific tasks and human values. AgentTuning [360] enhances LLMs' ability to execute complex real-world tasks by combining lightweight instruction adjustment datasets. Additionally, using in-context learning and continual learning methods, agents can quickly absorb new information and update knowledge bases, thereby better adapting to environmental changes and new task requirements.

5.2 Future direction

Envisioning the future of LLM agents necessitates addressing the challenges and trends currently shaping this field. This section delves into three significant future directions: the development of collective intelligence in AI agents, the deployment of MAS as reliable and efficient services, and the expansion of these systems' applications across various domains. By exploring these areas, we aim to enhance the capabilities of LLM agents, making them more sophisticated, reliable, and versatile in mimicking human perception and interaction.

Collective Intelligence in AI Agents. Collective intelligence emphasizes integrating diverse perspectives and decision-making through collaboration and competition among agents, thereby forming group wisdom that surpasses individual capabilities [20]. The key to constructing such systems lies in designing effective coordination mechanisms to avoid groupthink and cognitive biases while promoting cooperation and enhancing collective intellectual performance.

A potential strategy to achieve this balance is the use of decentralized learning algorithms, where agents can learn and update their knowledge bases independently while periodically sharing insights with the team. This approach ensures each agent maintains its individuality while benefiting from collective intelligence. Additionally, incorporating mechanisms for conflict resolution and consensus-building can help maintain harmony within the agent group, fostering more robust and adaptive collective intelligence.

Moreover, reinforcement learning algorithms [362] provides a powerful tool for achieving collective intelligence, allowing LLM-MA to adjust based on immediate feedback from the environment or humans. However, current research often focuses on individual agents' memory and evolution techniques, which may result in suboptimal collective performance due to individual optimization [363, 364]. This limits the potential for collective intelligence within agent networks. Consequently, achieving optimal collective intelligence through the coordinated adjustment of multiple agents remains a critical challenge.

LLM-based Agent System as Service. The introduction of LLM multi-agent systems as a service (AaaS) heralds a significant shift in the service model within the AI domain [365, 366]. This model offers intelligent agent systems as a service via cloud platforms, reducing technical barriers and enhancing service reliability and efficiency. Users can access advanced agent services on demand without the need to build and maintain complex infrastructure, which is particularly appealing to small and medium-sized enterprises and individual users.

In practical implementation, AaaS must consider the coordination and communication mechanisms of agents to ensure effective cooperation among different agents, providing a coherent service experience. Additionally, AaaS platforms must be highly configurable, allowing users to adjust the agents' behavior and functionality according to their needs. For instance, OpenAI's API service enables users to guide agent behavior through customized prompts to achieve specific tasks.

However, the successful implementation of AaaS also faces challenges. First, the decision-making process of agents needs to be transparent and interpretable to gain users' trust. Second, as the scale of service expands, ensuring system stability and response speed, especially in high concurrency scenarios, is a key issue. Furthermore, the personalization and intelligence level of agent services need continuous improvement to meet users' expectations for service quality.

Application Expansion. In multi-modal and dynamically changing environments, the future development of MAS will focus on enhancing their adaptability and flexibility. With the continuous advancement of AI technology, MAS will be able to more accurately understand complex data streams and respond quickly in changing environments. For example, by integrating advanced machine learning and deep learning algorithms, MAS will be able to process information from different sensors and data sources, achieving more refined situational awareness. In downstream applications, the expansion of

MAS will bring innovation to fields such as healthcare, traffic management, and environmental monitoring. Particularly in healthcare, MAS can provide more accurate diagnostic and treatment recommendations by analyzing patients' multi-modal health records. In traffic management, MAS can optimize traffic signal control and reduce congestion by analyzing real-time traffic flow and accident data.

However, MAS faces challenges in data fusion, real-time processing, and decision-making when realizing these application expansions. Future research needs to explore more efficient data processing frameworks and algorithms to ensure MAS can adapt to constantly changing environmental demands while maintaining high performance. Additionally, ensuring the security and privacy protection of MAS is an important aspect that cannot be overlooked in future development. Through continuous technological innovation and interdisciplinary collaboration, MAS is expected to play a greater role in multiple fields, bringing more convenience and value to society.

6 Conclusion

In this paper, we have systematically provided an overview of LLM-based multi-agent systems, comprehensively reviewing the current research studies in this domain. We began by elucidating the origin and definition of agents, tracing their developmental trajectory from single agents to multi-agent systems. Motivated by the workflow of multi-agent systems, we systematically proposed a general framework comprising five main components: profile, perception, agent's self-action (including memory, knowledge, agent's ability, and action), mutual interaction, and evolution. For each module, we discussed and summarized specific application methods and workflows. Subsequently, we introduced the wide-ranging applications of LLM-basedv multi-agent systems, categorizing them into two sections: problem-solving and world simulation. Finally, the paper delved into current challenges, such as the intrinsic constraints of LLMs, adaptation to dynamic environments, and potential developmental directions for LLM-based multi-agent systems, such as collective intelligence. Despite the fact that current research is still somewhat distant from achieving ideal, reliable, and autonomous system applications, we believe that LLM-based agents represent a significant step forward.

Abbreviations

LLMs	Large language models
MAS	Multi-agent systems
RL	Reinforcement learning
MLLMs	Multi-modal large language models
CoT	Chain-of-thought
VLMs	Visual language models
ViT	Vision transformer
VQVAE	Vector quantized variational autoencoder
Q-Former	Querying transformer
AST	Audio spectrogram transformer
FIFO	First-in-first-out
CNNs	Convolutional neural networks
RNNs	Recurrent neural networks
RAG	Retrieval-augmented generation
ToT	Tree of Thought
PDDL	Planning domain definition languages
ICL	In-context learning
LoRA	Low-rank adaptation
DMAS	Decentralized multi-agent communication framework
SOPs	Standardized operating procedures
EDA	Electronic design automation
RTL	Register-transfer level
GDSII	Graphic data system version II
FDTD	Finite-difference time-domain
PCSEL	Photonic crystal surface-emitting lasers
MAD	Multi-agent debate
DoT	Domain of thought

Acknowledgements
We extend our gratitude to the authors of the referenced works mentioned in this paper for their invaluable contributions, which have provided substantial material support for our study. This article exclusively represents the views and conclusions of its authors, and we hope it will offer insightful understanding and assistance to readers.

Authors' contributions
The completion of this paper was a result of the collaborative efforts of all authors. The specific contributions of each author are enumerated as follows: Xinyi Li: Conceptualization, Methodology, Validation, Investigation, Formal Analysis, Resources, Data Curation, Visualization, Writing - Original Draft. Sai Wang: Conceptualization, Methodology, Resources, Supervision, Writing - Review & Editing. Siqi Zeng: Investigation, Resources, Data Curation, Visualization, Writing - Original Draft. Yu Wu: Writing - Review & Editing, Supervision, Project Administration, Funding Acquisition. Yi Yang: Supervision; Project Administration.

Funding
This work was partially supported by the National Natural Science Foundation of China (Ref. No.: 62372341) and the Fundamental Research Funds for the Central Universities (Ref. No.: 2042024kf0040).

Availability of data and materials
The authors confirm that the data and materials supporting the findings of this study are available within the article. The readers can access the relevant data by referring to the cited references within the paper.

Code availability
Not applicable.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper, may be limited by the attached paper reports the research.

Received: 15 July 2024 Revised: 21 August 2024 Accepted: 28 August 2024
Published online: 08 October 2024

References

1. Y. Dong, X. Zhu, Z. Pan, L. Zhu, Y. Yang, C. ReLER, Villageragent: A graph-based multi-agent framework for coordinating complex task dependencies in minecraft. in Findings of the Association for Computational Linguistics ACL 2024 (Association for Computational Linguistics, Bangkok, Thailand and virtual meeting, 2024), pp. 16290–16314.
2. N. Mehta, M. Teruel, P.F. Sanz, X. Deng, A.H. Awadallah, J. Kiseleva, Improving grounded language understanding in a collaborative environment by interacting with agents through help feedback. in Findings of the Association for Computational Linguistics: EACL 2024 (Association for Computational Linguistics, St. Julian's, Malta, 2024), pp. 1306–1321.
3. Y. Xia, M. Shenoy, N. Jazdi, M. Weyrich, Towards autonomous system: flexible modular production system enhanced with large language model agents. in *2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)* (IEEE, Sinaia, Romania, 2023), pp. 1–8
4. R. Li, C. Zhang, S. Mao, H. Huang, M. Zhong, Y. Cui, X. Zhou, F. Yin, Z. Zhang, From english to pcsel: LLM helps design and optimize photonic crystal surface emitting lasers (2023). arXiv preprint (2023) arXiv:2104.12145
5. A. Ghafarollahi, M.J. Buehler, Protagents: Protein discovery via large language model multi-agent collaborations combining physics and machine learning. in The Twelfth International Conference on Learning Representations (Digital Discovery, Vienna, Austria, 2024). **3**, pp. 1389-1409
6. D.A. Boiko, R. MacKnight, G. Gomes, Emergent autonomous scientific research capabilities of large language models. arXiv preprint (2023) arXiv:2304.05332
7. A.M. Bran, S. Cox, O. Schilter, C. Baldassari, A.D. White, P. Schwaller, Chemcrow: Augmenting large-language models with chemistry tools.*Nat Mach Intell* **6**, 525–535 (2024). https://doi.org/10.1038/s42256-024-00832-8
8. A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian et al., Do as i can, not as i say: Grounding language in robotic affordances. in *Conference on robot learning* (PMLR, Atlanta, GA, USA, 2023), pp. 287–318
9. W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar et al., Inner monologue: Embodied reasoning through planning with language models. in Proceedings of The 6th Conference on Robot Learning. Proceedings of Machine Learning Research (PMLR, Aucklang, New Zealand, 2023), vol. 205, pp. 1769–1782
10. J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, T. Funkhouser, Tidybot: Personalized robot assistance with large language models. Auton. Robot. **47**(8), 1087–1102 (2023)
11. R. Gallotta, G. Todd, M. Zammit, S. Earle, A. Liapis, J. Togelius, G.N. Yannakakis, Large language models and games: A survey and roadmap. arXiv preprint arXiv:2402.18659 (2024)
12. K. Li, A.K. Hopkins, D. Bau, F. Viégas, H. Pfister, M. Wattenberg, Emergent world representations: Exploring a sequence model trained on a synthetic task. in The Eleventh International Conference on Learning Representations (Kigali, Rwanda, 2023)
13. N. Renella, M. Eger, Towards automated video game commentary using generative ai. in Proceedings of the Experimental Artificial Intelligence in Games Workshop co-located with the 19th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2023) (AAAI Press, Washington, DC, USA, 2023)
14. C. Gao, X. Lan, Z. Lu, J. Mao, J. Piao, H. Wang, D. Jin, Y. Li, S ^{3}: Social-network simulation system with large language model-empowered agents. Available at SSRN: https://ssrn.com/abstract=4607026 or https://doi.org/10.2139/ssrn.4607026
15. Z. Ma, Y. Mei, Z. Su, Understanding the benefits and challenges of using large language model-based conversational agents for mental well-being support. in *AMIA Annual Symposium Proceedings* (American Medical Informatics Association, New Orleans, LA, 2023). pp. 1105
16. S. Jinxin, Z. Jiabao, W. Yilei, W. Xingjiao, L. Jiawen, H. Liang, Cgmi: Configurable general multi-agent interaction framework. arXiv preprint (2023) arXiv:2308.12503
17. L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin et al., A survey on large language model based autonomous agents. Front. Comput. Sci. **18**(6), 186345 (2024)
18. T. Guo, X. Chen, Y. Wang, R. Chang, S. Pei, N.V. Chawla, O. Wiest, X. Zhang, Large language model based multi-agents: A survey of progress and challenges. in 33rd International Joint Conference on Artificial Intelligence (IJCAI 2024) (Jeju Island, South Korea, 2024). IJCAI; Cornell arxiv:2308.12503
19. Y. Cheng, C. Zhang, Z. Zhang, X. Meng, S. Hong, W. Li, Z. Wang, Z. Wang, F. Yin, J. Zhao et al., Exploring large language model based intelligent agents: Definitions, methods, and prospects. arXiv preprint (2024) arXiv:2401.03428
20. Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou et al., The rise and potential of large language model based agents: A survey. arXiv preprint (2023) arXiv:2309.07864
21. J. Hu, P. Bhowmick, I. Jang, F. Arvin, A. Lanzon, A decentralized cluster formation containment framework for multirobot systems. IEEE Trans. Robot. **37**(6), 1936–1955 (2021)
22. G. Weiss, *Multiagent systems: a modern approach to distributed artificial intelligence* (MIT Press, Cambridge, MA, United States, 1999), pp. 547
23. J. He, C. Treude, D. Lo, Llm-based multi-agent systems for software engineering: Vision and the road ahead. arXiv preprint (2024) arXiv:2404.04834
24. S. Hu, Y. Zhong, M. Gao, W. Wang, H. Dong, Z. Li, X. Liang, Y. Yang, X. Chang, Marllib: Extending rllib for multi-agent reinforcement learning (2022). arXiv preprint arXiv:2210.13708, 2022b
25. H.V.D. Parunak, Applications of distributed artificial intelligence in industry. Found. Distrib. Artif. Intell. **2**(1), 18 (1996)
26. P. Stone, M. Veloso, Multiagent systems: A survey from a machine learning perspective. Auton. Robot. **8**, 345–383 (2000)
27. K.S. Decker, Distributed problem-solving techniques: A survey. IEEE Trans. Syst. Man Cybern. **17**(5), 729–740 (1987)
28. J.S. Park, J. O'Brien, C.J. Cai, M.R. Morris, P. Liang, M.S. Bernstein, Generative agents: Interactive simulacra of human behavior. in *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (Association for Computing Machinery, New York, NY, United State, 2023), pp. 1–22
29. I. Dasgupta, C. Kaeser-Chen, K. Marino, A. Ahuja, S. Babayan, F. Hill, R. Fergus, Collaborating with language models for embodied reasoning. in NeurIPS 2022 Foundation Models for Decision Making Workshop (New Orleans, Louisiana, United States of America, 2022)
30. C. Qian, W. Liu, H. Liu, N. Chen, Y. Dang, J. Li, C. Yang, W. Chen, Y. Su, X. Cong, et al.: Chatdev: Communicative agents for software development. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Bangkok, Thailand, 2024), pp. 15174–15186
31. S. Hong, X. Zheng, J. Chen, Y. Cheng, J. Wang, C. Zhang, Z. Wang, S.K.S. Yau, Z. Lin, L. Zhou et al., Metagpt: Meta programming for multi-agent collaborative framework. in The Twelfth International Conference on Learning Representations (Vienna, Austria, 2024)
32. Y. Dong, X. Jiang, Z. Jin, G. Li, Self-collaboration code generation via chatgpt. ACM Transactions on Software Engineering and Methodology (New York, NY, USA, 2024), ISSN:1049-331X
33. Y. Chen, J. Arkin, Y. Zhang, N. Roy, C. Fan, Scalable multi-robot collaboration with large language models: Centralized or decentralized systems? in *2024 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, Yokohama, Japan, 2024), pp. 4311–4317
34. Z. Mandi, S. Jain, S. Song, Roco: Dialectic multi-robot collaboration with large language models. In: 2024 IEEE International Conference on Robotics and Automation (ICRA) (IEEE, Yokohama, Japan, 2024), pp. 286–299.
35. H. Zhang, W. Du, J. Shan, Q. Zhou, Y. Du, J.B. Tenenbaum, T. Shu, C. Gan, Building cooperative embodied agents modularly with large language models. in NeurIPS 2023 Foundation Models for Decision Making Workshop (New Orleans, Louisiana, United States, 2023)
36. Y. Du, S. Li, A. Torralba, J.B. Tenenbaum, I. Mordatch, Improving factuality and reasoning in language models through multiagent debate. in Proceedings of the 41st International Conference on Machine Learning. Proceedings of Machine Learning Research (Vienna, Austria, 2024). **235**, pp. 11733–11763. PMLR.
37. K. Xiong, X. Ding, Y. Cao, T. Liu, B. Qin, Examining the inter-consistency of large language models: An in-depth analysis via debate. in Findings of the Association for Computational Linguistics: EMNLP 2023

(Association for Computational Linguistics, Singapore, 2023), pp. 7572-7590
38. C.M. Chan, W. Chen, Y. Su, J. Yu, W. Xue, S. Zhang, J. Fu, Z. Liu, Chateval: Towards better llm-based evaluators through multi-agent debate. in The Twelfth International Conference on Learning Representations (Vienna, Austria, 2024)
39. X. Tang, A. Zou, Z. Zhang, Y. Zhao, X. Zhang, A. Cohan, M. Gerstein, Medagents: Large language models as collaborators for zero-shot medical reasoning. in ICLR 2024 Workshop on Large Language Model (LLM) Agents (Vienna, Austria, 2024)
40. J.S. Park, L. Popowski, C. Cai, M.R. Morris, P. Liang, M.S. Bernstein, Social simulacra: Creating populated prototypes for social computing systems. in *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology (Association for Computing Machinery, New York, NY, United State)*. **2022**, pp. 1–18
41. Z. Kaiya, M. Naim, J. Kondic, M. Cortes, J. Ge, S. Luo, G.R. Yang, A. Ahn, Lyfe agents: Generative agents for low-cost real-time social interactions. arXiv preprint (2023) arXiv: 2310.02172
42. C. Li, X. Su, C. Fan, H. Han, C. Xue, C. Zheng, Quantifying the impact of large language models on collective opinion dynamics. arXiv preprint (2023) arXiv:2308.03313
43. Y. Xu, S. Wang, P. Li, F. Luo, X. Wang, W. Liu, Y. Liu, Exploring large language models for communication games: An empirical study on werewolf. arXiv preprint (2023) arXiv:2309.04658
44. J. Light, M. Cai, S. Shen, Z. Hu, Avalonbench: Evaluating llms playing the game of avalon. in *NeurIPS 2023 Foundation Models for Decision Making Workshop* (2023) (New Orleans, United States, 2023)
45. G. Mukobi, H. Erlebach, N. Lauffer, L. Hammond, A. Chan, J. Clifton, Welfare diplomacy: Benchmarking language model cooperation. in *NeurIPS 2023 Socially Responsible Language Modelling Research workshop* (New Orleans, United States, 2023)
46. G.V. Aher, R.I. Arriaga, A.T. Kalai, Using large language models to simulate multiple humans and replicate human subject studies. in *International Conference on Machine Learning* (PMLR, Honolulu, Hawaii, USA, 2023), pp. 337–371
47. J. Zhang, X. Xu, S. Deng, Exploring collaboration mechanisms for llm agents: A social psychology view. *In ICLR 2024 Workshop on Large Language Model Agent*s (Vienna, Austria, 2023)
48. A. Zhang, Y. Chen, L. Sheng, X. Wang, T.S. Chua, On generative agents in recommendation. in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2024) (Association for Computing Machinery, New York, NY, United State), pp. 1807–1817
49. J. Zhang, Y. Hou, R. Xie, W. Sun, J. McAuley, W.X. Zhao, L. Lin, J.R. Wen, Agentcf: Collaborative learning with autonomous language agents for recommender systems. in *Proceedings of the ACM on Web Conference 2024* (Association for Computing Machinery, New York, NY, United States, Singapore, 2024), pp. 3679–3689
50. N. Li, C. Gao, Y. Li, Q. Liao, Large language model-empowered agents for simulating macroeconomic activities. arXiv preprint (2023) arXiv:2310.10436
51. Y. Li, Y. Yu, H. Li, Z. Chen, K. Khashanah, Tradinggpt: Multi-agent system with layered memory and distinct characters for enhanced financial trading performance. arXiv preprint (2023) arXiv:2309.03736
52. R. Williams, N. Hosseinichimeh, A. Majumdar, N. Ghaffarzadegan, Epidemic modeling with generative agents. arXiv preprint (2023) arXiv:2307.04986
53. C. Zhang, K. Yang, S. Hu, Z. Wang, G. Li, Y. Sun, C. Zhang, Z. Zhang, A. Liu, S.C. Zhu et al., Proagent: Building proactive cooperative ai with large language models. in Proceedings of AAAI Conference on Artificial Intelligent (Vancouver, Canada, 2024), **38**(16), 17591-17599. https://doi.org/10.1609/aaai.v38i16.29710
54. W. Li, D. Qiao, B. Wang, X. Wang, B. Jin, H. Zha, Semantically aligned task decomposition in multi-agent reinforcement learning. arXiv preprint (2023) arXiv:2305.10865
55. B. Yu, H. Kasaei, M. Cao, Co-navgpt: Multi-robot cooperative visual semantic navigation using large language models. arXiv preprint (2023) arXiv:2310.07937
56. M. Safdari, G. Serapio-García, C. Crepy, S. Fitz, P. Romero, L. Sun, M. Abdulhai, A. Faust, M. Matarić, Personality traits in large language models. arXiv preprint (2023) arXiv:2307.00184
57. S. Wang, C. Liu, Z. Zheng, S. Qi, S. Chen, Q. Yang, A. Zhao, C. Wang, S. Song, G. Huang, Avalon's game of thoughts: Battle against deception through recursive contemplation. arXiv preprint (2023) arXiv:2310.01320
58. D. Zhang, Z. Li, P. Wang, X. Zhang, Y. Zhou, X. Qiu, Speechagents: Human-communication simulation with multi-modal multi-agent systems. arXiv preprint (2024) arXiv:2401.03945
59. A. Zhang, L. Sheng, Y. Chen, H. Li, Y. Deng, X. Wang, T.S. Chua, On generative agents in recommendation. in Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (Association for Computing Machinery, New York, NY, USA, 2024). SIGIR '24, pp. 1807–1817. https://doi.org/10.1145/3626772.3657844
60. L. Wang, J. Zhang, X. Chen, Y. Lin, R. Song, W.X. Zhao, J.R. Wen, Recagent: A novel simulation paradigm for recommender systems. arXiv preprint (2023) arXiv:2306.0255
61. L.P. Argyle, E.C. Busby, N. Fulda, J.R. Gubler, C. Rytting, D. Wingate, Out of one, many: Using language models to simulate human samples. Polit. Anal. **31**(3), 337–351 (2023)
62. D.H. Hubel, T.N. Wiesel, Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. J. Physiol. **160**(1), 106 (1962)
63. N.K. Logothetis, D.L. Sheinberg, Visual object recognition. Ann. Rev. Neurosci. **19**(1), 577–621 (1996)
64. J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F.L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat et al., Gpt-4 technical report. arXiv preprint (2023) arXiv:2303.08774
65. J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler et al., Emergent abilities of large language models. Transactions on Machine Learning Research (2022).
66. S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y.T. Lee, Y. Li, S. Lundberg et al., Sparks of artificial general intelligence: Early experiments with gpt-4. arXiv preprint (2023) arXiv:2303.12712
67. R. Wang, P. Jansen, M.A. Côté, P. Ammanabrolu, Scienceworld: Is your agent smarter than a 5th grader? in Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022), pp. 11279–11298.
68. M. Shridhar, X. Yuan, M.A. Côté, Y. Bisk, A. Trischler, M. Hausknecht, Alfworld: Aligning text and embodied environments for interactive learning. International Conference on Learning Representation (2021)
69. Meta Fundamental AI Research Diplomacy Team (FAIR)†, A. Bakhtin, N. Brown, E. Dinan, G. Farina, C. Flaherty, D. Fried, A. Goff, J. Gray, H. Hu et al., Human-level play in the game of diplomacy by combining language models with strategic reasoning. Science **378**(6624), 1067–1074 (2022)
70. M. Fırat, S. Kuleli, What if gpt4 became autonomous: The auto-gpt project and use cases. J. Emerg. Comput. Technol. **3**(1), 1–6 (2023)
71. D. Driess, F. Xia, M.S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu et al., Palm-e: An embodied multimodal language model. in Proceedings of the 40th International Conference on Machine Learning (Honolulu, Hawaii, USA, 2023). ICML'23. JMLR.org 340, 20.
72. I. Kecskés, I. Kecskes, L.R. Horn, *Explorations in pragmatics: Linguistic, cognitive and intercultural aspects* (Mouton de Gruyter, New York, 2007). https://doi.org/10.1515/9783110198843
73. M.A. Mamun, H.M. Abdullah, M.G.R. Alam, M.M. Hassan, M.Z. Uddin, Affective social anthropomorphic intelligent system. Multimed. Tools Appl. **82**(23), 35059–35090 (2023)
74. A. Madasu, M. Firdaus, A. Ekbal, A unified framework for emotion identification and generation in dialogues. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop (Association for Computational Linguistics, Dubrovnik, Croatia, 2023), pp. 73–78.
75. C.H. Song, J. Wu, C. Washington, B.M. Sadler, W.L. Chao, Y. Su, Llm-planner: Few-shot grounded planning for embodied agents with large language models. in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (IEEE, Paris, France, 2023), pp. 2998–3009
76. Y. Bang, S. Cahyawijaya, N. Lee, W. Dai, D. Su, B. Wilie, H. Lovenia, Z. Ji, T. Yu, W. Chung et al., A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. in Proceedings of

the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers) (Association for Computational Linguistics, Nusa Dua, Bali, 2023), pp. 675–718.
77. Z. Zhang, H. Zhao, Advances in multi-turn dialogue comprehension: A survey. arXiv preprint (2021) arXiv:2103.03125
78. W. Tan, Z. Ding, W. Zhang, B. Li, B. Zhou, J. Yue, H. Xia, J. Jiang, L. Zheng, X. Xu et al., Towards general computer control: A multimodal agent for red dead redemption ii as a case study. in ICLR 2024 Workshop on Large Language Model (LLM) Agents (Vienna, Austria, 2024).
79. J. Yang, Y. Dong, S. Liu, B. Li, Z. Wang, C. Jiang, H. Tan, J. Kang, Y. Zhang, K. Zhou et al., Octopus: Embodied vision-language programmer from environmental feedback. arXiv preprint (2023) arXiv:2310.08588
80. S. Reed, K. Zolna, E. Parisotto, S.G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J.T. Springenberg et al., A generalist agent. Transactions on Machine Learning Research (2022).
81. S. Zheng, Y. Feng, Z. Lu et al., Steve-eye: Equipping llm-based embodied agents with visual perception in open worlds. in *The Twelfth International Conference on Learning Representations* (Kigali, Rwanda, 2023)
82. Y. Yang, Y. Zhuang, Y. Pan, Multiple knowledge representation for big data artificial intelligence: framework, applications, and case studies. Front. Inf. Technol. Electron. Eng. **22**(12), 1551–1558 (2021)
83. M. Cornia, M. Stefanini, L. Baraldi, R. Cucchiara, Meshed-memory transformer for image captioning. in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (IEEE, Seattle, WA, USA, 2020), pp. 10578–10587
84. K. Li, Y. He, Y. Wang, Y. Li, W. Wang, P. Luo, Y. Wang, L. Wang, Y. Qiao, Videochat: Chat-centric video understanding. arXiv preprint (2023) arXiv:2305.06355
85. Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, Z. Sui, A survey on in-context learning. arXiv preprint (2022) arXiv:2301.00234
86. J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q.V. Le, D. Zhou et al., Chain-of-thought prompting elicits reasoning in large language models. Adv. Neural Inf. Process. Syst. **35**, 24824–24837 (2022)
87. W. Ma, D. Wu, Y. Sun, T. Wang, S. Liu, J. Zhang, Y. Xue, Y. Liu, Combining fine-tuning and llm-based agents for intuitive smart contract auditing with justifications. in Proceedings of 47th International Conference on Software Engineering (Association for Computing Machinery, New York, NY, United States, 2024).
88. H. Gao, Y. Zhang, Memory sharing for large language model based agents. arXiv preprint (2024) arXiv:2404.09982
89. J. Lin, D. Fried, D. Klein, A. Dragan, Inferring rewards from language in context. in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Association for Computational Linguistics, Dublin, Ireland, 2022), pp. 8546–8560.
90. P.F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, D. Amodei, Deep reinforcement learning from human preferences. Adv. Neural Inf. Process. Syst. **30** (2017). pp. 4302-4310
91. C. Basu, M. Singhal, A.D. Dragan, Learning from richer human guidance: Augmenting comparison-based learning with feature queries. in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction* (Association for Computing Machinery, New York, NY, United States, 2018), pp. 132–140
92. T.R. Sumers, S. Yao, K. Narasimhan, T.L. Griffiths, Cognitive architectures for language agents. Transactions on Machine Learning Research (2024), ISSN 2835-8856.
93. J. Wei, M. Bosma, V.Y. Zhao, K. Guu, A.W. Yu, B. Lester, N. Du, A.M. Dai, Q.V. Le, Finetuned language models are zero-shot learners. in International Conference on Learning Representations (2022) (Virtual Event).
94. Y. Wu, L. Jiang, Y. Yang, Switchable novel object captioner. IEEE Trans. Pattern Anal. Mach. Intell. **45**(1), 1162–1173 (2022)
95. W. Li, L. Zhu, L. Wen, Y. Yang, Decap: Decoding clip latents for zero-shot captioning via text-only training. in International Conference on Learning Representations (2023) (Kigali, Rwanda, 2023).
96. S. Zhao, X. Wang, L. Zhu, Y. Yang, Test-time adaptation with clip reward for zero-shot generalization in vision-language models. in The Twelfth International Conference on Learning Representations (Vienna, Austria, 2024)
97. J. Chen, H. Guo, K. Yi, B. Li, M. Elhoseiny, Visualgpt: Data-efficient image captioning by balancing visual input and linguistic knowledge from pretraining. in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (IEEE, New Orleans, LA, USA, 2022), pp. 18030–18040
98. M. Choraria, N. Sekhar, Y. Wu, X. Zhang, P. Singhal, L.R. Varshney, Language grounded qformer for efficient vision language understanding. arXiv preprint (2023) arXiv:2311.07449
99. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly et al., An image is worth 16x16 words: Transformers for image recognition at scale. in International Conference on Learning Representations (2021) (Vienna, Austria).
100. A. Van Den Oord, O. Vinyals et al., Neural discrete representation learning. Adv. Neural Inf. Process. Syst. **30** (2017). pp. 6309-6318
101. S. Mehta, M. Rastegari, Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. in International Conference on Learning Representations (2022) (Virtual Event).
102. I.O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit et al., Mlp-mixer: An all-mlp architecture for vision. Adv. Neural Inf. Process. Syst. **34**, 24261–24272 (2021)
103. J. Li, D. Li, S. Savarese, S. Hoi, Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. in *International conference on machine learning* (PMLR, Honolulu, Hawaii, USA, 2023), pp. 19730–19742
104. W. Dai, J. Li, D. Li, A.M.H. Tiong, J. Zhao, W. Wang, B. Li, P.N. Fung, S. Hoi, Instructblip: Towards general-purpose vision-language models with instruction tuning. Adv. Neural Inf. Process. Syst. **36** (2024). pp. 49250-49267
105. A. de Wynter, Will gpt-4 run doom? arXiv preprint (2024) arXiv:2403.05468
106. D. Zhu, J. Chen, X. Shen, X. Li, M. Elhoseiny, Minigpt-4: Enhancing vision-language understanding with advanced large language models. in The Twelfth International Conference on Learning Representations (Vienna, Austria, 2024).
107. Y. Su, T. Lan, H. Li, J. Xu, Y. Wang, D. Cai, Pandagpt: One model to instruction-follow them all. in Proceedings of the 1st Workshop on Taming Large Language Models: Controllability in the era of Interactive Assistants! (Association for Computational Linguistics, Prague, Czech Republic, 2023), pp. 11-23.
108. Z. Peng, W. Wang, L. Dong, Y. Hao, S. Huang, S. Ma, F. Wei, Kosmos-2: Grounding multimodal large language models to the world. in The Twelfth International Conference on Learning Representations (Vienna, Austria, 2024).
109. Y. Zhu, Y. Wu, Y. Yang, Y. Yan, Saying the unseen: Video descriptions via dialog agents. IEEE Trans. Pattern Anal. Mach. Intell. **44**(10), 7190–7204 (2021)
110. L. Qian, J. Li, Y. Wu, Y. Ye, H. Fei, T.S. Chua, Y. Zhuang, S. Tang, Momentor: Advancing video large language model with fine-grained temporal reasoning. in Forty-first International Conference on Machine Learning (Vienna, Austria, 2024).
111. Z. Yang, G. Chen, X. Li, W. Wang, Y. Yang, Doraemongpt: Toward understanding dynamic scenes with large language models. in The Twelfth International Conference on Learning Representations (Vienna, Austria, 2024).
112. J.B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds et al., Flamingo: a visual language model for few-shot learning. Adv. Neural Inf. Process. Syst. **35**, 23716–23736 (2022)
113. X. Wang, Y. Zhang, O. Zohar, S. Yeung-Levy, Videoagent: Long-form video understanding with large language model as agent. arXiv preprint (2024) arXiv:2403.10517
114. X. Liu, Z. Zhu, H. Liu, Y. Yuan, M. Cui, Q. Huang, J. Liang, Y. Cao, Q. Kong, M.D. Plumbley et al., Wavjourney: Compositional audio creation with large language models. arXiv preprint (2023) arXiv:2307.14335
115. Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi et al., Audiolm: a language modeling approach to audio generation. IEEE/ACM Trans. Audio, Speech and Lang. Proc. **31**, 2523–2533 (2023)
116. F. Chen, M. Han, H. Zhao, Q. Zhang, J. Shi, S. Xu, B. Xu, X-llm: Bootstrapping advanced large language models by treating multi-modalities as foreign languages. arXiv preprint (2023) arXiv:2305.04160

117. H. Zhang, X. Li, L. Bing, Video-llama: An instruction-tuned audio-visual language model for video understanding. in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (Association for Computational Linguistics, Singapore, 2023), pp. 543-553.
118. Y. Tang, D. Shimada, J. Bi, C. Xu, Avicuna: Audio-visual llm with interleaver and context-boundary alignment for temporal referential dialogue. arXiv preprint (2024) arXiv:2403.16276
119. S. Han, Q. Zhang, Y. Yao, W. Jin, Z. Xu, C. He, Llm multi-agent systems: Challenges and open problems. arXiv preprint (2024) arXiv:2402.03578
120. Y. Gong, Y.A. Chung, J. Glass, Ast: Audio spectrogram transformer. arXiv preprint (2021) arXiv:2104.01778
121. W.N. Hsu, B. Bolte, Y.H.H. Tsai, K. Lakhotia, R. Salakhutdinov, A. Mohamed, Hubert: Self-supervised speech representation learning by masked prediction of hidden units. IEEE/ACM Trans. Audio Speech Lang. Process. **29**, 3451–3460 (2021)
122. K. Li, Z. Yang, L. Chen, Y. Yang, J. Xiao, Catr: Combinatorial-dependence audio-queried transformer for audio-visual video segmentation. in *Proceedings of the 31st ACM International Conference on Multimedia* (Association for Computing Machinery, New York, NY, United States, 2023), pp. 1485–1494
123. R. Huang, M. Li, D. Yang, J. Shi, X. Chang, Z. Ye, Y. Wu, Z. Hong, J. Huang, J. Liu et al., Audiogpt: Understanding and generating speech, music, sound, and talking head. in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI Press, Washington, DC, USA)*. **38** (2024), pp. 23802–23804
124. Y. Shen, K. Song, X. Tan, D. Li, W. Lu, Y. Zhuang, Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. Adv. Neural Inf. Process. Syst. **36** (2024). pp. 38154-38180
125. X. Shen, Z. Yang, X. Wang, J. Ma, C. Zhou, Y. Yang, Global-to-local modeling for video-based 3d human pose and shape estimation. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, Vancouver, BC, Canada, 2023), pp. 8887–8896
126. X. Pan, Z. Yang, J. Ma, C. Zhou, Y. Yang, Transhuman: A transformer-based human representation for generalizable neural human rendering. in Proceedings of the IEEE/CVF International conference on computer vision (IEEE, Paris, France, 2023), pp. 3544–3555
127. Z. Zhang, Z. Yang, Y. Yang, Sifu: Side-view conditioned implicit function for real-world usable clothed human reconstruction. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, Seattle, WA, USA, 2024), pp. 9936–9947
128. L.H. Marshall, H.W. Magoun, *Discoveries in the human brain: neuroscience prehistory, brain structure, and function* (Springer Science & Business Media, Humana Totowa, NJ, 2013)
129. X. Zhu, Y. Chen, H. Tian, C. Tao, W. Su, C. Yang, G. Huang, B. Li, L. Lu, X. Wang et al., Ghost in the minecraft: Generally capable agents for open-world enviroments via large language models with text-based knowledge and memory. arXiv preprint (2023) arXiv:2305.17144
130. A. Zhao, D. Huang, Q. Xu, M. Lin, Y.J. Liu, G. Huang, Expel: Llm agents are experiential learners. in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI Press, Washington, DC, USA)*. **38**(2024), pp. 19632–19642
131. H. Chase. Langchain: Building applications with llms through composability (2022). https://github.com/hwchase17/langchain.
132. N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, S. Yao, Reflexion: Language agents with verbal reinforcement learning. Adv. Neural Inf. Process. Syst. **36**(2024). pp. 8634-8652
133. C. Packer, V. Fang, S.G. Patil, K. Lin, S. Wooders, J.E. Gonzalez, Memgpt: Towards llms as operating systems. arXiv preprint (2023) arXiv:2310.08560
134. W. Zhong, L. Guo, Q. Gao, H. Ye, Y. Wang, Memorybank: Enhancing large language models with long-term memory. in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI Press, Washington, DC, USA)*. **38**(2024), pp. 19724–19731
135. J. Lin, H. Zhao, A. Zhang, Y. Wu, H. Ping, Q. Chen, Agentsims: An open-source sandbox for large language model evaluation. arXiv preprint (2023) arXiv:2308.04026
136. T.G. Karimpanal, L.B. Semage, S. Rana, H. Le, T. Tran, S. Gupta, S. Venkatesh, Lagr-seq: Language-guided reinforcement learning with sample-efficient querying. arXiv preprint (2023) arXiv:2308.13542
137. D. Zhang, L. Chen, S. Zhang, H. Xu, Z. Zhao, K. Yu, Large language models are semi-parametric reinforcement learning agents. Adv. Neural Inf. Process. Syst. **36** (2024). pp. 78227-78239
138. L. Zheng, R. Wang, X. Wang, B. An, Synapse: Trajectory-as-exemplar prompting with memory for computer control. in *The Twelfth International Conference on Learning Representations* (Messe Wien Exhibition and Congress Center, Vienna, Austria, 2023)
139. J. Kang, R. Laroche, X. Yuan, A. Trischler, X. Liu, J. Fu, Think before you act: Decision transformers with internal working memory. in The Twelfth International Conference on Learning Representations (Vienna, Austria, 2024).
140. M. Guo, J. Ainslie, D. Uthus, S. Ontanon, J. Ni, Y.H. Sung, Y. Yang, Longt5: Efficient text-to-text transformer for long sequences. in Findings of the Association for Computational Linguistics: NAACL 2022 (Association for Computational Linguistics, Seattle, United States, 2022), pp. 724–736.
141. J. Ainslie, T. Lei, M. de Jong, S. Ontañón, S. Brahma, Y. Zemlyanskiy, D. Uthus, M. Guo, J. Lee-Thorp, Y. Tay et al., Colt5: Faster long-range transformers with conditional computation. in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (Association for Computational Linguistics, Singapore, 2023), pp. 5085–5100.
142. A. Ruoss, G. Delétang, T. Genewein, J. Grau-Moya, R. Csordás, M. Bennani, S. Legg, J. Veness, Randomized positional encodings boost length generalization of transformers. in Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (Association for Computational Linguistics, Toronto, Canada, 2023), pp. 1889–1903.
143. C. Hu, J. Fu, C. Du, S. Luo, J. Zhao, H. Zhao, Chatdb: Augmenting llms with databases as their symbolic memory. arXiv preprint (2023) arXiv:2306.03901
144. Z. Huang, S. Gutierrez, H. Kamana, S. MacNeil, Memory sandbox: Transparent and interactive memory management for conversational agents. in *Adjunct Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (Association for Computing Machinery, New York, NY, United States, 2023), pp. 1–3
145. A. Modarressi, A. Imani, M. Fayyaz, H. Schütze, Ret-llm: Towards a general read-write memory for large language models. in ICLR 2024 Workshop: How Far Are We From AGI (Vienna, Austria, 2024)
146. D. Schuurmans, Memory augmented large language models are computationally universal. arXiv preprint (2023) arXiv:2301.04589
147. Y. Nie, H. Huang, W. Wei, X.L. Mao, Capturing global structural information in long document question answering with compressive graph selector network. in Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022), pp. 5036–5047.
148. A. Bertsch, U. Alon, G. Neubig, M. Gormley, Unlimiformer: Long-range transformers with unlimited length input. Adv. Neural Inf. Process. Syst. **36**(2024). pp. 35522-35543
149. P. Manakul, M.J. Gales, Sparsity and sentence structure in encoder-decoder attention of summarization systems. in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021), pp. 9359–9368.
150. X. Zhou, G. Li, Z. Liu, Llm as dba. arXiv preprint (2023) arXiv:2308.05481
151. T. Silver, V. Hariprasad, R.S. Shuttleworth, N. Kumar, T. Lozano-Pérez, L.P. Kaelbling, Pddl planning with pretrained large language models. in *NeurIPS 2022 foundation models for decision making workshop* (New Orleans, Louisiana, USA, 2022)
152. Y. Xie, C. Yu, T. Zhu, J. Bai, Z. Gong, H. Soh, Translating natural language to planning goals with large-language models. in The International Journal of Robotics Research (2020). **2019**, pp. 1
153. M. Tsimpoukelli, J.L. Menick, S. Cabi, S. Eslami, O. Vinyals, F. Hill, Multimodal few-shot learning with frozen language models. Adv. Neural Inf. Process. Syst. **34**, 200–212 (2021)

154. H. Liu, C. Li, Q. Wu, Y.J. Lee, Visual instruction tuning. Adv. Neural Inf. Process. Syst. **36**(2024). pp. 34892-34916
155. D. Zhang, S. Li, X. Zhang, J. Zhan, P. Wang, Y. Zhou, X. Qiu, Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities. in Findings of the Association for Computational Linguistics: EMNLP 2023 (Association for Computational Linguistics, Singapore, 2023), pp. 15757–15773.
156. H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar et al., Llama: Open and efficient foundation language models. arXiv preprint (2023) arXiv:2302.13971
157. C. Lyu, M. Wu, L. Wang, X. Huang, B. Liu, Z. Du, S. Shi, Z. Tu, Macaw-llm: Multi-modal language modeling with image, audio, video, and text integration. arXiv preprint (2023) arXiv:2306.09093
158. W. Yu, C. Tang, G. Sun, X. Chen, T. Tan, W. Li, L. Lu, Z. Ma, C. Zhang, Connecting speech encoder and large language model for asr. in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (IEEE, Seoul, Korea, 2024), pp. 12637–12641
159. P.K. Rubenstein, C. Asawaroengchai, D.D. Nguyen, A. Bapna, Z. Borsos, F.D.C. Quitry, P. Chen, D.E. Badawy, W. Han, E. Kharitonov et al., Audiopalm: A large language model that can speak and listen. arXiv preprint (2023) arXiv:2306.12925
160. Y. Ding, X. Zhang, S. Amiri, N. Cao, H. Yang, A. Kaminski, C. Esselink, S. Zhang, Integrating action knowledge and llms for task planning and situation handling in open worlds. Auton. Robot. **47**(8), 981–997 (2023)
161. K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano et al., Training verifiers to solve math word problems. arXiv preprint (2021) arXiv:2110.14168
162. K. Singhal, S. Azizi, T. Tu, S.S. Mahdavi, J. Wei, H.W. Chung, N. Scales, A. Tanwani, H. Cole-Lewis, S. Pfohl et al., Large language models encode clinical knowledge. *Nature* **620**, 172-180(2023). https://doi.org/10.1038/s41586-023-06291-2
163. K. Singhal, T. Tu, J. Gottweis, R. Sayres, E. Wulczyn, L. Hou, K. Clark, S. Pfohl, H. Cole-Lewis, D. Neal et al., Towards expert-level medical question answering with large language models. arXiv preprint (2023) arXiv:2305.09617
164. T. Tu, S. Azizi, D. Driess, M. Schaekermann, M. Amin, P.C. Chang, A. Carroll, C. Lau, R. Tanno, I. Ktena et al., Towards generalist biomedical ai. NEJM AI **1**(3), AIoa2300,138 (2024)
165. F.F. Xu, U. Alon, G. Neubig, V.J. Hellendoorn, A systematic evaluation of large language models of code. in *Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming* (Association for Computing Machinery, New York, NY, United States, 2022), pp. 1–10
166. A. Madaan, S. Zhou, U. Alon, Y. Yang, G. Neubig, Language models of code are few-shot commonsense learners. in Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022), pp. 1384–1403.
167. V. Pallagani, B.C. Muppasani, K. Roy, F. Fabiano, A. Loreggia, K. Murugesan, B. Srivastava, F. Rossi, L. Horesh, A. Sheth, On the prospects of incorporating large language models (llms) in automated planning and scheduling (aps). in *Proceedings of the International Conference on Automated Planning and Scheduling (AAAI Press, Washington, DC, USA)*. **34**(2024), pp. 432–444
168. A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H.W. Chung, C. Sutton, S. Gehrmann et al., Palm: Scaling language modeling with pathways. J. Mach. Learn. Res. **24**(240), 1–113 (2023)
169. R. Sun, S.Ö. Arik, A. Muzio, L. Miculicich, S. Gundabathula, P. Yin, H. Dai, H. Nakhost, R. Sinha, Z. Wang et al., Sql-palm: Improved large language model adaptation for text-to-sql (extended). arXiv preprint (2023) arXiv:2306.00739
170. X. Wang, Q. Yang, Y. Qiu, J. Liang, Q. He, Z. Gu, Y. Xiao, W. Wang, Knowledgpt: Enhancing large language models with retrieval and storage access on knowledge bases. arXiv preprint (2023) arXiv:2308.11761
171. R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, S. Jain, V. Kosaraju, W. Saunders et al., Webgpt: Browser-assisted question-answering with human feedback. arXiv preprint (2021) arXiv:2112.09332
172. L. Gao, A. Madaan, S. Zhou, U. Alon, P. Liu, Y. Yang, J. Callan, G. Neubig, Pal: Program-aided language models. in *International Conference on Machine Learning* (PMLR, Honolulu, Hawaii, USA, 2023), pp. 10764–10799
173. A. Parisi, Y. Zhao, N. Fiedel, Talm: Tool augmented language models. arXiv preprint (2022) arXiv:2205.12255
174. S.G. Patil, T. Zhang, X. Wang, J.E. Gonzalez, Gorilla: Large language model connected with massive apis. arXiv preprint (2023) arXiv:2305.15334
175. P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.T. Yih, T. Rocktäschel et al., Retrieval-augmented generation for knowledge-intensive nlp tasks. Adv. Neural Inf. Process. Syst. **33**, 9459–9474 (2020)
176. L. Zha, J. Zhou, L. Li, R. Wang, Q. Huang, S. Yang, J. Yuan, C. Su, X. Li, A. Su et al., Tablegpt: Towards unifying tables, nature language and commands into one gpt. arXiv preprint (2023) arXiv:2307.08674
177. Z. Luo, C. Xu, P. Zhao, X. Geng, C. Tao, J. Ma, Q. Lin, D. Jiang, Augmented large language models with parametric knowledge guiding. arXiv preprint (2023) arXiv:2305.04757
178. X. He, Y. Tian, Y. Sun, N.V. Chawla, T. Laurent, Y. LeCun, X. Bresson, B. Hooi, G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. arXiv preprint (2024) arXiv:2402.07630
179. X. Cheng, D. Luo, X. Chen, L. Liu, D. Zhao, R. Yan, Lift yourself up: Retrieval-augmented text generation with self-memory. Adv. Neural Inf. Process. Syst. **36** (2024). pp. 43780-43799
180. W. Yu, D. Iter, S. Wang, Y. Xu, M. Ju, S. Sanyal, C. Zhu, M. Zeng, M. Jiang, Generate rather than retrieve: Large language models are strong context generators. in The Eleventh International Conference on Learning Representations (Kigali, Rwanda, 2023)
181. Z. Shao, Y. Gong, Y. Shen, M. Huang, N. Duan, W. Chen, Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. in Findings of the Association for Computational Linguistics: EMNLP 2023 (Association for Computational Linguistics, Singapore, 2023), pp. 9248-9274
182. H. Trivedi, N. Balasubramanian, T. Khot, A. Sabharwal, Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. in Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Association for Computational Linguistics, Toronto, Canada, 2023), pp. 10014-10037
183. Z. Jiang, F.F. Xu, L. Gao, Z. Sun, Q. Liu, J. Dwivedi-Yu, Y. Yang, J. Callan, G. Neubig, Active retrieval augmented generation. in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (Association for Computational Linguistics, Singapore, 2023), pp. 7969-7992
184. A. Asai, Z. Wu, Y. Wang, A. Sil, H. Hajishirzi, Self-rag: Learning to retrieve, generate, and critique through self-reflection. in The Twelfth International Conference on Learning Representations (Vienna, Austria, 2024)
185. Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, H. Wang, Retrieval-augmented generation for large language models: A survey. arXiv preprint (2023) arXiv:2312.10997
186. S. Zhuang, B. Liu, B. Koopman, G. Zuccon, Open-source large language models are strong zero-shot query likelihood models for document ranking. in Findings of the Association for Computational Linguistics: EMNLP 2023 (Association for Computational Linguistics, Singapore, 2023), pp. 8807-8817
187. H. Yang, Z. Li, Y. Zhang, J. Wang, N. Cheng, M. Li, J. Xiao, Prca: Fitting black-box large language models for retrieval question answering via pluggable reward-driven contextual adapter. in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (Association for Computational Linguistics, Singapore, 2023), pp. 5364-5375
188. F. Xu, W. Shi, E. Choi, Recomp: Improving retrieval-augmented lms with compression and selective augmentation. in The Twelfth International Conference on Learning Representations (Vienna, Austria, 2024)
189. X. Du, H. Ji, Retrieval-augmented generative question answering for event argument extraction. in Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022), pp. 4649-4666
190. X. Li, Z. Liu, C. Xiong, S. Yu, Y. Gu, Z. Liu, G. Yu, Structure-aware language model pretraining improves dense retrieval on structured data. in Findings of the Association for Computational Linguistics: ACL 2023 (Association for Computational Linguistics, Toronto, Canada, 2023), pp. 11560-11574

191. J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A.A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska et al., Overcoming catastrophic forgetting in neural networks. Proc. Natl. Acad. Sci. **114**(13), 3521–3526 (2017)
192. B. Peng, M. Galley, P. He, H. Cheng, Y. Xie, Y. Hu, Q. Huang, L. Liden, Z. Yu, W. Chen et al., Check your facts and try again: Improving large language models with external knowledge and automated feedback. arXiv preprint (2023) arXiv:2302.12813
193. Y. Yao, P. Wang, B. Tian, S. Cheng, Z. Li, S. Deng, H. Chen, N. Zhang, Editing large language models: Problems, methods, and opportunities. in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (Association for Computational Linguistics, Singapore, 2023), pp. 10222-10240
194. X. Li, S. Li, S. Song, J. Yang, J. Ma, J. Yu, Pmet: Precise model editing in a transformer. in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI Press, Washington, DC, USA)*, vol. 38 (2024), pp. 18564–18572
195. E. Mitchell, C. Lin, A. Bosselut, C.D. Manning, C. Finn, Memory-based model editing at scale. in *International Conference on Machine Learning* (PMLR, Baltimore, Maryland, USA, 2022), pp. 15817–15831
196. J. Maynez, S. Narayan, B. Bohnet, R. McDonald, On faithfulness and factuality in abstractive summarization. in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (Association for Computational Linguistics, Online, 2022), pp. 1906-1919
197. V. Raunak, A. Menezes, M. Junczys-Dowmunt, The curious case of hallucinations in neural machine translation. in Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Association for Computational Linguistics, Online, 2021), pp. 1172-1183
198. Y. Zhang, Y. Li, L. Cui, D. Cai, L. Liu, T. Fu, X. Huang, E. Zhao, Y. Zhang, Y. Chen et al., Siren's song in the ai ocean: a survey on hallucination in large language models. arXiv preprint (2023) arXiv:2309.01219
199. Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y.J. Bang, A. Madotto, P. Fung, Survey of hallucination in natural language generation. ACM Comput. Surv. **55**(12), 1–38 (2023)
200. Z. Guo, M. Schlichtkrull, A. Vlachos, A survey on automated fact-checking. Trans. Assoc. Comput. Linguist. **10**, 178–206 (2022)
201. J. Thorne, A. Vlachos, Automated fact checking: Task formulations, methods and future directions. in Proceedings of the 27th International Conference on Computational Linguistics (Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018), pp. 3346-3359
202. S. Dhuliawala, M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz, J. Weston, Chain-of-verification reduces hallucination in large language models. in Findings of the Association for Computational Linguistics ACL 2024 (Association for Computational Linguistics, Bangkok, Thailand and virtual meeting, 2024), pp. 3563-3578
203. S. Huang, S. Mamidanna, S. Jangam, Y. Zhou, L.H. Gilpin, Can large language models explain themselves? a study of llm-generated self-explanations. arXiv preprint (2023) arXiv:2310.11207
204. C. Zhou, P. Liu, P. Xu, S. Iyer, J. Sun, Y. Mao, X. Ma, A. Efrat, P. Yu, L. Yu et al., Lima: Less is more for alignment. Adv. Neural Inf. Process. Syst. **36** (2024). pp. 55006-55021
205. L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray et al., Training language models to follow instructions with human feedback. Adv. Neural Inf. Process. Syst. **35**, 27730–27744 (2022)
206. Z. Wu, Y. Hu, W. Shi, N. Dziri, A. Suhr, P. Ammanabrolu, N.A. Smith, M. Ostendorf, H. Hajishirzi, Fine-grained human feedback gives better rewards for language model training. Adv. Neural Inf. Process. Syst. **36** (2024). pp. 59008-59033
207. S. Lin, J. Hilton, O. Evans, Truthfulqa: Measuring how models mimic human falsehoods. in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Association for Computational Linguistics, Dublin, Ireland, 2022), pp. 3214-3252
208. S. Zhang, L. Pan, J. Zhao, W.Y. Wang, Mitigating language model hallucination with interactive question-knowledge alignment. in Proceedings of the AAAI Conference on Artificial Intelligence, **38**(16), 18126-18134. https://doi.org/10.1609/aaai.v38i16.29770
209. Y.S. Chuang, Y. Xie, H. Luo, Y. Kim, J. Glass, P. He, Dola: Decoding by contrasting layers improves factuality in large language models. in The Twelfth International Conference on Learning Representations (Vienna, Austria, 2024)
210. W. Shi, X. Han, M. Lewis, Y. Tsvetkov, L. Zettlemoyer, S.W.t. Yih, Trusting your evidence: Hallucinate less with context-aware decoding. in Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers) (Association for Computational Linguistics, Mexico City, Mexico, 2024), pp. 783-791
211. R. Geirhos, J.H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, F.A. Wichmann, Shortcut learning in deep neural networks. Nat. Mach. Intell. **2**(11), 665–673 (2020)
212. Y. Lu, M. Bartolo, A. Moore, S. Riedel, P. Stenetorp, Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Association for Computational Linguistics, Dublin, Ireland, 2022), pp. 8086-8098
213. M. Du, F. He, N. Zou, D. Tao, X. Hu, Shortcut learning of large language models in natural language understanding. Commun. ACM **67**(1), 110–120 (2023)
214. R. Tang, D. Kong, L. Huang, H. Xue, Large language models can be lazy learners: Analyze shortcuts in in-context learning. in Findings of the Association for Computational Linguistics: ACL 2023 (Association for Computational Linguistics, Toronto, Canada, 2023), pp. 4645-4657
215. Y. Zhou, P. Xu, X. Liu, B. An, W. Ai, F. Huang, Explore spurious correlations at the concept level in language models for text classification. in Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Association for Computational Linguistics, Bangkok, Thailand, 2024), pp. 478-492
216. P.C. Wason, Reasoning about a rule. Q. J. Exp. Psychol. **20**(3), 273–281 (1968)
217. P.C. Wason, P.N. Johnson-Laird, *Psychology of reasoning: Structure and content*. Harvard University Press, Cambridge, MA, USA, 1972. **86**, pp. 246-252
218. K.M. Galotti, Approaches to studying formal and everyday reasoning. Psychol. Bull. **105**(3), 331 (1989)
219. J. Huang, K.C.C. Chang, Towards reasoning in large language models: A survey. in Findings of the Association for Computational Linguistics: ACL 2023 (Association for Computational Linguistics, Toronto, Canada, 2023), pp. 1049-1065
220. D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q. Le et al., Least-to-most prompting enables complex reasoning in large language models. in The Eleventh International Conference on Learning Representations (Kigali, Rwanda, 2023)
221. M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman et al., Do as i can, not as i say: Grounding language in robotic affordances. in Conference on Robot Learning (Atlanta, GA, United States, 2023), pp. 287–318. PMLR
222. B. Xu, Z. Peng, B. Lei, S. Mukherjee, Y. Liu, D. Xu, Rewoo: Decoupling reasoning from observations for efficient augmented language models. arXiv preprint (2023) arXiv:2305.18323
223. S.S. Raman, V. Cohen, E. Rosen, I. Idrees, D. Paulius, S. Tellex, Planning with large language models via corrective re-prompting. in *NeurIPS 2022 Foundation Models for Decision Making Workshop (New Orleans, Louisiana, USA, 2022)*
224. T. Kojima, S.S. Gu, M. Reid, Y. Matsuo, Y. Iwasawa, Large language models are zero-shot reasoners. Adv. Neural Inf. Process. Syst. **35**, 22199–22213 (2022)
225. Q. Lyu, S. Havaldar, A. Stein, L. Zhang, D. Rao, E. Wong, M. Apidianaki, C. Callison-Burch, Faithful chain-of-thought reasoning. in Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers) (Association for Computational Linguistics, Nusa Dua, Bali, 2023), pp. 305-329
226. Y. Wu, S.Y. Min, Y. Bisk, R. Salakhutdinov, A. Azaria, Y. Li, T. Mitchell, S. Prabhumoye, Plan, eliminate, and track–language models are good teachers for embodied agents. arXiv preprint (2023) arXiv:2305.02412
227. B.Y. Lin, Y. Fu, K. Yang, F. Brahman, S. Huang, C. Bhagavatula, P. Ammanabrolu, Y. Choi, X. Ren, Swiftsage: A generative agent with fast and slow thinking for complex interactive tasks. Adv. Neural Inf. Process. Syst. **36** (2024). pp. 23813-23825

228. Z. Zhang, A. Zhang, M. Li, A. Smola, Automatic chain of thought prompting in large language models. in The Eleventh International Conference on Learning Representations (Kigali, Rwanda, 2023)
229. X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, D. Zhou, Self-consistency improves chain of thought reasoning in language models. in The Eleventh International Conference on Learning Representations (Kigali, Rwanda, 2023)
230. S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, K. Narasimhan, Tree of thoughts: Deliberate problem solving with large language models. Adv. Neural Inf. Process. Syst. **36** (2024). pp. 11809-11822
231. B. Sel, A. Al-Tawaha, V. Khattar, L. Wang, R. Jia, M. Jin, Algorithm of thoughts: Enhancing exploration of ideas in large language models. in Forty-first International Conference on Machine Learning (Vienna, Austria, 2024)
232. Y. Wang, Z. Jiang, Z. Chen, F. Yang, Y. Zhou, E. Cho, X. Fan, X. Huang, Y. Lu, Y. Yang, Recmind: Large language model powered agent for recommendation. in Findings of the Association for Computational Linguistics: NAACL 2024 (Association for Computational Linguistics, Mexico City, Mexico, 2024), pp. 4351-4364
233. M. Besta, N. Blach, A. Kubicek, R. Gerstenberger, M. Podstawski, L. Gianinazzi, J. Gajda, T. Lehmann, H. Niewiadomski, P. Nyczyk et al., Graph of thoughts: Solving elaborate problems with large language models. in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI Press, Washington, DC, USA)*. **38**(2024), pp. 17682–17690
234. W. Huang, P. Abbeel, D. Pathak, I. Mordatch, Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. in *International Conference on Machine Learning* (PMLR, Baltimore, Maryland, USA, 2022), pp. 9118–9147
235. S. Hao, Y. Gu, H. Ma, J.J. Hong, Z. Wang, D.Z. Wang, Z. Hu, Reasoning with language model is planning with world model. in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (Association for Computational Linguistics, Singapore, 2023), pp. 8154-8173
236. K. Nottingham, P. Ammanabrolu, A. Suhr, Y. Choi, H. Hajishirzi, S. Singh, R. Fox, Do embodied agents dream of pixelated sheep: Embodied decision making using language guided world modelling. in *International Conference on Machine Learning* (PMLR, Honolulu, Hawaii, USA, 2023), pp. 26311–26325
237. J. Ruan, Y. Chen, B. Zhang, Z. Xu, T. Bao, G. Du, S. Shi, H. Mao, X. Zeng, R. Zhao, Tptu: Task planning and tool usage of large language model-based ai agents. in NeurIPS 2023 Foundation Models for Decision Making Workshop (New Orleans, Louisiana, United States of America, 2023)
238. E. Karpas, O. Abend, Y. Belinkov, B. Lenz, O. Lieber, N. Ratner, Y. Shoham, H. Bata, Y. Levine, K. Leyton-Brown et al., Mrkl systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning. arXiv preprint (2022) arXiv:2205.00445
239. B. Liu, Y. Jiang, X. Zhang, Q. Liu, S. Zhang, J. Biswas, P. Stone, Llm+ p: Empowering large language models with optimal planning proficiency. arXiv preprint (2023) arXiv:2304.11477
240. T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell et al., Language models are few-shot learners. Adv. Neural Inf. Process. Syst. **33**, 1877–1901 (2020)
241. Z. Zhao, E. Wallace, S. Feng, D. Klein, S. Singh, Calibrate before use: Improving few-shot performance of language models. in *International conference on machine learning* (PMLR, Virtual Event, 2021), pp. 12697–12706
242. A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever et al., Language models are unsupervised multitask learners. OpenAI Blog **1**(8), 9 (2019)
243. Z. Wang, S. Cai, G. Chen, A. Liu, X. Ma, Y. Liang, Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. arXiv preprint (2023) arXiv: 2302.01560
244. G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, A. Anandkumar, Voyager: An open-ended embodied agent with large language models. arXiv preprint (2023) arXiv: 2305.16291
245. Y. Liang, C. Wu, T. Song, W. Wu, Y. Xia, Y. Liu, Y. Ou, S. Lu, L. Ji, S. Mao et al., Taskmatrix. ai: Completing tasks by connecting foundation models with millions of apis. arXiv preprint (2023) arXiv: 2303.16434
246. Y. Song, W. Xiong, D. Zhu, C. Li, K. Wang, Y. Tian, S. Li, Restgpt: Connecting large language models with real-world applications via restful apis. arXiv preprint (2023) arXiv: 2306.06624
247. T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, E. Hambro, L. Zettlemoyer, N. Cancedda, T. Scialom, Toolformer: Language models can teach themselves to use tools. Adv. Neural Inf. Process. Syst. **36** (2024). pp. 68539-68551
248. R. Gong, Q. Huang, X. Ma, H. Vo, Z. Durante, Y. Noda, Z. Zheng, S.C. Zhu, D. Terzopoulos, L. Fei-Fei et al., Mindagent: Emergent gaming interaction. arXiv preprint (2023) arXiv: 2309.09971
249. M. Carroll, R. Shah, M.K. Ho, T. Griffiths, S. Seshia, P. Abbeel, A. Dragan, On the utility of learning about humans for human-ai coordination. Adv. Neural Inf. Process. Syst. **32**(2019), pp. 5174-5185
250. H. Hu, D. Yarats, Q. Gong, Y. Tian, M. Lewis, Hierarchical decision making by generating and following natural language instructions. Adv. Neural Inf. Process. Syst. **32**(2019), pp. 10025-10034
251. B. Hu, C. Zhao, P. Zhang, Z. Zhou, Y. Yang, Z. Xu, B. Liu, Enabling intelligent interactions between an agent and an llm: A reinforcement learning approach. arXiv preprint (2023) arXiv: 2306.03604
252. Z. Liu, Y. Zhang, P. Li, Y. Liu, D. Yang, Dynamic llm-agent network: An llm-agent collaboration framework with agent team optimization. arXiv preprint (2023) arXiv: 2310.02170
253. Y. Chen, J. Arkin, Y. Zhang, N. Roy, C. Fan, Scalable multi-robot collaboration with large language models: Centralized or decentralized systems? arXiv preprint (2023) arXiv: 2309.15943
254. Z. Hu, Z. Zhang, H. Li, C. Chen, H. Ding, Z. Wang, Attention-guided contrastive role representations for multi-agent reinforcement learning. arXiv preprint (2023) arXiv: 2312.04819
255. G. Li, H. Hammoud, H. Itani, D. Khizbullin, B. Ghanem, Camel: Communicative agents for "mind" exploration of large language model society. Adv. Neural Inf. Process. Syst. **36**(2024), pp. 51991-52008
256. F. Guo, Gpt agents in game theory experiments. Technical report (2023)
257. Q. Zhao, J. Wang, Y. Zhang, Y. Jin, K. Zhu, H. Chen, X. Xie, Competeai: Understanding the competition behaviors in large language model-based agents. arXiv preprint (2023) arXiv: 2310.17512
258. R. Hao, L. Hu, W. Qi, Q. Wu, Y. Zhang, L. Nie, Chatllm network: More brains, more intelligence. arXiv preprint (2023) arXiv: 2304.12998
259. X. Ning, Z. Lin, Z. Zhou, Z. Wang, H. Yang, Y. Wang, Skeleton-of-thought: Prompting llms for efficient parallel generation. in *The Twelfth International Conference on Learning Representations* (Vienna, Austria, 2024)
260. Q. Wu, G. Bansal, J. Zhang, Y. Wu, S. Zhang, E. Zhu, B. Li, L. Jiang, X. Zhang, C. Wang, Autogen: Enabling next-gen llm applications via multi-agent conversation framework. arXiv preprint (2023) arXiv: 2308.08155
261. S. Srivastava, C. Li, M. Lingelbach, R. Martín-Martín, F. Xia, K.E. Vainio, Z. Lian, C. Gokmen, S. Buch, K. Liu et al., Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments. in *Conference on robot learning* (PMLR, London, UK, 2021), pp. 477–490
262. X. Guo, K. Huang, J. Liu, W. Fan, N. Vélez, Q. Wu, H. Wang, T.L. Griffiths, M. Wang, Embodied llm agents learn to cooperate in organized teams. arXiv preprint (2024) arXiv: 2403.12482
263. Z. Wang, S. Mao, W. Wu, T. Ge, F. Wei, H. Ji, Unleashing cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. arXiv preprint (2023) arXiv: 2307.05300
264. T. Liang, Z. He, W. Jiao, X. Wang, Y. Wang, R. Wang, Y. Yang, Z. Tu, S. Shi, Encouraging divergent thinking in large language models through multi-agent debate. arXiv preprint (2023) arXiv: 2305.19118
265. Z. Liu, W. Yao, J. Zhang, L. Yang, Z. Liu, J. Tan, P.K. Choubey, T. Lan, J. Wu, H. Wang et al., Agentlite: A lightweight library for building and advancing task-oriented llm agent system. arXiv preprint (2024) arXiv: 2402.15538
266. W. Yao, S. Heinecke, J.C. Niebles, Z. Liu, Y. Feng, L. Xue, R. Murthy, Z. Chen, J. Zhang, D. Arpit et al., Retroformer: Retrospective large language agents with policy gradient optimization. arXiv preprint (2023) arXiv: 2308.02151
267. Y. Shu, H. Gu, P. Zhang, H. Zhang, T. Lu, D. Li, N. Gu, Rah! recsys-assistant-human: A human-central recommendation framework with large language models. arXiv preprint (2023) arXiv: 2308.09904
268. C. Colas, L. Teodorescu, P.Y. Oudeyer, X. Yuan, M.A. Côté, Augmenting autotelic agents with large language models. in *Conference on Lifelong Learning Agents* (PMLR, McGill University, Montréal, Québec, Canada, 2023), pp. 205–226

269. Y. Wu, Z. Jiang, A. Khan, Y. Fu, L. Ruis, E. Grefenstette, T. Rocktäschel, Chatarena: Multi-agent language game environments for large language models. https://github.com/chatarena/chatarena.
270. C. Fan, J. Chen, Y. Jin, H. He, Can large language models serve as rational players in game theory? a systematic analysis. in *Proceedings of the AAAI Conference on Artificial Intelligence* (AAAI, Washington, DC, USA, 2024), pp. 17960–17967
271. Z.J. Wang, D. Choi, S. Xu, D. Yang, Putting humans in the natural language processing loop: A survey. arXiv preprint (2021) arXiv: 2103.04044
272. K.A. Fischer, Reflective linguistic programming (rlp): A stepping stone in socially-aware agi (socialagi). arXiv preprint (2023) arXiv: 2305.12647
273. B. Chen, C. Shu, E. Shareghi, N. Collier, K. Narasimhan, S. Yao, Fireact: Toward language agent fine-tuning. arXiv preprint (2023) arXiv: 2310.05915
274. A. Brock, T. Lim, J.M. Ritchie, N. Weston, Freezeout: Accelerate training by progressively freezing layers. arXiv preprint (2017) arXiv: 1706.04983
275. Y. Liu, S. Agarwal, S. Venkataraman, Autofreeze: Automatically freezing model blocks to accelerate fine-tuning. arXiv preprint (2021) arXiv: 2102.01386
276. L. Zhu, L. Hu, J. Lin, S. Han, Lift: Efficient layer-wise fine-tuning for large model models. in *Proceedings of the 31st ACM International Conference on -Multimedia* (Association for Computing Machinery, New York, NY, United States, 2023), pp. 4678-4687
277. E.B. Zaken, S. Ravfogel, Y. Goldberg, Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. arXiv preprint (2021) arXiv: 2106.10199
278. W. Song, Z. Li, L. Zhang, H. Zhao, B. Du, Sparse is enough in fine-tuning pre-trained large language model. arXiv preprint (2023) arXiv: 2312.11875
279. N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, S. Gelly, Parameter-efficient transfer learning for nlp. in *International conference on machine learning* (PMLR, Long Beach, California, USA, 2019), pp. 2790–2799
280. J. Pfeiffer, I. Vulić, I. Gurevych, S. Ruder, Mad-x: An adapter-based framework for multi-task cross-lingual transfer. arXiv preprint (2020) arXiv: 2005.00052
281. J. He, C. Zhou, X. Ma, T. Berg-Kirkpatrick, G. Neubig, Towards a unified view of parameter-efficient transfer learning. arXiv preprint (2021) arXiv: 2110.04366
282. Z. Hu, L. Wang, Y. Lan, W. Xu, E.P. Lim, L. Bing, X. Xu, S. Poria, R.K.W. Lee, Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. arXiv preprint (2023) arXiv: 2304.01933
283. E.J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models. arXiv preprint (2021) arXiv: 2106.09685
284. T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, Qlora: Efficient finetuning of quantized llms. Adv. Neural Inf. Process. Syst. **36** (2024). pp. 10088-10115
285. X.L. Li, P. Liang, Prefix-tuning: Optimizing continuous prompts for generation. arXiv preprint (2021) arXiv: 2101.00190
286. Z.R. Zhang, C. Tan, H. Xu, C. Wang, J. Huang, S. Huang, Towards adaptive prefix tuning for parameter-efficient language model fine-tuning. arXiv preprint (2023) arXiv: 2305.15212
287. B. Lester, R. Al-Rfou, N. Constant, The power of scale for parameter-efficient prompt tuning. arXiv preprint (2021) arXiv: 2104.08691
288. X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, J. Tang, Gpt understands, too. arXiv preprint (2021) arXiv: 2103.10385
289. F. Petroni, T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A.H. Miller, S. Riedel, Language models as knowledge bases? arXiv preprint (2019) arXiv: 1909.01066
290. A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, S. Bowman, Superglue: A stickier benchmark for general-purpose language understanding systems. Adv. Neural Inf. Process. Syst. **32** (2019). pp. 3266-3280
291. Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon et al., Constitutional ai: Harmlessness from ai feedback. arXiv preprint (2022) arXiv: 2212.08073
292. A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang et al., Self-refine: Iterative refinement with self-feedback. Adv. Neural Inf. Process. Syst. **36** (2024). pp. 46534-46594
293. N. Shinn, B. Labash, A. Gopinath, Reflexion: an autonomous agent with dynamic memory and self-reflection. arXiv preprint (2023) arXiv: 2303.11366
294. R. Rafailov, A. Sharma, E. Mitchell, C.D. Manning, S. Ermon, C. Finn, Direct preference optimization: Your language model is secretly a reward model. Adv. Neural Inf. Process. Syst. **36** (2024). pp. 53728-53741
295. T. Shin, Y. Razeghi, R.L. Logan IV, E. Wallace, S. Singh, Autoprompt: Eliciting knowledge from language models with automatically generated prompts. arXiv preprint (2020) arXiv: 2010.15980
296. E. Brooks, L. Walls, R.L. Lewis, S. Singh, Large language models can implement policy iteration. Adv. Neural Inf. Process. Syst. **36** (2024). pp. 30349-30366
297. T. Carta, C. Romac, T. Wolf, S. Lamprier, O. Sigaud, P.Y. Oudeyer, Grounding large language models in interactive environments with online reinforcement learning. in *International Conference on Machine Learing* (PMLR, Honolulu, Hawaii, USA, 2023), pp. 3676–3713
298. Z. Yang, J. Liu, Y. Han, X. Chen, Z. Huang, B. Fu, G. Yu, Appagent: Multimodal agents as smartphone users. arXiv preprint (2023) arXiv: 2312.13771
299. A. Madaan, N. Tandon, P. Clark, Y. Yang, Memory-assisted prompt editing to improve gpt-3 after deployment. arXiv preprint (2022) arXiv: 2201.06009
300. Y. Zhang, S. Yang, C. Bai, F. Wu, X. Li, X. Li, Z. Wang, Towards efficient llm grounding for embodied multi-agent collaboration. arXiv preprint (2024) arXiv: 2405.14314
301. G. Chen, S. Dong, Y. Shu, G. Zhang, J. Sesay, B.F. Karlsson, J. Fu, Y. Shi, Autoagents: A framework for automatic agent generation. arXiv preprint (2023) arXiv: 2309.17288
302. W. Chen, Y. Su, J. Zuo, C. Yang, C. Yuan, C. Qian, C.M. Chan, Y. Qin, Y. Lu, R. Xie et al., Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. arXiv preprint (2023) arXiv: 2308.10848
303. H. Wu, Z. He, X. Zhang, X. Yao, S. Zheng, H. Zheng, B. Yu, Chateda: A large language model powered autonomous agent for eda. in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 43 (IEEE, Snowbird, UT, USA, 2024), pp. 3184-3197
304. S. Kang, J. Yoon, S. Yoo, Large language models are few-shot testers: Exploring llm-based general bug reproduction. in *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)* (IEEE, 2023), pp. 2312–2323
305. G. Deng, Y. Liu, V. Mayoral-Vilches, P. Liu, Y. Li, Y. Xu, T. Zhang, Y. Liu, M. Pinzger, S. Rass, Pentestgpt: An llm-empowered automatic penetration testing tool. arXiv preprint (2023) arXiv: 2308.06782
306. S. Sudhakaran, M. González-Duque, M. Freiberger, C. Glanois, E. Najarro, S. Risi, Mariogpt: Open-ended text2level generation through large language models. Adv. Neural Inf. Process. Syst. **36** (2024). pp. 54213-54227
307. J.J. Horton, Large language models as simulated economic agents: What can we learn from homo silicus? Technical report, National Bureau of Economic Research (2023)
308. E. Akata, L. Schulz, J. Coda-Forno, S.J. Oh, M. Bethge, E. Schulz, Playing repeated games with large language models. arXiv preprint (2023) arXiv: 2305.16867
309. J. Zhang, R. Xie, Y. Hou, W.X. Zhao, L. Lin, J.R. Wen, Recommendation as instruction following: A large language model empowered recommendation approach. arXiv preprint (2023) arXiv: 2305.07001
310. K. Bao, J. Zhang, Y. Zhang, W. Wang, F. Feng, X. He, Tallrec: An effective and efficient tuning framework to align large language model with recommendation. in *Proceedings of the 17th ACM Conference on Recommender Systems* (Association for Computing Machinery, New York, NY, United States, 2023), pp. 1007–1014
311. Y. Hou, J. Zhang, Z. Lin, H. Lu, R. Xie, J. McAuley, W.X. Zhao, Large language models are zero-shot rankers for recommender systems. in *European Conference on Information Retrieval* (Springer-Verlag, Berlin, Heidelberg, 2024), pp. 364–381
312. J. Liu, C. Liu, P. Zhou, R. Lv, K. Zhou, Y. Zhang, Is chatgpt a good recommender? a preliminary study. arXiv preprint (2023) arXiv: 2304.10149
313. Y. Gao, T. Sheng, Y. Xiang, Y. Xiong, H. Wang, J. Zhang, Chat-rec: Towards interactive and explainable llms-augmented recommender system. arXiv preprint (2023) arXiv: 2303.14524

314. S. Dai, N. Shao, H. Zhao, W. Yu, Z. Si, C. Xu, Z. Sun, X. Zhang, J. Xu, Uncovering chatgpt's capabilities in recommender systems. in *Proceedings of the 17th ACM Conference on Recommender Systems* (Association for Computing Machinery, New York, NY, United States, 2023), pp. 1126–1132
315. Y. Xi, W. Liu, J. Lin, J. Zhu, B. Chen, R. Tang, W. Zhang, R. Zhang, Y. Yu, Towards open-world recommendation with knowledge augmentation from large language models. arXiv preprint (2023) arXiv: 2306.10933
316. Q. Liu, N. Chen, T. Sakai, X.M. Wu, A first look at llm-powered generative news recommendation. arXiv preprint (2023) arXiv: 2305.06566
317. W. Wei, X. Ren, J. Tang, Q. Wang, L. Su, S. Cheng, J. Wang, D. Yin, C. Huang, Llmrec: Large language models with graph augmentation for recommendation. in *Proceedings of the 17th ACM International Conference on Web Search and Data Mining* (Association for Computing Machinery, New York, NY, United States, 2024), pp. 806–815
318. E. Ie, C.w. Hsu, M. Mladenov, V. Jain, S. Narvekar, J. Wang, R. Wu, C. Boutilier, Recsim: A configurable simulation platform for recommender systems. arXiv preprint (2019) arXiv: 1909.04847
319. N. Ghaffarzadegan, A. Majumdar, R. Williams, N. Hosseinichimeh, Generative agent-based modeling: Unveiling social system dynamics through coupling mechanistic models with generative artificial intelligence. arXiv preprint (2023) arXiv: 2309.11456
320. D. Zhou, Y. Li, F. Ma, X. Zhang, Y. Yang, Migc: Multi-instance generation controller for text-to-image synthesis. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, Seattle, WA, USA, 2024), pp. 6818–6828
321. Y. Cheng, L. Li, Y. Xu, X. Li, Z. Yang, W. Wang, Y. Yang, Segment and track anything. arXiv preprint (2023) arXiv: 2305.06558
322. C. Liang, F. Ma, L. Zhu, Y. Deng, Y. Yang, Caphuman: Capture your moments in parallel universes. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, Seattle, WA, USA, 2024), pp. 6400–6409
323. X. Dong, S.I. Yu, X. Weng, S.E. Wei, Y. Yang, Y. Sheikh, Supervision-by-registration: An unsupervised approach to improve the precision of facial landmark detectors. in *Proceedings of the IEEE conference on computer vision and pattern recognition* (IEEE, Salt Lake City, UT, USA, 2018), pp. 360–368
324. A. Salih, Z. Raisi-Estabragh, I.B. Galazzo, P. Radeva, S.E. Petersen, G. Menegaz, K. Lekadir, Commentary on explainable artificial intelligence methods: Shap and lime. arXiv preprint (2023) arXiv: 2305.02012
325. L. Gao, Z. Dai, P. Pasupat, A. Chen, A.T. Chaganty, Y. Fan, V.Y. Zhao, N. Lao, H. Lee, D.C. Juan et al., Rarr: Researching and revising what language models say, using language models. arXiv preprint (2022) arXiv: 2210.08726
326. F. Cardoso Durier da Silva, R. Vieira, A.C. Garcia, Can machines learn to detect fake news? a survey focused on social media, in *2021 IEEE 4th International Conference on Computing, Power and Communication Technologies* (Kuala Lumpur, Malaysia, 2021), pp.1-7
327. I.O. Gallegos, R.A. Rossi, J. Barrow, M.M. Tanjim, S. Kim, F. Dernoncourt, T. Yu, R. Zhang, N.K. Ahmed, Bias and fairness in large language models: A survey. Computational Linguistics 2024. **50**(3), 1097–1179.
328. H. Kotek, R. Dockum, D. Sun, Gender bias and stereotypes in large language models. in *Proceedings of The ACM Collective Intelligence Conference* (Association for Computing Machinery, New York, NY, United States, 2023), pp. 12–24
329. Y. Wan, G. Pu, J. Sun, A. Garimella, K.W. Chang, N. Peng, "kelly is a warm person, joseph is a role model": Gender biases in llm-generated reference letters. arXiv preprint (2023) arXiv: 2310.09219
330. Y. Li, M. Du, R. Song, X. Wang, Y. Wang, A survey on fairness in large language models. arXiv preprint (2023) arXiv: 2308.10149
331. K. Lu, P. Mardziel, F. Wu, P. Amancharla, A. Datta, Gender bias in neural natural language processing. in *Logic, language, and security* (Springer, Cham, 2020), pp. 189–202
332. R. Qian, C. Ross, J. Fernandes, E. Smith, D. Kiela, A. Williams, Perturbation augmentation for fairer nlp. arXiv preprint (2022) arXiv: 2205.12586
333. A. Zayed, P. Parthasarathi, G. Mordido, H. Palangi, S. Shabanian, S. Chandar, Deep learning on a healthy data diet: Finding important examples for fairness. in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37 (AAAI Press, Washington, DC, USA, 2023), pp. 14593–14601
334. H. Liu, J. Dacon, W. Fan, H. Liu, Z. Liu, J. Tang, Does gender matter? towards fairness in dialogue systems. arXiv preprint (2019) arXiv: 1910.10486
335. D. Saunders, R. Sallis, B. Byrne, First the worst: Finding better gender translations during beam search. arXiv preprint (2021) arXiv: 2104.07429
336. H. Dhingra, P. Jayashanker, S. Moghe, E. Strubell, Queer people are people first: Deconstructing sexual identity stereotypes in large language models. arXiv preprint (2023) arXiv: 2307.00101
337. E.K. Tokpo, T. Calders, Text style transfer for bias mitigation using masked language modeling. arXiv preprint (2022) arXiv: 2201.08643
338. Z. He, B.P. Majumder, J. McAuley, Detect and perturb: Neutral rewriting of biased and sensitive text via gradient-based decoding. arXiv preprint (2021) arXiv: 2109.11708
339. J. Devlin, M.W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint (2018) arXiv: 1810.04805
340. B. Buchanan, A. Lohn, M. Musser, K. Sedova, Truth, lies, and automation. Cent. Secur. Emerg. Technol. **1**(1), 2 (2021)
341. Y. Pan, L. Pan, W. Chen, P. Nakov, M.Y. Kan, W.Y. Wang, On the risk of misinformation pollution with large language models. arXiv preprint (2023) arXiv: 2305.13661
342. J. Yang, H. Xu, S. Mirzoyan, T. Chen, Z. Liu, W. Ju, L. Liu, M. Zhang, S. Wang, Poisoning scientific knowledge using large language models. bioRxiv (2023). https://doi.org/10.1101/2023.11.06.565928
343. P. Charan, H. Chunduri, P.M. Anand, S.K. Shukla, From text to mitre techniques: Exploring the malicious use of large language models for generating cyber attack payloads. arXiv preprint (2023) arXiv: 2305.15336
344. F. Heiding, B. Schneier, A. Vishwanath, J. Bernstein, Devising and detecting phishing: Large language models vs. smaller human models. arXiv preprint (2023) arXiv: 2308.12287
345. A. Happe, J. Cito, Getting pwn'd by ai: Penetration testing with large language models. in *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (Association for Computing Machinery, New York, NY, United States, 2023), pp. 2082–2086
346. P.V. Falade, Decoding the threat landscape: Chatgpt, fraudgpt, and wormgpt in social engineering attacks. arXiv preprint (2023) arXiv: 2310.05595
347. N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson et al., Extracting training data from large language models. in *30th USENIX Security Symposium (USENIX Security 21)* (USENIX, Berkeley, CA, 2021), pp. 2633–2650
348. G.M. Currie, Academic integrity and artificial intelligence: is chatgpt hype, hero or heresy? in *Seminars in Nuclear Medicine* (Elsevier, New York, NY, United States, 2023), pp. 719-730
349. L. Li, D. Song, X. Qiu, Text adversarial purification as defense against adversarial attacks. arXiv preprint (2022) arXiv: 2203.14207
350. A. Robey, E. Wong, H. Hassani, G.J. Pappas, Smoothllm: Defending large language models against jailbreaking attacks. arXiv preprint (2023) arXiv: 2310.03684
351. Z. Xi, T. Du, C. Li, R. Pang, S. Ji, J. Chen, F. Ma, T. Wang, Defending pre-trained language models as few-shot learners against backdoor attacks. Adv. Neural Inf. Process. Syst. **36**(2024), pp. 32748-32764
352. S. Goyal, S. Doddapaneni, M.M. Khapra, B. Ravindran, A survey of adversarial defenses and robustness in nlp. ACM Comput. Surv. **55**(14s), 1–39 (2023)
353. A. Kumar, C. Agarwal, S. Srinivas, S. Feizi, H. Lakkaraju, Certifying llm safety against adversarial prompting. arXiv preprint (2023) arXiv: 2309.02705
354. K. Siau, W. Wang, Artificial intelligence (ai) ethics: ethics of ai and ethical AI. J. Database Manag. (JDM) **31**(2), 74–87 (2020)
355. E. Prem, From ethical ai frameworks to tools: a review of approaches. AI Ethics **3**(3), 699–716 (2023)
356. Y. Talebirad, A. Nadiri, Multi-agent collaboration: Harnessing the power of intelligent llm agents. arXiv preprint (2023) arXiv: 2306.03314
357. M. Yue, J. Zhao, M. Zhang, L. Du, Z. Yao, Large language model cascades with mixture of thoughts representations for cost-efficient reasoning. arXiv preprint (2023) arXiv: 2310.03094
358. Z. Yang, L. Li, J. Wang, K. Lin, E. Azarnasab, F. Ahmed, Z. Liu, C. Liu, M. Zeng, L. Wang, Mm-react: Prompting chatgpt for multimodal reasoning and action. arXiv preprint (2023) arXiv: 2303.11381

359. R. Lou, K. Zhang, W. Yin, Is prompt all you need? no. a comprehensive and broader view of instruction learning. arXiv preprint (2023) arXiv: 2303.10475
360. A. Zeng, M. Liu, R. Lu, B. Wang, X. Liu, Y. Dong, J. Tang, Agenttuning: Enabling generalized agent abilities for llms. arXiv preprint (2023) arXiv: 2310.12823
361. A. Glaese, N. McAleese, M. Teębacz, J. Aslanides, V. Firoiu, T. Ewalds, M. Rauh, L. Weidinger, M. Chadwick, P. Thacker et al., Improving alignment of dialogue agents via targeted human judgements. arXiv preprint (2022) arXiv: 2209.14375
362. K. Zhang, Z. Yang, T. Başar, Multi-agent reinforcement learning: A selective overview of theories and algorithms. Handb. in Studies in Systems, Decision and Control. Springer, Cham. **325**, pp. 321-384(2021)
363. E. Ostrom, Tragedy of the commons. New Palgrave Dictionary Econ. **2**, 1–4 (2008)
364. E.I. Pas, S.L. Principio, Braess' paradox: Some new insights. Transp. Res. B Methodol. **31**(3), 265–276 (1997)
365. T. Sun, Y. Shao, H. Qian, X. Huang, X. Qiu, Black-box tuning for language-model-as-a-service. in *International Conference on Machine Learning* (PMLR, Baltimore, Maryland, USA, 2022), pp. 20841–20855
366. L. Yu, Q. Chen, J. Lin, L. He, Black-box prompt tuning for vision-language model as a service. in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence* (IJCAI, Montreal, Canada, 2023), pp. 1686–1694

Publisher's Note