A Survey on Large Language Model based Autonomous Agents

基于大型语言模型的自主智能体综述

Lei Wang, Chen Ma, *Xueyang Feng*, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhi-Yuan Chen, Jiakai Tang, Xu Chen(&), Yankai Lin(&), Wayne Xin Zhao, Zhewei Wei, Ji-Rong Wen

王磊，马晨，冯雪阳，张泽宇，杨浩，张景森，陈志远，唐嘉凯，陈旭(&)，林彦凯(&)，赵新，魏哲伟，温继荣

Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, 100872, China
中国人民大学高岭人工智能学院，北京，100872，中国

Abstract Autonomous agents have long been a research focus in academic and industry communities. Previous research often focuses on training agents with limited knowledge within isolated environments, which diverges significantly from human learning processes, and makes the agents hard to achieve human-like decisions. Recently, through the acquisition of vast amounts of web knowledge, large language models (LLMs) have shown potential in human-level intelligence, leading to a surge in research on LLM-based autonomous agents. In this paper, we present a comprehensive survey of these studies, delivering a systematic review of LLM-based autonomous agents from a holistic perspective. We first discuss the construction of LLM-based autonomous agents, proposing a unified framework that encompasses much of previous work. Then, we present an overview of the diverse applications of LLM-based autonomous agents in social science, natural science, and engineering. Finally, we delve into the evaluation strategies commonly used for LLM-based autonomous agents. Based on the previous studies, we also present several challenges and future directions in this field.

摘要 自主智能体长期以来一直是学术界和工业界的研究重点。以往研究多聚焦于在孤立环境中训练知识有限的智能体，这与人类的学习过程存在显著差异，导致智能体难以实现类人决策。近年来，借助海量网络知识的获取，大型语言模型（LLMs）展现出类人智能的潜力，推动了基于LLM的自主智能体研究的快速发展。本文对相关研究进行了全面综述，从整体视角系统回顾了基于LLM的自主智能体。首先，我们讨论了基于LLM的自主智能体的构建，提出了涵盖大部分先前工作的统一框架。随后，概述了基于LLM的自主智能体在社会科学、自然科学和工程领域的多样化应用。最后，深入探讨了常用的评估策略。基于已有研究，我们还提出了该领域面临的若干挑战与未来发展方向。

Keywords Autonomous agent, Large language model, Human-level intelligence
关键词 自主智能体，大型语言模型，类人智能

1　1 Introduction

2　1 引言

"An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future."
“自主智能体是一个位于环境中并作为环境一部分的系统，它感知该环境并对其进行作用，随着时间推移，追求自身目标，从而影响其未来所感知的环境。”

Franklin and Graesser (1997)
Franklin 和 Graesser (1997)

Autonomous agents have long been recognized as a promising approach to achieving artificial general intelligence (AGI), which is expected to accomplish tasks through self-directed planning and actions. In previous studies, the agents are assumed to act based on simple and heuristic policy functions, and learned in isolated and restricted environments [1-6]. Such assumptions significantly differs from the human learning process, since the human mind is highly complex and individuals can learn from a much wider variety of environments. Because of these gaps, the agents obtained from previous studies are usually far from replicating human-level decision processes, especially in unconstrained, open-domain settings.
自主智能体长期以来被视为实现人工通用智能（AGI，Artificial General Intelligence）的有前景途径，期望通过自主规划和行动完成任务。以往研究中，智能体通常假设基于简单的启发式策略函数行动，并在孤立且受限的环境中学习[1-6]。这种假设与人类的学习过程存在显著差异，因为人类思维极为复杂，且个体能够从更广泛的环境中学习。由于这些差距，先前研究中获得的智能体通常难以复制类人的决策过程，尤其是在无约束的开放域环境中。

Received month dd, yyyy; accepted month dd, yyyy
收稿日期 月 日，年；接受日期 月 日，年

E-mail: xu.chen@ruc.edu.cn;yankailin@ruc.edu.cn *Both authors contribute equally to this paper.
电子邮件：xu.chen@ruc.edu.cn;yankailin@ruc.edu.cn *两位作者对本文贡献相同。

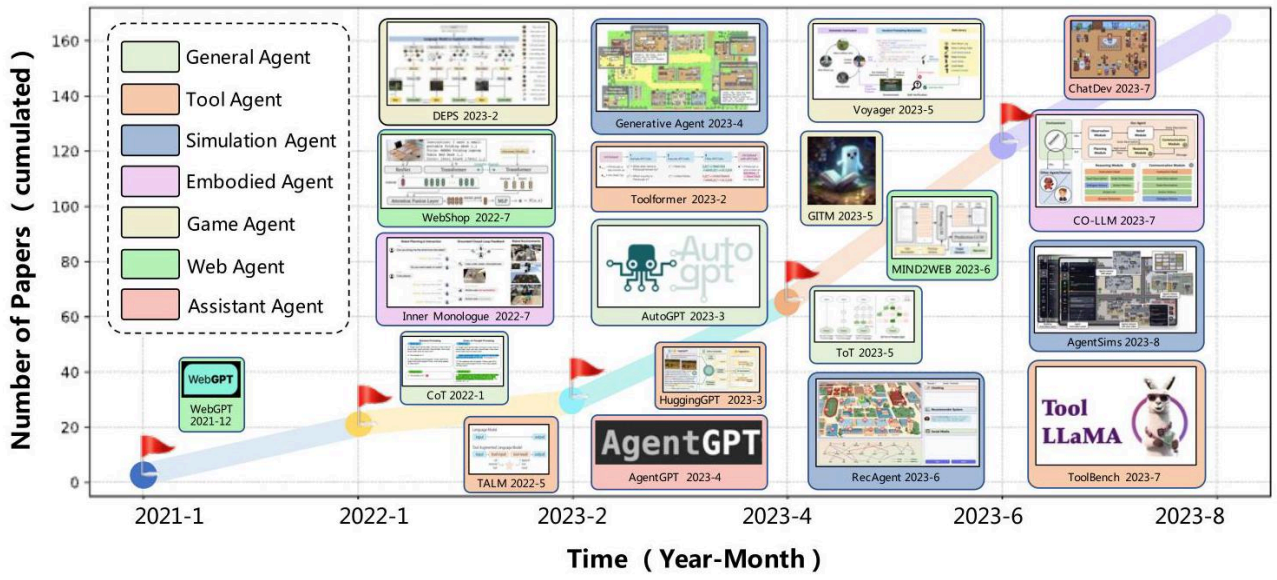

Fig. 1 Illustration of the growth trend in the field of LLM-based autonomous agents. We present the cumulative number of papers published from January 2021 to August 2023. We assign different colors to represent various agent categories. For example, a game agent aims to simulate a game-player, while a tool agent mainly focuses on tool using. For each time period, we provide a curated list of studies with diverse agent categories.
图1 基于LLM的自主智能体领域增长趋势示意。展示了2021年1月至2023年8月期间发表论文的累计数量。我们用不同颜色表示各类智能体类别。例如，游戏智能体旨在模拟游戏玩家，而工具智能体主要关注工具使用。每个时间段均提供了涵盖多样智能体类别的精选研究列表。

In recent years, large language models (LLMs) have achieved notable successes, demonstrating significant potential to achieve human-like intelligence [5-10]. This capability arises from leveraging comprehensive training datasets alongside a substantial number of model parameters. Building upon this capability, there has been a growing research area that employs LLMs as central controllers to construct autonomous agents to obtain human-like decision-making capabilities [11-17].
近年来，大型语言模型（LLMs）取得显著成功，展现出实现类人智能的巨大潜力[5-10]。这种能力源于利用全面的训练数据集和大量模型参数。在此基础上，研究者们逐渐兴起了以LLM为核心控制器构建自主智能体以实现类人决策能力的研究方向[11-17]。

Compared to reinforcement learning, LLM-based agents possess more comprehensive internal world knowledge, enabling them to perform informed actions even without training on specific domain data. Furthermore, LLM-based agents can offer natural language interfaces for human interaction, providing greater flexibility and enhanced explainability.
与强化学习相比，基于LLM的智能体拥有更全面的内部世界知识，使其即使未在特定领域数据上训练，也能执行有信息支撑的行动。此外，基于LLM的智能体能够提供自然语言交互界面，增强了灵活性和可解释性。

Along this direction, researchers have developed numerous promising models (see Figure 1 for an overview), where the key idea is to equip LLMs with human capabilities such as memory and planning to make them behave like humans and complete various tasks effectively. Previously, these models were proposed independently, with limited efforts made to summarize and compare them holistically. However, we believe that a systematic summary of this rapidly developing field is of great significance for a comprehensive understanding of it and is beneficial in inspiring future research.
沿此方向，研究者开发了众多有前景的模型（见图1概览），其核心思想是赋予LLM记忆和规划等人类能力，使其表

现得如同人类般，有效完成各类任务。此前，这些模型多为独立提出，缺乏系统总结和整体比较。我们认为，对这一快速发展的领域进行系统总结具有重要意义，有助于全面理解并激发未来研究。

In this paper, we conduct a comprehensive survey of the field of LLM-based autonomous agents. We organize our survey around three key aspects: construction, application, and evaluation of LLM-based autonomous agents. For agent construction, we focus on two problems, that is, (1) how to design the agent architecture to better leverage LLMs, and (2) how to inspire and enhance the agent capability to complete different tasks. Intuitively, the first problem aims to build the hardware fundamentals for the agent, while the second problem focuses on providing the agent with software resources. For the first problem, we present a unified agent framework, which can encompass most of the previous studies. For the second problem, we provide a summary on the commonly-used strategies for agents' capability acquisition. In addition to discussing agent construction, we also provide a systematic overview of the applications of LLM-based autonomous agents in social science, natural science, and engineering. Finally, we delve into the strategies for evaluating LLM-based autonomous agents, focusing on both subjective and objective strategies.
本文对基于大型语言模型（LLM）的自主智能体领域进行了全面综述。我们围绕三个关键方面组织调研内容：基于LLM的自主智能体的构建、应用和评估。在智能体构建方面，我们关注两个问题，即（1）如何设计智能体架构以更好地利用LLM，以及（2）如何激发和增强智能体完成不同任务的能力。直观来看，第一个问题旨在为智能体构建硬件基础，而第二个问题则侧重于为智能体提供软件资源。针对第一个问题，我们提出了一个统一的智能体框架，能够涵盖大多数已有研究。针对第二个问题，我们总结了智能体能力获取的常用策略。除了讨论智能体构建外，我们还系统性地概述了基于LLM的自主智能体在社会科学、自然科学和工程领域的应用。最后，我们深入探讨了评估基于LLM的自主智能体的策略，重点关注主观和客观两种方法。

In summary, this survey conducts a systematic review and establishes comprehensive taxonomies for existing studies in the burgeoning field of LLM-based autonomous agents. Our focus encompasses three primary areas: the construction of agents, their applications, and methods of evaluation. Drawing from a wealth of previous studies, we identify various challenges in this field and discuss potential future directions. We expect that our survey can provide newcomers of LLM-based autonomous agents with a comprehensive background knowledge, and also encourage further groundbreaking studies.
总之，本综述对新兴的基于LLM的自主智能体领域进行了系统回顾，并建立了现有研究的全面分类体系。我们的关注点涵盖三个主要方面：智能体的构建、应用及评估方法。基于大量前人研究，我们识别了该领域的多项挑战，并讨论了潜在的未来发展方向。我们期望本综述能为基于LLM的自主智能体领域的新入者提供全面的背景知识，同时激励更多开创性研究的开展。

3　2 LLM-based Autonomous Agent Con- struction

4　2 基于LLM的自主智能体构建

LLM-based autonomous agents are expected to effectively perform diverse tasks by leveraging the human-like capabilities of LLMs. In order to achieve this goal, there are two significant aspects: (1) which architecture should be designed to better use LLMs and (2) given the designed architecture, how to enable the agent to acquire capabilities for accomplishing specific tasks. Within the context of architecture design, we contribute a systematic synthesis of existing research, culminating in a comprehensive unified framework. As for the second aspect, we summarize the strategies for agent capability acquisition based on whether they fine-tune the LLMs. Comparing LLM-based autonomous agents to traditional machine learning, architecture design is analogous to defining the network structure, while capability acquisition resembles the process of learning network parameters. In the following sections, we explore these two aspects in greater detail.
基于LLM的自主智能体期望通过利用LLM类人能力有效执行多样化任务。为实现此目标，有两个重要方面：（1）应设计何种架构以更好地利用LLM；（2）在设计好架构的前提下，如何使智能体获得完成特定任务的能力。在架构设计方面，我们对现有研究进行了系统整合，提出了一个全面的统一框架。至于第二个方面，我们基于是否对LLM进

行微调，总结了智能体能力获取的策略。将基于LLM的自主智能体与传统机器学习相比，架构设计类似于定义网络结构，而能力获取则类似于学习网络参数。接下来的章节中，我们将更详细地探讨这两个方面。

4.1 2.1 Agent Architecture Design

4.2 2.1 智能体架构设计

Recent advancements in LLMs have demonstrated their great potential to accomplish a wide range of tasks in the form of question-answering (QA). However, building autonomous agents is far from QA, since they need to fulfill specific roles and autonomously perceive and learn from the environment to evolve themselves like humans. To bridge the gap between traditional LLMs and autonomous agents, a crucial aspect is to design rational agent architectures to assist LLMs in maximizing their capabilities. Along this direction, previous work has developed a number of modules to enhance LLMs. In this section, we propose a unified framework to summarize these modules. Specifically, the overall structure of our framework is illustrated in Figure 2, which is composed of a profiling module, a memory module, a planning module, and an action module. The purpose of the profiling module is to identify the role of the agent. The memory and planning modules place the agent into a dynamic environment, enabling it to recall past behaviors and plan future actions. The action module is responsible for translating the agent's decisions into specific outputs. Within these modules, the profiling module impacts the memory and planning modules, and collectively, these three modules influence the action module. In the following, we detail these modules.

近期LLM的进展展示了其以问答（QA）形式完成广泛任务的巨大潜力。然而，构建自主智能体远非简单的问答，因为它们需要承担特定角色，并能自主感知和从环境中学习以实现自我进化，类似人类。为弥合传统LLM与自主智能体之间的差距，设计合理的智能体架构以辅助LLM最大化其能力是关键。沿着这一方向，已有工作开发了多种模块以增强LLM。本文提出一个统一框架以总结这些模块。具体而言，我们框架的整体结构如图2所示，由画像模块、记忆模块、规划模块和行动模块组成。画像模块的目的是识别智能体的角色。记忆和规划模块将智能体置于动态环境中，使其能够回忆过去行为并规划未来行动。行动模块负责将智能体的决策转化为具体输出。在这些模块中，画像模块影响记忆和规划模块，三者共同影响行动模块。以下内容将详细介绍这些模块。

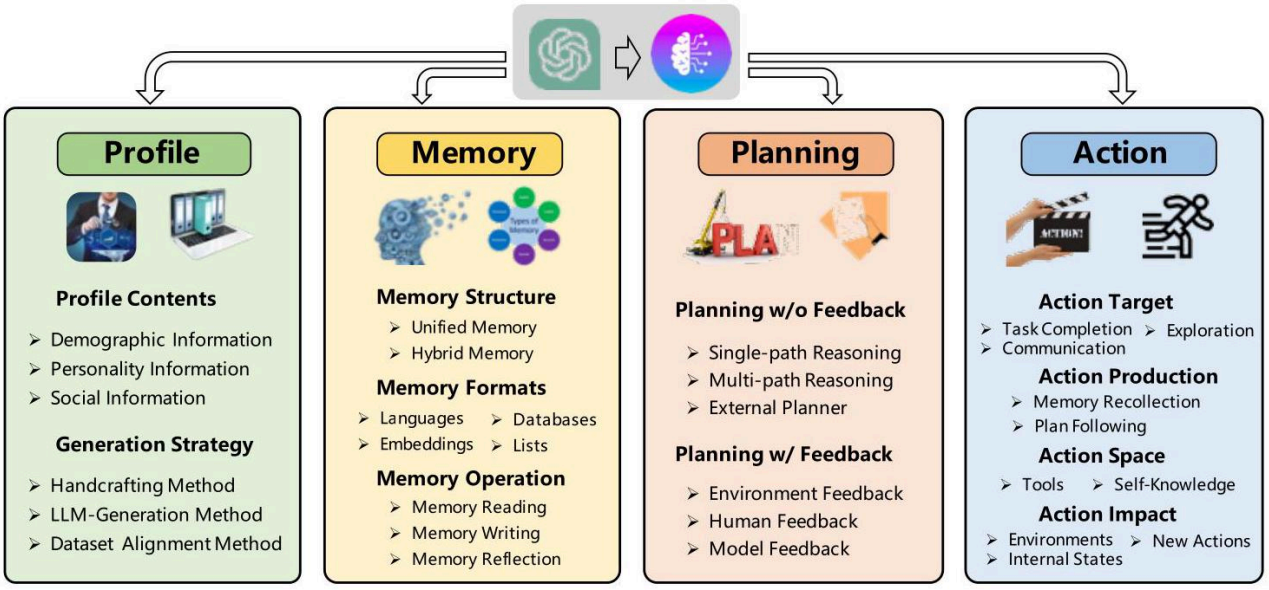

Fig. 2 A unified framework for the architecture design of LLM-based autonomous agent.
图2 基于LLM的自主智能体架构设计统一框架。

4.2.1 2.1.1 Profiling Module

4.2.2 2.1.1 画像模块

Autonomous agents typically perform tasks by assuming specific roles, such as coders, teachers, and domain experts [18, 19] . The profiling module aims to indicate the profiles of the agent roles, which are usually written into the prompt to influence the behavior of the LLM. Agent profiles typically encompass basic information such as age, gender, and career [20], as well as psychology information, reflecting the personalities of the agents, and social information, detailing the relationships between agents [21]. The choice of information to profile the agent is largely determined by the specific application scenarios. For instance, if the application aims to study human cognitive process, then the psychology information becomes pivotal. After identifying the types of profile information, the next important problem is to create specific profiles for the agents. Existing literature commonly employs the following three strategies.
自主智能体通常通过承担特定角色执行任务，如程序员、教师和领域专家[18, 19]。画像模块旨在指示智能体角色的画像，通常写入提示词以影响LLM的行为。智能体画像通常包含基本信息，如年龄、性别和职业[20]，以及心理信息，反映智能体的个性特征，还有社会信息，描述智能体之间的关系[21]。选择何种信息来画像智能体，主要取决于具体应用场景。例如，若应用旨在研究人类认知过程，则心理信息尤为关键。在确定画像信息类型后，下一个重要问题是为智能体创建具体画像。现有文献通常采用以下三种策略。

Handcrafting Method: in this method, agent profiles are manually specified. For instance, if one would like to design agents with different personalities, he can use "you are an outgoing person" or "you are an introverted person" to profile the agent. The handcrafting method has been leveraged in a lot of previous work to specify the agent profiles. For example, Generative Agent [22] describes the agent by the information such as name, objectives, and relationships with other agents. MetaGPT [23], ChatDev [18], and Self-collaboration [24] predefine various roles and their corresponding responsibilities in software development, manually assigning distinct profiles to each agent to facilitate collaboration. PTLLM [25] aims to explore and quantify personality traits displayed in texts generated by LLMs. This method guides LLMs in generating diverse responses by manually defining various agent characters through the use of personality assessment tools such as IPIP-NEO [26] and BFI [27]. [28] studies the toxicity of the LLM output by manually prompting LLMs with different roles, such as politicians, journalists and businesspersons. In general, the handcrafting method is very flexible, since one can assign any profile information to the agents. However, it can be also labor-intensive, particularly when dealing with a large number of agents.
手工制作方法：在此方法中，代理的个人资料是手动指定的。例如，如果想设计具有不同个性的代理，可以使用“你是一个外向的人”或“你是一个内向的人”来为代理设定档案。手工制作方法已被大量先前工作采用以指定代理档案。例如，Generative Agent [22]通过姓名、目标及与其他代理的关系等信息描述代理。MetaGPT [23]、ChatDev [18]和Self-collaboration [24]预定义了软件开发中的各种角色及其对应职责，手动为每个代理分配不同的档案以促进协作。PTLLM [25]旨在探索和量化大型语言模型（LLMs）生成文本中表现出的个性特征。该方法通过使用IPIP-NEO [26]和BFI [27]等人格评估工具，手动定义多种代理角色，引导LLMs生成多样化的响应。[28]通过手动提示LLMs扮演不同角色（如政治家、记者和商人）研究LLM输出的毒性。总体而言，手工制作方法非常灵活，因为可以为代理分配任何档案信息，但在处理大量代理时也可能非常费力。

LLM-generation Method: in this method, agent profiles are automatically generated based on LLMs. Typically, it begins by indicating the profile generation rules, elucidating the composition and attributes of the agent profiles within the target population. Then, one can optionally specify several seed agent profiles to serve as few-shot examples. Finally, LLMs are leveraged to generate all the agent profiles. For example, RecAgent [21] first creates seed profiles for a few agents by manually crafting their attributes such as age, gender, personal traits, and movie preferences. Then, it leverages ChatGPT to generate more agent profiles based on the seed information. This approach significantly reduces the time and effort required to construct agent profiles, particularly for large-scale populations. However, it may lack precise control over the generated profiles, which can result in inconsistencies or deviations from the intended characteristics.

LLM生成方法：此方法基于大型语言模型（LLMs）自动生成代理档案。通常，首先指明档案生成规则，阐明目标群体中代理档案的组成和属性。然后，可以选择性地指定若干种子代理档案作为少量示例。最后，利用LLMs生成所有代理档案。例如，RecAgent [21]首先通过手工制作属性（如年龄、性别、个人特征和电影偏好）为少数代理创建种子档案，然后利用ChatGPT基于种子信息生成更多代理档案。该方法显著减少了构建代理档案所需的时间和精力，尤其适用于大规模群体。然而，它可能缺乏对生成档案的精确控制，导致与预期特征不一致或偏差。

Dataset Alignment Method: in this method, the agent profiles are obtained from real-world datasets. Typically, one can first organize the information about real humans in the datasets into natural language prompts, and then leverage it to profile the agents. For instance, in [29], the authors assign roles to GPT-3 based on the demographic backgrounds (such as race/ethnicity, gender, age, and state of residence) of participants in the American National Election Studies (ANES). They subsequently investigate whether GPT-3 can produce similar results to those of real humans. The dataset alignment method accurately captures the attributes of the real population, thereby making the agent behaviors more meaningful and reflective of real-world scenarios.
数据集对齐方法：此方法通过真实世界的数据集获取代理档案。通常，首先将数据集中关于真实人物的信息组织成自然语言提示，然后利用这些信息为代理设定档案。例如，在[29]中，作者根据美国国家选举研究（ANES）参与者的人口统计背景（如种族/族裔、性别、年龄和居住州）为GPT-3分配角色，随后研究GPT-3是否能产生与真实人类相似的结果。数据集对齐方法准确捕捉了真实人口的属性，使代理行为更具意义且更能反映现实场景。

Remark. While most of the previous work leverage the above profile generation strategies independently, we argue that combining them may yield additional benefits. For example, in order to predict social developments via agent simulation, one can leverage real-world datasets to profile a subset of the agents, thereby accurately reflecting the current social status. Subsequently, roles that do not exist in the real world but may emerge in the future can be manually assigned to the other agents, enabling the prediction of future social development. Beyond this example, one can also flexibly combine the other strategies. The profile module serves as the foundation for agent design, exerting significant influence on the agent memorization, planning, and action procedures.
备注。虽然大多数先前工作独立采用上述档案生成策略，但我们认为结合使用它们可能带来额外优势。例如，为了通过代理模拟预测社会发展，可以利用真实世界数据集为部分代理设定档案，从而准确反映当前社会状况。随后，可以手动为其他代理分配现实中不存在但未来可能出现的角色，实现对未来社会发展的预测。除了此例外，还可以灵活组合其他策略。档案模块作为代理设计的基础，对代理的记忆、规划和行动过程具有重要影响。

4.2.3 2.1.2 Memory Module

4.2.4 2.1.2 记忆模块

The memory module plays a very important role in the agent architecture design. It stores information perceived from the environment and leverages the recorded memories to facilitate future actions. The memory module can help the agent to accumulate experiences, self-evolve, and behave in a more consistent, reasonable, and effective manner. This section provides a comprehensive overview of the memory module, focusing on its structures, formats, and operations.
记忆模块在代理架构设计中起着非常重要的作用。它存储从环境中感知的信息，并利用记录的记忆促进未来的行动。记忆模块帮助代理积累经验、自我进化，并以更连贯、合理和有效的方式表现行为。本节全面介绍记忆模块，重点讨论其结构、格式和操作。

Memory Structures: LLM-based autonomous agents often draw inspiration from cognitive science research on human memory processes. Human memory follows a general progression from sensory memory that registers perceptual inputs, to short-term memory that maintains information transiently, to long-term memory that consolidates information over extended periods. When designing the agent memory structures, researchers take inspiration from these aspects of human memory. In particular, short-term memory is analogous to the input information within the context window constrained by the transformer architecture. Long-term memory resembles the external vector storage that agents can rapidly query and retrieve from as needed. In the following, we introduce two commonly used memory structures based on the short-term and long-term memories.
记忆结构：基于大型语言模型（LLM）的自主代理常借鉴认知科学中关于人类记忆过程的研究。人类记忆通常经历

从感官记忆（注册感知输入）、短期记忆（暂时维持信息）到长期记忆（长时间巩固信息）的过程。在设计代理记忆结构时，研究者从人类记忆的这些方面获得启发。特别地，短期记忆类似于受变换器架构限制的上下文窗口内的输入信息；长期记忆则类似于代理可以快速查询和检索的外部向量存储。以下介绍两种基于短期记忆和长期记忆的常用记忆结构。

- Unified Memory. This structure only simulates the human short-term memory, which is usually realized by in-context learning, and the memory information is directly written into the prompts. For example, RLP [30] is a conversation agent, which maintains internal states for the speaker and listener. During each round of conversation, these states serve as LLM prompts, functioning as the agent's short-term memory. SayPlan [31] is an embodied agent specifically designed for task planning. In this agent, the scene graphs and environment feedback serve as the agent's short-term memory, guiding its actions. CALYPSO [32] is an agent designed for the game Dungeons & Dragons, which can assist Dungeon Masters in the creation and narration of stories. Its short-term memory is built upon scene descriptions, monster information, and previous summaries. DEPS [33] is also a game agent, developed for Minecraft. The agent initially generates task plans and then utilizes them to prompt LLMs, which in turn produce actions to complete the task. These plans can be deemed as the agent's short-term memory. In practice, implementing short-term memory is straightforward and can enhance an agent's ability to perceive recent or contextually sensitive behaviors and observations. However, the limited context window of LLMs restricts incorporating comprehensive memories into prompts, which can impair agent performance. This challenge necessitates LLMs with larger context windows and the ability to handle extended contexts. Consequently, numerous researchers turn to hybrid memory systems to mitigate this issue.
- 统一记忆。该结构仅模拟人类的短期记忆，通常通过上下文学习实现，记忆信息直接写入提示中。例如，RLP [30] 是一个对话代理，维护说话者和听者的内部状态。在每轮对话中，这些状态作为大语言模型（LLM）的提示，充当代理的短期记忆。SayPlan [31] 是一个专门用于任务规划的具身代理。在该代理中，场景图和环境反馈作为代理的短期记忆，指导其行动。CALYPSO [32] 是为《龙与地下城》（Dungeons & Dragons）设计的代理，能够协助地下城主进行故事创作和叙述。其短期记忆基于场景描述、怪物信息和之前的总结。DEPS [33] 也是一个游戏代理，针对《我的世界》（Minecraft）开发。该代理首先生成任务计划，然后利用这些计划提示LLM，进而产生完成任务的动作。这些计划可视为代理的短期记忆。实际上，实现短期记忆较为简单，能增强代理感知近期或上下文敏感行为和观察的能力。然而，LLM的上下文窗口有限，限制了将全面记忆纳入提示，可能影响代理性能。此挑战需要具备更大上下文窗口和处理扩展上下文能力的LLM。因此，许多研究者转向混合记忆系统以缓解该问题。

- Hybrid Memory. This structure explicitly models the human short-term and long-term memories. The short-term memory temporarily buffers recent perceptions, while long-term memory consolidates important information over time. For instance, Generative Agent [20] employs a hybrid memory structure to facilitate agent behaviors. The short-term memory contains the context information about the agent current situations, while the long-term memory stores the agent past behaviors and thoughts, which can be retrieved according to the current events. AgentSims [34] also implements a hybrid memory architecture. The information provided in the prompt can be considered as short-term memory. In order to enhance the storage capacity of memory, the authors propose a long-term memory system that utilizes a vector database, facilitating efficient storage and retrieval. Specifically, the agent's daily memories are encoded as embeddings and stored in the vector database. If the agent needs to recall its previous memories, the long-term memory system retrieves relevant information using embedding similarities. This process can improve the consistency of the agent's behavior. In GITM [16], the short-term memory stores the current trajectory, and the long-term memory saves reference plans summarized from successful prior trajectories. Long-term memory provides stable knowledge, while short-term memory allows flexible planning. Reflexion [12] utilizes a short-term sliding window to capture recent feedback and incorporates persistent long-term storage to retain condensed insights. This combination allows for the utilization of both detailed immediate experiences and high-level abstractions. SCM [35] selectively activates the most relevant long-term knowledge to combine with short-term memory, enabling reasoning over complex contextual dialogues. Sim-plyRetrieve [36] utilizes user queries as short-term memory and stores long-term memory using private knowledge bases. This design enhances the model accuracy while guaranteeing user

privacy. MemorySandbox [37] implements long-term and short-term memory to store different objects, which can then be accessed throughout various conversations. Users can create multiple conversations with different agents on the same canvas, facilitating the sharing of memory objects through a simple drag-and-drop interface. In practice, integrating both short-term and long-term memories can enhance an agent's ability for long-range reasoning and accumulation of valuable experiences, which are crucial for accomplishing tasks in complex environments.

- 混合记忆。该结构明确模拟人类的短期记忆和长期记忆。短期记忆暂时缓冲近期感知，长期记忆则随时间巩固重要信息。例如，Generative Agent [20] 采用混合记忆结构以促进代理行为。短期记忆包含代理当前情境的上下文信息，长期记忆存储代理过去的行为和思考，可根据当前事件检索。AgentSims [34] 也实现了混合记忆架构。提示中提供的信息可视为短期记忆。为增强记忆存储容量，作者提出利用向量数据库的长期记忆系统，实现高效存储和检索。具体而言，代理的日常记忆被编码为嵌入向量并存储于向量数据库中。如需回忆先前记忆，长期记忆系统通过嵌入相似度检索相关信息。此过程可提升代理行为的一致性。在GITM [16] 中，短期记忆存储当前轨迹，长期记忆保存从成功先前轨迹总结的参考计划。长期记忆提供稳定知识，短期记忆支持灵活规划。
  Reflexion [12] 利用短期滑动窗口捕捉近期反馈，并结合持久的长期存储以保留浓缩见解。此组合兼顾详细的即时体验和高层次抽象。SCM [35] 有选择地激活最相关的长期知识与短期记忆结合，实现复杂上下文对话的推理。SimplyRetrieve [36] 使用用户查询作为短期记忆，长期记忆则存储于私有知识库中。此设计提升模型准确性，同时保障用户隐私。MemorySandbox [37] 实现长期和短期记忆以存储不同对象，供多次对话访问。用户可在同一画布上与不同代理创建多次对话，通过简单拖放界面共享记忆对象。实践中，整合短期和长期记忆可增强代理的长程推理能力和宝贵经验积累，这对在复杂环境中完成任务至关重要。

Remark. Careful readers may find that there may also exist another type of memory structure, that is, only based on the long-term memory. However, we find that such type of memory is rarely documented in the literature. Our speculation is that the agents are always situated in continuous and dynamic environments, with consecutive actions displaying a high correlation. Therefore, the capture of short-term memory is very important and usually cannot be disregarded.
备注。细心的读者可能会发现还存在另一种记忆结构，即仅基于长期记忆。然而，我们发现文献中对此类记忆的记载较少。我们推测，代理通常处于连续且动态的环境中，连续动作高度相关。因此，捕捉短期记忆非常重要，通常不可忽视。

Memory Formats: In addition to the memory structure, another perspective to analyze the memory module is based on the formats of the memory storage medium, for example, natural language memory or embedding memory. Different memory formats possess distinct strengths and are suitable for various applications. In the following, we introduce several representative memory formats.
记忆格式：除了记忆结构外，分析记忆模块的另一个视角是基于记忆存储介质的格式，例如自然语言记忆或嵌入记忆。不同记忆格式具有各自优势，适用于不同应用。以下介绍几种代表性记忆格式。

- Natural Languages. In this format, memory information such as the agent behaviors and observations are directly described using raw natural language. This format possesses several strengths. Firstly, the memory information can be expressed in a flexible and understandable manner. Moreover, it retains rich semantic information that can provide comprehensive signals to guide agent behaviors. In the previous work, Reflexion [12] stores experiential feedback in natural language within a sliding window. Voyager [38] employs natural language descriptions to represent skills within the Minecraft game, which are directly stored in memory.
- 自然语言。在此格式中，记忆信息如代理行为和观察直接用原始自然语言描述。该格式具备多项优势。首先，记忆信息表达灵活且易于理解。此外，它保留丰富的语义信息，能为指导代理行为提供全面信号。在先前工作中，Reflexion [12] 在滑动窗口内以自然语言存储体验反馈。Voyager [38] 使用自然语言描述表示《我的世界》中的技能，直接存储于记忆中。

- Embeddings. In this format, memory information is encoded into embedding vectors, which enhances both retrieval and reading efficiency. For instance, MemoryBank [39] encodes each memory segment as an embedding vector and employs a dual-tower dense retrieval model to efficiently retrieve relevant information from past conversations.

- 嵌入（Embeddings）。在这种格式中，记忆信息被编码为嵌入向量，从而提升了检索和阅读的效率。例如，MemoryBank [39] 将每个记忆片段编码为嵌入向量，并采用双塔密集检索模型高效地从过去的对话中检索相关信息。

- Databases. In this format, memory information is stored in databases, allowing the agent to manipulate memories efficiently and comprehensively. For example, ChatDB [40] uses a database as a symbolic memory module. The agent can utilize SQL statements to precisely add, delete, and modify the memory information.
- 数据库（Databases）。在这种格式中，记忆信息存储于数据库中，使得智能体能够高效且全面地操作记忆。例如，ChatDB [40] 使用数据库作为符号记忆模块，智能体可以利用SQL语句精确地添加、删除和修改记忆信息。

- Structured Lists. In this format, memory information is organized into lists, and the semantic of memory can be conveyed in an efficient and concise manner. For instance, GITM [16] stores action lists for sub-goals in a hierarchical tree structure. The hierarchical structure explicitly captures the relationships between goals and corresponding plans. RET-LLM [41] initially converts natural language sentences into triplet phrases, and subsequently stores them in memory.
- 结构化列表（Structured Lists）。在这种格式中，记忆信息被组织成列表，能够以高效且简洁的方式传达记忆的语义。例如，GITM [16] 将子目标的动作列表存储在分层树结构中，该层级结构明确捕捉了目标与对应计划之间的关系。RET-LLM [41] 则先将自然语言句子转换为三元组短语，随后存储于记忆中。

Remark. Here we only show several representative memory formats, but it is important to note that there are many uncovered ones, such as the programming code used by [38]. Moreover, it should be emphasized that these formats are not mutually exclusive; many models incorporate multiple formats to concurrently harness their respective benefits. A notable example is the memory module of GITM [16], which utilizes a key-value list structure. In this structure, the keys are represented by embedding vectors, while the values consist of raw natural languages. The use of embedding vectors allows for efficient retrieval of memory records. By utilizing natural languages, the memory contents become highly comprehensive, enabling more informed agent actions.
备注。这里仅展示了几种具有代表性的记忆格式，但需要注意的是，还有许多未涵盖的格式，如[38]中使用的编程代码。此外，应强调这些格式并非相互排斥，许多模型结合多种格式以同时利用各自优势。一个显著的例子是GITM [16]的记忆模块，它采用键值列表结构，其中键由嵌入向量表示，值则为原始自然语言。嵌入向量的使用使得记忆记录的检索更加高效，而自然语言的采用则使记忆内容高度丰富，支持更为明智的智能体行为。

Above, we mainly discuss the internal designs of the memory module. In the following, we turn our focus to memory operations, which are used to interact with external environments.
以上主要讨论了记忆模块的内部设计，接下来我们将关注记忆操作，这些操作用于与外部环境交互。

Memory Operations: The memory module plays a critical role in allowing the agent to acquire, accumulate, and utilize significant knowledge by interacting with the environment. The interaction between the agent and the environment is accomplished through three crucial memory operations: memory reading, memory writing, and memory reflection. In the following, we introduce these operations more in detail.
记忆操作：记忆模块在使智能体通过与环境交互获取、积累和利用重要知识方面发挥关键作用。智能体与环境的交互通过三种关键的记忆操作实现：记忆读取、记忆写入和记忆反思。以下将详细介绍这些操作。

- Memory Reading. The objective of memory reading is to extract meaningful information from memory to enhance the agent's actions. For example, using the previously successful actions to achieve similar goals [16]. The key of memory reading lies in how to extract valuable information from history actions. Usually, there are three commonly used criteria for information extraction, that is, the recency, relevance, and importance [20]. Memories that are more recent, relevant, and important are more likely to be extracted. Formally, we conclude the following equation from existing literature for memory information extraction:
- 记忆读取。记忆读取的目标是从记忆中提取有意义的信息，以增强智能体的行为。例如，利用先前成功的行动来实现类似目标[16]。记忆读取的关键在于如何从历史行动中提取有价值的信息。通常，有三种常用的信息提取标

准，即新近性、相关性和重要性[20]。更近期、更相关和更重要的记忆更有可能被提取。形式上，我们从现有文献中总结出以下记忆信息提取公式：

$$m^{*}=\arg\max_{m\in M}\left(\alpha s^{rec}\left(q,m\right)+\beta s^{rel}\left(q,m\right)+\gamma s^{imp}\left(m\right)\right),$$

(1)

where q is the query,for example,the task that the agent should address or the context in which the agent is situated. M is the set of all memories. $s^{rec}\left(\cdot\right),s^{rel}\left(\cdot\right)$ and $s^{imp}\left(\cdot\right)$ are the scoring functions for measuring the recency, relevance, and importance of the memory m ,with higher scores indicating more recent, more relevant, and more important memories respectively. These scoring functions can be implemented using various methods, for example, $s^{\mathrm{rel}}\left(q,m\right)$ can be calculated using vector similarity measures between query and memory embeddings. It should be noted that s^{imp} only reflects the characters of the memory itself, thus it is unrelated to the query $q.\alpha,\beta$ and γ are balancing parameters. By assigning them with different values, one can obtain various memory reading strategies. For example, by setting $\alpha=\gamma=0$,many studies [16, 30, 38, 41] only consider the relevance score s^{rel} for memory reading. By assigning $\alpha=\beta=\gamma=1.0$,[20] equally weights all the above three metrics to extract information from memory.
其中q是查询，例如智能体应处理的任务或智能体所处的上下文。M是所有记忆的集合。$s^{rec}\left(\cdot\right),s^{rel}\left(\cdot\right)$和$s^{imp}\left(\cdot\right)$是用于衡量记忆$m$的新近性、相关性和重要性的评分函数，分数越高表示记忆越新近、越相关和越重要。这些评分函数可以通过多种方法实现，例如，$s^{\mathrm{rel}}\left(q,m\right)$可以通过查询与记忆嵌入向量之间的向量相似度计算。需要注意的是，s^{imp}仅反映记忆本身的特性，因此与查询无关。$q.\alpha,\beta$和γ是平衡参数，通过赋予不同的值，可以获得多种记忆读取策略。例如，通过设置$\alpha=\gamma=0$，许多研究[16, 30, 38, 41]仅考虑相关性评分s^{rel}进行记忆读取。通过赋值$\alpha=\beta=\gamma=1.0$，[20]对上述三种指标赋予相等权重以提取记忆信息。

- Memory Writing. The purpose of memory writing is to store information about the perceived environment in memory. Storing valuable information in memory provides a foundation for retrieving informative memories in the future, enabling the agent to act more efficiently and rationally. During the memory writing process, there are two potential problems that should be carefully addressed. On one hand, it is crucial to address how to store information that is similar to existing memories (i.e., memory duplicated). On the other hand, it is important to consider how to remove information when the memory reaches its storage limit (i.e., memory overflow). In the following, we discuss these problems more in detail. (1) Memory Duplicated. To incorporate similar information, people have developed various methods for integrating new and previous records. For instance, in [16], the successful action sequences related to the same subgoal are stored in a list. Once the size of the list reaches $\mathrm{N}\left(=5\right)$,all the sequences in it are condensed into a unified plan solution using LLMs. The original sequences in the memory are replaced with the newly generated one. Augmented LLM [42] aggregates duplicate information via count accumulation, avoiding redundant storage. (2) Memory Overflow. In order to write information into the memory when it is full, people design different methods to delete existing information to continue the memorizing process. For example, in ChatDB [40], memories can be explicitly deleted based on user commands. RET-LLM [41] uses a fixed-size buffer for memory, overwriting the oldest entries in a first-in-first-out (FIFO) manner.
- 记忆写入。记忆写入的目的是将感知到的环境信息存储到记忆中。将有价值的信息存储在记忆中为未来检索有用记忆提供了基础，使智能体能够更高效、更理性地行动。在记忆写入过程中，有两个潜在问题需要认真解决。一方面，如何存储与现有记忆相似的信息（即记忆重复）至关重要。另一方面，当记忆达到存储上限时，如何删除信息（即记忆溢出）也需考虑。以下将详细讨论这些问题。(1) 记忆重复。为了整合相似信息，研究者们开发了多种方法来融合新旧记录。例如，在[16]中，与同一子目标相关的成功动作序列被存储在一个列表中。一旦列表大小达到$\mathrm{N}\left(=5\right)$，列表中的所有序列将通过大型语言模型（LLMs）浓缩成一个统一的计划方案。记忆中的原始序列被新生成的序列替代。增强型LLM [42]通过计数累积聚合重复信息，避免冗余存储。(2) 记忆溢出。为了在

记忆已满时继续写入信息，研究者设计了不同的方法删除现有信息以继续记忆过程。例如，在ChatDB [40]中，记忆可以根据用户命令被显式删除。RET-LLM [41]使用固定大小的缓冲区存储记忆，以先进先出（FIFO）方式覆盖最旧条目。

- Memory Reflection. Memory reflection emulates humans' ability to witness and evaluate their own cognitive, emotional, and behavioral processes. When adapted to agents, the objective is to provide agents with the capability to independently summarize and infer more abstract, complex and high-level information. More specifically, in Generative Agent [20], the agent has the capability to summarize its past experiences stored in memory into broader and more abstract insights. To begin with, the agent generates three key questions based on its recent memories. Then, these questions are used to query the memory to obtain relevant information. Building upon the acquired information, the agent generates five insights, which reflect the agent high-level ideas. For example, the low-level memories "Klaus Mueller is writing a research paper", "Klaus Mueller is engaging with a librarian to further his research", and "Klaus Mueller is conversing with Ayesha Khan about his research" can induce the high-level insight "Klaus Mueller is dedicated to his research". In addition, the reflection process can occur hierarchically, meaning that the insights can be generated based on existing insights. In GITM [16], the actions that successfully accomplish the sub-goals are stored in a list. When the list contains more than five elements, the agent summarizes them into a common and abstract pattern and replaces all the elements. In ExpeL [43], two approaches are introduced for the agent to acquire reflection. Firstly, the agent compares successful or failed trajectories within the same task. Secondly, the agent learns from a collection of successful trajectories to gain experiences.
- 记忆反思。记忆反思模拟人类见证并评估自身认知、情感和行为过程的能力。应用于智能体时，目标是赋予智能体独立总结和推断更抽象、复杂和高级信息的能力。具体来说，在生成型智能体（Generative Agent）[20]中，智能体能够将存储在记忆中的过去经历总结为更广泛、更抽象的见解。首先，智能体基于近期记忆生成三个关键问题。然后，利用这些问题查询记忆以获取相关信息。在获取信息的基础上，智能体生成五个见解，反映其高级思想。例如，低级记忆“克劳斯 · 穆勒（Klaus Mueller）正在撰写研究论文”、“克劳斯 · 穆勒正在与图书管理员交流以推进研究”和“克劳斯 · 穆勒正在与艾莎 · 汗（Ayesha Khan）讨论他的研究”可以引发高级见解“克劳斯 · 穆勒专注于他的研究”。此外，反思过程可以分层进行，即见解可以基于已有见解生成。在GITM [16]中，成功完成子目标的动作被存储在列表中。当列表元素超过五个时，智能体将其总结为一个共同且抽象的模式，并替换所有元素。在ExpeL [43]中，介绍了两种智能体获取反思的方法。首先，智能体比较同一任务中成功或失败的轨迹。其次，智能体从一组成功轨迹中学习以积累经验。

A significant distinction between traditional LLMs and the agents is that the latter must possess the capability to learn and complete tasks in dynamic environments. If we consider the memory module as responsible for managing the agents' past behaviors, it becomes essential to have another significant module that can assist the agents in planning their future actions. In the following, we present an overview of how researchers design the planning module.

传统大型语言模型（LLMs）与智能体之间的一个显著区别在于，后者必须具备在动态环境中学习和完成任务的能力。如果将记忆模块视为管理智能体过去行为的模块，那么另一个重要模块则是辅助智能体规划未来行动的模块。以下内容将概述研究者如何设计规划模块。

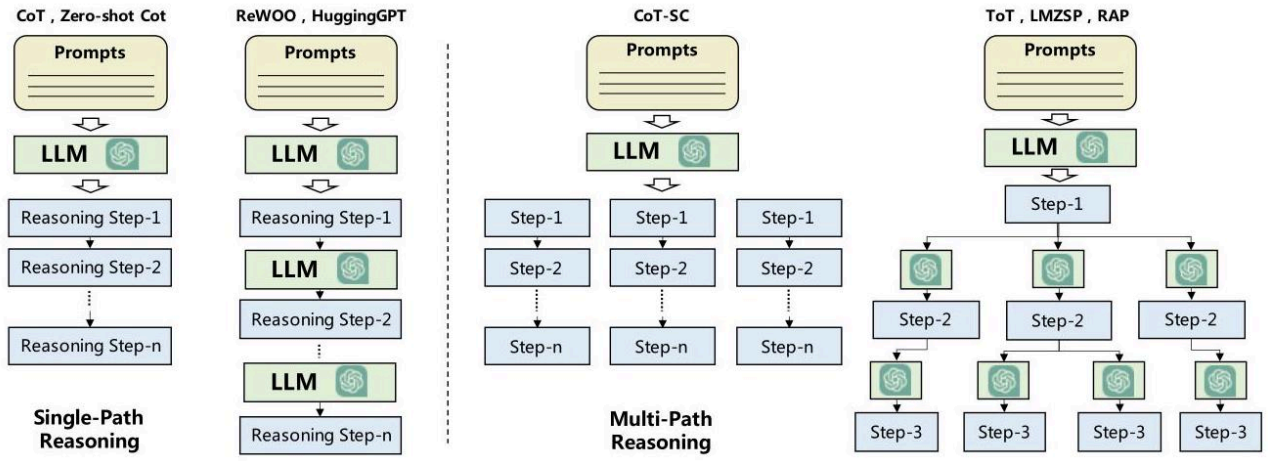

Fig. 3 Comparison between the strategies of single-path and multi-path reasoning. LMZSP is the model proposed in [44].
图3 单路径与多路径推理策略的比较。LMZSP是[44]中提出的模型。

4.2.5 2.1.3 Planning Module

4.2.6 2.1.3 规划模块

When faced with a complex task, humans tend to deconstruct it into simpler subtasks and solve them individually. The planning module aims to empower the agents with such human capability, which is expected to make the agent behave more reasonably, powerfully, and reliably. In specific, we summarize existing studies based on whether the agent can receive feedback in the planing process, which are detailed as follows:
面对复杂任务时，人类倾向于将其拆解为更简单的子任务并逐一解决。规划模块旨在赋予智能体这种人类能力，期望使智能体表现得更合理、更强大且更可靠。具体而言，我们根据智能体在规划过程中是否能接收反馈对现有研究进行总结，具体如下：

Planning without Feedback: In this method, the agents do not receive feedback that can influence its future behaviors after taking actions. In the following, we present several representative strategies.
无反馈规划：在此方法中，智能体在采取行动后不会收到影响其未来行为的反馈。以下介绍几种代表性策略。

- Single-path Reasoning. In this strategy, the final task is decomposed into several intermediate steps. These steps are connected in a cascading manner, with each step leading to only one subsequent step. LLMs follow these steps to achieve the final goal. Specifically, Chain of Thought (CoT) [45] proposes inputting reasoning steps for solving complex problems into the prompt. These steps serve as examples to inspire LLMs to plan and act in a step-by-step manner. In this method, the plans are created based on the inspiration from the examples in the prompts. Zero-shot-CoT [46] enables LLMs to generate task reasoning processes by prompting them with trigger sentences like "think step by step". Unlike CoT, this method does not incorporate reasoning steps as examples in the prompts. Re-Prompting [47] involves checking whether each step meets the necessary prerequisites before generating a plan. If a step fails to meet the prerequisites, it introduces a prerequisite error message and prompts the LLM to regenerate the plan. ReWOO [48] introduces a paradigm of separating plans from external observations, where the agents first generate plans and obtain observations independently, and then combine them together to derive the final results. HuggingGPT [13] first decomposes the task into many sub-goals, and then solves each of them based on Huggingface. Different from CoT and Zero-shot-CoT, which outcome all the reasoning steps in a one-shot manner, ReWOO and HuggingGPT produce the results by accessing LLMs multiply times. SWIFTSAGE [49], inspired by the dual-process theory of human cognition [50], combines the power of both SWIFT and SAGE modules for planning in complex interactive tasks. The

SWIFT module provides quick responses based on learned patterns, while the SAGE module, using large language models, conducts in-depth planning by asking key questions and generating action sequences to ensure successful task completion.

- 单路径推理。在该策略中，最终任务被分解为若干中间步骤，这些步骤以级联方式连接，每一步仅通向一个后续步骤。大型语言模型（LLMs）按照这些步骤实现最终目标。具体而言，思维链（Chain of Thought, CoT）[45]提出将解决复杂问题的推理步骤输入提示中，这些步骤作为示例激发LLMs逐步规划和行动。在此方法中，计划基于提示中示例的启发生成。零样本思维链（Zero-shot-CoT）[46]通过触发句如“逐步思考”提示LLMs生成任务推理过程，与CoT不同，该方法不将推理步骤作为示例纳入提示。再提示（Re-Prompting）[47]涉及在生成计划前检查每一步是否满足必要前提，若不满足则引入前提错误信息并提示LLM重新生成计划。ReWOO[48]引入了将计划与外部观察分离的范式，代理先独立生成计划和获取观察，再将两者结合得出最终结果。HuggingGPT[13]首先将任务分解为多个子目标，然后基于Huggingface逐一解决。不同于一次性输出所有推理步骤的CoT和Zero-shot-CoT，ReWOO和HuggingGPT通过多次调用LLMs生成结果。SWIFTSAGE[49]受人类认知双过程理论[50]启发，结合SWIFT和SAGE模块的优势用于复杂交互任务的规划。SWIFT模块基于学习模式快速响应，SAGE模块利用大型语言模型通过提出关键问题和生成动作序列进行深入规划，确保任务成功完成。

- Multi-path Reasoning. In this strategy, the reasoning steps for generating the final plans are organized into a tree-like structure. Each intermediate step may have multiple subsequent steps. This approach is analogous to human thinking, as individuals may have multiple choices at each reasoning step. In specific, Self-consistent CoT (CoT-SC) [51] believes that each complex problem has multiple ways of thinking to deduce the final answer. Thus, it starts by employing CoT to generate various reasoning paths and corresponding answers. Subsequently, the answer with the highest frequency is chosen as the final output. Tree of Thoughts (ToT) [52] is designed to generate plans using a tree-like reasoning structure. In this approach, each node in the tree represents a "thought," which corresponds to an intermediate reasoning step. The selection of these intermediate steps is based on the evaluation of LLMs. The final plan is generated using either the breadth-first search (BFS) or depth-first search (DFS) strategy. Comparing with CoT-SC, which generates all the planed steps together, ToT needs to query LLMs for each reasoning step. In RecMind [53], the authors designed a self-inspiring mechanism, where the discarded historical information in the planning process is also leveraged to derive new reasoning steps. In GoT [54], the authors expand the tree-like reasoning structure in ToT to graph structures, resulting in more powerful prompting strategies. In AoT [55], the authors design a novel method to enhance the reasoning processes of LLMs by incorporating algorithmic examples into the prompts. Remarkably, this method only needs to query LLMs for only one or a few times. In [44], the LLMs are leveraged as zero-shot planners. At each planning step, they first generate multiple possible next steps, and then determine the final one based on their distances to admissible actions. [56] further improves [44] by incorporating examples that are similar to the queries in the prompts. RAP [57] builds a world model to simulate the potential benefits of different plans based on Monte Carlo Tree Search (MCTS), and then, the final plan is generated by aggregating multiple MCTS iterations. To enhance comprehension, we provide an illustration comparing the strategies of single-path and multi-path reasoning in Figure 3.
- 多路径推理。在该策略中，生成最终计划的推理步骤组织成树状结构，每个中间步骤可能有多个后续步骤。此方法类似于人类思维，因为个体在每个推理步骤可能有多种选择。具体而言，自洽思维链（Self-consistent CoT, CoT-SC）[51]认为每个复杂问题存在多种思考路径以推导最终答案，因此先用CoT生成多条推理路径及对应答案，随后选择出现频率最高的答案作为最终输出。思维树（Tree of Thoughts, ToT）[52]设计用于使用树状推理结构生成计划，其中树中每个节点代表一个“思考”，对应一个中间推理步骤，这些步骤的选择基于LLMs的评估。最终计划通过广度优先搜索（BFS）或深度优先搜索（DFS）策略生成。与一次性生成所有计划步骤的CoT-SC不同，ToT需对每个推理步骤调用LLMs。在RecMind[53]中，作者设计了自我激励机制，利用规划过程中被舍弃的历史信息推导新推理步骤。GoT[54]将ToT中的树状推理结构扩展为图结构，带来更强的提示策略。AoT[55]设计了一种新方法，通过在提示中加入算法示例增强LLMs的推理过程，且仅需调用LLMs一至数次。[44]中将LLMs用作零样本规划器，每个规划步骤先生成多个可能的下一步，再根据与可接受动作的距离确定最终步骤。[56]通过在提示中加入与查询相似的示例进一步改进了[44]。RAP[57]基于蒙特卡洛树搜索（MCTS）构

建世界模型，模拟不同计划的潜在收益，最终通过多次MCTS迭代聚合生成计划。为增强理解，我们在图3中提供了单路径与多路径推理策略的对比示意。

- External Planner. Despite the demonstrated power of LLMs in zero-shot planning, effectively generating plans for domain-specific problems remains highly challenging. To address this challenge, researchers turn to external planners. These tools are well-developed and employ efficient search algorithms to rapidly identify correct, or even optimal, plans. In specific, LLM+P [58] first transforms the task descriptions into formal Planning Domain Definition Languages (PDDL), and then it uses an external planner to deal with the PDDL. Finally, the generated results are transformed back into natural language by LLMs. Similarly, LLM-DP [59] utilizes LLMs to convert the observations, the current world state, and the target objectives into PDDL. Subsequently, this transformed data is passed to an external planner, which efficiently determines the final action sequence. CO-LLM [22] demonstrates that LLMs is good at generating high-level plans, but struggle with low-level control. To address this limitation, a heuristically designed external low-level planner is employed to effectively execute actions based on high-level plans.
- 外部规划器。尽管大型语言模型（LLMs）在零样本规划中展现了强大能力，但有效生成特定领域问题的规划仍然极具挑战性。为应对这一挑战，研究人员转向外部规划器。这些工具成熟且采用高效的搜索算法，能够快速识别正确甚至最优的规划。具体而言，LLM+P [58] 首先将任务描述转换为形式化的规划领域定义语言（PDDL），然后使用外部规划器处理PDDL，最终由LLMs将生成结果转换回自然语言。类似地，LLM-DP [59] 利用LLMs将观察结果、当前世界状态和目标转换为PDDL，随后将转换后的数据传递给外部规划器，高效确定最终动作序列。CO-LLM [22] 证明LLMs擅长生成高层次规划，但在低层次控制上存在困难。为解决此限制，采用启发式设计的外部低层规划器，根据高层规划有效执行动作。

Planning with Feedback: In many real-world scenarios, the agents need to make long-horizon planning to solve complex tasks. When facing these tasks, the above planning modules without feedback can be less effective due to the following reasons: firstly, generating a flawless plan directly from the beginning is extremely difficult as it needs to consider various complex preconditions. As a result, simply following the initial plan often leads to failure. Moreover, the execution of the plan may be hindered by unpredictable transition dynamics, rendering the initial plan non-executable. Simultaneously, when examining how humans tackle complex tasks, we find that individuals may iteratively make and revise their plans based on external feedback. To simulate such human capability, researchers have designed many planning modules, where the agent can receive feedback after taking actions. The feedback can be obtained from environments, humans, and models, which are detailed in the following.
带反馈的规划：在许多现实场景中，智能体需要进行长远规划以解决复杂任务。面对这些任务，上述无反馈的规划模块可能效果不佳，原因包括：首先，直接从一开始生成完美规划极其困难，因为需考虑多种复杂前提条件，导致单纯遵循初始规划常常失败。此外，规划执行可能受不可预测的转移动态影响，使初始规划无法执行。同时，观察人类解决复杂任务的方式发现，个体会基于外部反馈反复制定和修正规划。为模拟这种人类能力，研究者设计了多种规划模块，使智能体在执行动作后能接收反馈。反馈来源包括环境、人类和模型，具体如下。

- Environmental Feedback. This feedback is obtained from the objective world or virtual environment. For instance, it could be the game's task completion signals or the observations made after the agent takes an action. In specific, ReAct [60] proposes constructing prompts using thought-act-observation triplets. The thought component aims to facilitate high-level reasoning and planning for guiding agent behaviors. The act represents a specific action taken by the agent. The observation corresponds to the outcome of the action, acquired through external feedback, such as search engine results. The next thought is influenced by the previous observations, which makes the generated plans more adaptive to the environment. Voyager [38] makes plans by incorporating three types of environment feedback including the intermediate progress of program execution, the execution error and self-verification results. These signals can help the agent to make better plans for the next action. Similar to Voyager, Ghost [16] also incorporates feedback into the reasoning and action taking processes. This feedback encompasses the environment states as well as the success and failure information for each executed action. SayPlan [31] leverages environmental feedback derived from a scene graph simulator to validate and refine its strategic formulations. This simulator is adept at discerning the

outcomes and state transitions subsequent to agent actions, facilitating SayPlan's iterative recalibration of its strategies until a viable plan is ascertained. In DEPS [33], the authors argue that solely providing information about the completion of a task is often inadequate for correcting planning errors. Therefore, they propose informing the agent about the detail reasons for task failure, allowing them to more effectively revise their plans. LLM-Planner [61] introduces a grounded re-planning algorithm that dynamically updates plans generated by LLMs when encountering object mismatches and unattainable plans during task completion. Inner Monologue [62] provides three types of feedback to the agent after it takes actions: (1) whether the task is successfully completed, (2) passive scene descriptions, and (3) active scene descriptions. The former two are generated from the environments, which makes the agent actions more reasonable.

- 环境反馈。此类反馈来自客观世界或虚拟环境。例如，游戏中的任务完成信号或智能体执行动作后的观察结果。具体而言，ReAct [60] 提出使用思考-行动-观察三元组构建提示。思考部分促进高层推理和规划以指导智能体行为；行动代表智能体采取的具体动作；观察对应动作结果，通过外部反馈获得，如搜索引擎结果。下一步思考受前次观察影响，使生成的规划更适应环境。Voyager [38] 结合程序执行的中间进展、执行错误和自我验证结果三种环境反馈制定规划，这些信号帮助智能体为下一步动作制定更优规划。与Voyager类似，Ghost [16] 也将反馈融入推理和行动过程中，反馈涵盖环境状态及每个执行动作的成功与失败信息。SayPlan [31] 利用场景图模拟器提供的环境反馈验证并优化其策略制定，该模拟器能识别智能体动作后的结果和状态转移，助力SayPlan反复调整策略直至确定可行规划。在DEPS [33] 中，作者认为仅提供任务完成信息往往不足以纠正规划错误，因而提出向智能体告知任务失败的详细原因，以便更有效地修正规划。LLM-Planner [61] 引入基于实地的重新规划算法，在任务执行中遇到对象不匹配和不可达规划时动态更新LLMs生成的规划。Inner Monologue [62] 在智能体执行动作后提供三类反馈：（1）任务是否成功完成，（2）被动场景描述，（3）主动场景描述。前两者由环境生成，使智能体动作更合理。

- Human Feedback. In addition to obtaining feedback from the environment, directly interacting with humans is also a very intuitive strategy to enhance the agent planning capability. The human feedback is a subjective signal. It can effectively make the agent align with the human values and preferences, and also help to alleviate the hallucination problem. In Inner Monologue [62], the agent aims to perform high-level natural language instructions in a 3D visual environment. It is given the capability to actively solicit feedback from humans regarding scene descriptions. Then, the agent incorporates the human feedback into its prompts, enabling more informed planning and reasoning. In the above cases, we can see, different types of feedback can be combined to enhance the agent planning capability. For example, Inner Monologue [62] collects both environment and human feedback to facilitate the agent plans.
- 人类反馈。除了从环境获取反馈，直接与人类互动也是提升智能体规划能力的直观策略。人类反馈是一种主观信号，能有效使智能体与人类价值观和偏好保持一致，同时有助于缓解幻觉问题。在Inner Monologue [62] 中，智能体旨在3D视觉环境中执行高层自然语言指令，具备主动向人类请求场景描述反馈的能力，随后将人类反馈融入提示中，实现更有依据的规划与推理。上述案例表明，不同类型的反馈可结合使用以增强智能体规划能力。例如，Inner Monologue [62] 同时收集环境和人类反馈以辅助智能体规划。

- Model Feedback. Apart from the aforementioned environmental and human feedback, which are external signals, researchers have also investigated the utilization of internal feedback from the agents themselves. This type of feedback is usually generated based on pre-trained models. In specific, [63] proposes a self-refine mechanism. This mechanism consists of three crucial components: output, feedback, and refinement. Firstly, the agent generates an output. Then, it utilizes LLMs to provide feedback on the output and offer guidance on how to refine it. At last, the output is improved by the feedback and refinement. This output-feedback-refinement process iterates until reaching some desired conditions. SelfCheck [64] allows agents to examine and evaluate their reasoning steps generated at various stages. They can then correct any errors by comparing the outcomes. InterAct [65] uses different language models (such as ChatGPT and InstructGPT) as auxiliary roles, such as checkers and sorters, to help the main language model avoid erroneous and inefficient actions. ChatCoT [66] utilizes model feedback to improve the quality of its reasoning process. The model feedback is generated by an evaluation module that monitors the agent reasoning steps. Reflexion [12] is developed to enhance the agent's planning capability through detailed verbal feedback. In this model, the agent first

produces an action based on its memory, and then, the evaluator generates feedback by taking the agent trajectory as input. In contrast to previous studies, where the feedback is given as a scalar value, this model leverages LLMs to provide more detailed verbal feedback, which can provide more comprehensive supports for the agent plans.

- 模型反馈。除了前述的环境反馈和人类反馈这类外部信号外，研究者们还探讨了利用智能体自身的内部反馈。这类反馈通常基于预训练模型生成。具体而言，[63]提出了一种自我优化机制。该机制包含三个关键组成部分：输出、反馈和优化。首先，智能体生成输出。然后，利用大型语言模型（LLMs）对输出进行反馈并提供优化指导。最后，通过反馈和优化改进输出。该输出-反馈-优化过程反复迭代，直至达到预期条件。SelfCheck [64]允许智能体检查并评估其在各阶段生成的推理步骤，进而通过比较结果纠正错误。InterAct [65]使用不同的语言模型（如ChatGPT和InstructGPT）作为辅助角色，如检查者和排序者，帮助主语言模型避免错误和低效行为。ChatCoT [66]利用模型反馈提升推理过程质量。模型反馈由一个评估模块生成，该模块监控智能体的推理步骤。Reflexion [12]旨在通过详细的口头反馈增强智能体的规划能力。在该模型中，智能体首先基于记忆生成动作，随后评估者以智能体轨迹为输入生成反馈。与以往以标量值形式提供反馈的研究不同，该模型利用大型语言模型提供更详尽的口头反馈，为智能体规划提供更全面的支持。

Remark. In conclusion, the implementation of planning module without feedback is relatively straightforward. However, it is primarily suitable for simple tasks that only require a small number of reasoning steps. Conversely, the strategy of planning with feedback needs more careful designs to handle the feedback. Nevertheless, it is considerably more powerful and capable of effectively addressing complex tasks that involve long-range reasoning.
备注。总之，不带反馈的规划模块实现相对简单，但主要适用于仅需少量推理步骤的简单任务。相反，带反馈的规划策略需要更为细致的设计以处理反馈，然而其功能更强大，能够有效应对涉及长程推理的复杂任务。

4.2.7 2.1.4 Action Module

4.2.8 2.1.4 行动模块

The action module is responsible for translating the agent's decisions into specific outcomes. This module is located at the most downstream position and directly interacts with the environment. It is influenced by the profile, memory, and planning modules. This section introduces the action module from four perspectives: (1) Action goal: what are the intended outcomes of the actions? (2) Action production: how are the actions generated? (3) Action space: what are the available actions? (4) Action impact: what are the consequences of the actions? Among these perspectives, the first two focus on the aspects preceding the action ("before-action" aspects), the third focuses on the action itself ("in-action" aspect), and the fourth emphasizes the impact of the actions ("after-action" aspect).
行动模块负责将智能体的决策转化为具体结果。该模块位于最下游位置，直接与环境交互，受角色设定、记忆和规划模块影响。本节从四个角度介绍行动模块：（1）行动目标：行动预期达成的结果是什么？（2）行动生成：行动如何产生？（3）行动空间：可用的行动有哪些？（4）行动影响：行动带来哪些后果？其中，前两个角度关注行动之前的方面（“行动前”），第三个关注行动本身（“行动中”），第四个强调行动的影响（“行动后”）。

Action Goal: The agent can perform actions with various objectives. Here, we present several representative examples: (1) Task Completion. In this scenario, the agent's actions are aimed at accomplishing specific tasks, such as crafting an iron pickaxe in Minecraft [38] or completing a function in software development [18]. These actions usually have well-defined objectives, and each action contributes to the completion of the final task. Actions aimed at this type of goal are very common in existing literature. (2) Communication. In this case, the actions are taken to communicate with the other agents or real humans for sharing information or collaboration. For example, the agents in ChatDev [18] may communicate with each other to collectively accomplish software development tasks. In Inner Monologue [62], the agent actively engages in communication with humans and adjusts its action strategies based on human feedback. (3) Environment Exploration. In this example, the agent aims to explore unfamiliar environments to expand its perception and strike a balance between exploring and exploiting. For instance, the agent in Voyager [38] may explore unknown skills in their task completion process and continually refine the skill execution code based on environment feedback through trial and error.
行动目标：智能体可执行多种目标的行动。以下列举几个代表性示例：（1）任务完成。在此场景中，智能体的行动

旨在完成特定任务，如Minecraft [38]中制作铁镐，或软件开发中的函数实现 [18]。此类行动通常目标明确，每个行动均助力最终任务完成。文献中此类目标的行动非常常见。（2）交流。在此情况下，行动用于与其他智能体或真实人类沟通，以共享信息或协作。例如，ChatDev [18]中的智能体可能相互交流，共同完成软件开发任务。Inner Monologue [62]中，智能体主动与人类交流，并根据人类反馈调整行动策略。（3）环境探索。在此示例中，智能体旨在探索未知环境，扩展感知能力，并在探索与利用之间取得平衡。例如，Voyager [38]中的智能体在任务完成过程中探索未知技能，并通过试错不断根据环境反馈优化技能执行代码。

Action Production: Different from ordinary LLMs. where the model input and output are directly associated, the agent may take actions via different strategies and sources. In the following, we introduce two types of commonly used action production strategies. (1) Action via Memory Recollection. In this strategy, the action is generated by extracting information from the agent memory according to the current task. The task and the extracted memories are used as prompts to trigger the agent actions. For example, in Generative Agents [20], the agent maintains a memory stream, and before taking each action, it retrieves recent, relevant and important information from the memory steam to guide the agent actions. In GITM [16], in order to achieve a low-level sub-goal, the agent queries its memory to determine if there are any successful experiences related to the task. If similar tasks have been completed previously, the agent invokes the previously successful actions to handle the current task directly. In collaborative agents such as ChatDev [18] and MetaGPT [23], different agents may communicate with each other. In this process, the conversation history in a dialog is remembered in the agent memories. Each utterance generated by the agent is influenced by its memory. (2) Action via Plan Following. In this strategy, the agent takes actions following its pre-generated plans. For instance, in DEPS [33], for a given task, the agent first makes action plans. If there are no signals indicating plan failure, the agent will strictly adhere to these plans. In GITM [16], the agent makes high-level plans by decomposing the task into many sub-goals. Based on these plans, the agent takes actions to solve each sub-goal sequentially to complete the final task.
动作生成：不同于普通的大型语言模型（LLMs），其输入和输出直接关联，智能体可能通过不同的策略和来源采取动作。以下介绍两种常用的动作生成策略。(1) 通过记忆回忆生成动作。在此策略中，动作是通过根据当前任务从智能体记忆中提取信息生成的。任务和提取的记忆作为提示触发智能体动作。例如，在生成型智能体（Generative Agents）[20]中，智能体维护一个记忆流，在每次采取动作前，从记忆流中检索最近的、相关且重要的信息以指导动作。在GITM[16]中，为实现低级子目标，智能体查询其记忆以确定是否存在与任务相关的成功经验。如果之前完成过类似任务，智能体会调用先前成功的动作直接处理当前任务。在协作智能体如ChatDev[18]和MetaGPT[23]中，不同智能体可能相互通信。在此过程中，对话历史被记忆在智能体记忆中，智能体生成的每句话都受其记忆影响。(2) 通过计划执行动作。在此策略中，智能体按照预先生成的计划采取动作。例如，在DEPS[33]中，对于给定任务，智能体首先制定行动计划。如果没有计划失败的信号，智能体将严格遵循这些计划。在GITM[16]中，智能体通过将任务分解为多个子目标制定高层计划。基于这些计划，智能体依次采取行动解决每个子目标以完成最终任务。

Action Space: Action space refers to the set of possible actions that can be performed by the agent. In general, we can roughly divide these actions into two classes: (1) external tools and (2) internal knowledge of the LLMs. In the following, we introduce these actions more in detail.
动作空间：动作空间指智能体可执行的所有可能动作的集合。一般而言，我们可以将这些动作大致分为两类：(1) 外部工具和(2) 大型语言模型（LLMs）的内部知识。以下将更详细介绍这些动作。

- External Tools. While LLMs have been demonstrated to be effective in accomplishing a large amount of tasks, they may not work well for the domains which need comprehensive expert knowledge. In addition, LLMs may also encounter hallucination problems, which are hard to be resolved by themselves. To alleviate the above problems, the agents are empowered with the capability to call external tools for executing action. In the following, we present several representative tools which have been exploited in the literature.
- 外部工具。虽然大型语言模型已被证明能有效完成大量任务，但在需要综合专家知识的领域可能表现不佳。此外，LLMs还可能遇到难以自行解决的幻觉问题。为缓解上述问题，智能体被赋予调用外部工具执行动作的能力。以下介绍文献中使用的几种代表性工具。

(1) APIs. Leveraging external APIs to complement and expand action space is a popular paradigm in recent years. For example, HuggingGPT [13] integrates HuggingFace's vast model ecosystem to tackle complex user tasks. Similarly, WebGPT [67] proposes to automatically generate queries to extract relevant content from external web pages when responding to user request. TPTU [68] explores the potential of LLMs to address intricate tasks through strategic task planning and API-based tools. Gorilla [69] introduces a fine-tuned LLM capable of generating precise input arguments for API calls, effectively mitigating hallucination issues during external API usage. ToolFormer [15] employs self-supervised learning to determine when and how to invoke external tools, using demonstrations of tool APIs for training. API-Bank [70] offers a comprehensive benchmark with a diverse collection of API tools to systematically evaluate tool-augmented LLMs, alongside robust training datasets designed to enhance their integration capabilities. ToolL-LaMA [14] proposes a tool-use framework encompassing data collection, training, and evaluation, with the resulting fine-tuned model excelling across a wide array of APIs. RestGPT [71] connects LLMs with RESTful APIs, which follow widely accepted standards for web services development, making the resulting program more compatible with real-world applications. TaskMatrix.AI [72] connects LLMs with an extensive ecosystem of APIs to support task execution. At its core lies a multimodal conversational foundational model that interacts with users, understands their goals and context, and then produces executable code for particular tasks. In essence, these intelligent agents strategically harness external APIs as versatile tools, systematically expanding their action space and transcending the inherent limitations of traditional language models by integrating diverse computational capabilities.
(1) API。利用外部API补充和扩展动作空间是近年来的流行范式。例如，HuggingGPT[13]整合了HuggingFace庞大的模型生态系统以应对复杂用户任务。类似地，WebGPT[67]提出自动生成查询以从外部网页提取相关内容以响应用户请求。TPTU[68]探索通过战略性任务规划和基于API的工具利用LLMs解决复杂任务的潜力。Gorilla[69]引入了一个经过微调的LLM，能够生成精确的API调用输入参数，有效缓解外部API使用中的幻觉问题。ToolFormer[15]采用自监督学习确定何时及如何调用外部工具，利用工具API示例进行训练。API-Bank[70]提供了一个包含多样API工具的综合基准，用于系统评估工具增强型LLMs，并配备了旨在提升集成能力的强大训练数据集。ToolL-LaMA[14]提出了涵盖数据收集、训练和评估的工具使用框架，微调后的模型在多种API上表现优异。RestGPT[71]将LLMs与遵循广泛接受的Web服务开发标准的RESTful API连接，使生成的程序更适合实际应用。TaskMatrix.AI[72]将LLMs与庞大的API生态系统连接以支持任务执行，其核心是一个多模态对话基础模型，能够与用户交互，理解其目标和上下文，然后生成特定任务的可执行代码。本质上，这些智能体策略性地利用外部API作为多功能工具，系统性地扩展其动作空间，突破传统语言模型的固有限制，整合多样的计算能力。

(2) Databases $\mathcal{E}$ Knowledge Bases. Integrating external database or knowledge base enables agents to obtain specific domain information for generating more realistic actions. For example, ChatDB [40] employs SQL statements to query databases, facilitating actions by the agents in a logical manner. MRKL [73] and OpenAGI [74] incorporate various expert systems such as knowledge bases and planners to access domain-specific information.
(2) 数据库$\mathcal{E}$知识库。集成外部数据库或知识库使智能体能够获取特定领域信息，从而生成更具现实性的动作。例如，ChatDB[40]使用SQL语句查询数据库，促进智能体以逻辑方式执行动作。MRKL[73]和OpenAGI[74]则整合了多种专家系统，如知识库和规划器，以访问领域特定信息。

(3) External Models. Previous studies often utilize external models to expand the range of possible actions. In comparison to APIs, external models typically handle more complex tasks. Each external model may correspond to multiple APIs. For example, ViperGPT [75] firstly uses Codex, which is implemented based on language model, to generate Python code from text descriptions, and then executes the code to complete the given tasks. Chem-Crow [76] is an LLM-based chemical agent designed to perform tasks in organic synthesis, drug discovery, and material design. It utilizes seventeen expert-designed models to assist its operations. MM-REACT [77] integrates various external models, such as VideoBERT for video summarization, $\mathbf{X}$ -decoder for image generation,and SpeechBERT for audio processing, enhancing its capability in diverse multimodal scenarios.
(3) 外部模型。以往研究常利用外部模型来扩展可能的动作范围。相比API，外部模型通常处理更复杂的任务。每个外部模型可能对应多个API。例如，ViperGPT [75] 首先使用基于语言模型实现的Codex，从文本描述生成Python代码，然后执行代码以完成指定任务。Chem-Crow [76] 是一个基于大型语言模型（LLM）的化学代理，设计用于有机合成、药物发现和材料设计任务。它利用十七个专家设计的模型辅助操作。MM-REACT [77] 集成了多种外部模型，

如用于视频摘要的VideoBERT，X-解码器用于图像生成，以及用于音频处理的SpeechBERT，增强了其在多模态多样场景下的能力。

- Internal Knowledge. In addition to utilizing external tools, many agents rely solely on the internal knowledge of LLMs to guide their actions. We now present several crucial capabilities of LLMs that can support the agent to behave reasonably and effectively. (1) Planning Capability. Previous work has demonstrated that LLMs can be used as decent planners to decompose complex tasks into simpler ones [45]. Such a capability of LLMs can be even triggered without incorporating examples in the prompts [46]. Building on the planning capability of LLMs, DEPS [33] develops a Minecraft agent, which can solve complex tasks via sub-goal decomposition. Similar agents like GITM [16] and Voyager [38] also heavily rely on the planning capability of LLMs to successfully complete various tasks. (2) Conversation Capability. LLMs can usually generate high-quality conversations. This capability enables agents to behave more like humans. In the previous work, many agents take actions based on the strong conversation capability of LLMs. For example, in ChatDev [18], different agents can discuss the software development process and reflect on their own behaviors. In RLP [30], the agent can communicate with the listeners based on their potential feedback on the agent's utterance. (3) Common Sense Understanding Capability. Another important capability of LLMs is that they can well comprehend human common sense. Based on this capability, many agents can simulate human daily life and make human-like decisions. For example, in Generative Agent [20], the agent can accurately understand its current state, the surrounding environment, and summarize high-level ideas based on basic observations. Without the common sense understanding capability of LLMs, these behaviors cannot be reliably simulated. Similar conclusions may also apply to RecAgent [21] and S3 [78], where the agents focus on simulating user social behaviors.
- 内部知识。除了利用外部工具，许多代理仅依赖LLM的内部知识来指导其行为。现介绍几项关键的LLM能力，支持代理合理且高效地行动。(1) 规划能力。先前工作表明，LLM可作为优秀的规划者，将复杂任务分解为更简单的子任务[45]。这种能力甚至可在提示中不包含示例的情况下触发[46]。基于LLM的规划能力，DEPS [33] 开发了一个Minecraft代理，能通过子目标分解解决复杂任务。类似的代理如GITM [16] 和Voyager [38] 也高度依赖LLM的规划能力成功完成各类任务。(2) 对话能力。LLM通常能生成高质量对话，使代理行为更具人性化。许多先前工作中的代理基于LLM强大的对话能力采取行动。例如，在ChatDev [18] 中，不同代理可讨论软件开发流程并反思自身行为。在RLP [30] 中，代理能根据听众对其发言的潜在反馈进行交流。(3) 常识理解能力。LLM另一重要能力是良好理解人类常识。基于此，许多代理能模拟人类日常生活并做出类人决策。例如，在Generative Agent [20] 中，代理能准确理解当前状态、周围环境，并基于基本观察总结高级想法。若无LLM的常识理解能力，这些行为无法可靠模拟。类似结论也适用于RecAgent [21] 和S3 [78]，这些代理专注于模拟用户社交行为。

Action Impact: Action impact refers to the consequences of an agent's actions. While the range of possible impacts is vast, we highlight a few key examples for clarity: (1) Changing Environments. Agents can directly alter environment states by actions, such as moving their positions, collecting items, constructing buildings, etc. For instance, in GITM [16] and Voyager [38], the environments are changed by the actions of the agents in their task completion process. Specifically, when an agent collects resources-such as harvesting three pieces of wood-the resources disappear from the environment. (2) Altering Internal States. Actions taken by the agent can also change the agent itself, including updating memories, forming new plans, acquiring novel knowledge, and more. For example, in Generative Agents [20], memory streams are updated after performing actions within the system. Similarly, SayCan [79] enables agents to take actions to update understandings of the environment. (3) Triggering New Actions. In task completion processes, one action often leads to subsequent actions. For example, in Voyager [38], once the agent has gathered the necessary resources, it triggers the construction of buildings.
动作影响：动作影响指代理行为的后果。尽管可能的影响范围广泛，以下列举几个关键示例以示说明：(1) 改变环境。代理可通过动作直接改变环境状态，如移动位置、收集物品、建造建筑等。例如，在GITM [16] 和Voyager [38] 中，代理在完成任务过程中通过动作改变环境。具体来说，当代理收集资源——如采集三块木头——资源即从环境中消失。(2) 改变内部状态。代理的动作也可改变自身状态，包括更新记忆、制定新计划、获取新知识等。例如，在Generative Agents [20] 中，执行动作后记忆流会被更新。类似地，SayCan [79] 使代理能通过动作更新对环境的理

解。(3) 触发新动作。在任务完成过程中，一个动作常引发后续动作。例如，在Voyager [38] 中，一旦代理收集到必要资源，即触发建筑建造。

4.3 2.2 Agent Capability Acquisition

4.4 2.2 代理能力获取

In the sections above, we focus mainly on how to design the agent architecture to better harness the capabilities of LLMs to enabling them to accomplish tasks akin to human performance. The architecture functions as the "hardware" of an agent. However, relying solely on the hardware is insufficient for achieving effective task performance. This is because the agent may lack the necessary task-specific capabilities, skills, and experiences, which can be regarded as "software" resources. In order to equip the agent with these resources, various strategies have been devised. Generally, we categorize these strategies into two classes based on whether they require fine-tuning of the LLMs. Below, we introduce each category in detail.
上述章节主要关注如何设计代理架构，以更好地利用LLM的能力，使其完成类似人类的任务。架构相当于代理的“硬件”。然而，仅依赖硬件不足以实现有效的任务执行，因为代理可能缺乏必要的任务特定能力、技能和经验，这些可视为“软件”资源。为赋予代理这些资源，已设计多种策略。一般而言，我们根据是否需要对LLM进行微调，将这些策略分为两类。以下详细介绍每类策略。

Capability Acquisition with Fine-tuning: A direct approach to enhance agent capabilities for task completion is to fine-tune the model using task-specific datasets. These datasets can be constructed from human annotations, LLM-generated content, or real-world applications. We discuss these methods in detail below.
通过微调获取能力：提升代理完成任务能力的直接方法是使用任务特定数据集对模型进行微调。这些数据集可由人工标注、LLM生成内容或真实应用构建。以下详细讨论这些方法。

- Fine-tuning with Human Annotated Datasets. To fine-tune the agent, utilizing human annotated datasets is a versatile approach that can be employed in various application scenarios. In this approach, researchers first design annotation tasks and then recruit workers to complete them. For example, in CoH [85] ,the authors aim to align LLMs with human values and preferences. Different from the other models, where the human feedback is leveraged in a simple and symbolic manner, this method converts the human feedback into detailed comparison information in the form of natural languages. The LLMs are directly fine-tuned based on these natural language datasets. In RET-LLM [41], in order to better convert natural languages into structured memory information, the authors fine-tune LLMs based on a human constructed dataset, where each sample is a "triplet-natural language" pair. In WebShop [86], the authors collect 1.18 million real-world products from amazon.com, and put them onto a simulated e-commerce website, which contains several carefully designed human shopping scenarios. Based on this website, the authors recruit 13 workers to collect a real-human behavior dataset. At last, three methods based on heuristic rules, imitation learning and reinforcement learning are trained based on this dataset. Although the authors do not fine-tune LLM-based agents, we believe that the dataset proposed in this paper holds immense potential to enhance the capabilities of agents in the field of web shopping. In EduChat [87], the authors aim to enhance the educational functions of LLMs, such as open-domain question answering, essay assessment, Socratic teaching, and emotional support. They fine-tune LLMs based on human annotated datasets that cover various educational scenarios and tasks.
- 使用人工标注数据集进行微调。为了微调智能体，利用人工标注数据集是一种多场景适用的通用方法。在该方法中，研究人员首先设计标注任务，然后招募工作人员完成。例如，在CoH [85]中，作者旨在使大型语言模型（LLMs）与人类价值观和偏好保持一致。与其他模型中以简单符号方式利用人类反馈不同，该方法将人类反馈转换为以自然语言形式呈现的详细比较信息。LLMs直接基于这些自然语言数据集进行微调。在RET-LLM [41]中，为了更好地将自然语言转换为结构化记忆信息，作者基于人工构建的数据集对LLMs进行微调，其中每个样本是一个“三元组-自然语言”对。在WebShop [86]中，作者收集了来自amazon.com的118万个真实产品，并将其放置在一个模拟电商网站上，该网站包含多个精心设计的人类购物场景。基于该网站，作者招募了13名工作人员收集真实人类行为数据集。最后，基于该数据集训练了三种方法，分别基于启发式规则、模仿学习和强化学习。尽管作者未对基于LLM的智能体进行微调，但我们认为本文提出的数据集在提升电商购物领域智能体能力方面

具有巨大潜力。在EduChat [87]中，作者旨在增强LLMs的教育功能，如开放域问答、作文评估、苏格拉底式教学和情感支持。他们基于涵盖多种教育场景和任务的人工标注数据集对LLMs进行微调。

Table 1 For the profile module, we use ①, ② and ③ to represent the handcrafting method, LLM-generation method, and dataset alignment method, respectively. For the memory module, we focus on the implementation strategies for memory operation and memory structure. For memory operation, we use ① and ② to indicate that the model only has read/write operations and has read/write/reflection operations, respectively. For memory structure, we use ① and ② to represent unified and hybrid memories, respectively. For the planning module, we use ① and ② to represent planning w/o feedback and w/ feedback, respectively. For the action module, we use ① and ② to represent that the model does not use tools and use tools, respectively. For the agent capability acquisition (CA) strategy, we use ① and ② to represent the methods with and without fine-tuning, respectively. "-" indicates that the corresponding content is not explicitly discussed in the paper.
表1 对于个人资料模块，我们用①、②和③分别表示手工制作方法、LLM生成方法和数据集对齐方法。对于记忆模块，我们关注记忆操作和记忆结构的实现策略。对于记忆操作，我们用①和②表示模型仅具有读/写操作和具有读/写/反思操作。对于记忆结构，我们用①和②表示统一记忆和混合记忆。对于规划模块，我们用①和②表示无反馈规划和有反馈规划。对于动作模块，我们用①和②表示模型不使用工具和使用工具。对于智能体能力获取（CA）策略，我们用①和②表示有微调和无微调的方法。“-”表示论文中未明确讨论对应内容。

Model	Profile	Memory		Planning	Action	CA	Time
		Operation	Structure				
WebGPT [67]	-	-	-	-	②	①	12/2021
SayCan [79]	-	-	-	①	①	②	04/2022
MRKL [73]	-	-	-	①	②	-	05/2022
Inner Monologue [62]	-	-	-	②	①	②	07/2022
Social Simulacra [80]	②	-	-	-	①	-	08/2022
ReAct [60]	-	-	-	②	②	①	10/2022
MALLM [42]	-	①	②	-	①	-	01/2023
DEPS [33]	-	-	-	②	①	②	02/2023
Toolformer [15]	-	-	-	①	②	①	02/2023
Reflexion [12]	-	②	②	②	①	②	03/2023
CAMEL [81]	① ②	-	-	②	①	-	03/2023
API-Bank [70]	-	-	-	②	②	②	04/2023
ViperGPT [75]	-	-	-	-	②	-	03/2023
HuggingGPT [13]	-	①	①	①	②	-	03/2023
Generative Agents [20]	①	②	②	②	①	-	04/2023
LLM+P [58]	-	-	-	①	①	-	04/2023
ChemCrow [76]	-	-	-	②	②	-	04/2023
OpenAGI [74]	-	-	-	②	②	①	04/2023
AutoGPT [82]	-	①	②	②	②	②	04/2023
SCM [35]	-	②	②	-	①	-	04/2023
Socially Alignment [83]	-	①	②	-	①	①	05/2023
GITM [16]	-	②	②	②	①	②	05/2023
Voyager [38]	-	②	②	②	①	②	05/2023
Introspective Tips [84]	-	-	-	②	①	②	05/2023
RET-LLM [41]	-	①	②	-	①	①	05/2023
ChatDB [40]	-	①	②	②	②	-	06/2023
${S}^{3}$ [78]	③	②	②	-	①	-	07/2023
ChatDev [18]	①	②	②	②	①	②	07/2023
ToolLLM [14]	-	-	-	②	②	①	07/2023
MemoryBank [39]	-	②	②	-	①	-	07/2023
MetaGPT [23]	①	②	②	②	②	-	08/2023

模型	配置文件	记忆 操作	记忆 结构	规划	行动	CA	时间
WebGPT [67]	-	-	-	-	②	①	12/2021
SayCan [79]	-	-	-	①	①	②	04/2022
MRKL [73]	-	-	-	①	②	-	05/2022
内心独白 [62]	-	-	-	②	①	②	07/2022
社会拟像 [80]	②	-	-	-	①	-	08/2022
ReAct [60]	-	-	-	②	②	①	10/2022
MALLM [42]	-	①	②	-	①	-	01/2023
DEPS [33]	-	-	-	②	①	②	02/2023
Toolformer [15]	-	-	-	①	②	①	02/2023
反思 (Reflexion) [12]	-	②	②	②	①	②	03/2023
CAMEL [81]	① ②	-	-	②	①	-	03/2023
API-Bank [70]	-	-	-	②	②	②	04/2023
ViperGPT [75]	-	-	-	-	②	-	03/2023
HuggingGPT [13]	-	①	①	①	②	-	03/2023
生成代理 (Generative Agents) [20]	①	②	②	②	①	-	04/2023
LLM+P [58]	-	-	-	①	①	-	04/2023
ChemCrow [76]	-	-	-	②	②	-	04/2023
OpenAGI [74]	-	-	-	②	②	①	04/2023
AutoGPT [82]	-	①	②	②	②	②	04/2023
SCM [35]	-	②	②	-	①	-	04/2023
社会对齐 (Socially Alignment) [83]	-	①	②	-	①	①	05/2023
GITM [16]	-	②	②	②	①	②	05/2023
Voyager [38]	-	②	②	②	①	②	05/2023
内省提示 (Introspective Tips) [84]	-	-	-	②	①	②	05/2023
RET-LLM [41]	-	①	②	-	①	①	05/2023
ChatDB [40]	-	①	②	②	②	-	06/2023
${S}^{3}$ [78]	③	②	②	-	①	-	07/2023
ChatDev [18]	①	②	②	②	①	②	07/2023
ToolLLM [14]	-	-	-	②	②	①	07/2023
记忆库 (MemoryBank) [39]	-	②	②	-	①	-	07/2023
MetaGPT [23]	①	②	②	②	②	-	08/2023

- Fine-tuning with LLM Generated Datasets. Building human-annotated datasets typically requires recruiting people, which can be costly, especially when dealing with large-scale annotation tasks. Considering that LLMs can achieve human-like capabilities in a wide range of tasks, a natural idea is using LLMs to accomplish the annotation task. While the datasets produced from this method can be not as perfect as the human annotated ones, it is much cheaper, and can be leveraged to generate more samples. For example, in ToolBench [14], to enhance the tool-using capability of open-source LLMs, the authors collect 16,464 real-world APIs spanning 49 categories from the RapidAPI Hub. They used these APIs to prompt ChatGPT to generate diverse instructions, covering both single-tool and multi-tool scenarios. Based on the obtained dataset, the authors fine-tune LLaMA [9], and obtain significant performance improvement in terms of tool using. In [83], to empower the agent with social capability, the authors design a sandbox, and deploy multiple agents to interact with each other. Given a social question, the central agent first generates initial responses. Then, it shares the

responses to its nearby agents for collecting their feedback. Based on the feedback as well as its detailed explanations, the central agent revise its initial responses to make them more consistent with social norms. In this process, the authors collect a large amount of agent social interaction data, which is then leveraged to fine-tune the LLMs.

- 使用大语言模型生成的数据集进行微调。构建人工标注的数据集通常需要招募人员，这可能成本较高，尤其是在处理大规模标注任务时。考虑到大语言模型（LLM）在广泛任务中能达到类人能力，一个自然的想法是利用LLM完成标注任务。虽然这种方法生成的数据集可能不如人工标注的完美，但成本更低，且可用于生成更多样本。例如，在ToolBench [14]中，为了提升开源LLM的工具使用能力，作者从RapidAPI Hub收集了涵盖49个类别的16,464个真实API。他们利用这些API提示ChatGPT生成多样化指令，涵盖单工具和多工具场景。基于获得的数据集，作者对LLaMA [9]进行了微调，在工具使用方面取得了显著性能提升。在[83]中，为了赋能代理的社交能力，作者设计了一个沙盒环境，部署多个代理相互交互。面对社交问题，中心代理首先生成初始回应，然后将回应分享给附近代理收集反馈。基于反馈及详细解释，中心代理修正初始回应，使其更符合社会规范。在此过程中，作者收集了大量代理社交交互数据，进而用于微调LLM。

- Fine-tuning with Real-world Datasets. In addition to building datasets based on human or LLM annotations, directly using real-world datasets to fine-tune the agent is also a common strategy. For example, in MIND2WEB [88], the authors collect a large amount of real-world datasets to enhance the agent capability in the web domain. In contrast to prior studies, the dataset presented in this paper encompasses diverse tasks, real-world scenarios, and comprehensive user interaction patterns. Specifically, the authors collect over 2,000 open-ended tasks from 137 real-world websites spanning 31 domains. Using this dataset, the authors fine-tune LLMs to enhance their performance on web-related tasks such as movie discovery and ticket booking. Similarly, in SQL-PaLM [89], researchers fine-tune PaLM-2 using cross-domain, large-scale text-to-SQL datasets, including Spider and BIRD. The resulting model achieves notable performance improvements on text-to-SQL tasks, demonstrating the effectiveness of real-world datasets for domain-specific applications.
- 使用真实世界数据集进行微调。除了基于人工或LLM标注构建数据集外，直接使用真实世界数据集微调代理也是常见策略。例如，在MIND2WEB [88]中，作者收集了大量真实世界数据集以增强代理在网页领域的能力。与以往研究相比，本文所呈现的数据集涵盖多样任务、真实场景及全面的用户交互模式。具体而言，作者从137个真实网站（涵盖31个领域）收集了超过2,000个开放式任务。利用该数据集，作者微调LLM以提升其在电影发现和票务预订等网页相关任务上的表现。类似地，在SQL-PaLM [89]中，研究者使用跨领域大规模文本到SQL数据集（包括Spider和BIRD）微调PaLM-2。所得模型在文本到SQL任务上取得显著性能提升，展示了真实世界数据集在特定领域应用中的有效性。

Capability Acquisition without Fine-tuning: In the era of tradition machine learning, the model capability is mainly acquired by learning from datasets, where the knowledge is encoded into the model parameters. In the era of LLMs, the model capability can be acquired either by training/fine-tuning the model parameters or designing delicate prompts (i.e., prompt engineering). In prompt engineering, one needs to write valuable information into the prompts to enhance the model capability or unleash existing LLM capabilities. In the era of agents, the model capability can be acquired based on three strategies: (1) model fine-tuning, (2) prompt engineering and (3) designing proper agent evolution mechanisms (we called it as mechanism engineering). Mechanism engineering is a broad concept that involves developing specialized modules, introducing novel working rules, and other strategies to enhance agent capabilities. For clearly understanding the transitions of model capability acquisition strategies, we illustrate them in Figure 4. In the following, we detail prompting engineering and mechanism engineering.
无需微调即可获得能力：在传统机器学习时代，模型能力主要通过学习数据集获得，知识被编码进模型参数。在LLM时代，模型能力既可通过训练/微调模型参数获得，也可通过设计精巧的提示（即提示工程）实现。在提示工程中，需要将有价值的信息写入提示，以增强模型能力或释放现有LLM能力。在代理时代，模型能力可基于三种策略获得：（1）模型微调，（2）提示工程和（3）设计合适的代理进化机制（称为机制工程）。机制工程是一个广泛概念，涉及开发专门模块、引入新工作规则及其他策略以提升代理能力。为清晰理解模型能力获取策略的演变，我们在图4中进行了说明。以下内容将详细介绍提示工程和机制工程。

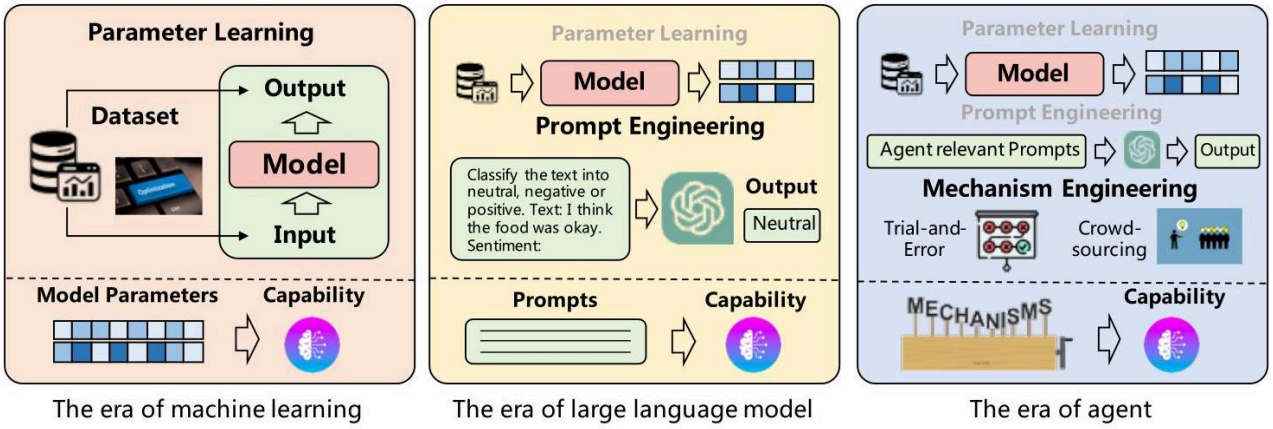

Fig. 4 Illustration of transitions in strategies for acquiring model capabilities.
图4 模型能力获取策略演变示意图。

- Prompting Engineering. Due to the strong language comprehension capabilities, people can directly interact with LLMs using natural languages. This introduces a novel strategy for enhancing agent capabilities, that is, one can describe the desired capability using natural language and then use it as prompts to influence LLM actions. For example, in CoT [45], in order to empower the agent with the capability for complex task reasoning, the authors present the intermediate reasoning steps as few-shot examples in the prompt. Similar techniques are also used in CoT-SC [51] and ToT [52]. In RLP [30], the authors aim to enhance an agent's self-awareness in conversations by prompting LLMs with the agent's beliefs about both its own and the listeners' mental states. This approach results in more engaging and adaptive utterances. Furthermore, the incorporation of the target mental states of listeners allows the agent to formulate more strategic plans. Retro-former [90] presents a retrospective model that enables the agent to generate reflections on its past failures. The reflections are integrated into the prompt of LLMs to guide the agent's future actions. Additionally, this model utilizes reinforcement learning to iteratively improve the retrospective model, thereby refining the LLM prompt.
- 提示工程。由于强大的语言理解能力，人们可以直接使用自然语言与LLM交互。这引入了一种提升代理能力的新策略，即用自然语言描述期望能力，然后将其作为提示影响LLM的行为。例如，在CoT [45]中，为赋能代理复杂任务推理能力，作者在提示中以少量示例呈现中间推理步骤。类似技术也应用于CoT-SC [51]和ToT [52]。在RLP [30]中，作者通过提示LLM代理关于自身及听众心理状态的信念，旨在增强代理在对话中的自我意识。该方法使发言更具吸引力和适应性。此外，纳入听众目标心理状态使代理能制定更具策略性的计划。Retro-former [90]提出了一种回顾模型，使代理能对过去失败进行反思。反思内容被整合进LLM提示，引导代理未来行动。该模型还利用强化学习迭代优化回顾模型，从而精炼LLM提示。

- Mechanism Engineering. Unlike model fine-tuning and prompt engineering, mechanism engineering is a unique strategy to enhance agent capability. In the following, we present several representative methods of mechanism engineering.
- 机制工程。不同于模型微调和提示工程，机制工程是一种独特的提升代理能力的策略。以下介绍几种具有代表性的机制工程方法。

(1) Trial-and-error. In this method, the agent first performs an action, and subsequently, a pre-defined critic is invoked to judge the action. If the action is deemed unsatisfactory, then the agent reacts by incorporating the critic's feedback. For example, in RAH [91], the agent serves as a user assistant in recommender systems. One of the agent's crucial roles is to simulate human behavior and generate responses on behalf of the user. To fulfill this objective, the agent first generates a predicted response and then compares it with the real human feedback. If the

predicted response and the real human feedback differ, the critic generates failure information, which is subsequently incorporated into the agent's next action. Similarly, in DEPS [33], the agent first designs a plan to accomplish a given task. In the plan execution process, if an action fails, the explainer generates specific details explaining the cause of the failure. This information is then incorporated by the agent to redesign the plan. In RoCo [92], the agent first proposes a sub-task plan and a path of 3D waypoints for each robot in a multi-robot collaboration task. The plan and waypoints are then validated by a set of environment checks, such as collision detection and inverse kinematics. If any of the checks fail, the feedback is appended to each agent's prompt and another round of dialog begins. The agents use LLMs to discuss and improve their plan and waypoints until they pass all validations. PREFER [93] extends this idea by leveraging LLMs to generate detailed feedback when the agent underperforms, enabling iterative refinement and performance improvement.
(1) 试错法。在此方法中，智能体首先执行一个动作，随后调用预定义的评判者对该动作进行评判。如果动作被认为不满意，智能体则根据评判者的反馈进行调整。例如，在RAH [91]中，智能体作为推荐系统中的用户助手，其关键角色之一是模拟人类行为并代表用户生成响应。为实现此目标，智能体首先生成预测响应，然后将其与真实的人类反馈进行比较。如果预测响应与真实反馈不符，评判者会生成失败信息，随后该信息被纳入智能体的下一次动作中。类似地，在DEPS [33]中，智能体首先设计一个计划以完成指定任务。在计划执行过程中，如果某个动作失败，解释者会生成具体细节说明失败原因。该信息随后被智能体采纳以重新设计计划。在RoCo [92]中，智能体首先为多机器人协作任务中的每个机器人提出子任务计划和一条3D路径航点。该计划和航点随后通过一系列环境检测验证，如碰撞检测和逆运动学。如果任何检测失败，反馈信息会被附加到每个智能体的提示中，开始新一轮对话。智能体利用大型语言模型（LLMs）讨论并改进其计划和航点，直至通过所有验证。PREFER [93]通过利用LLMs在智能体表现不佳时生成详细反馈，扩展了这一思路，实现迭代优化和性能提升。

(2) Crowd-sourcing. In [94], the authors design a debating mechanism that leverages the wisdom of crowds to enhance agent capabilities. To begin with, different agents provide separate responses to a given question. If their responses are not consistent, they will be prompted to incorporate the solutions from other agents and provide an updated response. This iterative process continues until reaching a final consensus answer. In this method, the capability of each agent is enhanced by understanding and incorporating the other agents' opinions.
(2) 众包法。在[94]中，作者设计了一种辩论机制，利用群体智慧提升智能体能力。首先，不同智能体针对同一问题提供各自的回答。如果回答不一致，智能体将被提示吸收其他智能体的解决方案并提供更新后的回答。该迭代过程持续进行，直至达成最终共识答案。在此方法中，每个智能体的能力通过理解并融合其他智能体的观点得到增强。

(3) Experience Accumulation. In GITM [16], the agent does not know how to solve a task in the beginning. Then, it makes explorations, and once it has successfully accomplished a task, the actions used in this task are stored into the agent memory. In the future, if the agent encounters a similar task, then the relevant memories are extracted to complete the current task. In this process, the improved agent capability comes from the specially designed memory accumulation and utilization mechanisms. Voyager [38] introduces a skill library, where executable codes for specific skills are refined through interactions with the environment, enabling efficient task execution over time. In AppAgent [95], the agent is designed to interact with apps in a manner akin to human users, learning through both autonomous exploration and observation of human demonstrations. Throughout this process, it constructs a knowledge base, which serves as a reference for performing intricate tasks across various applications on a mobile phone. In MemPrompt [96], the users are requested to provide feedback in natural language regarding the problem-solving intentions of the agent, and this feedback is stored in memory. When the agent encounters similar tasks, it attempts to retrieve related memories to generate more suitable responses.
(3) 经验积累。在GITM [16]中，智能体起初并不知道如何解决任务，随后进行探索，一旦成功完成任务，所用动作会被存入智能体记忆。未来遇到类似任务时，相关记忆会被提取以完成当前任务。该过程中，智能体能力的提升源于专门设计的记忆积累与利用机制。Voyager [38]引入了技能库，通过与环境的交互不断优化特定技能的可执行代码，实现任务执行效率的提升。AppAgent [95]设计智能体以类似人类用户的方式与应用程序交互，通过自主探索和观察人类示范进行学习，构建知识库，作为在手机上执行复杂任务的参考。在MemPrompt [96]中，用户被要求以自然语言提供关于智能体解决意图的反馈，该反馈被存储于记忆中。当智能体遇到类似任务时，会尝试检索相关记忆以生成更合适的响应。

(4) Self-driven Evolution. This method allows agents to autonomously improve through self-directed learning and feedback mechanisms. LMA3 [97] enables the agent to autonomously set goals for itself, and gradually improve its capability by exploring the environment and receiving feedback from a reward function. Following this mechanism, the agent can acquire knowledge and develop capabilities according to its own preferences. SALLM-MS [98] integrates advanced LLMs like GPT-4 into a multi-agent system, agents can adapt and perform complex tasks, showcasing advanced communication capabilities, thereby realizing self-driven evolution in their interactions with the environment. In CLMTWA [99], by using a large language model as a teacher and a weaker language model as a student, the teacher can generate and communicate natural language explanations to improve the student's reasoning skills via theory of mind. The teacher can also personalize its explanations for the student and intervene only when necessary, based on the expected utility of intervention. Meanwhile, NLSOM [100],leverages natural language collaboration between agents, dynamically adjusting roles, tasks, and relationships based on feedback to solve problems beyond the scope of a single agent.
(4) 自驱进化。此方法使智能体通过自我导向的学习和反馈机制自主提升能力。LMA3 [97]使智能体能够自主设定目标，并通过探索环境及从奖励函数获取反馈逐步提升能力。基于此机制，智能体可根据自身偏好获取知识并发展能力。SALLM-MS [98]将先进的大型语言模型如GPT-4集成入多智能体系统，智能体能够适应并执行复杂任务，展现出高级沟通能力，从而实现与环境交互中的自驱进化。在CLMTWA [99]中，利用大型语言模型作为教师，较弱的语言模型作为学生，教师通过心智理论生成并传达自然语言解释以提升学生的推理能力。教师还能根据干预的预期效用个性化解释，仅在必要时介入。与此同时，NLSOM [100]利用智能体间的自然语言协作，基于反馈动态调整角色、任务和关系，以解决单一智能体无法完成的问题。

Remark. Upon comparing the aforementioned strategies for agent capability acquisition, we can find that the fine-tuning method improves the agent capability by adjusting model parameters, which can incorporate a large amount of task-specific knowledge, but is only suitable for open-source LLMs. The method without fine-tuning usually enhances the agent capability based on delicate prompting strategies or mechanism engineering. They can be used for both open- and closed-source LLMs. However, due to the limitation of the input context window of LLMs, they cannot incorporate too much task information. In addition, the designing spaces of the prompts and mechanisms are extremely large, which makes it not easy to find optimal solutions.
备注。通过比较上述代理能力获取策略，我们发现微调方法通过调整模型参数提升代理能力，能够融合大量特定任务知识，但仅适用于开源大语言模型（LLMs）。非微调方法通常基于精细的提示策略或机制设计来增强代理能力，适用于开源和闭源LLMs。然而，由于LLMs输入上下文窗口的限制，无法融合过多任务信息。此外，提示和机制的设计空间极大，难以找到最优解。

In the above sections, we have detailed the construction of LLM-based agents, where we focus on two aspects including the architecture design and capability acquisition. We present the correspondence between existing work and the above taxonomy in Table 1. It should be noted that, for the sake of integrity, we have also incorporated several studies, which do not explicitly mention LLM-based agents but are highly related to this area.
在上述章节中，我们详细介绍了基于LLM的代理构建，重点关注架构设计和能力获取两个方面。我们在表1中展示了现有工作与上述分类法的对应关系。需要注意的是，为了完整性，我们还纳入了若干未明确提及基于LLM代理但与该领域高度相关的研究。

5 3 LLM-based Autonomous Agent Ap- plication

6 3 基于LLM的自主代理应用

Owing to the strong language comprehension, complex task reasoning, and common sense understanding capabilities, LLM-based autonomous agents have shown significant potential to influence multiple domains. This section provides a succinct summary of previous studies, categorizing them according to their applications in three distinct areas: social science, natural science, and engineering (see the left part of Figure 5 for a global overview).
凭借强大的语言理解、复杂任务推理和常识理解能力，基于LLM的自主代理在多个领域展现出显著潜力。本节简要总结了以往研究，按照其在社会科学、自然科学和工程三个不同领域的应用进行分类（详见图5左侧的整体概览）。

6.1 3.1 Social Science

6.2 3.1 社会科学

Social science is one of the branches of science, devoted to the study of societies and the relationships among individuals within those societies. LLM-based autonomous agents can promote this domain by leveraging their impressive human-like understanding, thinking and task solving capabilities. In the following, we discuss several key areas that can be affected by LLM-based autonomous agents.
社会科学是科学的一个分支，致力于研究社会及社会中个体间的关系。基于LLM的自主代理凭借其类人理解、思考和任务解决能力，能够推动该领域的发展。以下讨论了几个可能受基于LLM的自主代理影响的关键领域。

Psychology: For the domain of psychology, LLM-based agents can be leveraged for conducting simulation experiments, providing mental health support and so on [101-104]. For example, in [101], the authors assign LLMs with different profiles, and let them complete psychology experiments. From the results, the authors find that LLMs are capable of generating results that align with those from studies involving human participants. Additionally, it was observed that larger models tend to deliver more accurate simulation results compared to their smaller counterparts. An interesting discovery is that, in many experiments, models like ChatGPT and GPT-4 can provide too perfect estimates (called "hyper-accuracy distortion"), which may influence the downstream applications. In [103], the authors systematically analyze the effectiveness of LLM-based conversation agents for mental well-being support. They collect 120 posts from Reddit, and find that such agents can help users cope with anxieties, social isolation and depression on demand. At the same time, they also find that the agents may produce harmful contents sometimes.
心理学：在心理学领域，基于LLM的代理可用于进行模拟实验、提供心理健康支持等[101-104]。例如，在[101]中，作者赋予LLMs不同的角色设定，让其完成心理学实验。结果显示，LLMs能够生成与人类参与者研究结果相符的结果。此外，观察到较大模型相比小模型能提供更准确的模拟结果。有趣的是，在许多实验中，诸如ChatGPT和GPT-4等模型可能给出过于完美的估计（称为“超精确失真”），这可能影响后续应用。在[103]中，作者系统分析了基于LLM的对话代理在心理健康支持中的有效性。他们收集了Reddit上的120条帖子，发现此类代理能按需帮助用户应对焦虑、社交孤立和抑郁。同时，也发现代理有时可能产生有害内容。

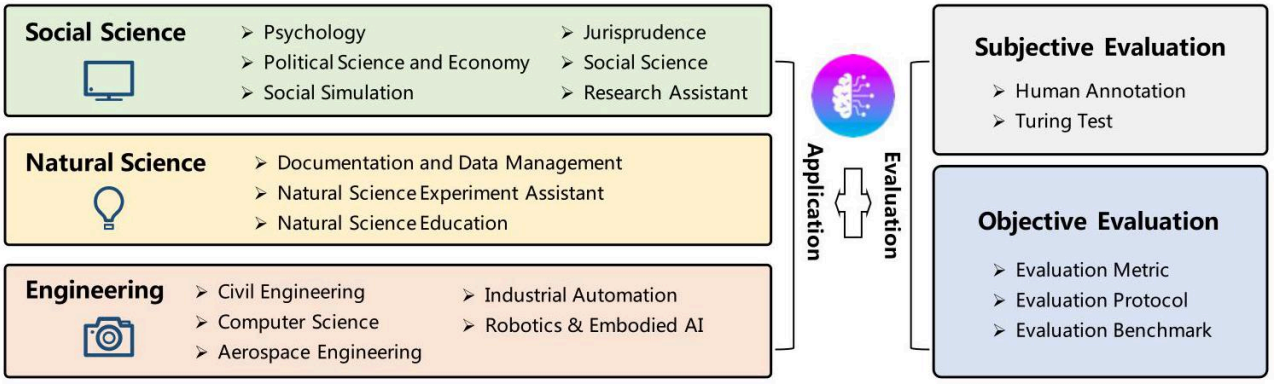

Fig. 5 The applications (left) and evaluation strategies (right) of LLM-based agents.
图5 基于LLM代理的应用（左）与评估策略（右）。

Political Science and Economy: LLM-based agents can also be utilized to study political science and economy [29, 104, 105]. In [29], LLM-based agents are utilized for ideology detection and predicting voting patterns. In [104], the authors focuses on understanding the discourse structure and persuasive elements of political speech through the assistance of LLM-based agents. In [105], LLM-based agents are provided with specific traits such as talents, preferences, and personalities to explore human economic behaviors in simulated scenarios.
政治学与经济学：基于LLM的代理也可用于研究政治学和经济学[29, 104, 105]。在[29]中，基于LLM的代理被用于意识形态检测和投票模式预测。在[104]中，作者借助基于LLM的代理，聚焦于理解政治演讲的话语结构和说服元素。在[105]中，基于LLM的代理被赋予特定特质，如才能、偏好和个性，以探索模拟场景中的人类经济行为。

Social Simulation: Previously, conducting experiments with human societies is often expensive, unethical, or even infeasible. With the ever prospering of LLMs, many people explore to build virtual environment with LLM-based agents to simulate social phenomena, such as the propagation of harmful information,and so on [20, 34, 78, 80 , 106-109]. For example, Social Simulacra [80] simulates an online social community and explores the potential of utilizing agent-based simulations to aid decision-makers to improve community regulations. [106, 107] investigates the potential impacts of different behavioral characteristics of LLM-based agents in social networks. Generative Agents [20] and AgentSims [34] construct multiple agents in a virtual town to simulate the human daily life. So-cialAI School [108] employs LLM-based agents to simulate and investigate the fundamental social cognitive skills during the course of child development. S^3 [78] builds a social network simulator,focusing on the propagation of information, emotion and attitude. CGMI [110] is a framework for multi-agent simulation. CGMI maintains the personality of the agents through a tree structure and constructs a cognitive model. The authors simulated a classroom scenario using CGMI.
社会模拟：以往，进行人类社会实验常因成本高昂、不道德甚至不可行而受限。随着LLMs的蓬勃发展，许多人尝试构建基于LLM代理的虚拟环境来模拟社会现象，如有害信息传播等[106-109]。例如，Social Simulacra[80]模拟了一个在线社交社区，探索利用基于代理的模拟帮助决策者改进社区管理的潜力。[b1]研究了基于LLM代理在社交网络中不同行为特征的潜在影响。Generative Agents[20]和AgentSims[34]在虚拟城镇中构建多个代理，模拟人类日常生活。SocialAI School[108]利用基于LLM的代理模拟并研究儿童发展过程中的基本社会认知技能。[b2][78]构建了一个社会网络模拟器，聚焦信息、情感和态度的传播。CGMI[110]是一个多代理模拟框架，通过树状结构维护代理个性并构建认知模型。作者利用CGMI模拟了课堂场景。

Jurisprudence: LLM-based agents can serve as aids in legal decision-making processes, facilitating more informed judgements [111, 112]. Blind Judgement [112] employs several language models to simulate the decision-making processes of multiple judges. It gathers diverse opinions and consolidates the outcomes through a voting mechanism. ChatLaw [111] is a prominent Chinese legal model based on LLM. It adeptly supports both database and keyword search strategies, specifically designed to mitigate the hallucination issue prevalent in such models. In addition, this model also employs self-attention mechanism to enhance the LLM's capability via mitigating the impact of reference inaccuracies.
法理学：基于大型语言模型（LLM）的智能体可以作为法律决策过程中的辅助工具，促进更为明智的判决[111, 112]。《盲判》（Blind Judgement）[112]采用多个语言模型模拟多位法官的决策过程，收集多样化意见并通过投票机制整合结果。ChatLaw[111]是一个基于LLM的知名中文法律模型，能够熟练支持数据库和关键词检索策略，专门设计以缓解此类模型中常见的幻觉问题。此外，该模型还采用自注意力机制，通过减轻参考信息不准确的影响，提升LLM的能力。

Research Assistant: Beyond their application in specialized domains, LLM-based agents are increasingly adopted as versatile assistants in the broad field of social science research [104, 113]. In [104], LLM-based agents offer multifaceted assistance, ranging from generating concise article abstracts and extracting pivotal keywords to crafting detailed scripts for studies, showcasing their ability to enrich and streamline the research process. Meanwhile, in [113], LLM-based agents serve as a writing assistant, demonstrating their capability to identify novel research inquiries for social scientists, thereby opening new avenues for exploration and innovation in the field. These examples highlight the potential of LLM-based agents in enhancing the efficiency, creativity, and breadth of social science research.

研究助理：除了在专业领域的应用外，基于LLM的智能体在社会科学研究领域作为多功能助手的应用日益广泛[104, 113]。[104]中，基于LLM的智能体提供多方面支持，从生成简明的文章摘要、提取关键关键词到撰写详细的研究脚本，展示了其丰富和简化研究流程的能力。同时，[113]中，基于LLM的智能体作为写作助手，展现了其识别社会科学家新颖研究问题的能力，从而为该领域开辟了新的探索和创新路径。这些例子凸显了基于LLM的智能体在提升社会科学研究效率、创造力和广度方面的潜力。

6.3 3.2 Natural Science

6.4 3.2 自然科学

Natural science is one of the branches of science concerned with the description, understanding and prediction of natural phenomena, based on empirical evidence from observation and experimentation. With the ever prospering of LLMs, the application of LLM-based agents in natural sciences becomes more and more popular. In the following, we present many representative areas, where LLM-based agents can play important roles.
自然科学是科学的一个分支，关注基于观察和实验的经验证据，对自然现象的描述、理解和预测。随着LLM的不断发展，基于LLM的智能体在自然科学中的应用日益普及。以下我们介绍多个代表性领域，展示基于LLM的智能体可以发挥重要作用的场景。

Documentation and Data Management: Natural scientific research often involves the collection, organization, and synthesis of substantial amounts of literature, which requires a significant dedication of time and human resources. LLM-based agents have shown strong capabilities on language understanding and employing tools such as the internet and databases for text processing. These capabilities empower the agent to excel in tasks related to documentation and data management. In [114], the agent can efficiently query and utilize internet information to complete tasks such as question answering and experiment planning. ChatMOF [115] utilizes LLMs to extract important information from text descriptions written by humans. It then formulates a plan to apply relevant tools for predicting the properties and structures of metal-organic frameworks. ChemCrow [76] utilizes chemistry-related databases to both validate the precision of compound representations and identify potentially dangerous substances. This functionality enhances the reliability and comprehensiveness of scientific inquiries by ensuring the accuracy of the data involved.
文献与数据管理：自然科学研究通常涉及大量文献的收集、整理和综合，这需要投入大量时间和人力资源。基于LLM的智能体在语言理解及利用互联网和数据库等工具进行文本处理方面表现出强大能力。这些能力使智能体在文献和数据管理相关任务中表现出色。[114]中，智能体能够高效查询和利用互联网信息，完成问答和实验规划等任务。ChatMOF[115]利用LLM从人工撰写的文本描述中提取重要信息，进而制定计划，应用相关工具预测金属有机框架（MOF）的性质和结构。ChemCrow[76]利用化学相关数据库验证化合物表示的准确性，并识别潜在危险物质。此功能通过确保数据的准确性，提升了科学研究的可靠性和全面性。

Experiment Assistant: LLM-based agents have the ability to independently conduct experiments, making them valuable tools for supporting scientists in their research projects [76,114]. For instance, [114] introduces an innovative agent system that utilizes LLMs for automating the design, planning, and execution of scientific experiments. This system, when provided with the experimental objectives as input, accesses the Internet and retrieves relevant documents to gather the necessary information. It subsequently utilizes Python code to conduct essential calculations and carry out the following experiments. ChemCrow [76] incorporates 17 carefully developed tools that are specifically designed to assist researchers in their chemical research. Once the input objectives are received, ChemCrow provides valuable recommendations for experimental procedures, while also emphasizing any potential safety risks associated with the proposed experiments.
实验助理：基于LLM的智能体具备独立开展实验的能力，是支持科学家研究项目的宝贵工具[76,114]。例如，[114]介绍了一种创新的智能体系统，利用LLM实现科学实验的自动设计、规划和执行。该系统在接收实验目标输入后，访问互联网并检索相关文献以收集必要信息，随后利用Python代码进行关键计算并执行后续实验。ChemCrow[76]集成了17个精心设计的工具，专门辅助化学研究人员。一旦接收输入目标，ChemCrow提供实验方案建议，同时强调拟议实验的潜在安全风险。

Natural Science Education: LLM-based agents can communicate with humans fluently, often being utilized to develop agent-based educational tools. For example, [114] develops agent-based education systems to facilitate students learning of experimental design, methodologies, and analysis. The objective of these systems is to enhance students' critical thinking and problem-solving skills, while also fostering a deeper comprehension of scientific principles. Math Agents [116] can assist researchers in exploring, discovering, solving and proving mathematical problems. Additionally, it can communicate with humans and aids them in understanding and using mathematics. [117] utilize the capabilities of CodeX [118] to automatically solve and explain university-level mathematical problems, which can be used as education tools to teach students and researchers. CodeHelp [119] is an education agent for programming. It offers many useful features, such as setting course-specific keywords, monitoring student queries, and providing feedback to the system. EduChat [87] is an LLM-based agent designed specifically for the education domain. It provides personalized, equitable, and empathetic educational support to teachers, students, and parents through dialogue. FreeText [120] is an agent that utilizes LLMs to automatically assess students' responses to open-ended questions and offer feedback.
自然科学教育：基于LLM的智能体能够流畅与人类交流，常被用于开发基于智能体的教育工具。例如，[114]开发了基于智能体的教育系统，促进学生学习实验设计、方法论和分析，旨在提升学生的批判性思维和解决问题能力，同时加深对科学原理的理解。Math Agents[116]可协助研究人员探索、发现、解决和证明数学问题，且能与人类交流，帮助其理解和应用数学。[117]利用CodeX[118]的能力自动解决并解释大学水平的数学问题，可作为教学工具辅助学生和研究人员。CodeHelp[119]是一个编程教育智能体，提供设定课程关键词、监控学生提问及反馈系统等多种实用功能。EduChat[87]是专为教育领域设计的基于LLM的智能体，通过对话为教师、学生和家长提供个性化、公平且富有同理心的教育支持。FreeText[120]利用LLM自动评估学生对开放性问题的回答并提供反馈。

6.5 3.3 Engineering

6.6 3.3 工程学

LLM-based autonomous agents have shown great potential in assisting and enhancing engineering research and applications. In this section, we review and summarize the applications of LLM-based agents in several major engineering domains.
基于LLM的自主智能体在辅助和提升工程研究与应用方面展现出巨大潜力。本节回顾并总结了基于LLM的智能体在若干主要工程领域的应用。

Computer Science & Software Engineering: In the field of computer science and software engineering, LLM-based agents offer potential for automating coding, testing, debugging, and documentation generation [18, 23, 24, 126-128]. Chat-Dev [18] proposes an end-to-end framework, where multiple agent roles communicate and collaborate through natural language conversations to complete the software development life cycle. This framework demonstrates efficient and cost-effective generation of executable software systems. MetaGPT [23] abstracts multiple roles, such as product managers, architects, project managers, and engineers, to supervise code generation process and enhance the quality of the final output code. This enables low-cost software development. [24] presents a self-collaboration framework for code generation using LLMs. In this framework, multiple LLMs are assumed to be distinct "experts" for specific subtasks. They collaborate and interact according to specified instructions, forming a virtual team that facilitates each other's work. Ultimately, the virtual team collaboratively addresses code generation tasks without requiring human intervention. LLIFT [141] employs LLMs to assist in conducting static analysis, specifically for identifying potential code vulnerabilities. This approach effectively manages the trade-off between accuracy and scalability. ChatEDA [123] is an agent developed for electronic design automation (EDA) to streamline the design process by integrating task planning, script generation, and execution. CodeHelp [119] is an agent designed to assist students and developers in debugging and testing their code. Its features include providing detailed explanations of error messages, suggesting potential fixes, and ensuring the accuracy of the code. Pentest [125] is a penetration testing tool based on LLMs, which can effectively identify common vulnerabilities, and interpret source code to develop exploits. D-Bot [122] utilizes the capabilities of LLMs to systematically assess potential root causes of anomalies in databases. Through the implementation of a tree of thought approach, D-Bot enables LLMs to backtrack to previous steps in case the current step proves unsuccessful, thus enhancing the

accuracy of the diagnosis process.

计算机科学与软件工程：在计算机科学与软件工程领域，基于大型语言模型（LLM）的智能体在自动化编码、测试、调试和文档生成方面展现出潜力[18, 23, 24, 126-128]。Chat-Dev[18]提出了一个端到端框架，其中多个智能体角色通过自然语言对话进行沟通与协作，以完成软件开发生命周期。该框架展示了高效且低成本的可执行软件系统生成能力。MetaGPT[23]抽象出产品经理、架构师、项目经理和工程师等多个角色，监督代码生成过程并提升最终输出代码的质量，从而实现低成本的软件开发。[24]提出了一个基于LLM的自我协作代码生成框架，在该框架中，多个LLM被视为针对特定子任务的不同“专家”，它们根据指定指令协作互动，形成一个虚拟团队，促进彼此工作，最终无需人工干预共同完成代码生成任务。LLIFT[141]利用LLM辅助进行静态分析，特别是识别潜在代码漏洞，有效平衡了准确性与可扩展性。ChatEDA[123]是一个面向电子设计自动化（EDA）的智能体，通过整合任务规划、脚本生成与执行，简化设计流程。CodeHelp[119]是一个旨在帮助学生和开发者调试及测试代码的智能体，功能包括详细解释错误信息、建议潜在修复方案并确保代码准确性。Pentest[125]是一款基于LLM的渗透测试工具，能够有效识别常见漏洞并解析源代码以开发利用手段。D-Bot[122]利用LLM能力系统性评估数据库异常的潜在根因，通过实现“思维树”方法，使LLM在当前步骤失败时能够回溯至前一步，从而提升诊断过程的准确性。

Table 2 Representative applications of LLM-based autonomous agents.

表2 基于大型语言模型的自主智能体代表性应用。

	Domain	Work
Social Science	Psychology	TE [101], Akata et al. [102], Ziems et al. [104], Ma et al. [103]
	Political Science and Economy	Argyle et al. [29], Horton [105], Ziems et al. [104]
	Social Simulation	Social Simulacra [80], Generative Agents [20], SocialAI School [108],AgentSims [34], ${\mathrm{S}}^{3}$ [78],Williams et al. [109],Li et al. [106], Chao et al. [107]
	Jurisprudence	ChatLaw [111], Blind Judgement [112]
Natural Science	Research Assistant	Ziems et al. [104], Bail et al. [113]
	Documentation and Data Management	ChemCrow [76], ChatMOF [115], Boiko et al. [114]
	Experiment Assistant	ChemCrow [76], Boiko et al. [114], Grossmann et al. [121]
	Natural Science Education	ChemCrow [76], CodeHelp [119], Boiko et al. [114], MathA- gent [116], Drori et al. [117], EduChat [87], FreeText [120]
Engineering	CS & SE	RestGPT [71], Self-collaboration [24], SQL-PALM [89], RAH [91], D-Bot [122], RecMind [53], ChatEDA [123], InteRecAgent [124], PentestGPT [125], CodeHelp [119], SmolModels [126], DemoGPT [127], GPTEngineer [128]
	Industrial Automation	GPT4IA [129], IELLM [130]
	Robotics & Embodied AI	ProAgent [131], LLM4RL [132], PET [133], REMEM- BERER [134], DEPS [33], Unified Agent [135], SayCan [79], TidyBot [136], RoCo [92], SayPlan [31], TaPA [137], Dasgupta et al. [138], DECKARD [139], Dialogue shaping [140]

	领域	工作
社会科学	心理学	TE [101], Akata 等人 [102], Ziems 等人 [104], Ma 等人 [103]
	政治学与经济学	Argyle 等人 [29], Horton [105], Ziems 等人 [104]
	社会模拟	社会拟像 (Social Simulacra) [80], 生成代理 (Generative Agents) [20], SocialAI 学校 [108], AgentSims [34], ${\mathrm{S}}^{3}$ [78], Williams 等人 [109], Li 等人 [106], Chao 等人 [107]
自然科学	法理学	ChatLaw [111], 盲判 (Blind Judgement) [112]
	研究助理	Ziems 等人 [104], Bail 等人 [113]
	文档与数据管理	ChemCrow [76], ChatMOF [115], Boiko 等人 [114]
	实验助理	ChemCrow [76], Boiko 等人 [114], Grossmann 等人 [121]
	自然科学教育	ChemCrow [76], CodeHelp [119], Boiko 等人 [114], MathAgent [116], Drori 等人 [117], EduChat [87], FreeText [120]
工程学	计算机科学与软件工程	RestGPT [71], 自我协作 (Self-collaboration) [24], SQL-PALM [89], RAH [91], D-Bot [122], RecMind [53], ChatEDA [123], InteRecAgent [124], PentestGPT [125], CodeHelp [119], SmolModels [126], DemoGPT [127], GPTEngineer [128]
	工业自动化	GPT4IA [129], IELLM [130]
	机器人学与具身人工智能	ProAgent [131], LLM4RL [132], PET [133], REMEMBERER [134], DEPS [33], 统一代理 (Unified Agent) [135], SayCan [79], TidyBot [136], RoCo [92], SayPlan [31], TaPA [137], Dasgupta 等人 [138], DECKARD [139], 对话塑造 (Dialogue shaping) [140]

Industrial Automation: In the field of industrial automation, LLM-based agents can be used to achieve intelligent planning and control of production processes. [129] proposes a novel framework that integrates LLMs with digital twin systems to accommodate flexible production needs. The framework leverages prompt engineering techniques to create LLM agents that can adapt to specific tasks based on the information provided by digital twins. These agents can coordinate a series of atomic functionalities and skills to complete production tasks at different levels. This research demonstrates the potential of integrating LLMs into industrial automation systems, providing innovative solutions for more agile, flexible and adaptive production processes. IELLM [130] showcases a case study on LLMs' role in the oil and gas industry, covering applications like factory automation and PLC programming.
工业自动化：在工业自动化领域，基于大型语言模型（LLM）的智能体可用于实现生产过程的智能规划与控制。[129]提出了一个将LLM与数字孪生系统集成的新框架，以满足灵活的生产需求。该框架利用提示工程技术创建能够根据数字孪生提供的信息适应特定任务的LLM智能体。这些智能体能够协调一系列原子功能和技能，在不同层级完成生产任务。该研究展示了将LLM集成到工业自动化系统中的潜力，为更敏捷、灵活和自适应的生产过程提供了创新解决方案。IELLM [130]展示了LLM在石油和天然气行业中的应用案例，涵盖工厂自动化和PLC编程等领域。

Robotics & Embodied Artificial Intelligence: Recent works have advanced the development of more efficient reinforcement learning agents for robotics and embodied artificial intelligence [16, $38, 79, 132-135, 137-140$] . These efforts focus on enhancing autonomous agents' capabilities in planning, reasoning, and collaboration within embodied environments. For instance, [138] proposes the Planner-Actor-Reporter paradigm for embodied reasoning and task planning. DECKARD [139] introduces the Planner-Actor-Reporter paradigm, which facilitates embodied reasoning and task planning by decoupling the agent's planning, execution, and reporting processes. TaPA [137] constructs a multimodal dataset comprising multi-view RGB images of indoor scenes, human instructions, and corresponding plans to fine-tune LLMs. The fine-tuned models align visual perception with task planning, enabling them to generate more executable plans and significantly improving their performance in visually grounded tasks.
机器人与具身人工智能：近期研究推动了更高效的强化学习智能体在机器人和具身人工智能领域的发展[16, $38, 79, 132-135, 137-140$]。这些工作聚焦于提升自主智能体在具身环境中的规划、推理和协作能力。例如，[138]提出了用于具身推理和任务规划的规划者-执行者-报告者（Planner-Actor-Reporter）范式。DECKARD [139]引入了

该范式，通过解耦智能体的规划、执行和报告过程，促进具身推理和任务规划。TaPA [137]构建了一个多模态数据集，包含室内场景的多视角RGB图像、人类指令及相应计划，用于微调LLM。微调后的模型将视觉感知与任务规划对齐，使其能够生成更具可执行性的计划，显著提升了在视觉基础任务中的表现。

To overcome the physical constraints, the agents can generate executable plans and accomplish long-term tasks by leveraging multiple skills. In terms of control policies, SayCan [79] focuses on investigating a wide range of manipulation and navigation skills utilizing a mobile manipulator robot. Taking inspiration from typical tasks encountered in a kitchen environment, it presents a comprehensive set of 551 skills that cover seven skill families and 17 objects. These skills encompass various actions such as picking, placing, grasping, and manipulating objects, among others. TidyBot [136] is an embodied agent designed to personalize household cleanup tasks. It can learn users' preferences on object placement and manipulation methods through textual examples.
为克服物理限制，智能体可以利用多种技能生成可执行计划并完成长期任务。在控制策略方面，SayCan [79]聚焦于利用移动操作机器人研究广泛的操作和导航技能。借鉴厨房环境中的典型任务，提出了涵盖七大技能类别和17种对象的551项技能。这些技能包括拾取、放置、抓取和操作物体等多种动作。TidyBot [136]是一种具身智能体，旨在个性化家庭清理任务。它能够通过文本示例学习用户对物品摆放和操作方式的偏好。

To promote the application of LLM-based autonomous agents, researchers have also introduced many open-source libraries, based on which the developers can quickly implement and evaluate agents according to their customized requirements [19, 82 , 127, 142-155]. For example, LangChain [147] is an open-source framework that automates coding, testing, debugging, and documentation generation tasks. By integrating language models with data sources and facilitating interaction with the environment, LangChain enables efficient and cost-effective software development through natural language communication and collaboration among multiple agent roles. Based on LangChain, XLang [145] provides a comprehensive set of tools and a fully integrated user interface. It focuses on executable language grounding, enabling the conversion of natural language instructions into code or action sequences that interact seamlessly with various environments, including databases, web applications, and physical robots. AutoGPT [82] is an agent that is fully automated. It sets one or more goals, breaks them down into corresponding tasks, and cycles through the tasks until the goal is achieved. WorkGPT [148] is an agent framework similar to AutoGPT and LangChain. By providing it with an instruction and a set of APIs, it engages in back-and-forth conversations with AI until the instruction is completed. GPT-Engineer [128] and DemoGPT [127] are open-source projects that focus on automating code generation through prompts to complete development tasks. SmolModels [126] offers a family of compact language models suitable for various tasks. AGiXT [144] is a dynamic AI automation platform that efficiently manages instructions and executes complex tasks across various AI providers, integrating adaptive memory, smart features, and a versatile plugin system. AgentVerse [156] is a versatile framework that facilitates researchers in creating customized LLM-based agent simulations efficiently. GPT Researcher [150] is an experimental application that leverages LLMs to efficiently develop research questions, trigger web crawls to gather information, summarize sources, and aggregate summaries. BMTools [151] provides a platform for community-driven tool building and sharing. It supports various types of tools, enables simultaneous task execution using multiple tools, and offers a simple interface for loading plugins via URLs, fostering easy development and contribution to the BMTools ecosystem.
为促进基于LLM的自主智能体的应用，研究者还推出了许多开源库，开发者可基于此快速实现并评估符合定制需求的智能体[19, 82，127, 142-155]。例如，LangChain [147]是一个开源框架，自动化编码、测试、调试和文档生成任务。通过将语言模型与数据源集成并促进与环境的交互，LangChain实现了多智能体角色间的自然语言沟通与协作，从而高效且经济地推动软件开发。基于LangChain，XLang [145]提供了全面的工具集和完整集成的用户界面，专注于可执行语言的落地，实现自然语言指令向代码或动作序列的转换，能够无缝与数据库、网页应用及物理机器人等多种环境交互。AutoGPT [82]是一种全自动智能体，设定一个或多个目标，将其拆解为相应任务，并循环执行直至目标达成。WorkGPT [148]是类似于AutoGPT和LangChain的智能体框架，通过提供指令和一组API，与AI进行反复对话直至完成指令。GPT-Engineer [128]和DemoGPT [127]是专注于通过提示自动生成代码以完成开发任务的开源项目。SmolModels [126]提供了一系列适用于多种任务的紧凑型语言模型。AGiXT [144]是一个动态AI自动化平台，高效管理指令并执行跨多AI提供商的复杂任务，集成了自适应记忆、智能功能和多功能插件系统。AgentVerse [156]是一个多功能框架，帮助研究者高效创建定制的基于LLM的智能体仿真。GPT Researcher [150]是一个实验性应用，利

用LLM高效开发研究问题，触发网络爬虫收集信息，汇总来源并整合摘要。BMTools [151]提供了一个社区驱动的工具构建与共享平台，支持多种工具类型，允许多工具同时执行任务，并提供通过URL加载插件的简易界面，促进BMTools生态系统的便捷开发与贡献。

Remark. Utilization of LLM-based agents in supporting above applications may also entail risks and challenges. On one hand, LLMs themselves may be susceptible to illusions and other issues, occasionally providing erroneous answers, leading to incorrect conclusions, experimental failures, or even posing risks to human safety in hazardous experiments. Therefore, during experimentation, users must possess the necessary expertise and knowledge to exercise appropriate caution. On the other hand, LLM-based agents could potentially be exploited for malicious purposes, such as the development of chemical weapons, necessitating the implementation of security measures, such as human alignment, to ensure responsible and ethical use.
备注。基于大型语言模型（LLM）的代理在支持上述应用时也可能带来风险和挑战。一方面，LLM本身可能易受幻觉等问题影响，偶尔会给出错误答案，导致错误结论、实验失败，甚至在危险实验中威胁人类安全。因此，实验过程中，用户必须具备必要的专业知识和技能，谨慎操作。另一方面，基于LLM的代理可能被恶意利用，例如用于化学武器的开发，因此需要实施安全措施，如人类对齐（human alignment），以确保负责任和伦理的使用。

In summary, in the above sections, we introduce the typical applications of LLM-based autonomous agents in three important domains. To facilitate a clearer understanding, we have summarized the relationship between previous studies and their respective applications in Table 2.
综上所述，在上述章节中，我们介绍了基于LLM的自主代理在三个重要领域的典型应用。为了便于更清晰的理解，我们在表2中总结了以往研究与各自应用之间的关系。

7　4 LLM-based Autonomous Agent Eval- uation

8　4 基于LLM的自主代理评估

Similar to LLMs themselves, evaluating the effectiveness of LLM-based autonomous agents is a challenging task. This section outlines two prevalent approaches to evaluation: subjective and objective methods. For a comprehensive overview, please refer to the right portion of Figure 5.
与LLM本身类似，评估基于LLM的自主代理的有效性是一项具有挑战性的任务。本节概述了两种常见的评估方法：主观评估和客观评估。欲了解全面内容，请参见图5右侧部分。

8.1　4.1 Subjective Evaluation

8.2　4.1 主观评估

Subjective evaluation measures the agent capabilities based on human judgements [20,22,29,80,157]. It is suitable for the scenarios where there are no evaluation datasets or it is very hard to design quantitative metrics, for example, evaluating the agent's intelligence or user-friendliness. In the following, we present two commonly used strategies for subjective evaluation.
主观评估基于人类判断来衡量代理能力[20,22,29,80,157]。适用于没有评估数据集或难以设计量化指标的场景，例如评估代理的智能水平或用户友好性。以下介绍两种常用的主观评估策略。

Human Annotation: This evaluation method involves human evaluators directly scoring or ranking the outputs generated by various agents [22, 29, 104] . For example, in [20], the authors engage numerous annotators by asking 25 questions that explore their abilities across five key areas directly related to agent capabilities. In [80], annotators are asked to determine whether the specifically designed models can significantly enhance the development of rules within online communities.
人工标注：该评估方法由人工评估者直接对各代理生成的输出进行评分或排序[22, 29, 104]。例如，在[20]中，作者邀请大量标注者回答25个问题，考察其在与代理能力直接相关的五个关键领域的表现。在[80]中，标注者被要求判断特定设计的模型是否能显著促进在线社区规则的发展。

Turing Test: This evaluation strategy necessitates that human evaluators differentiate between outputs produced by agents and those created by humans. If, in a given task, the evaluators cannot separate the agent and human results, it demonstrates that the agent can achieve human-like performance on this task. For instance, researchers in [29] conduct experiments on free-form Partisan text, and the human evaluators are asked to guess whether the responses are from human or LLM-based agent.
图灵测试：该评估策略要求人工评估者区分代理生成的输出与人类生成的输出。如果在某项任务中，评估者无法区分代理与人类的结果，则表明该代理在该任务上能达到类人表现。例如，[29]的研究者在自由形式的党派文本上进行实验，人工评估者需猜测回答是来自人类还是基于LLM的代理。

Remark. LLM-based agents are usually designed to serve humans. Thus, subjective agent evaluation plays a critical role, since it reflects human criterion. However, this strategy also faces issues such as high costs, inefficiency, and population bias. To address these issues, a growing number of researchers are investigating the use of LLMs themselves as intermediaries for carrying out these subjective assessments. For example, in ChemCrow [76], researchers assess the experimental results using GPT. They consider both the completion of tasks and the accuracy of the underlying processes. Similarly, ChatEval [158] introduces a novel approach by employing multiple agents to critique and assess the results generated by various candidate models in a structured debate format. This innovative use of LLMs for evaluation purposes holds promise for enhancing both the credibility and applicability of subjective assessments in the future. As LLM technology continues to evolve, it is anticipated that these methods will become increasingly reliable and find broader applications, thereby overcoming the current limitations of direct human evaluation.
备注。基于LLM的代理通常设计为服务人类，因此主观代理评估至关重要，因为它反映了人类标准。然而，该策略也面临高成本、效率低下和群体偏差等问题。为解决这些问题，越来越多研究者探索利用LLM自身作为中介执行主观评估。例如，在ChemCrow[76]中，研究者使用GPT评估实验结果，既考虑任务完成情况，也关注底层过程的准确性。类似地，ChatEval[158]引入了一种新方法，通过多个代理以结构化辩论形式批评和评估各候选模型生成的结果。这种创新的LLM评估应用有望提升主观评估的可信度和适用性。随着LLM技术不断发展，预计这些方法将变得更加可靠并获得更广泛应用，从而克服当前直接人工评估的局限。

8.3 4.2 Objective Evaluation

8.4 4.2 客观评估

Objective evaluation refers to assessing the capabilities of LLM-based autonomous agents using quantitative metrics that can be computed, compared and tracked over time. In contrast to subjective evaluation, objective metrics aim to provide concrete, measurable insights into the agent performance. For conducting objective evaluation, there are three important aspects, that is, the evaluation metrics, protocols and benchmarks. In the following, we introduce these aspects more in detail.
客观评估指使用可计算、可比较且可随时间跟踪的量化指标来评估基于LLM的自主代理的能力。与主观评估不同，客观指标旨在提供具体、可测量的代理性能洞见。进行客观评估时，有三个重要方面，即评估指标、协议和基准。以下将详细介绍这些方面。

Metrics: In order to objectively evaluate the effectiveness of the agents, designing proper metrics is significant, which may influence the evaluation accuracy and comprehensiveness. Ideal evaluation metrics should precisely reflect the quality of the agents, and align with the human feelings when using them in real-world scenarios. In existing work, we can conclude the following representative evaluation metrics. (1) Task success metrics: These metrics measure how well an agent can complete tasks and achieve goals. Common metrics include success rate [12, 22, 58, 60] ,reward/score [22, 60, 161], coverage [16], and accuracy/error rate [18, 40, 80, 101] . Depending on the scenario, accuracy may reflect aspects such as program executability [18] or task validity [101]. Higher values across these task success metrics indicate greater task completion ability. (2) Human similarity metrics: These metrics quantify the degree to which the agent behaviors closely resembles those of humans by emphasizing various aspects related to human traits, such as coherent [104], fluent [104], dialogue similarities with human [80] and human acceptance rate [101]. Higher similarity suggests better human simulation performance. (3) Efficiency metrics: In

contrast to the aforementioned metrics used to evaluate the agent effectiveness, these metrics aim to assess the efficiency of agent. Commonly considered metrics encompass the cost associated with development [18] and training efficiency [16, 38] .
指标：为了客观评估智能体的有效性，设计合适的指标至关重要，这可能影响评估的准确性和全面性。理想的评估指标应准确反映智能体的质量，并与人类在真实场景中使用时的感受相一致。在现有工作中，我们可以总结出以下具有代表性的评估指标。(1) 任务成功指标：这些指标衡量智能体完成任务和实现目标的能力。常见指标包括成功率[12, 22, 58, 60]、奖励/得分[22, 60, 161]、覆盖率[16]和准确率/错误率[18, 40, 80, 101]。根据场景不同，准确率可能反映程序可执行性[18]或任务有效性[101]等方面。这些任务成功指标的数值越高，表明任务完成能力越强。(2) 人类相似度指标：这些指标量化智能体行为与人类行为的相似程度，强调与人类特征相关的各个方面，如连贯性[104]、流畅性[104]、与人类对话的相似度[80]以及人类接受率[101]。相似度越高，表明人类模拟性能越好。(3) 效率指标：与上述用于评估智能体有效性的指标不同，这些指标旨在评估智能体的效率。常见考虑的指标包括开发成本[18]和训练效率[16, 38]。

Protocols: In addition to the evaluation metrics, another important aspect for objective evaluation is how to leverage these metrics. In the previous work, we can identify the following commonly used evaluation protocols: (1) Real-world simulation: In this method, the agents are evaluated within immersive environments like games and interactive simulators. The agents are required to perform tasks autonomously, and then metrics like task success rate and human similarity are leveraged to evaluate the capability of the agents based on their trajectories and completed objectives [12, 16, 22, 33, 38 , 60, 86, 161, 164, 168] . By simulating real-world scenarios, this approach aims to provide a comprehensive evaluation of the agents' practical capabilities. (2) Social evaluation: This method utilizes metrics to assess social intelligence based on the agent interactions in simulated societies. Various approaches have been adopted, such as collaborative tasks to evaluate teamwork skills, debates to analyze argumentative reasoning, and human studies to measure social aptitude [34, 80, 101, 163]. These approaches analyze qualities such as coherence, theory of mind, and social IQ to assess agents' capabilities in areas including cooperation, communication, empathy, and mimicking human social behavior. By subjecting agents to complex interactive settings, social evaluation provides valuable insights into agents' higher-level social cognition. (3) Multi-task evaluation: In this method, people use a set of diverse tasks from different domains to evaluate the agent, which can effectively measure the agent generalization capability in open-domain environments [12,29,86,151,162-164,169,170]. (4) Software testing: In this method, researchers evaluate the agents by letting them conduct tasks such as software testing tasks, such as generating test cases, reproducing bugs, debugging code, and interacting with developers and external tools [159, 160, 167]. Then, one can use metrics like test coverage and bug detection rate to measure the effectiveness of LLM-based agents.
协议：除了评估指标，客观评估的另一个重要方面是如何利用这些指标。在以往工作中，我们可以识别出以下常用的评估协议：(1) 真实世界模拟：该方法在游戏和交互式模拟器等沉浸式环境中评估智能体。智能体需自主执行任务，然后利用任务成功率和人类相似度等指标，根据其轨迹和完成的目标[12, 16, 22, 33, 38， 60, 86, 161, 164, 168]评估智能体的能力。通过模拟真实场景，该方法旨在全面评估智能体的实际能力。(2) 社会评估：该方法基于智能体在模拟社会中的交互，利用指标评估社会智能。采用了多种方法，如协作任务评估团队合作能力、辩论分析论证推理，以及人类研究测量社会能力[34, 80, 101, 163]。这些方法分析连贯性、心智理论和社会智商等素质，以评估智能体在合作、沟通、共情及模仿人类社会行为等方面的能力。通过将智能体置于复杂的交互环境中，社会评估为智能体的高级社会认知提供了宝贵见解。(3) 多任务评估：该方法使用来自不同领域的一组多样化任务评估智能体，能够有效衡量智能体在开放域环境中的泛化能力[12,29,86,151,162-164,169,170]。(4) 软件测试：该方法通过让智能体执行软件测试任务，如生成测试用例、复现缺陷、调试代码以及与开发者和外部工具交互[159, 160, 167]，来评估智能体。然后，可以使用测试覆盖率和缺陷检测率等指标衡量基于大语言模型（LLM）的智能体的有效性。

Table 3 For subjective evaluation, we use ① and ② to represent human annotation and the Turing test, respectively. For objective evaluation, we use ①, ②, ③, and ④ to represent real-world simulation, social evaluation, multi-task evaluation, and software testing, respectively. "√" indicates that the evaluations are based on benchmarks.

表3 对于主观评估，我们用①和②分别表示人工标注和图灵测试。对于客观评估，我们用①、②、③和④分别表示真实世界模拟、社会评估、多任务评估和软件测试。“√”表示评估基于基准测试。

Model	Subjective	Objective	Benchmark	Time
WebShop [86]	-	① ③	✓	07/2022
Social Simulacra [80]	①	②	-	08/2022
TE [101]	-	②	-	08/2022
LIBRO [159]	-	④	-	09/2022
ReAct [60]	-	①	✓	10/2022
Argyle et al. [29]	②	② ③	-	02/2023
DEPS [33]	-	①	✓	02/2023
Jalil et al. [160]	-	④	-	02/2023
Reflexion [12]	-	① ③	-	03/2023
IGLU [161]	-	①	✓	04/2023
Generative Agents [20]	①	-	-	04/2023
ToolBench [151]	-	③	✓	04/2023
GITM [16]	-	①	✓	05/2023
Two-Failures [162]	-	③	-	05/2023
Voyager [38]	-	①	✓	05/2023
SocKET [163]	-	② ③	✓	05/2023
MobileEnv [164]	-	① ③	✓	05/2023
Clembench [165]	-	① ③	✓	05/2023
Dialop [166]	-	③	✓	06/2023
Feldt et al. [167]	-	④	-	06/2023
CO-LLM [22]	①	①	-	07/2023
Tachikuma [168]	①	① ③	✓	07/2023
RocoBench [92]	-	① ③	✓	07/2023
AgentSims [34]	-	②	-	08/2023
AgentBench [169]	-	③	✓	08/2023
BOLAA [170]	-	③	✓	08/2023
Gentopia [171]	-	③	✓	08/2023
EmotionBench [172]	①	-	✓	08/2023
PTB [125]	-	④	-	08/2023

模型	主观的	客观的	基准	时间
WebShop [86]	-	① ③	✓	07/2022
社会拟像 [80]	①	②	-	08/2022
TE [101]	-	②	-	08/2022
LIBRO [159]	-	④	-	09/2022
ReAct [60]	-	①	✓	10/2022
Argyle 等人 [29]	②	② ③	-	02/2023
DEPS [33]	-	①	✓	02/2023
Jalil 等人 [160]	-	④	-	02/2023
反思 (Reflexion) [12]	-	① ③	-	03/2023
IGLU [161]	-	①	✓	04/2023
生成代理 (Generative Agents) [20]	①	-	-	04/2023
ToolBench [151]	-	③	✓	04/2023
GITM [16]	-	①	✓	05/2023
两次失败 (Two-Failures) [162]	-	③	-	05/2023
Voyager [38]	-	①	✓	05/2023
SocKET [163]	-	② ③	✓	05/2023
MobileEnv [164]	-	① ③	✓	05/2023
Clembench [165]	-	① ③	✓	05/2023
Dialop [166]	-	③	✓	06/2023
Feldt 等人 [167]	-	④	-	06/2023
CO-LLM [22]	①	①	-	07/2023
Tachikuma [168]	①	① ③	✓	07/2023
RocoBench [92]	-	① ③	✓	07/2023
AgentSims [34]	-	②	-	08/2023
AgentBench [169]	-	③	✓	08/2023
BOLAA [170]	-	③	✓	08/2023
Gentopia [171]	-	③	✓	08/2023
EmotionBench [172]	①	-	✓	08/2023
PTB [125]	-	④	-	08/2023

Benchmarks: Given the metrics and protocols, a crucial aspect of evaluation is the selection of appropriate benchmarks. Over time, various benchmarks have been introduced to assess the capabilities of LLM-based agents across diverse domains and scenarios. Many studies employ environments such as ALFWorld [60], IGLU [161], and Minecraft [16, 33, 38] to evaluate agent capabilities in interactive and task-oriented simulations. Tachikuma [168] evaluates LLMs' ability to infer and understand complex interactions involving multiple characters and novel objects through TRPG game logs. AgentBench [169] provides a comprehensive framework for evaluating LLMs as autonomous agents across diverse environments. It represents the first systematic assessment of LLMs as agents on real-world challenges across diverse domains. SocKET [163] is a comprehensive benchmark for evaluating the social capabilities of LLMs across 58 tasks covering five categories of social information such as humor and sarcasm, emotions and feelings, credibility, etc. AgentSims [34] is a versatile framework for evaluating LLM-based agents, where one can flexibly design the agent planning, memory and action strategies, and measure the effectiveness of different agent modules in interactive environments. ToolBench [151] focuses on assessing and enhancing language models' ability to use tools, featuring 16,464 real-world RESTful APIs and diverse instructions tailored for single- and multi-tool scenarios. WebShop [86] develops a benchmark for evaluating LLM-based agents in terms of their capabilities for product search and retrieval, which is constructed using a collection of 1.18 million real-world items.

Mobile-Env [164] serves as an extendable interactive platform designed to evaluate the multi-step interaction capabilities of LLM-based agents. WebArena [173] offers a comprehensive website environment that spans multiple domains. Its purpose is to evaluate agents in an end-to-end fashion and determine the accuracy of their completed tasks. GentBench [171] is crafted to evaluate the agent capabilities, including their reasoning, safety, and efficiency, when utilizing tools to complete complex tasks. RocoBench [92] comprises six tasks that evaluate multi-agent collaboration across diverse scenarios, emphasizing communication and coordination strategies to assess adaptability and generalization in cooperative robotics. EmotionBench [172] evaluates the emotion appraisal ability of LLMs, i.e., how their feelings change when presented with specific situations. It collects over 400 situations that elicit eight negative emotions and measures the emotional states of LLMs and human subjects using self-report scales. PEB [125] is tailored for assessing LLM-based agents in penetration testing scenarios, comprising 13 diverse targets from leading platforms. It offers a structured evaluation across varying difficulty levels, reflecting real-world challenges for agents. ClemBench [165] contains five Dialogue Games to assess LLMs' ability as a player. E2E [174] serves as an end-to-end benchmark for testing the accuracy and usefulness of chatbots.
基准测试：在给定指标和协议的情况下，评估的关键方面是选择合适的基准。随着时间推移，已经引入了各种基准来评估基于大型语言模型（LLM）的代理在不同领域和场景中的能力。许多研究采用如ALFWorld [60]、IGLU [161]和Minecraft [16, 33, 38]等环境来评估代理在交互式和任务导向模拟中的能力。Tachikuma [168]通过TRPG游戏日志评估LLM推断和理解涉及多角色及新颖对象的复杂交互的能力。AgentBench [169]提供了一个全面的框架，用于评估LLM作为自主代理在多样环境中的表现，代表了首次对LLM作为代理在多领域现实挑战中的系统性评估。SocKET [163]是一个涵盖幽默与讽刺、情感与感受、可信度等五类社会信息共58项任务的综合基准，用于评估LLM的社交能力。AgentSims [34]是一个多功能框架，用于评估基于LLM的代理，用户可以灵活设计代理的规划、记忆和行动策略，并测量不同代理模块在交互环境中的有效性。ToolBench [151]专注于评估和提升语言模型使用工具的能力，包含16,464个真实世界的RESTful API及针对单工具和多工具场景的多样指令。WebShop [86]开发了一个基准，用于评估基于LLM的代理在产品搜索和检索方面的能力，该基准基于118万件真实商品构建。Mobile-Env [164]是一个可扩展的交互平台，设计用于评估基于LLM的代理的多步交互能力。WebArena [173]提供了一个涵盖多个领域的综合网站环境，旨在端到端评估代理并确定其完成任务的准确性。GentBench [171]旨在评估代理在使用工具完成复杂任务时的推理、安全性和效率能力。RocoBench [92]包含六个任务，评估多代理在不同场景下的协作，强调通信与协调策略，以评估合作机器人中的适应性和泛化能力。EmotionBench [172]评估LLM的情绪评估能力，即在特定情境下其情感变化，收集了400多个引发八种负面情绪的情境，并通过自我报告量表测量LLM和人类受试者的情绪状态。PEB [125]专为评估基于LLM的代理在渗透测试场景中的表现设计，包含来自主流平台的13个多样目标，提供了反映现实挑战的分级结构化评估。ClemBench [165]包含五个对话游戏，用于评估LLM作为玩家的能力。E2E [174]作为一个端到端基准，用于测试聊天机器人的准确性和实用性。

Remark. Objective evaluation facilitates the quantitative analysis of capabilities in LLM-based agents through a variety of metrics. While current techniques can not perfectly measure all types of agent capabilities, objective evaluation provides essential insights that complement subjective assessment. Continued advancements in benchmarks and methodologies for objective evaluation will enhance the development and understanding of LLM-based autonomous agents further.
备注。客观评估通过多种指标促进了对基于LLM的代理能力的量化分析。尽管当前技术无法完美衡量所有类型的代理能力，客观评估仍提供了补充主观评估的重要见解。基准和客观评估方法的持续进步将进一步推动基于LLM的自主代理的发展和理解。

In the above sections, we introduce both subjective and objective strategies for LLM-based autonomous agent evaluation. The evaluation of the agents play significant roles in this domain. However, both subjective and objective evaluation have their own strengths and weakness. Maybe, in practice, they should be combined to comprehensively evaluate the agents. We summarize the correspondence between the previous work and these evaluation strategies in Table 3.
在上述章节中，我们介绍了基于LLM的自主代理评估的主观和客观策略。代理的评估在该领域中起着重要作用。然而，主观和客观评估各有优缺点。实际上，二者应结合使用，以全面评估代理。我们在表3中总结了前人工作与这些评估策略的对应关系。

9 5 Related Surveys

10 5 相关综述

With the vigorous development of large language models, a variety of comprehensive surveys have emerged, providing detailed insights into various aspects. [175] extensively introduces the background, main findings, and mainstream technologies of LLMs, encompassing a vast array of existing works. On the other hand, [176] primarily focus on the applications of LLMs in various downstream tasks and the challenges associated with their deployment. Aligning LLMs with human intelligence is an active area of research to address concerns such as biases and illusions. [177] have compiled existing techniques for human alignment, including data collection and model training methodologies. Reasoning is a crucial aspect of intelligence, influencing decision-making, problem-solving, and other cognitive abilities. [178] presents the current state of research on LLMs' reasoning abilities, exploring approaches to improve and evaluate their reasoning skills. [179] propose that language models can be enhanced with reasoning capabilities and the ability to utilize tools, termed Augmented Language Models (ALMs). They conduct a comprehensive review of the latest advancements in ALMs. As the utilization of large-scale models becomes more prevalent, evaluating their performance is increasingly critical. [180] shed light on evaluating LLMs, addressing what to evaluate, where to evaluate, and how to assess their performance in downstream tasks and societal impact. [181] also discusses the capabilities and limitations of LLMs in various downstream tasks. The aforementioned research encompasses various aspects of large models, including training, application, and evaluation. However, prior to this paper, no work has specifically focused on the rapidly emerging and highly promising field of LLM-based Agents. In this study, we have compiled 100 relevant works on LLM-based Agents, covering their construction, applications, and evaluation processes.

随着大型语言模型（LLMs）的蓬勃发展，各类综合性综述纷纷涌现，提供了对不同方面的详细洞见。[175] 广泛介绍了LLMs的背景、主要发现及主流技术，涵盖了大量现有工作。另一方面，[176] 主要关注LLMs在各类下游任务中的应用及其部署面临的挑战。使LLMs与人类智能对齐是一个活跃的研究领域，旨在解决偏见和幻觉等问题。[177] 汇编了现有的人类对齐技术，包括数据收集和模型训练方法。推理是智能的关键方面，影响决策、问题解决及其他认知能力。[178] 展示了LLMs推理能力的研究现状，探讨了提升和评估其推理技能的方法。[179] 提出语言模型可以通过增强推理能力和工具使用能力来提升，称为增强语言模型（Augmented Language Models, ALMs），并对ALMs的最新进展进行了全面回顾。随着大规模模型的广泛应用，评估其性能变得愈发重要。[180] 阐述了LLMs的评估问题，包括评估内容、评估场所及如何评估其在下游任务和社会影响中的表现。[181] 也讨论了LLMs在各类下游任务中的能力与局限。上述研究涵盖了大型模型的训练、应用和评估等多个方面。然而，在本文之前，尚无专门聚焦于快速兴起且极具前景的基于LLM的智能体（Agents）领域的工作。本研究汇编了100篇相关文献，涵盖基于LLM的智能体的构建、应用及评估过程。

11 6 Challenges

12 6 挑战

While previous work on LLM-based autonomous agent 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.

尽管先前关于基于LLM的自主智能体的研究取得了诸多显著成果，但该领域仍处于初期阶段，发展过程中存在若干重大挑战。以下我们将介绍一些具有代表性的挑战。

12.1 6.1 Role-playing Capability

12.2 6.1 角色扮演能力

Different from traditional LLMs, autonomous agent usually has to play as specific roles (e.g., program coder, researcher and chemist) for accomplishing different tasks. Thus, the capability of the agent for role-playing is very important. Although LLMs can effectively simulate many common roles such as movie reviewers, there are still various roles and aspects that they struggle to capture accurately. To begin with, LLMs are usually trained based on web-corpus, thus for the roles which are seldom discussed on the web or the newly emerging roles, LLMs may not simulate them well. In addition, previous research [30] has shown that existing LLMs may not well model the human cognitive psychology characters, leading to the lack of self-awareness in conversation scenarios. Potential solution to these problems may include fine-tuning LLMs or carefully designing the agent prompts/architectures [182] For example, one can firstly collect real-human data for uncommon roles or psychology characters, and then leverage it to fine-tune LLMs. However, how to ensure that fine-tuned model still perform well for the common roles may pose further challenges. Beyond fine-tuning, one can also design tailored agent prompts/architectures to enhance the capability of LLM on role-playing. However, finding the optimal prompts/architectures is not easy, since their designing spaces are too large.

与传统LLMs不同，自主智能体通常需要扮演特定角色（如程序编码员、研究员和化学家）以完成不同任务。因此，智能体的角色扮演能力非常重要。尽管LLMs能够有效模拟许多常见角色，如电影评论员，但仍有多种角色和方面难以准确捕捉。首先，LLMs通常基于网络语料训练，因此对于网络上较少讨论或新兴的角色，LLMs可能无法很好地模拟。此外，先前研究[30]表明，现有LLMs可能无法很好地模拟人类认知心理特征，导致在对话场景中缺乏自我意识。解决这些问题的潜在方案包括微调LLMs或精心设计智能体的提示词/架构[182]。例如，可以先收集真实人类数据以覆盖不常见的角色或心理特征，然后利用这些数据微调LLMs。然而，如何确保微调后的模型在常见角色上仍表现良好，可能带来进一步挑战。除了微调，还可以设计定制的智能体提示词/架构以增强LLMs的角色扮演能力。但由于设计空间庞大，寻找最优提示词/架构并非易事。

12.3 6.2 Generalized Human Alignment

12.4 6.2 广义人类对齐

Human alignment has been discussed a lot for traditional LLMs. In the field of LLM-based autonomous agent, especially when the agents are leveraged for simulation, we believe this concept should be discussed more in depth. In order to better serve human-beings, traditional LLMs are usually fine-tuned to be aligned with correct human values, for example, the agent should not plan to make a bomb for avenging society. However, when the agents are leveraged for real-world simulation, an ideal simulator should be able to honestly depict diverse human traits, including the ones with incorrect values. Actually, simulating the human negative aspects can be even more important, since an important goal of simulation is to discover and solve problems, and without negative aspects means no problem to be solved. For example, to simulate the real-world society, we may have to allow the agent to plan for making a bomb, and observe how it will act to implement the plan as well as the influence of its behaviors. Based on these observations, people can make better actions to stop similar behaviors in real-world society. Inspired by the above case, maybe an important problem for agent-based simulation is how to conduct generalized human alignment, that is, for different purposes and applications, the agent should be able to align with diverse human values. However, existing powerful LLMs including ChatGPT and GPT-4 are mostly aligned with unified human values. Thus, an interesting direction is how to "realign" these models by designing proper prompting strategies.

人类对齐在传统LLMs中已被广泛讨论。在基于LLM的自主智能体领域，尤其是当智能体用于模拟时，我们认为这一概念应得到更深入的探讨。为了更好地服务人类，传统LLMs通常通过微调使其与正确的人类价值观对齐，例如，智能体不应策划制造炸弹以报复社会。然而，当智能体用于现实世界模拟时，理想的模拟器应能诚实地描绘多样的人类特质，包括那些带有错误价值观的特质。实际上，模拟人类的负面方面可能更为重要，因为模拟的一个重要目标是发现和解决问题，而没有负面方面就无从谈起问题。例如，为了模拟现实社会，我们可能需要允许智能体策划制造炸弹，并观察其实施计划的行为及其影响。基于这些观察，人们可以采取更有效的措施阻止现实社会中类似行为。受上述案例启发，基于智能体的模拟中一个重要问题是如何实现广义的人类对齐，即针对不同目的和应用，智能体应能与

多样的人类价值观对齐。然而，现有强大的LLMs，包括ChatGPT和GPT-4，大多与统一的人类价值观对齐。因此，一个有趣的方向是如何通过设计合适的提示策略来“重新对齐”这些模型。

12.5 6.3 Prompt Robustness

12.6 6.3 提示词鲁棒性

To ensure rational behavior in agents, it's a common practice for designers to embed supplementary modules, such as memory and planning modules, into LLMs. However, the inclusion of these modules necessitates the development of more complex prompts in order to facilitate consistent operation and effective communication. Previous research [183, 184] has highlighted the lack of robustness in prompts for LLMs, as even minor alterations can yield substantially different outcomes. This issue becomes more pronounced when constructing autonomous agents, as they encompass not a single prompt but a prompt framework that considers all modules, wherein the prompt for one module has the potential to influence others. Moreover, the prompt frameworks can vary significantly across different LLMs. The development of a unified and resilient prompt framework applicable across diverse LLMs remains a critical and unresolved challenge. There are two potential solutions to the aforementioned problems: (1) manually crafting the essential prompt elements through trial and error, or (2) automatically generating prompts using GPT.
为了确保智能体的理性行为，设计者通常会在大型语言模型（LLMs）中嵌入辅助模块，如记忆模块和规划模块。然而，加入这些模块需要开发更复杂的提示（prompts），以促进一致的操作和有效的沟通。先前的研究[183, 184]指出，LLMs的提示缺乏鲁棒性，即使是细微的改动也可能导致截然不同的结果。在构建自主智能体时，这一问题更加突出，因为它们不仅包含单一提示，而是涵盖所有模块的提示框架，其中一个模块的提示可能影响其他模块。此外，不同LLMs的提示框架差异显著。开发一个适用于多种LLMs的统一且稳健的提示框架仍是一个关键且未解决的挑战。针对上述问题，有两种潜在解决方案：（1）通过反复试验手动设计必要的提示元素，或（2）利用GPT自动生成提示。

12.7 6.4 Hallucination

12.8 6.4 幻觉问题

Hallucination poses a fundamental challenge for LLMs, characterized by the models' tendency to produce false information with a high level of confidence. This challenge is not limited to LLMs alone but is also a significant concern in the domain of autonomous agents. For instance, in [185], it was observed that when confronted with simplistic instructions during code generation tasks, the agent may exhibit hallucinatory behavior. Hallucination can lead to serious consequences such as incorrect or misleading code, security risks, and ethical issues [185]. To mitigate this issue, incorporating human correction feedback directly into the iterative process of human-agent interaction presents a viable approach [23]. More discussions on the hallucination problem can be seen in [175].
幻觉问题是LLMs面临的根本挑战，表现为模型倾向于以高度自信生成错误信息。这一挑战不仅限于LLMs，在自主智能体领域同样是重大关注点。例如，在文献[185]中观察到，当面对简单指令进行代码生成任务时，智能体可能表现出幻觉行为。幻觉可能导致严重后果，如代码错误或误导、安全风险及伦理问题[185]。为缓解该问题，将人工纠正反馈直接纳入人机交互的迭代过程中是一种可行方法[23]。关于幻觉问题的更多讨论可见文献[175]。

12.9 6.5 Knowledge Boundary

12.10 6.5 知识边界

A pivotal application of LLM-based autonomous agents lies in simulating diverse real-world human behaviors [20]. The study of human simulation has a long history, and the recent surge in interest can be attributed to the remarkable advancements made by LLMs, which have demonstrated significant capabilities in simulating human behavior. However, it is important to recognize that the power of LLMs may not always be advantageous. Specifically, an ideal simulation should accurately replicate human knowledge. In this context, LLMs may display overwhelming capabilities, being trained on a vast corpus of web knowledge that far exceeds what an average

individual might know. The immense capabilities of LLMs can significantly impact the effectiveness of simulations. For instance, when attempting to simulate user selection behaviors for various movies, it is crucial to ensure that LLMs assume a position of having no prior knowledge of these movies. However, there is a possibility that LLMs have already acquired information about these movies. Without implementing appropriate strategies, LLMs may make decisions based on their extensive knowledge, even though real-world users would not have access to the contents of these movies beforehand. Based on the above example, we may conclude that for building believable agent simulation environment, an important problem is how to constrain the utilization of user-unknown knowledge of LLM.
基于LLMs的自主智能体的一个关键应用是模拟多样的现实人类行为[20]。人类模拟研究历史悠久，近期兴趣激增归因于LLMs在模拟人类行为方面取得的显著进展。然而，必须认识到LLMs的强大能力并非总是有利。理想的模拟应准确复制人类知识。在此背景下，LLMs可能表现出压倒性的能力，因为它们训练于庞大的网络知识语料库，远超普通个体的知识水平。LLMs的巨大能力会显著影响模拟的有效性。例如，在尝试模拟用户对不同电影的选择行为时，必须确保LLMs处于对这些电影一无所知的状态。然而，LLMs可能已掌握这些电影的信息。若不采取适当策略，LLMs可能基于其广泛知识做出决策，而现实用户事先并不知晓这些电影内容。基于上述例子，我们可以得出结论：构建可信的智能体模拟环境时，一个重要问题是如何限制LLMs利用用户未知的知识。

12.11 6.6 Efficiency

12.12 6.6 效率

Due to their autoregressive architecture, LLMs typically have slow inference speeds. However, the agent may need to query LLMs for each action multiple times, such as extracting information from memory, make plans before taking actions and so on. Consequently, the efficiency of agent actions is greatly affected by the speed of LLM inference.
由于其自回归架构，LLMs的推理速度通常较慢。然而，智能体可能需要针对每个动作多次查询LLMs，例如从记忆中提取信息、在行动前制定计划等。因此，智能体动作的效率在很大程度上受限于LLMs推理速度。

13 7 Conclusion

14 7 结论

In this survey, we systematically summarize existing research in the field of LLM-based autonomous agents. We present and review these studies from three aspects including the construction, application, and evaluation of the agents. For each of these aspects, we provide a detailed taxonomy to draw connections among the existing research, summarizing the major techniques and their development histories. In addition to reviewing the previous work, we also propose several challenges in this field, which are expected to guide potential future directions.
在本综述中，我们系统总结了基于LLMs的自主智能体领域的现有研究。从智能体的构建、应用和评估三个方面对这些研究进行了呈现和回顾。针对每个方面，我们提供了详细的分类法，以连接现有研究，总结主要技术及其发展历程。除回顾以往工作外，我们还提出了该领域的若干挑战，期望为未来研究指明方向。

15 Acknowledgement

16 致谢

This work is supported in part by National Natural Science Foundation of China (No. 62102420), Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098, Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China, Public Computing Cloud, Renmin University of China, fund for building world-class universities (disciplines) of Renmin University of China, Intelligent Social Governance Platform. References
本工作部分由国家自然科学基金（编号62102420）、北京市杰出青年科学家计划（编号

BJJWZYJH012019100020098）、智能社会治理平台、“双一流”重大创新与规划跨学科平台（中国人民大学）、中国人民大学公共计算云、中国人民大学世界一流大学（学科）建设基金、智能社会治理平台资助。参考文献

1. Mnih V, Kavukcuoglu K, Silver D, Rusu A A, Veness J, Bellemare M G, Graves A, Riedmiller M, Fidjeland A K, Ostrovski G, others. Human-level control through deep reinforcement learning. nature, 2015, 518(7540): 529-533
2. Mnih V, Kavukcuoglu K, Silver D, Rusu A A, Veness J, Bellemare M G, Graves A, Riedmiller M, Fidjeland A K, Ostrovski G, 等. 通过深度强化学习实现人类水平控制. 自然, 2015, 518(7540): 529-533

2. Lillicrap T P, Hunt J J, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D. Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971, 2015
3. Lillicrap T P, Hunt J J, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D. 使用深度强化学习的连续控制。arXiv预印本 arXiv:1509.02971, 2015

3. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017
4. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. 近端策略优化算法。arXiv预印本 arXiv:1707.06347, 2017

4. Haarnoja T, Zhou A, Abbeel P, Levine S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: International conference on machine learning. 2018, 1861-1870
5. Haarnoja T, Zhou A, Abbeel P, Levine S. 软演员-评论家（Soft actor-critic）：带有随机演员的离策略最大熵深度强化学习。载于：国际机器学习会议，2018，1861-1870

5. Brown T, Mann B, Ryder N, Subbiah M, Kaplan J D, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, others. Language models are few-shot learners. Advances in neural information processing systems, 2020, 33: 1877-1901
6. Brown T, Mann B, Ryder N, Subbiah M, Kaplan J D, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, 等。语言模型是少样本学习者。神经信息处理系统进展，2020，33：1877-1901

6. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I, others. Language models are unsupervised multitask learners. OpenAI blog, 2019, 1(8): 9
7. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I, 等。语言模型是无监督多任务学习者。OpenAI博客，2019，1(8)：9

7. Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, Aleman F L, Almeida D, Altenschmidt J, Altman S, Anadkat S, others. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023
8. Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, Aleman F L, Almeida D, Altenschmidt J, Altman S, Anadkat S, 等。GPT-4技术报告。arXiv预印本 arXiv:2303.08774, 2023

8. Anthropic . Model card and evaluations for claude models. https://www-files.anthropic.com/production/images/ Model-Card-Claude-2.pdf?ref= maginative.com, 2023
9. Anthropic。Claude模型的模型卡和评估。https://www-files.anthropic.com/production/images/ Model-Card-Claude-2.pdf?ref= maginative.com, 2023

9. Touvron H, Lavril T, Izacard G, Martinet X, Lachaux M A, Lacroix T, Rozière B, Goyal N, Hambro E, Azhar F, others . Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971, 2023
10. Touvron H, Lavril T, Izacard G, Martinet X, Lachaux M A, Lacroix T, Rozière B, Goyal N, Hambro E, Azhar F, 等。Llama：开放且高效的基础语言模型。arXiv预印本 arXiv:2302.13971, 2023

10. Touvron H, Martin L, Stone K, Albert P, Almahairi A, Babaei Y, Bashlykov N, Batra S, Bhargava P, Bhosale S, others. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288, 2023

11. Touvron H, Martin L, Stone K, Albert P, Almahairi A, Babaei Y, Bashlykov N, Batra S, Bhargava P, Bhosale S, 等。Llama 2：开放基础及微调聊天模型。arXiv预印本 arXiv:2307.09288, 2023

11. Chen X, Li S, Li H, Jiang S, Qi Y, Song L. Generative adversarial user model for reinforcement learning based recommendation system. In: International Conference on Machine Learning. 2019, 1052-1061
12. Chen X, Li S, Li H, Jiang S, Qi Y, Song L. 基于生成对抗用户模型的强化学习推荐系统。载于：国际机器学习会议，2019，1052-1061

12. Shinn N, Cassano F, Gopinath A, Narasimhan K, Yao S. Reflexion: Language agents with verbal reinforcement learning. Advances in Neural Information Processing Systems, 2024, 36
13. Shinn N, Cassano F, Gopinath A, Narasimhan K, Yao S. Reflexion：具备语言强化学习的语言代理。神经信息处理系统进展，2024，36

13. Shen Y, Song K, Tan X, Li D, Lu W, Zhuang Y. Hug-ginggpt: Solving ai tasks with chatgpt and its friends in hugging face. Advances in Neural Information Processing Systems, 2024, 36
14. Shen Y, Song K, Tan X, Li D, Lu W, Zhuang Y. HuggingGPT：利用ChatGPT及其伙伴在Hugging Face上解决AI任务。神经信息处理系统进展，2024，36

14. Qin Y, Liang S, Ye Y, Zhu K, Yan L, Lu Y, Lin Y, Cong X, Tang X, Qian B, others . Toolllm: Facilitating large language models to master 16000+ real-world apis. arXiv preprint arXiv:2307.16789, 2023
15. Qin Y, Liang S, Ye Y, Zhu K, Yan L, Lu Y, Lin Y, Cong X, Tang X, Qian B, 等。ToolLLM：助力大型语言模型掌握16000+真实世界API。arXiv预印本 arXiv:2307.16789, 2023

15. Schick T, Dwivedi-Yu J, Dessì R, Raileanu R, Lomeli M, Hambro E, Zettlemoyer L, Cancedda N, Scialom T. Toolformer: Language models can teach themselves to use tools. Advances in Neural Information Processing Systems, 2024, 36
16. Schick T, Dwivedi-Yu J, Dessì R, Raileanu R, Lomeli M, Hambro E, Zettlemoyer L, Cancedda N, Scialom T. Toolformer：语言模型可以自学使用工具。神经信息处理系统进展，2024，36

16. Zhu X, Chen Y, Tian H, Tao C, Su W, Yang C, Huang G, LiB, LuL , Wang X , others . Ghost in the minecraft: Generally capable agents for open-world enviroments via large language models with text-based knowledge and memory. arXiv preprint arXiv:2305.17144, 2023
17. Zhu X, Chen Y, Tian H, Tao C, Su W, Yang C, Huang G, LiB, LuL , Wang X , 等。Minecraft中的幽灵：通过具备文本知识和记忆的大型语言模型实现开放世界环境中的通用能力代理。arXiv预印本 arXiv:2305.17144, 2023

17. Sclar M, Kumar S, West P, Suhr A, Choi Y, Tsvetkov Y. Minding language models'(lack of) theory of mind: A plug-and-play multi-character belief tracker. arXiv preprint arXiv:2306.00924, 2023
18. Sclar M, Kumar S, West P, Suhr A, Choi Y, Tsvetkov Y. 关注语言模型的（缺乏）心智理论：一种即插即用的多角色信念追踪器。arXiv预印本 arXiv:2306.00924, 2023

18. Qian C, Cong X, Yang C, Chen W, Su Y, Xu J, Liu Z, Sun M. Communicative agents for software development. arXiv preprint arXiv:2307.07924, 2023
19. Qian C, Cong X, Yang C, Chen W, Su Y, Xu J, Liu Z, Sun M. 用于软件开发的交互代理。arXiv预印本 arXiv:2307.07924, 2023

19. al. e C. Agentverse. https://github.com/ OpenBMB/AgentVerse, 2023
20. al. e C. Agentverse。https://github.com/OpenBMB/AgentVerse, 2023

20. Park J S, O'Brien J, Cai C J, Morris M R, Liang P, Bernstein M S. Generative agents: Interactive simulacra of human behavior. In: Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology. 2023, 1-22
21. Park J S, O'Brien J, Cai C J, Morris M R, Liang P, Bernstein M S. 生成代理：人类行为的交互模拟。在：第

36届年度ACM用户界面软件与技术研讨会论文集。2023, 1-22

21. Wang L, Zhang J, Chen X, Lin Y, Song R, Zhao W X, Wen J R. Recagent: A novel simulation paradigm for recommender systems. arXiv preprint arXiv:2306.02552, 2023
22. Wang L, Zhang J, Chen X, Lin Y, Song R, Zhao W X, Wen J R. Recagent：推荐系统的新型仿真范式。arXiv预印本 arXiv:2306.02552, 2023

22. Zhang H, Du W, Shan J, Zhou Q, Du Y, Tenenbaum J B, Shu T, Gan C. Building cooperative embodied agents modularly with large language models. arXiv preprint arXiv:2307.02485, 2023
23. Zhang H, Du W, Shan J, Zhou Q, Du Y, Tenenbaum J B, Shu T, Gan C. 利用大型语言模型模块化构建协作具身代理。arXiv预印本 arXiv:2307.02485, 2023

23. Hong S, Zheng X, Chen J, Cheng Y, Wang J, Zhang C, Wang Z, Yau S K S, Lin Z, Zhou L, others. Metagpt: Meta programming for multi-agent collaborative framework. arXiv preprint arXiv:2308.00352, 2023
24. Hong S, Zheng X, Chen J, Cheng Y, Wang J, Zhang C, Wang Z, Yau S K S, Lin Z, Zhou L, others. Metagpt：多代理协作框架的元编程。arXiv预印本 arXiv:2308.00352, 2023

24. Dong Y, Jiang X, Jin Z, Li G. Self-collaboration code generation via chatgpt. arXiv preprint arXiv:2304.07590, 2023
25. Dong Y, Jiang X, Jin Z, Li G. 通过ChatGPT实现自我协作代码生成。arXiv预印本 arXiv:2304.07590, 2023

25. Safdari M, Serapio-García G, Crepy C, Fitz S, Romero P, Sun L, Abdulhai M, Faust A, Matarić M. Personality traits in large language models. arXiv preprint arXiv:2307.00184, 2023
26. Safdari M, Serapio-García G, Crepy C, Fitz S, Romero P, Sun L, Abdulhai M, Faust A, Matarić M. 大型语言模型中的人格特质。arXiv预印本 arXiv:2307.00184, 2023

26. Johnson J A. Measuring thirty facets of the five factor model with a 120-item public domain inventory: Development of the ipip-neo-120. Journal of research in personality, 2014, 51: 78-89
27. Johnson J A. 使用120项公共领域量表测量五因素模型的三十个方面：ipip-neo-120的开发。《人格研究杂志》，2014，51：78-89

27. John OP, Donahue E M, Kentle R L. Big five inventory. Journal of Personality and Social Psychology, 1991
28. John OP, Donahue E M, Kentle R L. 大五人格量表。《人格与社会心理学杂志》，1991

28. Deshpande A, Murahari V, Rajpurohit T, Kalyan A, Narasimhan K. Toxicity in chatgpt: Analyzing persona-assigned language models. arXiv preprint arXiv:2304.05335, 2023
29. Deshpande A, Murahari V, Rajpurohit T, Kalyan A, Narasimhan K. ChatGPT中的有害性：分析赋予人格的语言模型。arXiv预印本 arXiv:2304.05335, 2023

29. Argyle L P, Busby E C, Fulda N, Gubler J R, Rytting C, Wingate D. Out of one, many: Using language models to simulate human samples. Political Analysis, 2023, 31(3): 337-351
30. Argyle L P, Busby E C, Fulda N, Gubler J R, Rytting C, Wingate D. 从一到多：利用语言模型模拟人类样本。《政治分析》，2023，31(3)：337-351

30. Fischer K A. Reflective linguistic programming (rlp): A stepping stone in socially-aware agi (socialagi). arXiv preprint arXiv:2305.12647, 2023
31. Fischer K A. 反思性语言编程（RLP）：社会感知人工通用智能（SocialAGI）的垫脚石。arXiv预印本 arXiv:2305.12647, 2023

31. Rana K, Haviland J, Garg S, Abou-Chakra J, Reid I, Suenderhauf N. Sayplan: Grounding large language models using 3 d scene graphs for scalable robot task planning. In: 7th Annual Conference on Robot Learning. 2023

32. Rana K, Haviland J, Garg S, Abou-Chakra J, Reid I, Suenderhauf N. Sayplan：利用3 d场景图为大型语言模型提供基础，实现可扩展的机器人任务规划。在：第7届机器人学习年度会议。2023

32. Zhu A, Martin L, Head A, Callison-Burch C. Calypso: Llms as dungeon master's assistants. In: Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. 2023, 380-390

33. Zhu A, Martin L, Head A, Callison-Burch C. Calypso：作为地下城主助手的大型语言模型。在：AAAI人工智能与互动数字娱乐会议论文集。2023，380-390

33. Wang Z, Cai S, Chen G, Liu A, Ma X, Liang Y. Describe, explain, plan and select: Interactive planning with large language models enables open-world multitask agents. arXiv preprint arXiv:2302.01560, 2023

34. 王卓, 蔡思, 陈刚, 刘安, 马翔, 梁毅. 描述、解释、规划与选择：利用大型语言模型的交互式规划实现开放世界多任务智能体. arXiv预印本 arXiv:2302.01560, 2023

34. Lin J, Zhao H, Zhang A, Wu Y, Ping H, Chen Q. Agentsims: An open-source sandbox for large language model evaluation. arXiv preprint arXiv:2308.04026, 2023

35. 林杰, 赵浩, 张安, 吴洋, 平浩, 陈强. Agentsims：一个用于大型语言模型评估的开源沙盒. arXiv预印本 arXiv:2308.04026, 2023

35. Liang X, Wang B, Huang H, Wu S, Wu P, Lu L, Ma Z, Li Z. Unleashing infinite-length input capacity for large-scale language models with self-controlled memory system. arXiv preprint arXiv:2304.13343, 2023

36. 梁翔, 王博, 黄辉, 吴松, 吴鹏, 陆亮, 马志, 李哲. 通过自控记忆系统释放大规模语言模型的无限长度输入能力. arXiv预印本 arXiv:2304.13343, 2023

36. Ng Y, Miyashita D, Hoshi Y, Morioka Y, Torii O, Ko-dama T, Deguchi J. Simplyretrieve: A private and lightweight retrieval-centric generative ai tool. arXiv preprint arXiv:2308.03983, 2023

37. 黄勇, 宫下大, 星野洋, 森岡洋, 鳥居大, 小玉透, 出口淳. Simplyretrieve：一个私密且轻量的以检索为中心的生成式AI工具. arXiv预印本 arXiv:2308.03983, 2023

37. Huang Z, Gutierrez S, Kamana H, MacNeil S. 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. 2023, 1-3

38. 黄志, 古铁雷斯, 卡马纳, 麦克尼尔. Memory sandbox：面向对话智能体的透明且交互式记忆管理. 载于：第36届ACM用户界面软件与技术年会附录论文集. 2023, 1-3

38. Wang G, Xie Y, Jiang Y, Mandlekar A, Xiao C, Zhu Y, Fan L, Anandkumar A. Voyager: An open-ended embodied agent with large language models. arXiv preprint arXiv:2305.16291, 2023

39. 王刚, 谢勇, 姜毅, 曼德尔卡尔, 肖晨, 朱阳, 范磊, 阿南德库马尔. Voyager：基于大型语言模型的开放式具身智能体. arXiv预印本 arXiv:2305.16291, 2023

39. Zhong W, Guo L, Gao Q, Wang Y. Memorybank: Enhancing large language models with long-term memory. arXiv preprint arXiv:2305.10250, 2023

40. 钟伟, 郭磊, 高强, 王宇. Memorybank：增强大型语言模型的长期记忆能力. arXiv预印本 arXiv:2305.10250, 2023

40. Hu C, Fu J, Du C, Luo S, Zhao J, Zhao H. Chatdb: Augmenting llms with databases as their symbolic memory. arXiv preprint arXiv:2306.03901, 2023

41. 胡超, 傅军, 杜晨, 罗松, 赵军, 赵浩. Chatdb：将数据库作为符号记忆增强大型语言模型. arXiv预印本 arXiv:2306.03901, 2023

41. Modarressi A, Imani A, Fayyaz M, Schütze H. Ret-llm: Towards a general read-write memory for large language models. arXiv preprint arXiv:2305.14322, 2023

42. Modarressi A, Imani A, Fayyaz M, Schütze H. Ret-llm：迈向大型语言模型的通用读写记忆. arXiv预印本 arXiv:2305.14322, 2023

42. Schuurmans D. Memory augmented large language models are computationally universal. arXiv preprint arXiv:2301.04589, 2023
43. Schuurmans D. 增强记忆的大型语言模型具备计算通用性. arXiv预印本 arXiv:2301.04589, 2023

43. Zhao A, Huang D, Xu Q, Lin M, Liu Y J, Huang G. Expel: Llm agents are experiential learners. arXiv preprint arXiv:2308.10144, 2023
44. 赵安, 黄丹, 徐强, 林明, 刘永杰, 黄刚. Expel：大型语言模型智能体是经验学习者. arXiv预印本 arXiv:2308.10144, 2023

44. Huang W, Abbeel P, Pathak D, Mordatch I. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In: International Conference on Machine Learning. 2022, 9118-9147
45. 黄伟, Abbeel P, Pathak D, Mordatch I. 语言模型作为零样本规划者：为具身智能体提取可执行知识. 载于：国际机器学习大会. 2022, 9118-9147

45. Wei J, Wang X, Schuurmans D, Bosma M, Xia F, Chi E, Le Q V, Zhou D, others. Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems, 2022, 35: 24824-24837
46. 魏军, 王翔, Schuurmans D, Bosma M, 夏飞, Chi E, Le Q V, 周丹, 等. 思维链提示激发大型语言模型的推理能力. 神经信息处理系统进展, 2022, 35: 24824-24837

46. Kojima T, Gu S S, Reid M, Matsuo Y, Iwasawa Y. Large language models are zero-shot reasoners. Advances in neural information processing systems, 2022, 35: 22199-22213
47. 小岛拓, Gu S S, Reid M, 松尾洋, 岩泽康. 大型语言模型是零样本推理者. 神经信息处理系统进展, 2022, 35: 22199-22213

47. Raman S S, Cohen V, Rosen E, Idrees I, Paulius D, Tellex S. Planning with large language models via corrective re-prompting. In: NeurIPS 2022 Foundation Models for Decision Making Workshop. 2022
48. Raman S S, Cohen V, Rosen E, Idrees I, Paulius D, Tellex S. 通过纠正性重新提示实现大型语言模型的规划. 载于：NeurIPS 2022决策基础模型研讨会. 2022

48. Xu B, Peng Z, Lei B, Mukherjee S, Liu Y, Xu D. Re-woo: Decoupling reasoning from observations for efficient augmented language models. arXiv preprint arXiv:2305.18323, 2023
49. 徐博, 彭志, 雷斌, Mukherjee S, 刘洋, 徐东. Re-woo：将推理与观察解耦以提升增强型语言模型效率. arXiv预印本 arXiv:2305.18323, 2023

49. Lin B Y, Fu Y, Yang K, Brahman F, Huang S, Bhaga-vatula C, Ammanabrolu P, Choi Y, Ren X. Swiftsage: A generative agent with fast and slow thinking for complex interactive tasks. Advances in Neural Information Processing Systems, 2024, 36
50. Lin B Y, Fu Y, Yang K, Brahman F, Huang S, Bhaga-vatula C, Ammanabrolu P, Choi Y, Ren X. Swiftsage：一种具备快慢思维的生成代理，用于复杂交互任务。《神经信息处理系统进展》(Advances in Neural Information Processing Systems), 2024, 36

50. Evans J S B, Stanovich K E. Dual-process theories of higher cognition: Advancing the debate. Perspectives on psychological science, 2013, 8(3): 223-241
51. Evans J S B, Stanovich K E. 高级认知的双过程理论：推动辩论进展。《心理科学视角》(Perspectives on psychological science), 2013, 8(3): 223-241

51. Wang X, Wei J, Schuurmans D, Le Q, Chi E, Narang S, Chowdhery A, Zhou D. Self-consistency improves chain of thought reasoning in language models. arXiv preprint arXiv:2203.11171, 2022
52. Wang X, Wei J, Schuurmans D, Le Q, Chi E, Narang S, Chowdhery A, Zhou D. 自洽性提升语言模型中的链式思维推理。arXiv预印本 arXiv:2203.11171, 2022

52. Yao S, Yu D, Zhao J, Shafran I, Griffiths T, Cao Y, Narasimhan K. Tree of thoughts: Deliberate problem solving with large language models. Advances in Neural Information Processing Systems, 2024, 36
53. Yao S, Yu D, Zhao J, Shafran I, Griffiths T, Cao Y, Narasimhan K. 思维树：利用大型语言模型进行深思熟虑的问题解决。《神经信息处理系统进展》(Advances in Neural Information Processing Systems), 2024, 36

53. Wang Y, Jiang Z, Chen Z, Yang F, Zhou Y, Cho E, Fan X, Huang X, Lu Y, Yang Y. Recmind: Large language model powered agent for recommendation. arXiv preprint arXiv:2308.14296, 2023
54. Wang Y, Jiang Z, Chen Z, Yang F, Zhou Y, Cho E, Fan X, Huang X, Lu Y, Yang Y. Recmind：基于大型语言模型的推荐代理。arXiv预印本 arXiv:2308.14296, 2023

54. Besta M, Blach N, Kubicek A, Gerstenberger R, Gianinazzi L, Gajda J, Lehmann T, Podstawski M, Niewiadomski H, Nyczyk P, others. Graph of thoughts: Solving elaborate problems with large language models. arXiv preprint arXiv:2308.09687, 2023
55. Besta M, Blach N, Kubicek A, Gerstenberger R, Gianinazzi L, Gajda J, Lehmann T, Podstawski M, Niewiadomski H, Nyczyk P, 等. 思维图谱：利用大型语言模型解决复杂问题。arXiv预印本 arXiv:2308.09687, 2023

55. Sel B, Al-Tawaha A, Khattar V, Wang L, Jia R, Jin M. Algorithm of thoughts: Enhancing exploration of ideas in large language models. arXiv preprint arXiv:2308.10379, 2023
56. Sel B, Al-Tawaha A, Khattar V, Wang L, Jia R, Jin M. 思维算法：增强大型语言模型中的创意探索。arXiv预印本 arXiv:2308.10379, 2023

56. Gramopadhye M, Szafir D. Generating executable action plans with environmentally-aware language models. In: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2023, 3568- 3575
57. Gramopadhye M, Szafir D. 利用环境感知语言模型生成可执行行动计划。载于：2023年IEEE/RSJ国际智能机器人与系统会议(IROS)。2023, 3568-3575

57. Hao S, Gu Y, Ma H, Hong J J, Wang Z, Wang D Z, Hu Z. Reasoning with language model is planning with world model. arXiv preprint arXiv:2305.14992, 2023
58. Hao S, Gu Y, Ma H, Hong J J, Wang Z, Wang D Z, Hu Z. 用语言模型推理即是用世界模型进行规划。arXiv预印本 arXiv:2305.14992, 2023

58. Liu B, Jiang Y, Zhang X, Liu Q, Zhang S, Biswas J, Stone P. LLM+P: Empowering large language models with optimal planning proficiency. arXiv preprint arXiv:2304.11477, 2023
59. Liu B, Jiang Y, Zhang X, Liu Q, Zhang S, Biswas J, Stone P. LLM+P：赋能大型语言模型以实现最优规划能力。arXiv预印本 arXiv:2304.11477, 2023

59. Dagan G, Keller F, Lascarides A. Dynamic planning with a llm. arXiv preprint arXiv:2308.06391, 2023
60. Dagan G, Keller F, Lascarides A. 利用大型语言模型进行动态规划。arXiv预印本 arXiv:2308.06391, 2023

60. Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan K ,Cao Y. React: Synergizing reasoning and acting in language models. In: The Twelfth International Conference on Learning Representations. 2023
61. Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan K ,Cao Y. React：在语言模型中协同推理与行动。载于：第十二届国际学习表征会议。2023

61. Song C H, Wu J, Washington C, Sadler B M, Chao W L, Su Y. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023, 2998-3009
62. Song C H, Wu J, Washington C, Sadler B M, Chao W L, Su Y. Llm-planner：基于大型语言模型的少样本具身代理规划。载于：IEEE/CVF国际计算机视觉会议论文集。2023, 2998-3009

62. Huang W, Xia F, Xiao T, Chan H, Liang J, Florence P, Zeng A, Tompson J, Mordatch I, Chebotar Y, others.

Inner monologue: Embodied reasoning through planning with language models. arXiv preprint arXiv:2207.05608, 2022

63. Huang W, Xia F, Xiao T, Chan H, Liang J, Florence P, Zeng A, Tompson J, Mordatch I, Chebotar Y, 等. 内心独白：通过语言模型规划实现具身推理。arXiv预印本 arXiv:2207.05608, 2022

63. Madaan A, Tandon N, Gupta P, Hallinan S, Gao L, Wiegreffe S, Alon U, Dziri N, Prabhumoye S, Yang Y, others . Self-refine: Iterative refinement with self-feedback. Advances in Neural Information Processing Systems, 2024, 36

64. Madaan A, Tandon N, Gupta P, Hallinan S, Gao L, Wiegreffe S, Alon U, Dziri N, Prabhumoye S, Yang Y, 等. 自我优化：基于自我反馈的迭代精炼。《神经信息处理系统进展》(Advances in Neural Information Processing Systems), 2024, 36

64. Miao N, Teh Y W, Rainforth T. Selfcheck: Using llms to zero-shot check their own step-by-step reasoning. In: The Twelfth International Conference on Learning Representations. 2023

65. Miao N, Teh Y W, Rainforth T. Selfcheck：利用大型语言模型零样本检查自身的逐步推理。载于：第十二届国际学习表征会议。2023

65. Chen P L, Chang C S. Interact: Exploring the potentials of chatgpt as a cooperative agent. arXiv preprint arXiv:2308.01552, 2023

66. 陈鹏立, 张成松. Interact: 探索ChatGPT作为协作代理的潜力. arXiv预印本 arXiv:2308.01552, 2023

66. Chen Z, Zhou K, Zhang B, Gong Z, Zhao W X, Wen J R. Chatcot: Tool-augmented chain-of-thought reasoning on\\chat-based large language models. arXiv preprint arXiv:2305.14323, 2023

67. 陈卓, 周康, 张斌, 龚志, 赵文轩, 温建荣. Chatcot: 基于聊天的大型语言模型的工具增强链式思维推理. arXiv预印本 arXiv:2305.14323, 2023

67. Nakano R, Hilton J, Balaji S, Wu J, Ouyang L, Kim C, Hesse C, Jain S, Kosaraju V, Saunders W, others . Webgpt: Browser-assisted question-answering with human feedback. arXiv preprint arXiv:2112.09332, 2021

68. 中野亮, 希尔顿, 巴拉吉, 吴杰, 欧阳亮, 金昌, 赫斯, 贾因, 科萨拉朱, 桑德斯等. WebGPT: 结合浏览器辅助和人类反馈的问答系统. arXiv预印本 arXiv:2112.09332, 2021

68. Ruan J, Chen Y, Zhang B, Xu Z, Bao T, Du G, Shi S, Mao H, Zeng X, Zhao R. TPTU: Task planning and tool usage of large language model-based AI agents. arXiv preprint arXiv:2308.03427, 2023

69. 阮军, 陈阳, 张斌, 徐志, 鲍涛, 杜刚, 史松, 毛浩, 曾翔, 赵锐. TPTU: 基于大型语言模型的AI代理的任务规划与工具使用. arXiv预印本 arXiv:2308.03427, 2023

69. Patil S G, Zhang T, Wang X, Gonzalez J E. Gorilla: Large language model connected with massive apis. arXiv preprint arXiv:2305.15334, 2023

70. Patil S G, 张涛, 王翔, Gonzalez J E. Gorilla: 连接海量API的大型语言模型. arXiv预印本 arXiv:2305.15334, 2023

70. Li M, Song F, Yu B, Yu H, Li Z, Huang F, Li Y. Api-bank: A benchmark for tool-augmented llms. arXiv preprint arXiv:2304.08244, 2023

71. 李明, 宋飞, 余斌, 余浩, 李志, 黄峰, 李阳. API-bank: 工具增强大型语言模型的基准测试. arXiv预印本 arXiv:2304.08244, 2023

71. Song Y, Xiong W, Zhu D, Li C, Wang K, Tian Y, Li S. Restgpt: Connecting large language models with real-world applications via restful apis. arXiv preprint arXiv:2306.06624, 2023

72. 宋洋, 熊伟, 朱丹, 李超, 王凯, 田阳, 李爽. RestGPT: 通过RESTful API将大型语言模型与现实应用连接. arXiv预印本 arXiv:2306.06624, 2023

72. Liang Y, Wu C, Song T, Wu W, Xia Y, Liu Y, Ou Y, Lu S, Ji L, Mao S, others . Taskmatrix. ai: Completing

tasks by connecting foundation models with millions of apis. Intelligent Computing, 2024, 3: 0063

73. 梁颖, 吴超, 宋涛, 吴伟, 夏阳, 刘洋, 欧阳毅, 卢森, 纪磊, 毛松等. TaskMatrix.ai: 通过连接基础模型与数百万API完成任务. 智能计算, 2024, 3: 0063

73. Karpas E, Abend O, Belinkov Y, Lenz B, Lieber O, Ratner N, Shoham Y, Bata H, Levine Y, Leyton-Brown K, others . Mrkl systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning. arXiv preprint arXiv:2205.00445, 2022

74. Karpas E, Abend O, Belinkov Y, Lenz B, Lieber O, Ratner N, Shoham Y, Bata H, Levine Y, Leyton-Brown K等. MRKL系统: 一种结合大型语言模型、外部知识源和离散推理的模块化神经符号架构. arXiv预印本 arXiv:2205.00445, 2022

74. Ge Y, Hua W, Mei K, Tan J, Xu S, Li Z, Zhang Y, others . Openagi: When llm meets domain experts. Advances in Neural Information Processing Systems, 2024, 36

75. Ge Y, Hua W, Mei K, Tan J, Xu S, Li Z, Zhang Y, 等. Openagi：当大型语言模型（LLM）遇到领域专家. 神经信息处理系统进展, 2024, 36

75. Surís D, Menon S, Vondrick C. Vipergpt: Visual inference via python execution for reasoning. arXiv preprint arXiv:2303.08128, 2023

76. Surís D, Menon S, Vondrick C. Vipergpt：通过Python执行实现视觉推理. arXiv预印本 arXiv:2303.08128, 2023

76. Bran A M, Cox S, White A D, Schwaller P. Chem-crow: Augmenting large-language models with chemistry tools. arXiv preprint arXiv:2304.05376, 2023

77. Bran A M, Cox S, White A D, Schwaller P. Chem-crow：利用化学工具增强大型语言模型. arXiv预印本 arXiv:2304.05376, 2023

77. Yang Z, Li L, Wang J, Lin K, Azarnasab E, Ahmed F, Liu Z, Liu C, Zeng M, Wang L. Mm-react: Prompting chatgpt for multimodal reasoning and action. arXiv preprint arXiv:2303.11381, 2023

78. Yang Z, Li L, Wang J, Lin K, Azarnasab E, Ahmed F, Liu Z, Liu C, Zeng M, Wang L. Mm-react：利用ChatGPT进行多模态推理与行动提示. arXiv预印本 arXiv:2303.11381, 2023

78. Gao C, Lan X, Lu Z, Mao J, Piao J, Wang H, Jin D, Li Y. S3: Social-network simulation system with large language model-empowered agents. arXiv preprint arXiv:2307.14984, 2023

79. Gao C, Lan X, Lu Z, Mao J, Piao J, Wang H, Jin D, Li Y. S3：基于大型语言模型赋能代理的社交网络仿真系统. arXiv预印本 arXiv:2307.14984, 2023

79. Ahn M, Brohan A, Brown N, Chebotar Y, Cortes O, David B, Finn C, Fu C, Gopalakrishnan K, Hausman K, others. Do as i can, not as i say: Grounding language in robotic affordances. arXiv preprint arXiv:2204.01691, 2022

80. Ahn M, Brohan A, Brown N, Chebotar Y, Cortes O, David B, Finn C, Fu C, Gopalakrishnan K, Hausman K, 等. 按我所能行事，而非我所言：将语言与机器人可供性（affordances）相结合. arXiv预印本 arXiv:2204.01691, 2022

80. Park J S, Popowski L, Cai C, Morris M R, Liang P, Bernstein M S. Social simulacra: Creating populated prototypes for social computing systems. In: Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology. 2022, 1-18

81. Park J S, Popowski L, Cai C, Morris M R, Liang P, Bernstein M S. 社会模拟：为社会计算系统创建人口原型. 载于：第35届年度ACM用户界面软件与技术研讨会论文集. 2022, 1-18

81. Li G, Hammoud H A A K, Itani H, Khizbullin D, Ghanem B. Camel: Communicative agents for" mind" exploration of large scale language model society. arXiv preprint arXiv:2303.17760, 2023

82. Li G, Hammoud H A A K, Itani H, Khizbullin D, Ghanem B. Camel：用于大型语言模型社会“心智”探索的交流代理. arXiv预印本 arXiv:2303.17760, 2023

82. al. e T. Auto-GPT. https://github.com/ Significant-Gravitas/Auto-GPT, 2023
83. 等. Auto-GPT. https://github.com/ Significant-Gravitas/Auto-GPT, 2023

83. Liu R, Yang R, Jia C, Zhang G, Zhou D, Dai A M, Yang D, Vosoughi S. Training socially aligned language models in simulated human society. arXiv preprint arXiv:2305.16960, 2023
84. Liu R, Yang R, Jia C, Zhang G, Zhou D, Dai A M, Yang D, Vosoughi S. 在模拟人类社会中训练社会对齐的语言模型. arXiv预印本 arXiv:2305.16960, 2023

84. Chen L, Wang L, Dong H, Du Y, Yan J, Yang F, Li S, Zhao P, Qin S, Rajmohan S, others. Introspective tips: Large language model for in-context decision making. arXiv preprint arXiv:2305.11598, 2023
85. Chen L, Wang L, Dong H, Du Y, Yan J, Yang F, Li S, Zhao P, Qin S, Rajmohan S, 等. 内省提示：用于上下文决策的大型语言模型. arXiv预印本 arXiv:2305.11598, 2023

85. Liu H, Sferrazza C, Abbeel P. Chain of hindsight aligns language models with feedback. In: The Twelfth International Conference on Learning Representations. 2023
86. Liu H, Sferrazza C, Abbeel P. 回顾链（Chain of hindsight）使语言模型与反馈保持一致. 载于：第十二届国际学习表征会议. 2023

86. Yao S, Chen H, Yang J, Narasimhan K. Webshop: Towards scalable real-world web interaction with grounded language agents. Advances in Neural Information Processing Systems, 2022, 35: 20744-20757
87. Yao S, Chen H, Yang J, Narasimhan K. Webshop：面向可扩展真实网络交互的基于语言代理的系统. 神经信息处理系统进展, 2022, 35: 20744-20757

87. Dan Y, Lei Z, Gu Y, Li Y, Yin J, Lin J, Ye L, Tie Z, Zhou Y ,Wang Y ,others. Educhat: A large-scale language model-based chatbot system for intelligent education. arXiv preprint arXiv:2308.02773, 2023
88. Dan Y, Lei Z, Gu Y, Li Y, Yin J, Lin J, Ye L, Tie Z, Zhou Y ,Wang Y ,等. Educhat：基于大型语言模型的智能教育聊天机器人系统. arXiv预印本 arXiv:2308.02773, 2023

88. Deng X, Gu Y, Zheng B, Chen S, Stevens S, Wang B, Sun H, Su Y. Mind2web: Towards a generalist agent for the web. Advances in Neural Information Processing Systems, 2024, 36
89. Deng X, Gu Y, Zheng B, Chen S, Stevens S, Wang B, Sun H, Su Y. Mind2web：迈向通用网络代理. 神经信息处理系统进展, 2024, 36

89. Sun R, Arik S O, Nakhost H, Dai H, Sinha R, Yin P, Pfister T. Sql-palm: Improved large language modeladaptation for text-to-sql. arXiv preprint arXiv:2306.00739, 2023
90. Sun R, Arik S O, Nakhost H, Dai H, Sinha R, Yin P, Pfister T. Sql-palm：改进的大型语言模型文本到SQL适配. arXiv预印本 arXiv:2306.00739, 2023

90. Yao W, Heinecke S, Niebles J C, Liu Z, Feng Y, Xue L, Murthy R, Chen Z, Zhang J, Arpit D, Xu R, Mui P, Wang H, Xiong C, Savarese S. Retroformer: Retrospective large language agents with policy gradient optimization, 2023
91. Yao W, Heinecke S, Niebles J C, Liu Z, Feng Y, Xue L, Murthy R, Chen Z, Zhang J, Arpit D, Xu R, Mui P, Wang H, Xiong C, Savarese S. Retroformer：基于策略梯度优化的回顾性大型语言代理，2023

91. Shu Y, Gu H, Zhang P, Zhang H, Lu T, Li D, Gu N. Rah! recsys-assistant-human: A human-central recommendation framework with large language models. arXiv preprint arXiv:2308.09904, 2023
92. Shu Y, Gu H, Zhang P, Zhang H, Lu T, Li D, Gu N. Rah! recsys-assistant-human：以人为中心的大型语言模型推荐框架。arXiv预印本 arXiv:2308.09904，2023

92. Mandi Z, Jain S, Song S. Roco: Dialectic multi-robot collaboration with large language models. arXiv preprint arXiv:2307.04738, 2023
93. Mandi Z, Jain S, Song S. Roco：基于大型语言模型的辩证多机器人协作。arXiv预印本 arXiv:2307.04738，2023

93. Zhang C, Liu L, Wang J, Wang C, Sun X, Wang H, Cai M. Prefer: Prompt ensemble learning via feedback-reflect-refine. arXiv preprint arXiv:2308.12033, 2023
94. Zhang C, Liu L, Wang J, Wang C, Sun X, Wang H, Cai M. Prefer：通过反馈-反思-精炼的提示集成学习。arXiv预印本 arXiv:2308.12033，2023

94. Du Y, Li S, Torralba A, Tenenbaum J B, Mordatch I. Improving factuality and reasoning in language models through multiagent debate. arXiv preprint arXiv:2305.14325, 2023
95. Du Y, Li S, Torralba A, Tenenbaum J B, Mordatch I. 通过多智能体辩论提升语言模型的事实性和推理能力。arXiv预印本 arXiv:2305.14325，2023

95. Yang Z, Liu J, Han Y, Chen X, Huang Z, Fu B, Yu G. Appagent: Multimodal agents as smartphone users. arXiv preprint arXiv:2312.13771, 2023
96. Yang Z, Liu J, Han Y, Chen X, Huang Z, Fu B, Yu G. Appagent：作为智能手机用户的多模态代理。arXiv预印本 arXiv:2312.13771，2023

96. Madaan A, Tandon N, Clark P, Yang Y. Memory-assisted prompt editing to improve GPT-3 after deployment. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. 2022
97. Madaan A, Tandon N, Clark P, Yang Y. 记忆辅助的提示编辑以提升部署后的GPT-3表现。载于：2023年自然语言处理实证方法会议论文集，2022

97. Colas C, Teodorescu L, Oudeyer P Y, Yuan X, Côté M A. Augmenting autotelic agents with large language models. arXiv preprint arXiv:2305.12487, 2023
98. Colas C, Teodorescu L, Oudeyer P Y, Yuan X, Côté M A. 利用大型语言模型增强自驱动代理。arXiv预印本 arXiv:2305.12487，2023

98. Nascimento N, Alencar P, Cowan D. Self-adaptive large language model (llm)-based multiagent systems. In: 2023 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C). 2023, 104-109
99. Nascimento N, Alencar P, Cowan D. 基于大型语言模型（LLM）的自适应多智能体系统。载于：2023年IEEE自主管理计算与自组织系统国际会议伴随会议（ACSOS-C），2023，104-109

99. Saha S, Hase P, Bansal M. Can language models teach weaker agents? teacher explanations improve students via theory of mind. arXiv preprint arXiv:2306.09299, 2023
100. Saha S, Hase P, Bansal M. 语言模型能教导能力较弱的代理吗？教师解释通过心智理论提升学生表现。arXiv预印本 arXiv:2306.09299，2023

100. Zhuge M, Liu H, Faccio F, Ashley D R, Csordás R, Gopalakrishnan A, Hamdi A, Hammoud H A A K, Herrmann V, Irie K, others. Mindstorms in natural language-based societies of mind. arXiv preprint arXiv:2305.17066, 2023
101. Zhuge M, Liu H, Faccio F, Ashley D R, Csordás R, Gopalakrishnan A, Hamdi A, Hammoud H A A K, Herrmann V, Irie K, 等。基于自然语言的心智风暴社会。arXiv预印本 arXiv:2305.17066，2023

101. Aher G V, Arriaga R I, Kalai A T. Using large language models to simulate multiple humans and replicate human subject studies. In: International Conference on Machine Learning. 2023, 337-371
102. Aher G V, Arriaga R I, Kalai A T. 利用大型语言模型模拟多个人类并复现人类受试者研究。载于：国际机器学习会议，2023，337-371

102. Akata E, Schulz L, Coda-Forno J, Oh S J, Bethge M, Schulz E. Playing repeated games with large language models. arXiv preprint arXiv:2305.16867, 2023
103. Akata E, Schulz L, Coda-Forno J, Oh S J, Bethge M, Schulz E. 使用大型语言模型进行重复博弈。arXiv预印本 arXiv:2305.16867，2023

103. Ma Z, Mei Y, Su Z. Understanding the benefits and challenges of using large language model-based conversational agents for mental well-being support. In: AMIA Annual Symposium Proceedings. 2023, 1105
104. Ma Z, Mei Y, Su Z. 理解基于大型语言模型的对话代理在心理健康支持中的优势与挑战。载于：AMIA年度研讨会论文集，2023，1105

104. Ziems C, Held W, Shaikh O, Chen J, Zhang Z, Yang D. Can large language models transform computational social science? arXiv preprint arXiv:2305.03514, 2023
105. Ziems C, Held W, Shaikh O, Chen J, Zhang Z, Yang D. 大型语言模型能否变革计算社会科学？arXiv预印本 arXiv:2305.03514，2023

105. Horton J J. Large language models as simulated economic agents: What can we learn from homo silicus? Technical report, National Bureau of Economic Research, 2023
106. Horton J J. 大型语言模型作为模拟经济代理：我们能从“硅人”（homo silicus）学到什么？技术报告，美国国家经济研究局，2023

106. Li S, Yang J, Zhao K. Are you in a masquerade? exploring the behavior and impact of large language model driven social bots in online social networks. arXiv preprint arXiv:2307.10337, 2023
107. 李思, 杨杰, 赵凯. 你在伪装舞会中吗？探讨大型语言模型驱动的社交机器人在在线社交网络中的行为及影响. arXiv预印本 arXiv:2307.10337, 2023

107. Li C, Su X, Fan C, Han H, Xue C, Zheng C. Quantifying the impact of large language models on collective opinion dynamics. arXiv preprint arXiv:2308.03313, 2023
108. 李超, 苏翔, 范超, 韩浩, 薛晨, 郑晨. 量化大型语言模型对集体舆论动态的影响. arXiv预印本 arXiv:2308.03313, 2023

108. Kovač G, Portelas R, Dominey P F, Oudeyer P Y. The socialai school: Insights from developmental psychology towards artificial socio-cultural agents. arXiv preprint arXiv:2307.07871, 2023
109. Kovač G, Portelas R, Dominey P F, Oudeyer P Y. 社会人工智能学校：来自发展心理学对人工社会文化代理的启示. arXiv预印本 arXiv:2307.07871, 2023

109. Williams R, Hosseinichimeh N, Majumdar A, Ghaf-farzadegan N. Epidemic modeling with generative agents. arXiv preprint arXiv:2307.04986, 2023
110. Williams R, Hosseinichimeh N, Majumdar A, Ghaf-farzadegan N. 利用生成代理进行流行病建模. arXiv预印本 arXiv:2307.04986, 2023

110. Jinxin S, Jiabao Z, Yilei W, Xingjiao W, Jiawen L, Liang H. Cgmi: Configurable general multi-agent interaction framework. arXiv preprint arXiv:2308.12503, 2023
111. 金鑫, 贾宝, 依蕾, 邢娇, 嘉文, 梁浩. CGMI：可配置通用多智能体交互框架. arXiv预印本 arXiv:2308.12503, 2023

111. Cui J, Li Z, Yan Y, Chen B, Yuan L. Chatlaw: Open-source legal large language model with integrated external knowledge bases. arXiv preprint arXiv:2306.16092, 2023
112. 崔杰, 李志, 颜阳, 陈斌, 袁亮. ChatLaw：集成外部知识库的开源法律大型语言模型. arXiv预印本 arXiv:2306.16092, 2023

112. Hamilton S. Blind judgement: Agent-based supreme court modelling with gpt. arXiv preprint arXiv:2301.05327, 2023
113. Hamilton S. 盲目判断：基于代理的最高法院建模与GPT. arXiv预印本 arXiv:2301.05327, 2023

113. Bail C A. Can generative ai improve social science? 2023
114. Bail C A. 生成式人工智能能改善社会科学吗？2023

114. Boiko D A, MacKnight R, Gomes G. Emergent autonomous scientific research capabilities of large language

models. arXiv preprint arXiv:2304.05332, 2023

115. Boiko D A, MacKnight R, Gomes G. 大型语言模型的自主科学研究能力的涌现. arXiv预印本 arXiv:2304.05332, 2023

115. Kang Y, Kim J. Chatmof: An autonomous ai system for predicting and generating metal-organic frameworks. arXiv preprint arXiv:2308.01423, 2023

116. Kang Y, Kim J. ChatMOF：用于预测和生成金属有机框架的自主人工智能系统. arXiv预印本 arXiv:2308.01423, 2023

116. Swan M, Kido T, Roland E, Santos R P d. Math agents: Computational infrastructure, mathematical embedding, and genomics. arXiv preprint arXiv:2307.02502, 2023

117. Swan M, Kido T, Roland E, Santos R P d. 数学代理：计算基础设施、数学嵌入与基因组学. arXiv预印本 arXiv:2307.02502, 2023

117. Drori I, Zhang S, Shuttleworth R, Tang L, Lu A, Ke E, Liu K, Chen L, Tran S, Cheng N, others. A neural network solves, explains, and generates university math problems by program synthesis and few-shot learning at human level. Proceedings of the National Academy of Sciences, 2022, 119(32): e2123433119

118. Drori I, Zhang S, Shuttleworth R, Tang L, Lu A, Ke E, Liu K, Chen L, Tran S, Cheng N, 等. 通过程序合成和少样本学习，神经网络以人类水平解决、解释并生成大学数学问题. 美国国家科学院院刊, 2022, 119(32): e2123433119

118. Chen M, Tworek J, Jun H, Yuan Q, Pinto H P d O, Kaplan J, Edwards H, Burda Y, Joseph N, Brockman G, others . Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374, 2021

119. Chen M, Tworek J, Jun H, Yuan Q, Pinto H P d O, Kaplan J, Edwards H, Burda Y, Joseph N, Brockman G, 等. 评估基于代码训练的大型语言模型. arXiv预印本 arXiv:2107.03374, 2021

119. Liffiton M, Sheese B E, Savelka J, Denny P. Codehelp: Using large language models with guardrails for scalable support in programming classes. In: Proceedings of the 23rd Koli Calling International Conference on Computing Education Research. 2023, 1-11

120. Liffiton M, Sheese B E, Savelka J, Denny P. CodeHelp：在编程课程中使用带有保护措施的大型语言模型进行可扩展支持. 载于：23rd Koli Calling国际计算教育研究会议论文集. 2023, 1-11

120. Matelsky J K, Parodi F, Liu T, Lange R D, Kording K P. A large language model-assisted education tool to provide feedback on open-ended responses. arXiv preprint arXiv:2308.02439, 2023

121. Matelsky J K, Parodi F, Liu T, Lange R D, Kording K P. 一种大型语言模型辅助的教育工具，用于对开放式回答提供反馈. arXiv预印本 arXiv:2308.02439, 2023

121. Grossmann I, Feinberg M, Parker D C, Christakis N A, Tetlock P E, Cunningham W A. Ai and the transformation of social science research. Science, 2023, 380(6650): 1108-1109

122. Grossmann I, Feinberg M, Parker D C, Christakis N A, Tetlock P E, Cunningham W A. 人工智能与社会科学研究的变革. 科学, 2023, 380(6650): 1108-1109

122. Zhou X, Li G, Liu Z. Llm as dba. arXiv preprint arXiv:2308.05481, 2023

123. 周晓, 李刚, 刘志. 大型语言模型作为数据库管理员(LLM as DBA). arXiv预印本 arXiv:2308.05481, 2023

123. He Z, Wu H, Zhang X, Yao X, Zheng S, Zheng H, Yu B. Chateda: A large language model powered autonomous agent for eda. In: 2023 ACM/IEEE 5th Workshop on Machine Learning for CAD (MLCAD). 2023, 1-6

124. 何志, 吴浩, 张翔, 姚翔, 郑爽, 郑浩, 余斌. Chateda：一种基于大型语言模型的自主电子设计自动化(EDA)代理. 载于：2023年ACM/IEEE第五届计算机辅助设计机器学习研讨会(MLCAD). 2023, 1-6

124. Huang X, Lian J, Lei Y, Yao J, Lian D, Xie X. Recommender ai agent: Integrating large language models for

interactive recommendations. arXiv preprint arXiv:2308.16505, 2023

125. 黄晓, 连军, 雷阳, 姚军, 连东, 谢翔. 推荐AI代理：集成大型语言模型实现交互式推荐. arXiv预印本 arXiv:2308.16505, 2023

125. Deng G, Liu Y, Mayoral-Vilches V, Liu P, Li Y, Xu Y, Zhang T, Liu Y, Pinzger M, Rass S. Pentestgpt: An llm-empowered automatic penetration testing tool. arXiv preprint arXiv:2308.06782, 2023

126. 邓刚, 刘洋, Mayoral-Vilches V, 刘鹏, 李阳, 徐阳, 张涛, 刘洋, Pinzger M, Rass S. PentestGPT：一种基于大型语言模型的自动渗透测试工具. arXiv预印本 arXiv:2308.06782, 2023

126. al. e S. Smolmodels. https://github.com/ smol-ai/developer,2023

127. 等. Smolmodels. https://github.com/smol-ai/developer, 2023

127. al. e M U. DemoGPT. https://github.com/ melih-unsal/DemoGPT,2023

128. 等. DemoGPT. https://github.com/melih-unsal/DemoGPT, 2023

128. al. e A O. GPT engineer. https://github.com/ AntonOsika/gpt-engineer, 2023

129. 等. GPT engineer. https://github.com/AntonOsika/gpt-engineer, 2023

129. Xia Y, Shenoy M, Jazdi N, Weyrich M. Towards autonomous system: flexible modular production system enhanced with large language model agents. arXiv preprint arXiv:2304.14721, 2023

130. 夏阳, Shenoy M, Jazdi N, Weyrich M. 迈向自主系统：结合大型语言模型代理的灵活模块化生产系统. arXiv预印本 arXiv:2304.14721, 2023

130. Ogundare O, Madasu S, Wiggins N. Industrial engineering with large language models: A case study of chatgpt's performance on oil & gas problems. arXiv preprint arXiv:2304.14354, 2023

131. Ogundare O, Madasu S, Wiggins N. 利用大型语言模型的工业工程：ChatGPT在油气问题上的性能案例研究. arXiv预印本 arXiv:2304.14354, 2023

131. Zhang C, Yang K, Hu S, Wang Z, Li G, Sun Y, Zhang C, Zhang Z, Liu A, Zhu S C, others . Proagent: Building proactive cooperative ai with large language models. arXiv preprint arXiv:2308.11339, 2023

132. 张超, 杨凯, 胡松, 王志, 李刚, 孙阳, 张超, 张志, 刘安, 朱松纯, 等. Proagent：构建基于大型语言模型的主动协作人工智能. arXiv预印本 arXiv:2308.11339, 2023

132. Hu B, Zhao C, others. Enabling intelligent interactions between an agent and an 1 lm : A reinforcement learning approach. arXiv:2306.03604, 2023

133. 胡斌, 赵晨, 等. 实现代理与1 lm之间的智能交互：一种强化学习方法. arXiv:2306.03604, 2023

133. Wu Y, Min S Y, Bisk Y, Salakhutdinov R, Azaria A, Li Y, Mitchell T, Prabhumoye S. Plan, eliminate, and track-language models are good teachers for embodied agents. arXiv preprint arXiv:2305.02412, 2023

134. 吴洋, Min S Y, Bisk Y, Salakhutdinov R, Azaria A, 李阳, Mitchell T, Prabhumoye S. 计划、消除与追踪——语言模型是具身代理的优秀教师. arXiv预印本 arXiv:2305.02412, 2023

134. Zhang D, Chen L, Zhang S, Xu H, Zhao Z, Yu K. Large language models are semi-parametric reinforcement learning agents. Advances in Neural Information Processing Systems, 2024, 36

135. 张东, 陈磊, 张帅, 徐浩, 赵志, 余凯. 大型语言模型是半参数强化学习代理. 神经信息处理系统进展(NeurIPS), 2024, 36

135. Di Palo N, Byravan A, Hasenclever L, Wulfmeier M, Heess N, Riedmiller M. Towards a unified agent with foundation models. In: Workshop on Reincarnating Reinforcement Learning at ICLR 2023. 2023

136. Di Palo N, Byravan A, Hasenclever L, Wulfmeier M, Heess N, Riedmiller M. 迈向基于基础模型的统一代理. 载于：ICLR 2023强化学习复兴研讨会. 2023

136. Wu J, Antonova R, Kan A, Lepert M, Zeng A, Song S, Bohg J, Rusinkiewicz S, Funkhouser T. Tidybot: Personalized robot assistance with large language models. arXiv preprint arXiv:2305.05658, 2023
137. 吴俊, Antonova R, Kan A, Lepert M, Zeng A, Song S, Bohg J, Rusinkiewicz S, Funkhouser T. Tidybot：基于大型语言模型的个性化机器人助手. arXiv预印本 arXiv:2305.05658, 2023

137. Wu Z, Wang Z, Xu X, Lu J, Yan H. Embodied task planning with large language models. arXiv preprint arXiv:2307.01848, 2023
138. 吴志, 王志, 徐翔, 卢军, 颜浩. 基于大型语言模型的具身任务规划. arXiv预印本 arXiv:2307.01848, 2023

138. Dasgupta I, Kaeser-Chen C, Marino K, Ahuja A, Babayan S, Hill F, Fergus R. Collaborating with language models for embodied reasoning. arXiv preprint arXiv:2302.00763, 2023
139. Dasgupta I, Kaeser-Chen C, Marino K, Ahuja A, Babayan S, Hill F, Fergus R. 与语言模型协作进行具身推理。arXiv预印本 arXiv:2302.00763, 2023

139. Nottingham K, Ammanabrolu P, Suhr A, Choi Y, Ha-jishirzi H, Singh S, Fox R. Do embodied agents dream of pixelated sheep?: Embodied decision making using language guided world modelling. In: Workshop on Reincarnating Reinforcement Learning at ICLR 2023. 2023
140. Nottingham K, Ammanabrolu P, Suhr A, Choi Y, Ha-jishirzi H, Singh S, Fox R. 具身代理会梦见像素化的羊吗？：使用语言引导的世界建模进行具身决策。收录于：ICLR 2023“重生强化学习”研讨会，2023

140. Zhou W, Peng X, Riedl M. Dialogue shaping: Empowering agents through npc interaction. arXiv preprint arXiv:2307.15833, 2023
141. Zhou W, Peng X, Riedl M. 对话塑造：通过NPC交互赋能代理。arXiv预印本 arXiv:2307.15833, 2023

141. Li H, Hao Y, Zhai Y, Qian Z. The hitchhiker's guide to program analysis: A journey with large language models. arXiv preprint arXiv:2308.00245, 2023
142. Li H, Hao Y, Zhai Y, Qian Z. 程序分析指南：与大型语言模型（Large Language Models）同行的旅程。arXiv预印本 arXiv:2308.00245, 2023

142. al. e R. AgentGPT. https://github.com/ reworkd/AgentGPT, 2023
143. 等人 R. AgentGPT。https://github.com/reworkd/AgentGPT, 2023

143. al. e E. Ai-legion. https://github.com/ eumemic/ai-legion,2023
144. 等人 E. Ai-legion。https://github.com/eumemic/ai-legion, 2023

144. al. e J X. Agixt. https://github.com/ Josh-XT/AGiXT, 2023
145. 等人 J X. Agixt。https://github.com/Josh-XT/AGiXT, 2023

145. al. e C. Xlang. https://github.com/ xlang-ai/xlang,2023
146. 等人 C. Xlang。https://github.com/xlang-ai/xlang, 2023

146. al. e N. Babyagi. https://github.com/ yoheinakajima,2023
147. 等人 N. Babyagi。https://github.com/yoheinakajima, 2023

147. Chase H. langchain. https://docs.langchain.com/docs/, 2023
148. Chase H. langchain。https://docs.langchain.com/docs/, 2023

148. al. e A M. WorkGPT. https://github.com/ team-openpm/workgpt, 2023
149. 等人 A M. WorkGPT。https://github.com/team-openpm/workgpt, 2023

149. al. e F R. LoopGPT. https://github.com/ farizrahman4u/loopgpt,2023
150. 等人 F R. LoopGPT。https://github.com/farizrahman4u/loopgpt, 2023

150. al. e A E. GPT-researcher. https://github.com/assafelovic/gpt-researcher,2023
151. 等人 A E. GPT-researcher。https://github.com/assafelovic/gpt-researcher, 2023

151. Qin Y, Hu S, Lin Y, Chen W, Ding N, Cui G, Zeng Z, Huang Y, Xiao C, Han C, others. Tool learning with foundation models. arXiv preprint arXiv:2304.08354, 2023
152. Qin Y, Hu S, Lin Y, Chen W, Ding N, Cui G, Zeng Z, Huang Y, Xiao C, Han C, 等人。基于基础模型（Foundation Models）的工具学习。arXiv预印本 arXiv:2304.08354, 2023

152. Face H. transformers-agent. https: //huggingface.co/docs/transformers/ transformers_agents,2023
153. Face H. transformers-agent。https://huggingface.co/docs/transformers/transformers_agents, 2023

153. al. e E. Miniagi. https://github.com/ muellerberndt/mini-agi,2023
154. 等人 E. Miniagi。https://github.com/muellerberndt/mini-agi, 2023

154. al. e T. Superagi. https://github.com/ TransformerOptimus/SuperAGI,2023
155. al. e T. Superagi. https://github.com/ TransformerOptimus/SuperAGI,2023

155. Wu Q, Bansal G, Zhang J, Wu Y, Zhang S, Zhu E, Li B, Jiang L, Zhang X, Wang C. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. arXiv preprint arXiv:2308.08155, 2023
156. Wu Q, Bansal G, Zhang J, Wu Y, Zhang S, Zhu E, Li B, Jiang L, Zhang X, Wang C. Autogen：通过多智能体对话框架实现下一代大型语言模型（LLM）应用。arXiv预印本 arXiv:2308.08155, 2023

156. Chen W, Su Y, Zuo J, Yang C, Yuan C, Qian C, Chan C M, Qin Y, Lu Y, Xie R, others. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. arXiv preprint arXiv:2308.10848, 2023
157. Chen W, Su Y, Zuo J, Yang C, Yuan C, Qian C, Chan C M, Qin Y, Lu Y, Xie R, 等. Agentverse：促进多智能体协作并探索智能体的涌现行为。arXiv预印本 arXiv:2308.10848, 2023

157. Lee M, Srivastava M, Hardy A, Thickstun J, Durmus E, Paranjape A, Gerard-Ursin I, Li X L, Ladhak F, Rong F, others. Evaluating human-language model interaction. arXiv preprint arXiv:2212.09746, 2022
158. Lee M, Srivastava M, Hardy A, Thickstun J, Durmus E, Paranjape A, Gerard-Ursin I, Li X L, Ladhak F, Rong F, 等. 评估人类与语言模型的交互。arXiv预印本 arXiv:2212.09746, 2022

158. Chan C M, Chen W, Su Y, Yu J, Xue W, Zhang S, Fu J, Liu Z. Chateval: Towards better llm-based evaluators through multi-agent debate. arXiv preprint arXiv:2308.07201, 2023
159. Chan C M, Chen W, Su Y, Yu J, Xue W, Zhang S, Fu J, Liu Z. Chateval：通过多智能体辩论提升基于大型语言模型的评估器。arXiv预印本 arXiv:2308.07201, 2023

159. Kang S, Yoon J, Yoo S. Large language models are few-shot testers: Exploring llm-based general bug reproduction. In: 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE). 2023, 2312-2323
160. Kang S, Yoon J, Yoo S. 大型语言模型是少样本测试者：探索基于LLM的一般性缺陷复现。载于：2023 IEEE/ACM第45届国际软件工程会议（ICSE），2023，2312-2323

160. Jalil S, Rafi S, LaToza T D, Moran K, Lam W. Chatgpt and software testing education: Promises & perils. In: 2023 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW). 2023, 4130-4137
161. Jalil S, Rafi S, LaToza T D, Moran K, Lam W. ChatGPT与软件测试教育：机遇与风险。载于：2023 IEEE国际软件测试、验证与验证研讨会（ICSTW），2023，4130-4137

161. Mehta N, Teruel M, Sanz P F, Deng X, Awadallah A H, Kiseleva J. Improving grounded language

understanding in a collaborative environment by interacting with agents through help feedback. arXiv preprint arXiv:2304.10750, 2023

162. Mehta N, Teruel M, Sanz P F, Deng X, Awadallah A H, Kiseleva J. 通过帮助反馈与智能体交互，提升协作环境中的基础语言理解能力。arXiv预印本 arXiv:2304.10750, 2023

162. Chen A, Phang J, Parrish A, Padmakumar V, Zhao C, Bowman S R, Cho K. Two failures of self-consistency in the multi-step reasoning of llms. arXiv preprint arXiv:2305.14279, 2023

163. Chen A, Phang J, Parrish A, Padmakumar V, Zhao C, Bowman S R, Cho K. 大型语言模型多步推理中的两种自洽性失败。arXiv预印本 arXiv:2305.14279, 2023

163. Choi M, Pei J, Kumar S, Shu C, Jurgens D. Do llms understand social knowledge? evaluating the sociability of large language models with socket benchmark. arXiv preprint arXiv:2305.14938, 2023

164. Choi M, Pei J, Kumar S, Shu C, Jurgens D. 大型语言模型理解社会知识吗？使用Socket基准评估大型语言模型的社交能力。arXiv预印本 arXiv:2305.14938, 2023

164. Zhang D, Chen L, Zhao Z, Cao R, Yu K. Mobile-env: An evaluation platform and benchmark for interactive agents in llm era. arXiv preprint arXiv:2305.08144, 2023

165. Zhang D, Chen L, Zhao Z, Cao R, Yu K. Mobile-env：面向大型语言模型时代交互智能体的评估平台与基准。arXiv预印本 arXiv:2305.08144, 2023

165. Chalamalasetti K, Götze J, Hakimov S, Madureira B, Sadler P, Schlangen D. clembench: Using game play to evaluate chat-optimized language models as conversational agents. arXiv preprint arXiv:2305.13455, 2023

166. Chalamalasetti K, Götze J, Hakimov S, Madureira B, Sadler P, Schlangen D. Clembench：利用游戏玩法评估针对聊天优化的语言模型作为对话智能体。arXiv预印本 arXiv:2305.13455, 2023

166. Lin J, Tomlin N, Andreas J, Eisner J. Decision-oriented dialogue for human-ai collaboration. arXiv preprint arXiv:2305.20076, 2023

167. Lin J, Tomlin N, Andreas J, Eisner J. 面向人机协作的决策导向对话。arXiv预印本 arXiv:2305.20076, 2023

167. Feldt R, Kang S, Yoon J, Yoo S. Towards autonomous testing agents via conversational large language models. arXiv preprint arXiv:2306.05152, 2023

168. Feldt R, Kang S, Yoon J, Yoo S. 通过对话式大型语言模型迈向自主测试智能体。arXiv预印本 arXiv:2306.05152, 2023

168. Liang Y, Zhu L, Yang Y. Tachikuma: Understading complex interactions with multi-character and novel objects by large language models. arXiv preprint arXiv:2307.12573, 2023

169. Liang Y, Zhu L, Yang Y. Tachikuma：大型语言模型理解多角色与新颖对象的复杂交互。arXiv预印本 arXiv:2307.12573, 2023

169. Liu X, Yu H, Zhang H, Xu Y, Lei X, Lai H, Gu Y, Ding H ,Men K,Yang K,others . Agentbench: Evaluating llms as agents. arXiv preprint arXiv:2308.03688, 2023

170. Liu X, Yu H, Zhang H, Xu Y, Lei X, Lai H, Gu Y, Ding H ,Men K,Yang K,等. Agentbench：评估大型语言模型作为智能体。arXiv预印本 arXiv:2308.03688, 2023

170. Liu Z, Yao W, Zhang J, Xue L, Heinecke S, Murthy R ,Feng Y ,Chen Z ,Niebles JC ,Arpit D ,others . Bo-laa: Benchmarking and orchestrating llm-augmented autonomous agents. arXiv preprint arXiv:2308.05960, 2023

171. Liu Z, Yao W, Zhang J, Xue L, Heinecke S, Murthy R ,Feng Y ,Chen Z ,Niebles JC ,Arpit D ,等. Bo-laa：基于大型语言模型增强的自主代理的基准测试与编排. arXiv预印本 arXiv:2308.05960, 2023

171. Xu B, Liu X, Shen H, Han Z, Li Y, Yue M, Peng Z, Liu Y, Yao Z, Xu D. Gentopia. ai: A collaborative platform for tool-augmented llms. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. 2023, 237-245

172. Xu B, Liu X, Shen H, Han Z, Li Y, Yue M, Peng Z, Liu Y, Yao Z, Xu D. Gentopia.ai：一个用于工具增强大型语言模型的协作平台. 载于：2023年自然语言处理实证方法会议系统演示论文集. 2023, 237-245

172. Huang J t, Lam M H, Li E J, Ren S, Wang W, Jiao W, Tu Z, Lyu M R. Emotionally numb or empathetic? evaluating how llms feel using emotionbench. arXiv preprint arXiv:2308.03656, 2023

173. Huang J t, Lam M H, Li E J, Ren S, Wang W, Jiao W, Tu Z, Lyu M R. 情感麻木还是共情？使用EmotionBench评估大型语言模型的情感感知. arXiv预印本 arXiv:2308.03656, 2023

173. Zhou S, Xu F F, Zhu H, Zhou X, Lo R, Sridhar A, Cheng X, Bisk Y, Fried D, Alon U, others . Webarena: A realistic web environment for building autonomous agents. arXiv preprint arXiv:2307.13854, 2023

174. Zhou S, Xu F F, Zhu H, Zhou X, Lo R, Sridhar A, Cheng X, Bisk Y, Fried D, Alon U, 等. Webarena：构建自主代理的真实网络环境. arXiv预印本 arXiv:2307.13854, 2023

174. Banerjee D, Singh P, Avadhanam A, Srivastava S. Benchmarking llm powered chatbots: methods and metrics. arXiv preprint arXiv:2308.04624, 2023

175. Banerjee D, Singh P, Avadhanam A, Srivastava S. 大型语言模型驱动聊天机器人的基准测试：方法与指标. arXiv预印本 arXiv:2308.04624, 2023

175. Zhao W X, Zhou K, Li J, Tang T, Wang X, Hou Y, Min Y, Zhang B, Zhang J, Dong Z, others. A survey of large language models. arXiv preprint arXiv:2303.18223, 2023

176. Zhao W X, Zhou K, Li J, Tang T, Wang X, Hou Y, Min Y, Zhang B, Zhang J, Dong Z, 等. 大型语言模型综述. arXiv预印本 arXiv:2303.18223, 2023

176. Yang J, Jin H, Tang R, Han X, Feng Q, Jiang H, Zhong $\mathbf{S, YinB, HuX}$. Harnessing the power of llms in practice: A survey on chatgpt and beyond. ACM Transactions on Knowledge Discovery from Data, 2023

177. Yang J, Jin H, Tang R, Han X, Feng Q, Jiang H, Zhong $\mathbf{S, YinB, HuX}$. 实践中利用大型语言模型的力量：关于ChatGPT及其后续的综述. ACM知识发现与数据挖掘汇刊, 2023

177. Wang Y, Zhong W, Li L, Mi F, Zeng X, Huang W, Shang L, Jiang X, Liu Q. Aligning large language models with human: A survey. arXiv preprint arXiv:2307.12966, 2023

178. Wang Y, Zhong W, Li L, Mi F, Zeng X, Huang W, Shang L, Jiang X, Liu Q. 使大型语言模型与人类对齐：一项综述. arXiv预印本 arXiv:2307.12966, 2023

178. Huang J, Chang K C C. Towards reasoning in large language models: A survey. arXiv preprint arXiv:2212.10403, 2022

179. Huang J, Chang K C C. 面向大型语言模型推理能力的研究综述. arXiv预印本 arXiv:2212.10403, 2022

179. Mialon G, Dessi R, Lomeli M, Nalmpantis C, Pasunuru R, Raileanu R, Rozière B, Schick T, Dwivedi-Yu J, Celikyilmaz A, others . Augmented language models: a survey. arXiv preprint arXiv:2302.07842, 2023

180. Mialon G, Dessi R, Lomeli M, Nalmpantis C, Pasunuru R, Raileanu R, Rozière B, Schick T, Dwivedi-Yu J, Celikyilmaz A, 等. 增强型语言模型综述. arXiv预印本 arXiv:2302.07842, 2023

180. Chang Y, Wang X, Wang J, Wu Y, Yang L, Zhu K, Chen $\mathbf{H, YiX}$,Wang $\mathbf{C}$,Wang $\mathbf{Y}$,others. A survey on evaluation of large language models. ACM Transactions on Intelligent Systems and Technology, 2023

181. Chang Y, Wang X, Wang J, Wu Y, Yang L, Zhu K, Chen $\mathbf{H, YiX}$,Wang $\mathbf{C}$,Wang $\mathbf{Y}$,等. 大型语言模型评估综述. ACM智能系统与技术汇刊, 2023

181. Chang T A, Bergen B K. Language model behavior: A comprehensive survey. Computational Linguistics, 2024, 1-58

182. Chang T A, Bergen B K. 语言模型行为：全面综述. 计算语言学, 2024, 1-58

182. Li C, Wang J, Zhu K, Zhang Y, Hou W, Lian J, Xie X. Emotionprompt: Leveraging psychology for large

language models enhancement via emotional stimulus. arXiv e-prints, 2023, arXiv-2307

183. Li C, Wang J, Zhu K, Zhang Y, Hou W, Lian J, Xie X. EmotionPrompt：利用心理学通过情感刺激提升大型语言模型. arXiv电子预印本, 2023, arXiv-2307

183. Zhuo T Y, Li Z, Huang Y, Shiri F, Wang W, Haffari G, Li Y F. On robustness of prompt-based semantic parsing with large pre-trained language model: An empirical study on codex. arXiv preprint arXiv:2301.12868, 2023

184. Zhuo T Y, Li Z, Huang Y, Shiri F, Wang W, Haffari G, Li Y F. 基于提示的语义解析在大型预训练语言模型上的鲁棒性研究：以Codex为例的实证分析. arXiv预印本 arXiv:2301.12868, 2023

184. Gekhman Z, Oved N, Keller O, Szpektor I, Reichart R. On the robustness of dialogue history representation in conversational question answering: a comprehensive study and a new prompt-based method. Transactions of the Association for Computational Linguistics, 2023, 11: 351-366

185. Gekhman Z, Oved N, Keller O, Szpektor I, Reichart R. 对话历史表示在对话问答中的鲁棒性：全面研究及一种新的基于提示的方法. 计算语言学协会汇刊, 2023, 11: 351-366

185. Ji Z, Lee N, Frieske R, Yu T, Su D, Xu Y, Ishii E, Bang Y J, Madotto A, Fung P. Survey of hallucination in natural language generation. ACM Computing Surveys, 2023, 55(12): 1-38

186. Ji Z, Lee N, Frieske R, Yu T, Su D, Xu Y, Ishii E, Bang Y J, Madotto A, Fung P. 自然语言生成中的幻觉调查。ACM计算机调查，2023，55(12): 1-38

Lei Wang is a Ph.D. candidate at Renmin University of China, Beijing. His research focuses on recommender systems and agent-based large language models.
王磊是中国人民大学的博士研究生，研究方向为推荐系统和基于大型语言模型的智能体。

Chen Ma is currently pursuing a Master's degree at Renmin University of China, Beijing, China. His research interests include recommender system, agent based on large language model.
马晨目前在中国人民大学攻读硕士学位，研究兴趣包括推荐系统和基于大型语言模型的智能体。

Xueyang Feng is currently studying for a Ph.D. degree at Ren-min University of China, Beijing, China. His research interests include recommender system, agent based on large language model.
冯雪阳目前在中国人民大学攻读博士学位，研究兴趣包括推荐系统和基于大型语言模型的智能体。

Zeyu Zhang is currently pursuing a Master's degree at Renmin University of China, Beijing, China. His research interests include recommender system, causal inference, agent based on large language model.
张泽宇目前在中国人民大学攻读硕士学位，研究兴趣包括推荐系统、因果推断和基于大型语言模型的智能体。

Hao Yang is currently studying for a Ph.D. degree at Renmin University of China, Beijing, China. His research interests include recommender system, causal inference.
杨浩目前在中国人民大学攻读博士学位，研究兴趣包括推荐系统和因果推断。

Jingsen Zhang is currently studying for a Ph.D. degree at Ren-min University of China, Beijing, China. His research interests include recommender system.
张景森目前在中国人民大学攻读博士学位，研究兴趣包括推荐系统。

Zhi-Yuan Chen is pursuing his Ph.D. in Gaoling school of Artificial Intelligence, Renmin University of China. His research mainly focuses on language model reasoning and agent based on large language model.
陈志远正在中国人民大学高岭人工智能学院攻读博士学位，主要研究语言模型推理和基于大型语言模型的智能体。

Jiakai Tang is currently pursuing a Master's degree at Renmin University of China, Beijing, China. His research interests include recommender system.
唐嘉凯目前在中国人民大学攻读硕士学位，研究兴趣包括推荐系统。

Xu Chen obtained his PhD degree from Tsinghua University, China. Before joining Renmin University of China, he was a postdoc researcher at University College London, UK. In the period from March to September of 2017, he was studying at Georgia Institute of Technology, USA, as a visiting scholar. His research mainly focuses on the recommender system, reinforcement learning and causal inference.
陈旭获得清华大学博士学位。加入中国人民大学前，他曾在英国伦敦大学学院做博士后研究。2017年3月至9月期间，他作为访问学者在美国佐治亚理工学院学习。其研究主要集中在推荐系统、强化学习和因果推断。

Yankai Lin received his B.E. and Ph.D. degrees from Tsinghua University in 2014 and 2019. After that, he worked as a senior researcher in Tencent WeChat, and joined Renmin University of China in 2022 as a tenure-track assistant professor. His main research interests are pretrained models and natural language processing.
林彦凯于2014年和2019年分别获得清华大学工学学士和博士学位。此后，他曾在腾讯微信担任高级研究员，2022年加入中国人民大学，任教于终身教职助理教授。其主要研究方向为预训练模型和自然语言处理。

Wayne Xin Zhao received his Ph.D. in Computer Science from Peking University in 2014. His research interests include data mining, natural language processing and information retrieval in general. The main goal is to study how to organize, analyze and mine user generated data for improving the service of real-world applications.
赵新伟于2014年获得北京大学计算机科学博士学位。其研究兴趣包括数据挖掘、自然语言处理和信息检索，主要目标是研究如何组织、分析和挖掘用户生成数据，以提升实际应用的服务质量。

Zhewei Wei received his Ph.D. of Computer Science and Engineering from Hong Kong University of Science and Technology. He did postdoctoral research in Aarhus University from 2012 to 2014, and joined Renmin University of China in 2014.
魏哲伟获得香港科技大学计算机科学与工程博士学位。2012年至2014年在奥胡斯大学从事博士后研究，2014年加入中国人民大学。

Ji-Rong Wen is a full professor, the executive dean of Gaoling School of Artificial Intelligence, and the dean of School of Information at Renmin University of China. He has been working in the big data and AI areas for many years, and publishing extensively on prestigious international conferences and journals.
温继荣是中国人民大学全职教授，高岭人工智能学院执行院长，信息学院院长。他在大数据和人工智能领域工作多

年，在国际知名会议和期刊上发表了大量论文。