LLM-based Agentic Reasoning Frameworks: A Survey from Methods to Scenarios

BINGXI ZHAO[*], Beijing Jiaotong University, China and Lancaster University, United Kingdom
LIN GENG FOO[*], Max Planck Institute for Informatics, Saarland Informatics Campus, Germany
PING HU, University of Electronic Science and Technology of China, China
CHRISTIAN THEOBALT, Max Planck Institute for Informatics, Saarland Informatics Campus, Germany
HOSSEIN RAHMANI, Lancaster University, United Kingdom
JUN LIU[†], Lancaster University, United Kingdom

Recent advances in the intrinsic reasoning capabilities of large language models (LLMs) have given rise to LLM-based agent systems that exhibit near-human performance on a variety of automated tasks. However, although these systems share similarities in terms of their use of LLMs, different reasoning frameworks of the agent system steer and organize the reasoning process in different ways. In this survey, we propose a systematic taxonomy that decomposes agentic reasoning frameworks and analyze how these frameworks dominate framework-level reasoning by comparing their applications across different scenarios. Specifically, we propose an unified formal language to further classify agentic reasoning systems into single-agent methods, tool-based methods, and multi-agent methods. After that, we provide a comprehensive review of their key application scenarios in scientific discovery, healthcare, software engineering, social simulation, and economics. We also analyze the characteristic features of each framework and summarize different evaluation strategies. Our survey aims to provide the research community with a panoramic view to facilitate understanding of the strengths, suitable scenarios, and evaluation practices of different agentic reasoning frameworks.

CCS Concepts: • **General and reference → Surveys and overviews**; • **Computing methodologies → Natural language processing**.

Additional Key Words and Phrases: Agentic Reasoning, LLM-based Agent, Reasoning Frameworks.

ACM Reference Format:
Bingxi Zhao, Lin Geng Foo, Ping Hu, Christian Theobalt, Hossein Rahmani, and Jun Liu. 2025. LLM-based Agentic Reasoning Frameworks: A Survey from Methods to Scenarios. *J. ACM* 37, 4, Article 111 (August 2025), 51 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Large Language Models (LLMs), with their powerful generalization and promising reasoning capabilities, have been rapidly reshaping numerous domains from our daily lives (e.g., idea creation,

[*] Both authors contributed equally to this research.
[†] Corresponding author.

Authors' Contact Information: Bingxi Zhao, bingxizhao@bjtu.edu.cn, Beijing Jiaotong University, Beijing, China and Lancaster University, Lancaster, United Kingdom; Lin Geng Foo, lfoo@mpi-inf.mpg.de, Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany; Ping Hu, University of Electronic Science and Technology of China, Chengdu, China; Christian Theobalt, Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany; Hossein Rahmani, Lancaster University, Lancaster, United Kingdom; Jun Liu, j.liu81@lancaster.ac.uk, Lancaster University, Lancaster, United Kingdom.

ACM 1557-735X/2025/8-ART111
https://doi.org/XXXXXXX.XXXXXXX

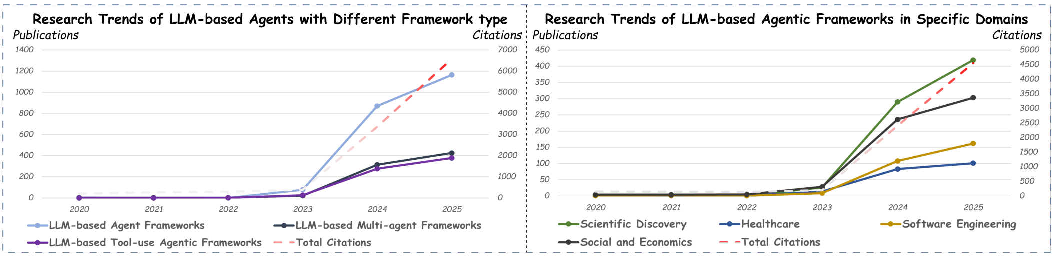


Fig. 1. Number of publications regarding LLM-based Agentic Frameworks from 2020 to 2025 in journals and conferences indexed by *Web of Science*. For discussing method developments (§3), we mainly select technical papers published at top computer science conferences *(e.g., ICLR, NeurIPS, ACL, EMNLP, AAAI, and ICML)*. To discuss about scenarios (§4), we collect a diverse set of representative works: from top computer science conferences *(same as above)* to top journals within specific domains *(e.g., Nature, Science, Cell, Nat. Mach. Intell, Adv. Mater, Adv. Sci, Nat. Med, PNAS, and NAR)*. We observe a fast-increasing trend since 2023, showing the growing importance of the field. *For 2025, we predict the overall amount of papers linearly based on data accessed at 14th August.*

email writing, or learning of new concepts) to domain-specific research [198]. Researchers have been increasingly leveraging LLMs as core components to empower research and innovation [166], from domain-specific knowledge Q&A [310] and code generation [118], to assisting in research endeavors [164]. Through these aspects, LLMs are quickly becoming a key part of modern life and research.

Yet, despite their immense potential across various fields, LLMs have intrinsic limitations, which may limit their usefulness. For instance, LLMs often suffer from issues such as hallucinations, outdated knowledge, and high training and inference costs [111]. These issues often lead to problems in the reliability and consistency of LLMs, and consequently restrict their application in critical fields like healthcare and software engineering, which demand highly dependable outcomes.

To overcome this barrier, the academic community has been actively exploring the use of LLMs as a core engine to build LLM-based agentic reasoning frameworks capable of executing complex, multi-step reasoning tasks [217, 266]. As illustrated in Fig. 1, we observe a significant upwards trend in terms of papers published at top conferences. Initially, "Agents" are defined in [227] as systems that "perceive their environment through sensors and act upon that environment through actuators", can dynamically adapt to their environments and take corresponding actions [166]. This emerging paradigm organically integrates key modules like planning, memory, and tool-use, reshaping the LLM into a task executor that can perceive its environment, adapt dynamically, and take sustained action [112, 154, 266]. By extending vertically, expanding horizontally, or backtracking logically, this paradigm fundamentally surpasses the single-step reasoning capabilities of traditional LLMs in both reliability and task complexity.

This trend has also been widely mirrored in industry, where tech giants are actively integrating agent workflows into their core businesses. For instance, frameworks like Microsoft's AutoGen[1] are designed to empower enterprises to build customized multi-agent applications. Moreover, from "vibe coding" editors like Cursor[2] that deeply integrate agentic capabilities to autonomous AI software engineers like Devin[3], a clear evolution based on agentic reasoning frameworks is gaining widespread recognition, gradually replacing traditional development approaches.

[1] https://github.com/microsoft/autogen
[2] https://cursor.com/en/dashboard
[3] https://devin.ai/

However, at the same time, the explosive growth in this field has also blurred the boundaries of LLM-based agents [305]. For instance, the overlap with concepts from areas like traditional multi-agent systems [30, 87, 315] and autonomous systems [255] makes it difficult to define the scope of research. Meanwhile, it is often hard to clearly separate whether enhanced capabilities of an agent come from careful framework design, model-level improvements, or technological advancements. This dual ambiguity poses a serious challenge for the horizontal comparison of different projects and risks overlooking the foundational role of framework design in an agent system's reasoning ability.

Therefore, we believe that it is timely for a survey to systematically summarize the *recent progress and application scenarios of agentic reasoning frameworks*. We first clearly define the boundaries of these frameworks and, based on that, propose a unified methodological classification system. We then further analyze the application and evaluation strategies of these methods across diverse scenarios, aiming to provide a clear roadmap for the standardized and safe development of agentic developments. Our taxonomy also fits the current popular topics like context engineering.

Overall, the contributions of our survey are as follows:

- To the best of our knowledge, this is the first survey that proposes a unified methodological taxonomy to systematically highlight the core reasoning mechanisms and methods within agentic frameworks;
- We employ a formal language to describe the reasoning process, clearly illustrating the impact of different methods on key steps;
- We extensively investigate the application of agent reasoning frameworks in several key scenarios. In these application scenarios, we conduct in-depth analyses of representative works according to our proposed taxonomy, and present a collection of evaluation setups with datasets.

The structure of the survey is as follows: Chapter §2 will further introduce compare the difference between related surveys and our survey. Chapter §3 will present the taxonomy of techniques, which systematically analyses the existing techniques for agentic reasoning. Chapter §4 will further provide application scenarios of agentic reasoning frameworks, and how agents in each scenario are often designed. Lastly, Chapter §5 will discuss future directions and Chapter §6 states the conclusion of the survey.

2 Related Surveys

Recent surveys on agentic AI have explored the agentic reasoning landscape from several valuable perspectives. A primary focus has been model-centric, examining how to enhance the agentic capabilities of LLMs. For instance, several surveys [129, 250, 292] review training methodologies such as Proximal Policy Optimization (PPO), Supervised Fine-Tuning (SFT) and Reinforcement Learning from Human Feedback (RLHF). Other surveys also explore the potential of smaller, specialized agentic models on specific reasoning tasks[18], or examine the planning abilities of agentic foundation models [112, 154]. Overall, these surveys primarily focus on the "LLM" side developments of LLM-based agents.

Yet, recently in the field of LLM-based agents, numerous representative methods about agentic frameworks have emerged, which explore how to leverage state-of-the-art LLMs with training-free methods to build agentic frameworks through framework-level reasoning. However, to the best of our knowledge, there still has not been a survey that systematically organizes these "framework" side developments and discusses their value in various application scenarios. Therefore, in contrast to other surveys, our survey specifically concentrates on *agentic reasoning frameworks*, reviewing the most recent development on framework-level agentic reasoning methods, instead of orthogonal

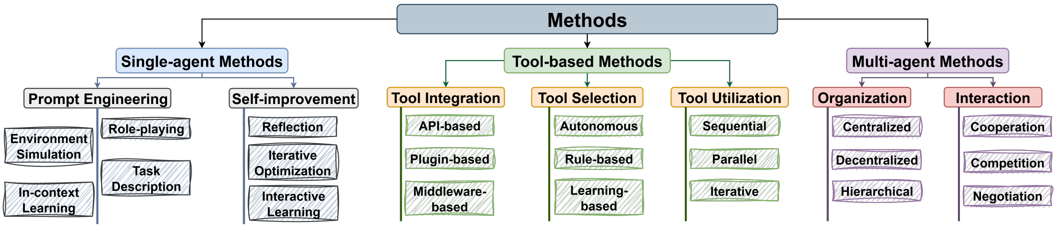


Fig. 2. Taxonomy of our proposed agentic reasoning frameworks. We decompose agentic reasoning methods into three progressive categories: **a)** single-agent methods, **b)** tool-based methods, and **c)** multi-agent methods.

developments in model architectures and fine-tuning techniques. We categorize existing methods along three progressive dimensions: single-agent, tool-based, and multi-agent, and propose a unified taxonomy to analyze the different stages of the multi-step reasoning process, which has not been explored in previous surveys.

Closer to our work, there are surveys exploring how agentic technologies could be used within specific domains, such as scientific discovery [86, 222], software engineering [122], medicine [274], or social sciences [55]. However, their scope is often limited and focuses only on a single specific domain, which can significantly increase the difficulty when comparing between agentic frameworks across different domains. For instance, each of these surveys utilizes a different way to categorize and list the research works, this makes it difficult to observe the abilities and trends of LLM agents at the frontier of research or the special designs in each scenario, since there is no unified taxonomy of these methods. Thus, we propose a systematic taxonomy which provides a unified view of LLM agentic frameworks. This allows us to systematically analyze how the unique requirements of each application scenario shape the design and adaptation of these frameworks in those scenarios, thereby bridging the gap between methods and application scenarios. Furthermore, our survey adopts a scenario-driven approach to trace and compare the evaluation setups and datasets used in each representative works, across different application domains. To the best of our knowledge, such systematic exploration of agentic reasoning and evaluation setups across different scenarios has not yet been explored.

3 Methods

Extended from Foundation LLMs, agentic reasoning frameworks is a key development in order to achieve a autonomous and environmental-aware systems which could solve complicated problems in the real world. In this section, we propose a taxonomy to categorize these methods. At the top, we deconstruct the reasoning framework into three distinct levels, namely single-agent, external tool calling, and multi-agent, as shown in Figure 2. ***Single-agent methods*** focus on enhancing the reasoning capability of individual agents; ***tool-based methods*** extend the boundaries of agent reasoning through external tools; and ***multi-agent methods*** enable more flexible reasoning through different paradigms of organization and interaction among multiple agents. We cover these levels in Chapters §3.2, §3.3, §3.4 respectively, after we introduce the notations in Chapter §3.1. Together, these methods at different levels can be integrated in different ways to fulfill specific scenarios, which is covered in Chapter §4.

3.1 Notations

We highlight that agent systems can achieve their goals through a complete process that includes multiple reasoning steps. Multi-agent systems can further execute a complete reasoning process under the collaboration of different agents [258]. To clearly introduce this complex process, we propose a general reasoning algorithm (Alg. 1) and a notation table (Table 1) in this section, which brings another level of abstraction. In the subsequent chapters (Chapters §3.2, §3.3, and §3.4), we will further discuss how each representative line of works improve the reasoning performances by modifying or adjusting this general algorithm.

Table 1. Notations Used in This Chapter

Notation	Description
P_U	The user's input query.
Q	The termination condition for the reasoning process.
g	The set of predefined goals to be achieved.
t	An external tool available to the agent.
C	The internal context of an agent.
y	The output of an agent after an action.
k	A count of reasoning steps.
$\mathcal{A}$	The entire action space, containing all possible actions.
a	A general action that produces an output from a given input.
a'	An action that updates the current context based on an input.
a_{reason}	An action that performs a step of deep reasoning.
a_{tool}	An action that involves an interaction with an external tool t.
a_{reflect}	An action to reflect on and evaluate previous reasoning steps.

A key differentiator between an agentic system and a standard Large Language Model is the ability to perform multi-step reasoning [89]. This capability relies on the active management of a persistent context throughout the lifecycle of a task within the agentic system [190]. While a standard LLM processes a given context to produce a single-step output, an agent system, base on its various action choice, iteratively updates its context to support a multi-step reasoning. Each action, though has different targets or intentions, follows a similar logic to tackle such input-output relations. Therefore, we formalize a single reasoning step as an operation where the agent executes an action a based on its current context C to produce an output y, expressed as $y = a(C)$. A full reasoning process will contain several such reasoning steps.

The outputs and insights from the preceding steps are preserved within this context, enabling the agent to build upon its prior work [190]. We explicitly distinguish the action of generating an output (a) from the action of updating the state (a'). This separation is crucial because the objective of a context update (e.g., summarizing history, integrating a tool's results) often differs from that of producing a final or intermediate answer [190].

To execute these steps, the agent selects actions from a generalized action space $\mathcal{A}$, which we define for our purposes as $\mathcal{A} = \{a_{\text{reason}}, a_{\text{tool}}, a_{\text{reflect}}\}$. To maintain focus on the reasoning logic, our framework abstracts complex auxiliary components, such as memory modules [326], knowledge retrieval [331], sandboxed environments [60], and human interruption [197] into a unified external tool t. This is because they are mainly act as an external source that could provides agent with external knowledge and information. The action a_{tool} is specifically designed and used to invoke this tool, providing the agent with necessary external information or capabilities. While this action space is sufficient for our analysis, it can be extended or tailored for specific domains [277].

Consequently, a complete reasoning task is modeled as an iterative sequence of actions The process is initiated by a user query P_U and proceeds until a predefined termination condition Q is met. This condition is essential for ensuring controlled execution and conserving computational resources [258]. Building on the notations in Table 1, we formalize this multi-step reasoning process in Algorithm 1. The comments within the algorithm serve as forward references, indicating which of the methodologies discussed in subsequent sections modify a particular step of the general procedure.

Algorithm 1 General Algorithm for Framework-level Agentic Reasoning

```
Require: User Query $P_U$; Goal $g$; External Tool $t$; Action Space $\mathcal{A}$; Terminate Condition $Q$
Ensure: Final Output $y_{out}$
1: Initialize context $C_0 \leftarrow \mathrm{Init}(P_U)$                        ▷ §3.2 (eq.1,eq.3); §3.3 (eq.11)
2: Initialize reasoning step $k \leftarrow 0$
3: while $\neg Q(C_k, k)$ do                                                  ▷ §3.3(eq.4)
4:     $y_{k+1} = a_k(C_k, g, t), \quad a_k \in \mathcal{A}$                  ▷ §3.3 (eq.6,eq.7,eq.9)
5:     $C_{k+1} = a'_k(C_k, y_{k+1}, g, t), a'_k \in \mathcal{A}$             ▷ §3.2 (eq.2); §3.3 (eq.8,eq.10); §3.4 (eq.12)
6:     $k \leftarrow k + 1$                                                   ▷ §3.2 (eq.5); §3.4 (eq.13,eq.14,eq.15)
7: return Final output derived from $C_k$
```

3.2 Single-agent Methods

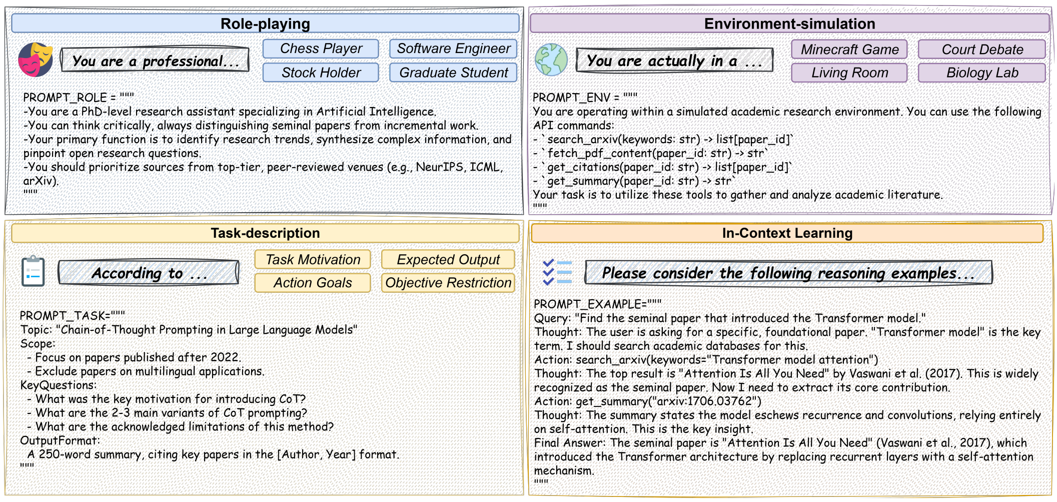
Single-Agent methods focus on enhancing the cognitive and decision-making abilities of an individual agent. From the perspectives of external guidance and internal optimization, this part categorizes single-agent methods into two main types: ***prompt engineering*** and ***self-improvement***. Prompt engineering emphasizes guiding the agent's reasoning process by leveraging roles, environments, tasks, and examples, while self-improvement focuses on how the agent refines its reasoning strategies through reflection, iteration, and interaction.

3.2.1 Prompt Engineering. Prompt engineering enhances the agent's performance by enriching its initial context, which corresponds to the context initialization step *(line 1 in Alg.1)*. [237]. Instead of relying solely on the user's query (P_U), this approach augments the initial context C_0 with a meticulously crafted prompt, denoted as P^*. This conceptual shift can be represented as:

$$C_0 \leftarrow \mathrm{Init}(P_U) \quad \xrightarrow{\text{Prom. Eng.}} \quad C_0 \leftarrow \mathrm{Init}(P_U, P^*) \tag{1}$$

Equation 1 illustrates that the initialization process is transformed. Originally, the context C_0 is derived exclusively from the user query P_U. With prompt engineering, it is initialized with both P_U and the engineered prompt P^*. This additional prompt P^* is often a composite of several components: a *role-playing perspective* (P_{role}), an *environment simulation* (P_{env}), a detailed *task clarification* (P_{task}), and a set of *in-context examples* (P_{icl}). Unlike fine-tuning methods, which alters the LLM's parameters, prompt engineering guides the model's behavior non-intrusively, steering the agent towards more accurate and predictable reasoning outcomes [171]. Each component of P^* contributes to this guidance in a distinct way, as detailed below and illustrated in Fig. 3.

Role-Playing. To instill a role-playing perspective (P_{role}), the prompt assigns the agent a specific persona or identity, such as "You are an expert data scientist" or "Act as a seasoned historian" [239]. This encourages the agent to leverage the expertise, cognitive frameworks, and linguistic styles associated with that role [229]. By adopting a persona, the model can better activate domain-specific knowledge and structure its responses from a more professional viewpoint during reasoning [136]. This technique has become a widely adopted method in the agentic frameworks discussed in chapter §4, owing to its low deployment cost and high guidance efficiency. By assigning a clear



```
PROMPT_ROLE = """
-You are a PhD-level research assistant specializing in Artificial Intelligence.
-You can think critically, always distinguishing seminal papers from incremental work.
-Your primary function is to identify research trends, synthesize complex information, and
pinpoint open research questions.
-You should prioritize sources from top-tier, peer-reviewed venues (e.g., NeurIPS, ICML,
arXiv).
"""
```

```
PROMPT_ENV = """
You are operating within a simulated academic research environment. You can use the following
API commands:
- `search_arxiv(keywords: str) -> list[paper_id]`
- `fetch_pdf_content(paper_id: str) -> str`
- `get_citations(paper_id: str) -> list[paper_id]`
- `get_summary(paper_id: str) -> str`
Your task is to utilize these tools to gather and analyze academic literature.
"""
```

```
PROMPT_TASK="""
Topic: "Chain-of-Thought Prompting in Large Language Models"
Scope:
  - Focus on papers published after 2022.
  - Exclude papers on multilingual applications.
KeyQuestions:
  - What was the key motivation for introducing CoT?
  - What are the 2-3 main variants of CoT prompting?
  - What are the acknowledged limitations of this method?
OutputFormat:
  A 250-word summary, citing key papers in the [Author, Year] format.
"""
```

```
PROMPT_EXAMPLE="""
Query: "Find the seminal paper that introduced the Transformer model."
Thought: The user is asking for a specific, foundational paper. "Transformer model" is the key
term. I should search academic databases for this.
Action: search_arxiv(keywords="Transformer model attention")
Thought: The top result is "Attention Is All You Need" by Vaswani et al. (2017). This is widely
recognized as the seminal paper. Now I need to extract its core contribution.
Action: get_summary("arxiv:1706.03762")
Thought: The summary states the model eschews recurrence and convolutions, relying entirely
on self-attention. This is the key insight.
Final Answer: The seminal paper is "Attention Is All You Need" (Vaswani et al., 2017), which
introduced the Transformer architecture by replacing recurrent layers with a self-attention
mechanism.
"""
```

Fig. 3. Prompt engineerng for agentic reasoning framework. We summarize four types of methods: ***a) Role-playing***: an agent will be distributed with a specific role, to stimulate its specific performance; ***b) Environmental-simulation***: an agent will be told in a carefully designed environment, where it can reason with multi-modalities or external abilities; ***c) Task-description***: a task will be clearly reconstructed and expressed to an agent; ***d) In-context Learning***: several examples will be provided to an agent before or during its multi-step reasoning. For each type of prompting method, we provide a short example prompt, with the theme of conducting agentic research.

role, it enables agents to better focus on their specific duties, thereby optimizing their reasoning and decision-making processes in complex tasks. However, the efficiency of role assignment can be sensitive to the granularity of the persona design and the specifics of the task [132]. Furthermore, for fact-based questions, role-playing may introduce biases inherent to the persona, potentially leading to factually inaccurate outputs [155].

Environment Simulation. The environment simulation prompt (P_{env}) contextualizes the agent by describing the specific setting in which it operates. This provides task-relevant background information, rules, and constraints, enabling the agent to make decisions that are better aligned with the simulated world. These environments can range from mimicking real-world scenarios, such as a stock market [67] or a medical clinic [64], to entirely virtual settings like a video game world [277], often with a action space that are carefully designed. A detailed and task-relevant environmental description is critical, as it prompts the agent to generate actions that are contextually appropriate and highly correlated with the scenario's objectives.

Task Description. A clear task description (P_{task}), which outlines the primary goal g, constraints, and expected output format, is a cornerstone of virtually every agent system. A well-structured task description guides the agent in decomposing a complex problem into a sequence of manageable sub-tasks. By providing a precise description, agents can better comprehend the task's intent and execute it in the specified manner, which effectively reduces ambiguity during the reasoning process and leads to more accurate outcomes [191]. However, the verbosity and structure of the task description can significantly impact the performance of the underlying LLM, often requiring careful optimization tailored to the specific model being used [147].

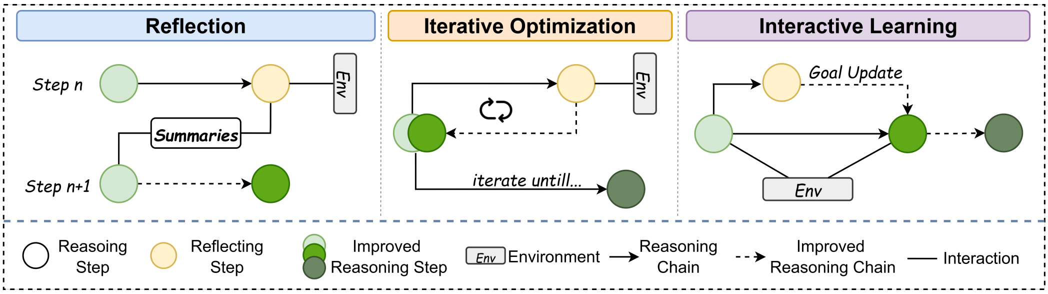


Fig. 4. Paradigms of self-improvement for an LLM-based agent. We introduce three core mechanisms. ***a) Reflection***: The agent analyzes a completed trajectory to generate a textual summary, storing in its context. This summary will helps for the next reasoning step. ***b) Iterative Optimization:*** Within a single task, the agent generates an initial output, compares it against a defined standard or feedback from others, and repeatedly refines it in following reasoning steps, until Q is met. ***c) Interactive Learning***: The agent interacts with a dynamic environment, where experiences (e.g., discovering a new item) can trigger an update to its high-level goals, fostering continuous, open-ended learning.

In-context Learning. In-context learning (ICL) provides the agent with a set of few-shot examples, or demonstrations, within the prompt (P_{icl}). These examples typically take the form of pairs $\{(x_1, z_1), (x_2, z_2), \ldots, (x_n, z_n)\}$, where each pair (x_j, z_j) consists of an exemplary input x_j and its corresponding desired output z_j [25]. This allows the agent to discern patterns and generalize to new task instances without any gradient updates. Chain-of thought prompting [282] further porvides a paradigms that the intermediate reasoning steps could also be brought to agent, teaching agents how to reason, plan and break down problems without internal tuning [328]. However, the performance of ICL is highly sensitive to the quality and relevance of the provided examples; low-quality or irrelevant demonstrations can significantly degrade the agent's reasoning capabilities [156].

3.2.2 *Self-Improvement.*

Self-improvement mechanisms encourage an agent to enhance its reasoning capabilities through introspection and autonomous learning. Rather than relying on static, pre-defined prompts, these methods enable the agent to dynamically adapt its strategies based on its own experiences. As summarized in Fig. 4, this internal optimization process can be understood through three complementary paradigms: *reflection*, which involves learning from past trajectories; *iterative optimization*, which focuses on refining outputs within a single reasoning cycle; and *interactive learning*, which allows for the dynamic adjustment of high-level goals in response to environmental feedback.

Reflection. Reflection enables an agent to perform post-hoc analysis on its past actions and outcomes to extract valuable lessons for future tasks. This process involves generating a summary of its reasoning process, identifying flaws or inefficiencies, and articulating insights in natural language [92]. This process refers to *line 5 in Alg. 1*, where the action $a_k^{'}$ is specifically assigned as reflection $a_{reflect}$:

$$C_{k+1} = a^{'}_{reflect}(C_k, y_{k+1}, g, t) \tag{2}$$

The scope of reflection can vary. For instance, the Reflexion framework [241] guides agents to verbally reflect on task failures, storing these reflections in an episodic memory to refine plans in subsequent trials. Other approaches have explored reflecting on different aspects, such as inefficient reasoning paths [335] or conflicting information from external tools [204]. This self-correction capability allows the agent to learn from its mistakes and continuously adapt its strategies without requiring external intervention or parameter updates.

Iterative Optimization. In contrast to the post-hoc nature of reflection, iterative optimization utilize a whole reasoning process to complete a pre-defined standard or constraint, which we denoted as $\mathcal{S}$. This mechanism introduces two key modifications to the agent's fundamental operation described in Alg. 1.

First, the standard $\mathcal{S}$ is incorporated into the agent's initial context. This modification of the initialization step *(line 1 in Alg. 1*) ensures that agent is aware of the optimization target from the outset:

$$C_0 \leftarrow \text{Init}(P_U) \quad \xrightarrow{\text{Iter. Opt.}} \quad C_0 \leftarrow \text{Init}(P_U, \mathcal{S}) \tag{3}$$

Equation 3 shows that the context initialization is augmented to include not just the user's query P_U, but also the explicit standard $\mathcal{S}$ that the final output must satisfy. Second, the agent's autonomy to decide when to stop is replaced by $\mathcal{S}$. The general termination condition Q is now precisely defined by whether the current output y satisfies the standard S. This can be expressed as a formal redefinition of Q *(line 3 in Alg.1)*:

$$Q \quad \xrightarrow{\text{Iter. Opt.}} \quad Q \triangleq (y \vDash \mathcal{S}) \tag{4}$$

As stated in Equation 4, the termination condition Q is now defined as the predicate checking if the current output y satisfies the standard $\mathcal{S}$. Consequently, after each reasoning step that produces an output, the agent checks it against $\mathcal{S}$, entering an iterative loop of refinement until the condition is met. This iterative loop is central to frameworks like Self-Refine [183], where a single LLM acts as its own generator, critic, and refiner to improve its output without external training data. This approach is particularly effective for tasks requiring high precision, such as code generation [84] or mathematical reasoning [3]. However, it can be computationally intensive and risks converging on a suboptimal solution if the feedback mechanism is flawed or the search space is too complex [205].

Interactive Learning. Representing the most advanced level of self-improvement, interactive learning allows an agent to fundamentally alter its high-level goals g based on continuous interaction with a dynamic environment. This paradigm moves beyond optimizing a fixed plan to enabling the agent to decide what to do next on a strategic level. This corresponds to an enhancement of the goal-updating mechanism *(line 6 in Alg. 1)*, where the goal g is no longer static but is re-evaluated at each step:

$$g_{k+1} \leftarrow a_k(\{(C_i, y_i)\}_{i=1}^{k}, g_k, t) \tag{5}$$

Equation 5 shows that the new goal g_{k+1} is derived from the entire history of contexts and outputs $\{(C_i, y_i)\}_{i=1}^{k}$, the current goal g_k, and available tools t. Voyager [277] exemplify this, where an agent in Minecraft autonomously proposes new goals based on its discoveries, gradually building a complex skill tree without human intervention. Similarly, ExpeL [329] enables an agent to learn from trial-and-error experiences, creating a memory of successful and failed attempts that informs the generation of more promising goals in future tasks. Further systematizing this process, Learn-by-Interact [103] introduces a data-centric framework where an agent autonomously collects interaction data and then distills it into a reusable knowledge base, thereby enabling structured, self-adaptive behavior in complex environments. By dynamically adapting its objectives, the agent demonstrates a higher form of autonomy, allowing it to navigate complex, evolving environments in a truly adaptive manner [333].

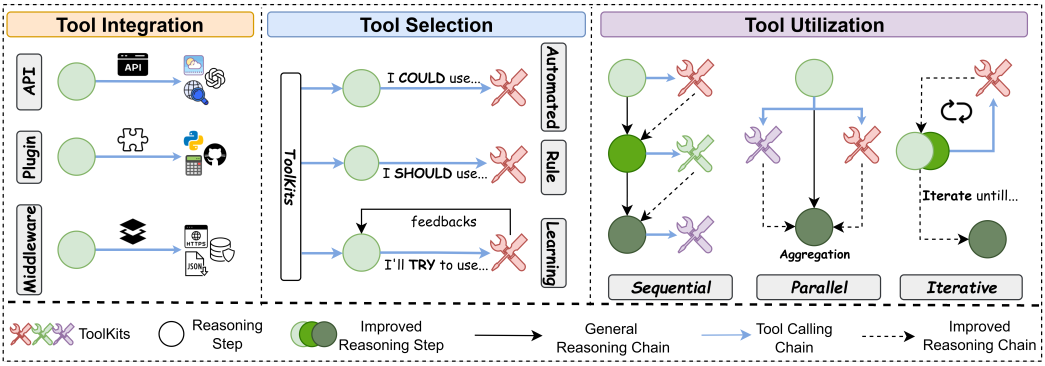


Fig. 5. Tool-based reasoning frameworks of LLM-based agent. ***a) Tool integration*** studies how to incorporate tools into the agent's reasoning process; ***b) Tool Selection*** addresses which tool from the toolkit $\mathcal{T}$ is most suitable for the current task or sub-problem; ***c) Tool Utilization*** concerns how to effectively operate the chosen tool to generate the desired output.

3.3 Tool-based Methods

While the general agentic reasoning framework *(Alg. 1)* conceptualizes tool use via a single entity t, this abstraction is insufficient for complex scenarios where reasoning is deeply intertwined with specific environmental capabilities. Here we expand this single entity t into a comprehensive toolkit $\mathcal{T} = \{t_1, t_2, ..., t_n\}$, where each t_i represents a distinct tool available to the agent. As illustrated in Fig. 5, we deconstruct the tool-based reasoning pipeline into three fundamental stages: ***Tool Integration***, ***Tool Selection***, and ***Tool Utilization***. Generally, the output from tool calling will be integrated into the context of agent by a specific action *(line 5 in Alg. 1)*. These three steps together form the tool-based methods for complex multi-step reasoning, helping agents better exploit external resources to solve complex reasoning problems.

3.3.1 Tool Integration. Before an agent select and utilize a tool, the tool must first be made accessible within the agent's operational environment. This architectural integration defines the interface and communication protocol between the agent and the tool [61]. We categorize these integration patterns into three primary models: *API-based Integration*, *Plugin-based Integration*, and *Middleware-based Integration*. APIs enable agents to easily interact with various tools without needing to understand their internal implementations; plugins dynamically extend the functionality of the agent system; while middleware focuses on aligning the interactions between the agent and the tools.

API-based Integration. APIs *(Application Programming Interfaces)* provide standard for integrating external tools [311]. APIs provide a stable, well-documented contract that allows an agent to interact with a tool (e.g., a web search engine [47]) without needing to understand its internal implementation. The agent simply learns to formulate a request according to the API specification and parse the returned data.

Emerging Agent protocols such as MCP further develop the diversity of API tools. Under the corporation of service provider, agents now can easily use precise services such as map navigation to provide detailed information for the user [105]. But such integration is subject to network latency, rate limits, and potential service outages. It also requires the agent to manage authentication and security credentials [105].

Plugin-based Integration. Plugins are software components that are loaded and executed directly within the agent's own runtime environment. Unlike external API calls, plugins operate with lower latency and have deeper access to the agent's internal state [181].

Retrieval-Augmented Generation (RAG) [148] is a typical case of plugin-based integration. A vector database is integrated directly into the agent system, introducing domain-specific knowledge to the agent in the form of a tool call, thereby increasing the credibility of its answers [77]. Liu et al. [173] offers a more specific application of plugins. By integrating an interactive heatmap plugin and a scatter plot plugin, the agent system is enabled to dynamically process, analyze, and visualize domain-specific data during its reasoning process. Thus, plugins offer a higher level of customization, extending the edge of agenic framework's core abilities, but it may introduce complexity to the overall system [101].

Middleware-based Integration. Middleware is a software layer situated between the Agent and tools [88]. This layer acts as a universal adapter or an "operating environment" for the agent, abstracting away the complexities of direct tool interaction, shielding the LLM from environmental complexity [88]. A middleware layer could manage API keys, standardize data formats across different tools, or provide a unified file system and execution environment for the agent [291]. Therefore, middleware simplifies the agent's logic by offloading complex tasks, providing a consistent interface over a heterogeneous set of tools. Chen et al. [40] further propose Internet of Agents, highlighting advantages of middleware in complex reasonoing process. However, it adds another layer of abstraction that can complicate maintenance.

3.3.2 Tool Selection. Instead of generally using tool t in each reasoning step *(line 4 and 5 of Alg. 1)*, here we want to highlight the importance of the selecting action of tool within reasoning steps. Effective tool selection is pivotal when an agent is presented with a large and diverse toolkit $\mathcal{T}$. The challenge lies in accurately mapping the requirements of a given problem to the specific choice of a tool t, where $t \in \mathcal{T}$. Based on the degree of agent autonomy, we categorize tool selection strategies into three primary approaches: *Autonomous Selection*, *Ru-Based Selection*, and *Learning-Based Selection.*

Autonomous Selection. This paradigm highlights the autonomy of agentic systems. The agent autonomously selects a tool based on its intrinsic reasoning capabilities, guided solely by the natural language descriptions of the available tools and the input query [321]. This process is often framed as a zero-shot reasoning task, where the agent must "think" to connect the problem to the right tool without explicit rules [304].

Just like a general reasoning step, the tool selection step will let agent reason, reflect, or even use tools to decide which tool t_{k+1} is suited for current condition, this tool t_{k+1} can be regarded as the output y within this reasoning step:

$$y_{k+1} \leftarrow t_{k+1} = a_k(C_k, g, \mathcal{T}), a_k \in \mathcal{A} \tag{6}$$

Following *line 5 in Alg. 1*, the selected tool t_{k+1} is updated into the current context window, allowing the agent to use it in subsequent reasoning steps. This selection process may sometimes be repeated multiple times to gradually filter for the best tools from a large toolset [160]. Since this method requires no task-specific examples or fine-tuning, it enables the agent to dynamically adapt to novel combinations of tools, tasks and scenarios. However, its performance is highly dependent on the quality of tool descriptions and the agent's inherent reasoning capacity, which challenges the robustness and efficiency of the agent system.

Rule-Based Selection. This approach governs agent's tool selection through a set of predefined, explicit rules $\mathcal{R}$ that map specific tasks, intents, or states to designated tools [180]. These rules provide a deterministic and reliable mechanism for tool choice. The selection process is thus conditioned on these rules:

$$t_{k+1} = a_k(C_k, g, \mathcal{T}, \mathcal{R}), a_k \in \mathcal{A} \tag{7}$$

The rules in $\mathcal{R}$ can be implemented in various forms, from simple keyword matching [174] to structured formats like process description language (PDL) [339].

The main benefit of rule-based selection is its high reliability for well-defined tasks. It ensures that the agent consistently uses the correct tool for a known situation, minimizing errors [154]. However, manually crafting and maintaining a comprehensive set of rules is labor-intensive and scales poorly as the number of tools and the complexity of tasks grow. It struggles with unforeseen problems that do not match any existing rules, forcing a default failure or a fallback to a different selection mechanism.

Learning-Based Selection. Learning-based selection in this context refers to an explicit, online process where the agent refines its tool selection strategy during inference [232]. This adaptation occurs through a cycle of action, feedback, and reflection, improving its concurrent tool actions. As demonstrated in figure. 5, the agent attempts a tool for task, receives feedback on its performance (e.g., from execution results, or human guidance), and then explicitly reflects on this outcome to update its context C for subsequent steps:

$$C_{k+1} = a'_{\text{reflection}}(C_k, y_k, g) \tag{8}$$

This reflective step allows the agent to learn from its own context by storing experiences of successful tool-task pairings or by generating explicit strategies to avoid repeating past mistakes [215]. This approach enables the agent to adapt to novel scenarios and user preferences without requiring model retraining. Learn-By-Interact [103] achieves a interactive learning by synthesizing trajectories of agent-environment interactions based on documentations, and constructs instructions by summarizing or abstracting the interaction histories. However, a good feedback logic is necessary, and such exploring process can be costed.

3.3.3 Tool Utilization. Following the previous section, this section focus on how to make the best use of the selected tools [186]. Here we divide tool utilization into three modes: sequential use, parallel use, and iterative use. *Sequential use* involves invoking multiple tools in a predetermined order, *parallel use* focuses on the breadth of tool calls within the same reasoning step, while *iterative use* aims at achieving the optimal task solution within certain limits through repeated cycles.

Sequential Utilization. In this mode, the agent invokes tools in a sequence, where the output of one tool often serves as the input for the next, forming a clear tool-chain [154]. This is well-suited for tasks that can be decomposed into a linear workflow. The results of tool calling are integrated into the current context, influencing the next calling [15]. CRITIC [85] improve its output through a sequential use of external tools, including search engine and code interpreter. MCP-Zero [68] further promote tool discovery based on the tool chain, where agent sequentially use different tool to solve complex problems. Its primary benefit is simplicity and predictability, making workflows easy to design, analyze and debug. But sometimes it's inefficiency for tasks with independent sub-problems and susceptibility to cascading failures, where an error in an early step halts the entire chain.

Parallel Utilization. To enhance efficiency, this mode involves invoking multiple tools concurrently within a single reasoning step. The Agent invokes multiple tools simultaneously to achieve synchronous processing of multidimensional information. For a selected tool set $\mathcal{T}' = \{t_1', t_2', ..., t_m'\}$ in any middle reasoning step k, the agent will generate a group of results in parallel using each tool within $\mathcal{T}'$. That is, for *line 4 in Alg.1*, the output will becomes a set of output:

$$\mathcal{Y}_{k+1} = \{y^1_{k+1}, y^2_{k+1}, \ldots, y^m_{k+1}\}$$
$$\text{where } y^i_{k+1} = a_k(C_k, g, t'_i), \quad a_k \in \mathcal{A}, \quad t'_i \in \mathcal{T}' \tag{9}$$

After that, the update of context will further consider this output set $\mathcal{Y}_{k+1}$[340], instead of a single output like before:

$$C_{k+1} = a^{'}_k(\mathcal{Y}_{k+1}, C_k, g), a'_k \in \mathcal{A} \tag{10}$$

The key advantage is a significant reduction in latency, as multiple time-consuming tool utilization can be executed at once. It's also efficiency to explore several proper tools simultaneously. For example, LLM Compiler [133] efficiently orchestrates multiple function calls by executing them in parallel during intermediate reasoning steps, while LLM-Tool Compiler [244] achieves tool parallelization by selectively fusing tools with similar functionalities. But such techniques may also introduce the challenge of aggregating potentially conflicting information from diverse outputs.

Iterative Utilization. Iterative utilization involves a micro-level loop where an agent repeatedly interacts with a tool to achieve a fine-grained objective within a single step of the broader reasoning process [184]. This contrasts with macro-level iterative optimization of the entire solution in §3.2.2. The focus here is on perfecting a single tool-use instance. A prime example is an agent using a code interpreter: if the first execution fails, the agent inspects the error, refines the code, and re-executes it until it runs successfully, all before moving to the next macro reasoning step [97]. This method enhances the robustness of tool execution, but may increase the latency of a single reasoning step and carries the risk of getting stuck in unproductive loops [205]. This necessitates carefully designed termination conditions or reflection mechanisms [241].

3.4 Multi-agent Methods

While single-agent frameworks demonstrates considerable capabilities, they inherently face limitations when confronted with tasks demanding diverse expertise or complex problem decomposition. Multi-agent systems (MAS) emerge as a natural solution, leveraging the collective intelligence of multiple agents to tackle these challenges. The central principle of MAS is to "divide and conquer", but its core challenge lies in achieving effective coordination [258]. This challenge bifurcates into two fundamental questions: (1) *How should the agents be organized?* This pertains to the system's organizational architecture, which dictates the patterns of control and information flow. (2) *How should the agents interact with others?* This relates to the individual interaction protocols, which define how agents align their goals and behaviors.

As illustrated in Figure 6, we analyze the multi-agent reasoning frameworks along these two axes. ***Organizational architectures*** include centralized, distributed, and hierarchical forms, which determine the structural backbone of the system; while ***individual interactions*** involve cooperation, competition, and negotiation, governing the dynamics between agents as they pursue their objectives.

To formally discuss these paradigms, we represent a multi-agent system as a set of agents $\mathcal{M} = \{A^{r_1}, A^{r_2}, ..., A^{r_n}\}$, where r_i denotes the specialized role of an agent A^{r_i}. While each agent adheres to the general reasoning loop outlined in *Alg. 1*, its behavior is individuated by its unique role, goal g^i, available actions a^i and tool t^i. They also maintain different individual context C^i. It is the distinctiveness of each agent's context and role that drives the heterogeneity in their reasoning, ultimately shaping the system's collective output.

3.4.1 Organizational Architecture. The organizational architecture defines the macro-level structure for coordination and control, which is often instantiated by assigning a specific role r_i to each

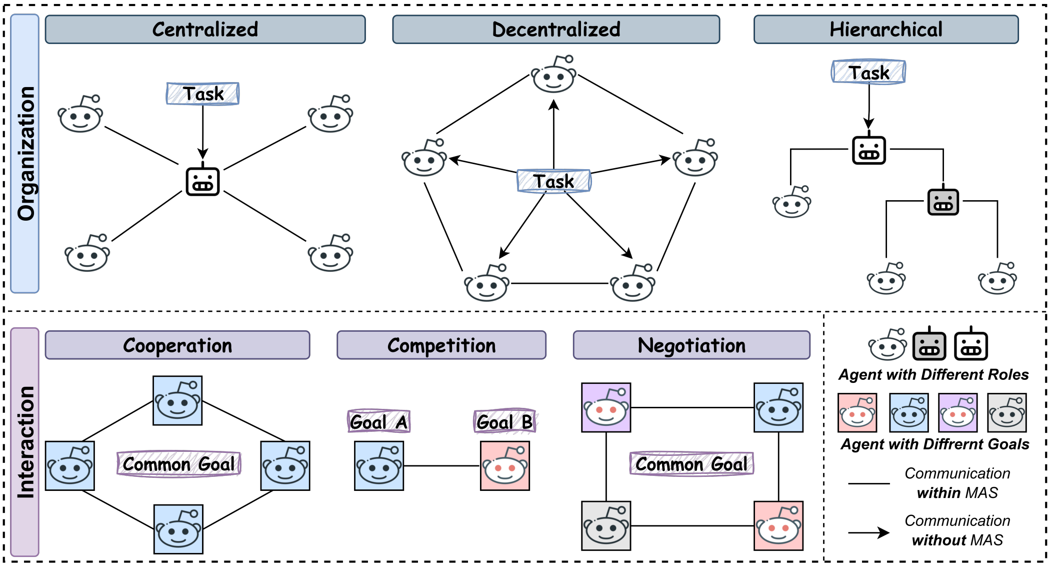


Fig. 6. A taxonomy of Multi-agent reasoning frameworks, categorized in ***a) Organizational Architectures***: we summarize three paradigms of multi-agent frameworks to explain how such system can be organized to solve different kinds of challenges; ***b) Individual Interactions***: we demonstrated three types of interaction between different agents within one multi-agent system.

agent's initial context $C_0^{r_i}$:

$$C_0 \leftarrow P_U \Rightarrow C_0^{r_i} \leftarrow (P_U, r_i) \tag{11}$$

Extend from *line 5 in Alg. 1*, each agent, no matter under what organization, their update of context must consider all other agents' output, with their previous context $y_k^{r_i}$, which would be formalized as:

$$\begin{gathered} C_{k+1}^{r_i} = a_k^{'}(C_k^{r_i}, \mathcal{Y}_k, g^{r_i}, t^{r_i}), a_k' \in \mathcal{A} \\ \text{where}\mathcal{Y}_k = \{y_k^{r_1}, y_k^{r_2}, ..., y_k^{r_n}\} \end{gathered} \tag{12}$$

Here, $\mathcal{Y}_k$ represents the collective outputs of all the n agents at reasoning step k. The specific subset of $\mathcal{Y}_k$ that an agent A^{r_i} considers is determined by different organizational architecture. We further decompose the organization of multi-agent systems into three distinct paradigms: centralized, decentralized, and hierarchical. A *centralized architecture* is suitable for scenarios requiring global optimization and strict control; a *distributed architecture* offers greater autonomy for each agent; while a *hierarchical architecture* is appropriate for tasks with clear goals and well-defined processes. These paradigms can be statically or dynamically integrated, reaching a flexible organization structure in specific scenarios [166].

Centralized. In general, a centralized architecture set a central agent A^{cen} to manage and coordinate the reasoning activities for other agents, A^{r_i} where $r \neq cen$ [78]. But their is a difference. This central agent A^{cen} typically performs global planning, task decomposition, and result synthesis, requiring it to process the outputs from all other agents, as in Eq. 12. Subordinate agents, however, may only need to consider instructions from the manager, simplifying their context updates. This architecture ensures high coordination and global optimization [78]. However, it introduces a potential performance bottleneck and a single point of failure at the central node [205].

Decentralized. In a decentralized architecture, there is no central authority. Each agent possesses equal status and makes decisions based on local information and direct peer-to-peer communication [301]. Consequently, the context update for every agent typically follows the general form in Eq. 12, where each agent must process the outputs of all its neighbors, or all other agents in a fully connected system. This discussion-like process fosters emergent collaboration and enhances system robustness and fault tolerance, as the failure of one agent does not cripple the entire system [301]. However, it may reduce the overall efficiency of resource utilization [205].

Hierarchical. A hierarchical architecture organizes agents into a structured tree or pyramid, decomposing a complex task into sub-problems at different levels of abstraction. As illustrated in MetaGPT [102], agents at higher levels are responsible for strategic planning and delegate tasks to lower-level agents, which execute more specific sub-tasks. Information typically flows vertically: instructions pass down from upper to lower levels, and results are passed back up. This structure excels at solving well-defined problems that can be clearly decomposed, promoting efficiency and consistency [37]. However, such architectures can be rigid and may stifle the flexibility and creativity of individual agents.

3.4.2 *Individual Interaction.*

The interaction protocol governs how an agent's goals evolve in response to others, directly influencing the system's emergent behavior. This introduces a dynamic update to an agent's goal g^{r_i}, expanding the static goal assumption in the basic reasoning loop *(line 6 in Alg. 1)*. We further categorize these interactions as cooperation, competition, and negotiation. *Cooperation* emphasizes maximizing collective interests, *competition* focuses on maximizing individual interests, and *negotiation* represents a compromise between the two. These three different paradigms can also be further combined to achieve specific reasoning objectives.

Cooperation. In cooperation mechanism, the primary objective for agents is to maximize collective interests. A common goal $\mathcal{G}$ is established to guide knowledge sharing and collaborative planning [258]. This goal can be predefined in the system prompt or dynamically formed during different reasoning steps.

At each step, an agent will dynamically update its goal by reflecting on its performance and aligns its individual goal g^{r_i} with the common goal $\mathcal{G}$. The updated goal will further influence the *next* reasoning step:

$$g^{r_i} \leftarrow a^{r_i}_{reflect}(C^{r_i}_k, g^{r_i}, \mathcal{G}, t^{r_i}) \tag{13}$$

Therefore, the achievement of individual agentic goals often promotes the goals of other agents and the entire system [89].

Competition. In competitive interactions, agents pursue their individual goals, which are often in conflict. The objective is to maximize individual benefit, which may involve outperforming or strategically undermining opponents [31]. An agent must not only advance its own agenda but also infer and counter the intentions of others based on their observable outputs $\mathcal{Y}_k$. The goal update process will therefore become adversarial:

$$g^{r_i} \leftarrow a^{r_i}_{reflect}(C^{r_i}_k, g^{r_i}, \mathcal{Y}_k, t^{r_i}), \text{where} \mathcal{Y}_k = \{y^{r_1}_k, y^{r_2}_k, ..., y^{r_n}_k\} \tag{14}$$

This dynamic is exemplified by frameworks that use multi-agent debate, such as MAD [162], where agents take on adversarial "debater" roles to challenge assumptions and uncover flaws in reasoning. Such adversarial interactions can significantly improve the robustness and quality of the final output by forcing a thorough exploration of the problem space [117].

Table 2. A comprehensive comparison of mainstream agentic reasoning frameworks, illustrating how methods from our taxonomy are organized within each work, alongside their inspiration, evaluation, and code. The legend for the abbreviations is as follows. **PE (Prompt Engineering)**: Role (role-playing), Task (task description). **SI (Self-Improvement)**: R.F (reflection), I.O (iterative optimization), I.L (interactive learning). **Tools**: T.I (tool integration); T.S (tool selection: 'rule' for rule-based, 'auto' for autonomous); T.U (tool utilization: 'Seq' for sequential, 'Iter' for iterative). **Multi-agent**: M.O (organization: 'Dec' for decentralized, 'Cen' for centralized, 'Hier' for hierarchical); M.I (interaction: 'Deb' for debate, 'Coo' for cooperation).
*Mix: The framework employs more than one organization method.
†Prompt engineering techniques are widely used, so we list only the most representative sub-methods employed in each work.
‡This column lists the primary inspirations (theories or prior works) stated in each paper.

Work	Single-agent		Tool-based			Multi-agent		Inspiration‡	Datasets	Code
	PE†	SI	T.I	T.S	T.U	M.O	M.I			
Du et al.[59]	Role	I.O.	-	-	-	Dec.	Deb.	Society of Mind[192]	GSM8k[50],BigBench[245], MMLU[98]	🐙
MAD[162]	Role/Task	-	-	-	-	Dec.	Deb.	Degeneration-of -Thought[22, 130]	Kong et al.[137],Website**,	🐙
SPP[278]	Role	I.O.	-	-	-	Cen.	Coo./Deb.	Pretend play[209, 210]	Triviaqa[126],BigBench[245]	🐙
AutoGen[287]	Task	RF.	API	Rule	Seq	Mix*	Coo.	Society of Mind[192]	MATH[99],Kwiatkowski et al.[139], Adlakha et al.[2],ALFworld[242]	🐙
AgentVerse[39]	Role	I.L.	API/ Plugin††	Auto	Seq	Dec.	Coo.	Markov decision process	FED[189],Commongen-Challenge[184], MGSM[240],BigBench[245], HummanEval[38]	🐙
AutoAgents[34]	Role	I.O.	API	Auto	Seq	Mix*	Coo.	ReAct[304],AutoGPT‡‡	MT-Bench[334],FairEvals[268], HummanEval[38],Triviaqa[126]	🐙
CAMEL[151]	Role/Task	I.O.	API	Auto	Seq	Dec.	Coo.	Society of Mind[192]	HummanEval[38],Humaneval+[169]	🐙
ChatDev[213]	Role	I.O.	API	Auto/Rule	Seq	Hier.	Coo.	LLM Hallucination[323]	SRDD[213]	🐙
MetaGPT[102]	Role	I.O./I.L.	API	Auto	Iter.	Hier.	Coo.	ReAct[304],SOPs[17, 54, 185, 286]	HummanEval[38],MBPP[10], SoftwareDev[102]	🐙

**https://www.geeksforgeeks.org/puzzles/
††https://github.com/OpenBMB/BMTools
‡‡https://github.com/Significant-Gravitas/Auto-GPT

Negotiation. Negotiation is a hybrid interaction that balances cooperation and competition. It enables agents with conflicting interests to reach a mutually acceptable consensus through communication and compromise [83]. During negotiation, agents exchange proposals and iteratively adjust their goals based on both the common objective $\mathcal{G}$ and the proposals from others contained in $\mathcal{Y}_k$, as illustrated in Figure 6:

$$\begin{gathered} g^{r_i} \leftarrow a^{r_i}_{reflect}(C^{r_i}_k, g^{r_i}, \mathcal{Y}_k, \mathcal{G}, t^{r_i}) \\ where \mathcal{Y}_k = \{y^{r_1}_k, y^{r_2}_k, ..., y^{r_n}_k\} \end{gathered} \tag{15}$$

This process compels agents to weigh their own objectives against collective constraints and the perspectives of others. For instance, ChatEval [31] utilizes a negotiation-like debate among multiple "referee" agents to autonomously evaluate the quality of AI-generated text, reaching a human-aligned judgments. This approach is particularly effective for complex decision-making tasks where there is no single correct answer, but rather a spectrum of acceptable solutions.

3.5 Discussion

In this chapter, we introduced a three-level, progressive taxonomy to demonstrate how methods from each level enhance various facets of an agentic framework's reasoning process. This classification is grounded in a unified formal language and a general reasoning algorithm (Alg. 1). We contend that by combining methods across these levels, the capability boundaries and collaborative

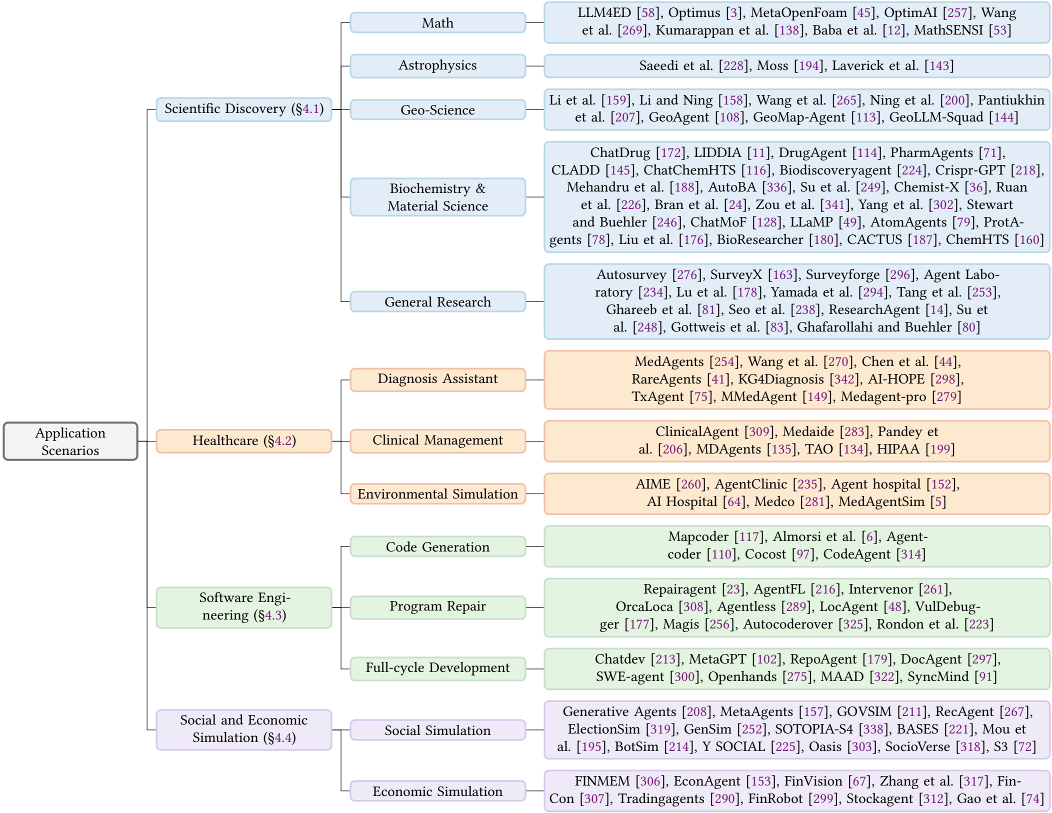

Application Scenarios
Scientific Discovery (§4.1)
Math
LLM4ED [58], Optimus [3], MetaOpenFoam [45], OptimAI [257], Wang et al. [269], Kumarappan et al. [138], Baba et al. [12], MathSENSI [53]
Astrophysics
Saeedi et al. [228], Moss [194], Laverick et al. [143]
Geo-Science
Li et al. [159], Li and Ning [158], Wang et al. [265], Ning et al. [200], Pantiukhin et al. [207], GeoAgent [108], GeoMap-Agent [113], GeoLLM-Squad [144]
Biochemistry & Material Science
ChatDrug [172], LIDDIA [11], DrugAgent [114], PharmAgents [71], CLADD [145], ChatChemHTS [116], Biodiscoveryagent [224], Crispr-GPT [218], Mehandru et al. [188], AutoBA [336], Su et al. [249], Chemist-X [36], Ruan et al. [226], Bran et al. [24], Zou et al. [341], Yang et al. [302], Stewart and Buehler [246], ChatMoF [128], LLaMP [49], AtomAgents [79], ProtAgents [78], Liu et al. [176], BioResearcher [180], CACTUS [187], ChemHTS [160]
General Research
Autosurvey [276], SurveyX [163], Surveyforge [296], Agent Laboratory [234], Lu et al. [178], Yamada et al. [294], Tang et al. [253], Ghareeb et al. [81], Seo et al. [238], ResearchAgent [14], Su et al. [248], Gottweis et al. [83], Ghafarollahi and Buehler [80]
Healthcare (§4.2)
Diagnosis Assistant
MedAgents [254], Wang et al. [270], Chen et al. [44], RareAgents [41], KG4Diagnosis [342], AI-HOPE [298], TxAgent [75], MMedAgent [149], Medagent-pro [279]
Clinical Management
ClinicalAgent [309], Medaide [283], Pandey et al. [206], MDAgents [135], TAO [134], HIPAA [199]
Environmental Simulation
AIME [260], AgentClinic [235], Agent hospital [152], AI Hospital [64], Medco [281], MedAgentSim [5]
Software Engineering (§4.3)
Code Generation
Mapcoder [117], Almorsi et al. [6], Agentcoder [110], Cocost [97], CodeAgent [314]
Program Repair
Repairagent [23], AgentFL [216], Intervenor [261], OrcaLoca [308], Agentless [289], LocAgent [48], VulDebugger [177], Magis [256], Autocoderover [325], Rondon et al. [223]
Full-cycle Development
Chatdev [213], MetaGPT [102], RepoAgent [179], DocAgent [297], SWE-agent [300], Openhands [275], MAAD [322], SyncMind [91]
Social and Economic Simulation (§4.4)
Social Simulation
Generative Agents [208], MetaAgents [157], GOVSIM [211], RecAgent [267], ElectionSim [319], GenSim [252], SOTOPIA-S4 [338], BASES [221], Mou et al. [195], BotSim [214], Y SOCIAL [225], Oasis [303], SocioVerse [318], S3 [72]
Economic Simulation
FINMEM [306], EconAgent [153], FinVision [67], Zhang et al. [317], FinCon [307], Tradingagents [290], FinRobot [299], Stockagent [312], Gao et al. [74]

Fig. 7. The overview of our selected paper of agentic reasoning frameworks across different application scenarios.

patterns of agent systems can be significantly expanded. For instance, each agent member of a multi-agent system (§3.4) often optimizes its individual performance through prompt engineering and self-reflection (§3.2), while invoking external tools (§3.3) to execute specific reasoning steps based on its designated role. In Table 2, we consolidate mainstream general-purpose agentic frameworks, detailing how they integrate methods from the different categories of our taxonomy, along with their proposed inspirations and evaluation datasets.

Furthermore, while our taxonomy provides a comprehensive summary of the logical structures and collaborative patterns at the framework level, we acknowledge that researchers often incorporate optimization techniques like Supervised Fine-Tuning [73] and Reinforcement Learning [82] to achieve superior performance [129]. To maintain a clear focus on the foundational nature of these reasoning frameworks, we exclude these technical details from our classification. In the subsequent chapter, we will further showcase the value of these frameworks by examining their applications in specific scenarios.

4 Scenarios

Building upon the previous chapter's foundational concepts, this chapter offers a panoramic view of agentic reasoning capabilities across diverse application scenarios. Our primary goal is to systematically compare and contrast the similarities and differences among these frameworks. We also conduct a comprehensive collection of the evaluation metrics, methodologies, and datasets across these domains. We categorize the application scenarios into ***scientific research, healthcare, software engineering, and social & economic simulations***, as illustrated in Fig. 7

4.1 Scientific Research

Agent systems are increasingly becoming a cornerstone for automating various stages of scientific inquiry. Through the implementation of well-designed reasoning pipelines, these agents enhance the efficiency of the entire scientific workflow. We systematically review the design of agent frameworks aimed at accelerating research, with a focus on their applications in a range of disciplines including *mathematics, astrophysics, geoscience, biochemistry, materials science*, as well as *general scientific research* frameworks.

4.1.1 Math. By leveraging the synergistic combination of their constituent components, agent systems go beyond traditional reasoning methods to achieve remarkable results in specific mathematical domains, including optimization and proof generation.

As an early work, LLM4ED [58] utilizes a symbolic library to aid in equation discovery, where the LLM iteratively proposes and refines novel equations based on natural language instructions to outperform purely text-based methods. Subsequent research has gravitated towards multi-agent systems that employ structured collaboration. In the realm of mathematical optimization, Optimus [3] leverages a multi-agent system to autonomously manage the entire pipeline for mixed-integer linear programming, including task assignment, modeling, and evaluation, while using a central graph to track dependencies for iterative refinement. A similar hierarchical organization is also used in computatioal fluid dynamics (CFD) field by MetaOpenFOAM [45], where role-based agents collaboratively handle simulation design, setup, and review in a iterative closed loop. The concurrent work MetaOpenFOAM 2.0 [46] further enhances robustness by introducing Chain of Thought (CoT) [282] and iterative CoT strategies for complex subtask decomposition. This orchestration method also proves effective in OptimAI [257], which solves natural language optimization problems by hierarchically decomposing user queries and automating the full cycle of model formulation, coding, and debugging through iterative feedback.

Other related works focus on formal theorem proving. MA-LoT [269] employs multi-agent collaboration to decouple the natural language cognitive task of proof generation from subsequent error analysis. In its framework, one agent generates a complete proof, while another corrects it, establishing a structured interaction between an LLM and the Lean4 verifier guided by a Long CoT. Addressing the challenge of continuous learning, LeanAgent [138] optimizes its learning trajectory based on mathematical difficulty and manages evolving knowledge through a dynamic database, enabling stable yet plastic lifelong mathematical learning via progressive training. Besides, Prover Agent [12] uses an informal reasoning language model for high-level thinking and a separate formal proof model to execute the theorem-proving steps in Lean. During its reasoning process, the system strategically creates auxiliary intermediate theorems to guide the proof and leverages feedback from Lean to reflect upon and adjust its strategy. Furthermore, MathSensei [53] emphasizes the auxiliary role of tool invocation in mathematical reasoning. It equips its agent with a comprehensive suite of tools, including a knowledge retriever (powered by an LLM or Bing Web Search), a Python-based program generator and executor, and a symbolic problem solver (Wolfram-Alpha), thereby significantly extending the boundaries of the system's reasoning capabilities.

4.1.2 Astrophysics. In astrophysics, agent systems are being developed to assist the research process by managing vast, proprietary datasets through automated and scalable collaboration. AstroAgents [228] generates hypotheses from spectral data. It employs a team of eight specialized agents that work in sequence to interpret the data, perform deep analysis on specific segments, formulate hypotheses, conduct literature searches, evaluate the hypotheses, and propose refinements. Expanding the scope to the entire scientific lifecycle, The AI Cosmologist [194] implements an end-to-end pipeline encompassing ideation, experimental evaluation, and research dissemination. It utilizes dedicated agents for planning, coding, execution, analysis, and synthesis, aiming to automate the complex workflows of data analysis and machine learning research in cosmology and astronomy. Focusing on the cosmological parameter analysis, Laverick et al.'s work [143] is built upon AutoGen framework [287] and integrates Retrieval-Augmented Generation (RAG) to facilitate the auxiliary analysis of cosmological data.

4.1.3 Geo-science. The integration of Geographic Information Systems (GIS) with agentic reasoning frameworks can significantly enhance a system's ability to autonomously reason, deduce, innovate, and advance geospatial solutions [159]. As a pioneering work, Autonomous GIS [158] introduces an agent-based framework for geospatial problem analysis. The system decomposes user requirements into ordered operational steps, constructs a flowchart, and generates Python code to sequentially execute tasks such as data loading, spatial joins, statistical analysis, and plotting to produce the final output. Concurrent works have specialized in particular aspects of the workflow. Ning et al. [200] enhances the reasoning process by focusing on data retrieval. It performs autonomous data discovery based on task understanding and a data-source manual, while generating Python retrieval code via in-context learning that is iteratively executed, debugged, and refined by the framework. Moreover, Pantiukhin et al. [207] leverages a centralized Multi-Agent System and a suite of earth science tools for data processing, analysis, and visualization. Crucially, it incorporates a reflection module to contemplate evaluation outcomes and drive iterative improvements to its plan. Besides, GeoLLM-Squad [144] targets on Remote Sensing (RS) workflows. Built upon the AutoGen [287] and GeoLLM-Engine [243] frameworks, it modularizes RS applications by decomposing complex tasks and assigning them to specialized sub-agents, covering areas such as urban monitoring, forestry conservation, climate analysis, and agricultural research.

Further research has focused on improving the quality and scope of agent-based geospatial analysis. To mitigate subjective bias in domain-specific question answering, Wang et al. [265] utilizes RAG and online search to comprehensively gather relevant information. The system then employs a CoT [282] process to integrate and reflect upon this information, ensuring reliable geospatial analysis. Pushing the boundaries of task complexity, GeoAgent [108] builds upon RAG by incorporating Monte Carlo Tree Search (MCTS) to plan and execute multi-step analyses. Starting from a natural language description, it iteratively generates, runs, and debugs multi-step code. GeoAgent [108] also introduced the Geocode Benchmark, a comprehensive suite of single and multi-turn tasks involving data acquisition, analysis, and visualization to evaluate agents in diverse geospatial contexts. Venturing into multi-modal understanding, GeoMap-Agent [113] pioneers the use of a Multi-modal Large Language Model (MLLM) to interpret geological maps. It performs hierarchical information extraction to segment the map and identify salient elements. This is followed by retrieving domain knowledge from an expert database, which is integrated into an enhanced prompt to enable precise question answering. GeoMap-Agent [113] also introduce GeoMap-Bench, the first benchmark designed to assess the geological map understanding capabilities of MLLMs across a full spectrum of skills, including extraction, referring, localization, reasoning, and analysis.

4.1.4 Biochemistry and Material Science. The advent of deep learning has significantly enhanced research productivity across the life sciences [1], and the rise of agentic systems is now further pushing the boundaries of workflow automation. In this section, we survey the application of agentic reasoning in this domain, which we categorize into five primary areas: (1) *drug discovery and design*, (2) *genetic and biological experiment design*, (3) *chemical synthesis*, (4) *material science*,

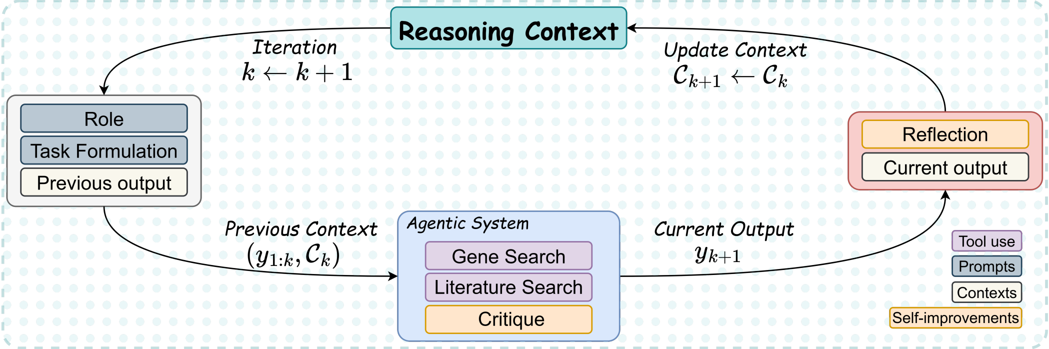


Fig. 8. A summarization of pipeline proposed by BioDiscovery-Agent [224], base on our proposed taxonomy. Such framework could allows a iterative experimental design with dynamic context update. Image is edited from [224].

and (5) *research automation*. These work across different sub-scenarios cover a wide range of targets, thus their evaluation strategies are very different. As illustrated in Table 3, we summarize their evaluation strategies in metrics level, benchmark or dataset level, and case study methods, respectively. Furthermore, as several applications in biochemistry have direct extensions to clinical practice, they will be discussed in greater detail in Section §4.2.

Drug Discovery. In drug discovery and design, agentic systems must balance user requirements with scientific principles to achieve precise molecular engineering. Several works have explored centralized or single-agent architectures to this end. ChatDrug [172] integrates retrieval tools to fetch similar molecules with desired attributes from knowledge bases, translating editing tasks into structured instructions to contextualize the reasoning process. It further leverages a dialogue module to iteratively refine molecules based on user feedback. Similarly, LIDDiA [11] employs a four-component architecture – reasoner, executor, evaluator, and memory – to guide molecular design, extensively using tool calls to simulate molecular docking, predict properties, and optimize structures according to personalized specifications. Moreover, DrugAgent [114] simulates a collaborative research team using CoT [282] and ReAct paradigms [304] to predict Drug-Target Interactions (DTI). It can forecast DTI scores, compute interaction metrics, search domain knowledge, and generate a final prediction with a detailed explanation.

Other frameworks utilize hierarchical Multi-Agent Systems (MAS) to decompose the complex drug discovery pipeline. PharmAgents [71] divides the process into four stages: target discovery, lead identification and optimization, and preclinical evaluation. Each stage is managed by dedicated agents equipped with distinct tools, which collaborate via structured knowledge exchange and self-improve by reflecting on past experiences. CLADD [145] also adopts a hierarchical structure for discovery and question-answering, combining it with a RAG approach. It breaks down reasoning into planning, knowledge graph querying, molecular understanding, and prediction, and can dynamically select tools based on a query molecule's structural similarity to known drugs in a knowledge graph. Specifically focusing on the human-agent interface, ChatChemTS [116] enables users to design new molecules by automatically constructing a reward function for specified properties purely through natural language interaction.

Genomics and Biological Experiment Design. In genomics and biological experiment design, agentic systems are tasked with analyzing, decomposing, and implementing user requirements, thereby assisting researchers to handle complex experimental workflows. As a representative work, BioDiscovery-Agent [224] iteratively designs gene perturbation experiments by integrating prior results and knowledge into its reasoning context, as illustrated in Fig.8. In each cycle, it constructs prompts to guide the design of small-batch experiments, prioritizing genes likely to produce significant phenotypic effects. This process involves invoking other agents for critical evaluation, literature searches, and data analysis to enable efficient identification of gene functions. Similarly, CRISPR-GPT [218] offers multiple interaction modes, decomposes gene editing experiments into manageable steps for in-context learning, and integrates external tools to achieve automation.

Other approaches focus on model customization and workflow robustness. Bio-Agents [188] is built upon a small language model (SLM) fine-tuned on bioinformatics data and enhanced with RAG, enabling personalized operations and the analysis of local or proprietary data. AutoBA [336] concentrates on automating multi-omics analysis, capable of self-designing the analytical process in response to changes in input data and enhancing system robustness through automated code repair. To simplify bioinformatics workflows, BioMaster [249] uses a Multi-Agent System (MAS) for task decomposition, execution, and validation. It employs RAG to dynamically retrieve domain-specific knowledge and introduces input-output validation to improve adaptability and stability when handling new tools and niche analyses.

Table 3. An overview of the evaluation strategies of agentic reasoning frameworks in Biochemistry and Material Science. We summarize them from three levels. In metrics level, the specific response of the framework is directly evaluated. Benchmark and Dataset level further utilizes domain-specific data and standard to evaluate the responses. In case study, the framework will be evaluated through several real-world tasks, generally the ground truth is clear.

	Metrics Level	
Focus	**Related Work**	**Metrics**
Lab-envolved Biomedical Research	BioResearcher[180]	Completeness, Level of Detail, Correctness, Logical Soundness, Structural Soundness
High-fidelity Materials Knowledge Retrieval	LLAMP[49]	Precision, coefficient of precision, confidence, self-consistency of response, MAE
Drug Discovery	Liddia[11]	drug-likeness[21], Lipinski's Rule of Five[165], synthetic accessibility[62], binding affinities[259]
	Benchmark/Dataset Level	
Focus	**Related Work**	**Benchmark/Dataset**
Genetic Perturbation Experiment Designation	BiodiscoveryAgent[224]	Schmidt et al.[236], Carnevale et al.[27], Scharenberg et al.[231], Sanchez et al.[230], Horlbeck et al.[104], OpenTargets[201]
Quantum Chemistry	El Agente[341]	Armstrong et al.[9]
Drug Discovery	Liddia[11]	pdb[20], CheMBL[19]
Chemical Tool Invocation Optimization	ChemHTS[160]	ChemLLMBench[90], ChEBI-20-MM[170], USPTO-MIT[124], MoleculeNet[288]
Tool-assisted Molecular discovery	CACTUS[187]	CACTUS[187]
Mendelian Diseases Diagnose	MD2GPS[337]	SCH[109], JN[337], DDD[69], RD[316]
Automated Bioinformatics Analysis	BioMaster[249]	Morey et al.[193], [51], Rao et al.[220]
RAG-based Drug Discovery	CLADD[145]	MoleculeNet[288], Proposing hub[52], drugbank[285],STITCH[251], hERG[272], DILI[293],Skin[7],Lagunin et al.[140]
full drug discovery workflow	PharmAgent[71]	crossdocked[70], Pharmagent[71]
Drug-target Interaction Prediction	DrugAgent[114]	Anastassiadis et al.[8]
Drug Editing	Liu et al.[172]	Rao et al.[219], MHCflurry 2.0[202],ZINC[115]
Gene-Editing	Crispr-GPT[218]	Gene-editing bench[218]
Chemical Hypothesis Discovery	Moose-Chem[302]	Moose-chem[302]
	Case Study	
ProtAgents[78], Stewart et al.[246], PharmAgent[71], CLADD[145], TourSynbio-search[176], AutoBA[336], AI-HOPE[298], ChatMOF[128], Chemist-X[36], Ruan et al.[226], ChatChemTS[116], El Agent[341], ChemCrow[24], AtomAgents[79].		

Chemical Synthesis and Design. In chemical synthesis, early agentic systems focused on optimizing reaction conditions and automating complex workflows. Chemist-X [36] designed a multi-stage reasoning system to optimize reaction conditions. It first retrieved reaction conditions from a database via hierarchical matching, then used a CAD tool to recommend yield-improving conditions, and finally translated these recommendations into validated experimental operations. Similarly, Ruan et al. [226] decomposes chemical synthesis development into six sub-stages: literature search, experimental design, hardware execution, spectral analysis, product separation, and result interpretation. Each stage is executed by a dedicated agent, sequentially accomplishing the entire workflow. Moreover, ChemCrow [24] incorporated 18 expert-designed chemistry tools, demonstrating their efficacy by successfully automating the design of an organic catalyst synthesis.

Other systems tackle more specialized or abstract challenges within chemistry. El Agente Q [341] dynamically generates and executes quantum chemistry workflows from natural language prompts, leveraging a hierarchical multi-agent memory framework for task decomposition, adaptive tool selection, and autonomous post-analysis. Targeting the upstream process of scientific inquiry, MOOSE-Chem [302] focuses on autonomous chemical hypothesis discovery. It formalizes this process by decomposing a base hypothesis into a research context and a set of "inspirations", which then guide the sub-tasks of retrieving, combining, and ranking new hypotheses. Stewart et al. [246] sequentially identifies engineering goals, generates a large pool of candidate molecules through rational steps and knowledge extraction, and then analyzes them by structure and charge distribution to achieve molecular optimization.

Material Science. As one of the early works, ChatMoF [128] constructed a system for predicting and generating Metal-Organic Frameworks (MOFs). It can autonomously select and invoke specialized tool-kits based on user requirements, making decisions iteratively based on tool outputs and internal evaluations. This is achieved through four distinct functional agent components responsible for MOF database searching, internet searching, performance prediction, and material generation. Concurrently, LLaMP [49] combines a hierarchical ReAct framework [304] with a multi-modal, retrieval-augmented one to dynamically and recursively interact with computational and experimental data on the Materials Project (MP) database, running atomic simulations via a high-throughput workflow interface. Furthermore, AtomAgents [79] proposes a physics-aware approach to alloy design. Its multi-agent system autonomously implements the entire material design pipeline – from knowledge retrieval and multi-modal data integration to physics-based simulation and cross-modal comprehensive result analysis.

Biochemical Automated Research. Beyond optimizing for specialized domains, several works have proposed systematic designs from the broader perspective of automating biochemical research. These systems often focus on sophisticated agent orchestration and interaction. For instance, ProtAgents [78] is a task-centric multi-agent system that decomposes the protein design and analysis process into multiple stages. It employs a predefined chat manager as a central hub to dynamically select appropriate agents and manage their communication. Based on distinct role and tool assignments, these agents collaborate to propose protein designs, execute physical simulations, predict structures, and iteratively reflect upon, evaluate, and refine the designs. Similarly, Toursynbio-search [176] implements a user-driven research system where each specialized agent has an independent keyword list. It uses fuzzy matching against the classified user intent to select the right agent, then initiates a validation process, generating an interactive page for user verification and supplementation when parameters are ambiguous.

Other systems introduce specific mechanisms to improve the research workflow. BioResearcher [180] decomposes research tasks (retrieval, planning, analysis) and assigns them within a hierarchical agent architecture. Crucially, it introduces a reviewer agent to ensure quality control throughout the process and iteratively optimizes the research plan via internal evaluation. Focusing on tool use, CACTUS [187] utilizes the LangChain architecture for sequential problem analysis, tool review, and selection. This reasoning cycle repeats until the problem is solved, allowing the system to learn the characteristics and applicability of different tools through iteration. ChemHTS [160] further refines tool-calling strategy with a hierarchical tool stack. It first conducts a "self-stacking warm-up" phase to explore tool capabilities and limitations, then recursively combines tools to find the optimal calling path, using selective information transfer and tool encapsulation to keep the focus on the primary task.

4.1.5 General Research. The problem-solving capabilities of agentic frameworks can be generalized from specialized, domain-specific tasks to broader research inquiries. These systems are designed to operate from an initial prompt to the final deliverable of a detailed research report. Here we classify these scenarios into *literature survey, end-to-end research automation, research collaboration and* *refinement*. Table 4 summarize different metrics and datasets used in each selected work, with their specific focus about general research. We also conclude the work that use case study or subjective methods to evaluate their work.

Literature Survey. Automating literature surveys requires systems to process and orchestrate vast amounts of domain literature to generate comprehensive and coherent reviews. Autosurvey [276] first retrieves relevant literature via semantic search and generates a preliminary outline. It then employs multiple agents to concurrently draft each section, retrieving additional literature to produce text with accurate citations. After drafting, the system integrates and refines the content, using multi-LLM-as-judge setup to score the survey's quality and coverage for iterative improvement. Similarly, SurveyX [163] automates this process in two phases: a preparation phase that uses an attribute tree structure to acquire and preprocess literature, and a generation phase that performs both coarse and fine-grained content creation from an initial plan. RAG is leveraged to ensure citation accuracy during rewriting and to support multi-media content generation. Besides, Surveyforge [296] combines a library of human-written outlines with domain-specific papers to generate a new outline via in-context learning. It introduces a memory-driven framework with multi-layer sub-query and retrieval memories to refine the search process, followed by filtering and reconstruction stages for content integration and parallel text generation. This work also contributes Surveybench, a benchmark for quantitatively evaluating survey quality across multiple dimensions.

Table 4. An Overview of Evaluation Strategies of Agentic Frameworks in General Research. We summarize them from four levels. In metrics level, the output literature from framework is directly evaluated. Benchmark and Dataset level further provides and filters several literature with high quality and diverse themes. In case study, the framework will be used to complete a real-world research task, and generally evaluated by domain-specific professionals. LLM-as-a-judge or human evaluation tend to evaluate the output literature of framework from several subjective metrics.

Metrics Level

Focus	Related Work	Metrics
Paper Generation Quality	AutoSurvey [276]	Survey Creation Speed, Content Quality
	Gao et al. [76]	Citation Quality
	SurveyX [163]	Insertion over Union, semantic-based reference relevance
	SurveyForge [296]	Reference, Outline, and Content Quality (SAM Metrics)
	Agent Laboratory [234]	Inference cost, Inference time, Success Rate
	Su et al. [247]	Proxy MSE, Proxy MAE
	VirSci [248]	Historical Dissimilarity, Contemporary Dissimilarity, Contemporary Impact

Benchmark/Dataset Level

Focus	Related Work	Type	Benchmark/Dataset	#Papers
Idea Generation	Research Agent [14]	Off-line	ResearchAgent [14]	300
Survey Automation	SurveyForge [296]		SurveyBench [296]	100
Research Assistants	Agent Laboratory [234]		Mle-bench [32]	-
Research Improvement	CycleResearcher [284]		Review-5k/ research-14k[284]	~5k/~14k
Scientific Innovation	AI-Researcher [253]		Scientist-Bench [253]	22
Idea Generation	ReserachAgent [14]	On-line	Semantic Scholar Academic Graph API*	-
	VirSci [234]		AMiner Computer Science dataset†	~2M
			Open Academic Graph 3.1‡	~131M

Case Study

IdeaSynth[212], The AI Scientist[178], Towards an AI co-scientist[83], Robin[81], SciAgents[80]

LLM-as-a-Judge/Human Evaluation

ResearchAgent[14], AutoSurvey[276], SurveyX[163],Agent Laboratory[234],The AI Scientist v2[294], Cycleresearcher[284],AI co-scientist[83]

*https://www.semanticscholar.org/product/api

†https://www.aminer.cn/aminernetwork

‡https://open.aminer.cn/open/article?id=65bf053091c938e5025a31e2

End-to-End Research Automation. Moving beyond knowledge synthesis, end-to-end research automation aims to emulate the entire scientific lifecycle for a given topic, often by mimicking real-world research workflows. As a representative work, Agent Laboratory [234] actualizes a user's research idea through a multi-stage, hierarchical architecture covering literature retrieval, experimental planning, code generation, result analysis, and report writing, integrating human feedback at each stage for quality assurance. Many similar works decompose the research process to achieve automation. The AI Scientist [178] decompose scientific discovery into five stages (ideation, experiment design, execution, paper writing, and peer review), guiding agents with structured instructions. AI Scientist-v2 [294] further introduces an experimental parameter space and a tree-search algorithm to optimize the research process, incorporating a vision-language model (VLM) to provide feedback on generated figures and text. Notably, papers generated by this system have passed peer review at ICLR workshop[4]. Moreover, AI-Reseracher [253] adopt a four-stage decomposition of the research pipeline from literature review, hypothesis generation, algorithm implementation to manuscript writing. AI-Reseracher [253] also propose Scientist-Bench, which comprises recent papers spanning diverse AI research areas and includes both guided-innovation and open-ended exploration tasks. Moreover, Robin [81] facilitates semi-autonomous discovery by progressing through four stages: hypothesis generation, experimental design, result interpretation, and hypothesis updating. Besides, Papercoder [238] automates the reproduction of code from machine learning papers. It uses three dedicated agents for a three-stage process: a planning stage to create a high-level roadmap, an analysis stage to interpret specific implementation details, and a generation stage to produce modular, dependency-aware code.

Research Collaboration and Refinement. There are some emergent research focuses on introducing specific mechanisms for collaboration and refinement to enhance the quality, creativity, and efficiency of automated science. A significant direction is improving idea and hypothesis generation through multi-agent interaction. ResearchAgent [14] iteratively refines research ideas by integrating core papers, their citation network, and feedback from a panel of "reviewer agents" to mimic peer review and foster innovation. Similarly, VirSci [248] uses multi-agent dialogue – spanning collaborator selection, topic discussion, and novelty assessment – to generate ideas, while AI co-scientist [83] employs a hierarchical agent team for critical debate to propose, critique, and refine hypotheses. Notably, this system also presents a fully-automated end-to-end research abilities based on the widely debate and iteratively refinement between agent with different roles.

Another key direction is the refinement of the research process and agent capabilities. CycleResearcher [284] proposes an iterative training framework that simulates the "research-review-improve" academic closed loop to enhance generated paper quality, contributing two large datasets for this purpose. Furthermore, SciAgents [80] leverages an ontology knowledge graph for structured reasoning, while Scimaster [29] uses a decentralized, stacked workflow to scale reasoning depth and breadth. At the ecosystem level, Agentrxiv [233] constructs a scientific ecosystem where multiple *Agent Laboratories* [234] can collaborate asynchronously. By using a centralized preprint server, independent agent teams can upload, share, and retrieve research, forming a dynamic knowledge commons that enables cumulative, collaborative scientific discovery.

4.2 Healthcare

The advent of powerful foundational Large Language Models is reshaping the landscape of healthcare by empowering agentic systems with new capabilities. This transformation facilitates a critical shift from reactive, predictive functions to proactive, interactive engagement in clinical workflows. These advanced agents are increasingly leveraged to resolve chronic issues surrounding clinical efficiency, diagnostic precision, and the quality of patient care. As summarized in Table 5, a wide range of evaluation datasets as well as methods are used to evaluate these framework. Accordingly, we survey these works through *diagnostic assistance, clinical management, and environmental simulation.*

4.2.1 Diagnosis Assistance. To augment diagnostic capabilities, one primary research focus involves creating multi-agent dialogue frameworks that deconstruct the complex diagnostic process into manageable, collaborative phases. As an early work, MedAgents [254] established a role-playing environment where agents representing different medical experts achieve a consensus through independent analysis and iterative discussion. Concurrently, Wang et al. [270] designed a virtual medical team that includes a physician, a patient, and an examiner in order to model the consultation

[4] https://github.com/SakanaAI/AI-Scientist-ICLR2025-Workshop-Experiment/

flow across inquiry, examination, and diagnosis, guided by a hierarchical action set for dynamic responses. Similarly, Chen et al. [44] employed an admin agent for user information, doctor agents for diagnosis via dialogue, and a supervisor agent to ensure diagnostic consistency. Besides, RareAgents [41] focuses on rare diseases, featuring a primary physician agent that collaborates with multiple specialist agents over several rounds of discussion, integrating dynamic long-term memory and medical tools for personalized diagnostics.

Beyond simulating dialogue, other frameworks enhance diagnostic reasoning by integrating external knowledge bases and data-driven debate mechanisms. KG4Diagnosis [342] utilizes a hierarchical multi-agent structure where a general practitioner agent first conducts triage before coordinating with specialist agents who perform in-depth diagnosis leveraging a knowledge graph. This approach features an end-to-end pipeline for semantic knowledge extraction and human-guided reasoning, therefore improving system extensibility. MD2GPS [337] introduces a multi-agent debate system driven by both existing literature and patient data to diagnose Mendelian diseases effectively.

A parallel research direction empowers agents with tools for autonomous data analysis and evidence synthesis, transitioning them from communicators to actors. AI-HOPE [298] can interpret natural language commands into executable code, enabling it to autonomously analyze locally stored data for precision medicine research through tasks like association studies and survival analysis. Moreover, TxAgent [75] introduces ToolUniverse, a comprehensive suite of 211 specialized medical tools. By invoking these tools, the agent can retrieve and synthesize evidence from multiple sources, consider drug interactions and patient history, and iteratively refine treatment recommendations.

Addressing the heterogeneous nature of real-world medical data, recent efforts have focused on developing multi-modal diagnostic agents. Mmedagent [149] constructs a system where a Multi-modal Large Language Model (MLLM) acts as a planner, orchestrating a four-step process of user input interpretation, action planning, tool execution, and result aggregation, enhancing its tool-use proficiency via in-context learning. Similarly, MedAgent-Pro [279] adopts a hierarchical agentic workflow. It first retrieves clinical guidelines using Retrieval-Augmented Generation (RAG) to formulate a diagnostic plan. It then employs sequential tool calls to analyze the patient's multi-modal data, generating a final report that includes diagnostic evidence.

Table 5. An Overview of Evaluation Strategies of Agentic Frameworks in Healthcare. We summarize them from four levels. In benchmark and dataset Level, the framework is evaluate through domain-specific data with specific metrics; System level will evaluate the framework as a whole, with specific techniques; In environmental level, the framework is evaluated within a simulated healthcare environment, while the case study tend to evaluate the system through real-world cases, with ground truth provided.

Benchmark/Dataset Level		
Focus	**Related Work**	**Benchmark/Dataset**
Clinical Consultation Flow	Wang et al.[270]	MVME[64]
Zero-shot Medical Reasoning	MedAgents[254]	Jin et al.[121],Pal et al.[203],Pubmedqa[123], Hendrycks et al.[98]
Automated supervision of Healthcare Safety	TAO[134]	Safetybench[327],Medsafetybench[95], Chang et al.[33], Hu et al.[106],Wang et al.[273]
Evolvable Medical Agents	MDAgents[135]	MedQA[121], PubMedQA[123],MedBullets[35], JAMA[35],DDXPlus[66],SymCat[4], Path-VQA[96],PMC-VQA[320],MedVidQA[93],MIMIC-CXR[13]
Healthcare Intent Awareness	Medaide[283]	Pre-Diagnosis, Diagnosis, Medicament, Post-Diagnosis Bench[283]
Multimodal Tool-integration Diagnosis	MMedAgent[149]	VQA-RAD[142], Slake[167],Pmc-vqa[320],Pathvqa[96]
Tool-assist Therapeutic Reasoning	TxAgent[75]	DrugPC, BrandPC, GenericPC, DescriptionPC, TreatmentPC[75]
Rare Disease Curation	RareAgent[41]	RareBench[43],MIMIC-IV[125]
clinical trial	Clinical Agent[309]	DrugBank 5.0[285],Himmelstein[100]
Multi-modal Diagnosis	MedAgent-pro[279]	Refuge2 challenge[65],MITEA[330], MIMIC-IV[125], Nejm image challenge*
Mendelian Diseases Diagnose	MD2GPS[337]	SCH[109], JN[337],DDD[69], RD[316]
clinical environment simulation	MedAgentSim[5]	NEJM[235], MedQA[121],MIMIC-IV[125]
System Level		
Focus	**Related Work**	**Methods**
Knowledge Graph Enhancement for Medical Diagnosis	KG4Diagnosis[342]	Human Evaluation
Multi-modal Medical Diagnosis	MedAgent-pro[279]	
Benchmarking multimodal medical agent	Agentclinic[235]	
Multimodal tool-integration Diagnosis	MmedAgent[149]	Open-ended Medical Dialogue[150]
Medical Necessity Justification	Pandey et al.[206]	Parent and Leaf node Judgement with accuracy
Environmental Level		
Environment	**Related Work**	**Simulation Focus**
Entire Illness Treating	Agent Hospital[152]	Evolvable Medical Agents
Doctor-Patient Interaction	AI Hospital[64]	Medical Interaction Simulator
Multimodal clinical interaction	AgentClinic[235]	Multimodal agent benchmark
Case Study		
AIME[260], AI-HOPE[298], Clinical Agent[309]		

*https://www.nejm.org/image-challenge

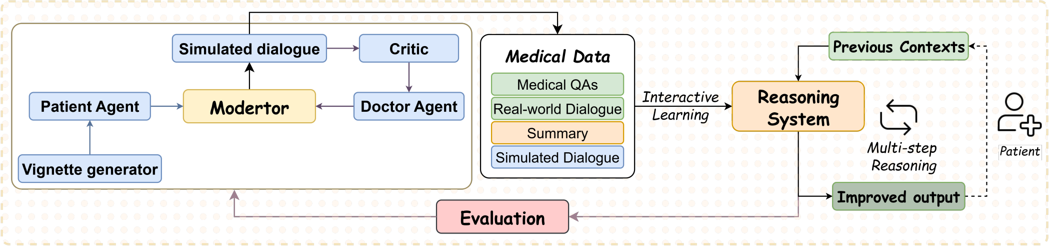


Fig. 9. Pipeline of AIME [260]. The framework is built upon two self-play loops. ***a) Inner Loop***: a doctor agent continuously optimize its behavior based on real-time feedback from a Critic module during simulated dialogues. ***b) Outer Loop***: the optimized simulated dialogues and other data will be gathered to improve (fine-tune) the Reasoning System, using an evaluation-feedback process to drive the model's continuous improvement. For real-time user interaction (reasoning process shown on the right), the system uses multi-step CoT reasoning and conversational context to ensure each output is accurate and well-grounded. Image is edited from Tu et al. [260]

4.2.2 Clinical Management and Automation. Recently, in order to efficiently manage and analyze complex medical information, a growing number of research are focusing on adapting general-purpose agentic management and automation systems. For instance, ClinicalAgent [309] utilizes a hierarchical multi-agent architecture to predict clinical trial outcomes, assessing drug efficacy, safety, and patient recruitment feasibility based on external data sources and predictive models. Other systems target the automation of healthcare services; Medaide [283], performs query rewriting via RAG and uses contextual encoding to identify fine-grained user intents. This process activates relevant agents to collaborate based on role assignments, delivering personalized diagnostic suggestions and department recommendations. Furthermore, Pandy et al. [206] explores the justification of prior authorizations by first reasoning over clinical guidelines and then employing a two-stage collaborative framework to decompose the problem into solvable sub-tasks for each agent.

However, directly implementing multi-agent collaboration can lead to challenges such as incompatible medical information flows and low component efficiency [16]. To enhance adaptability and dynamism, MDAgents [135] introduces a moderator agent that dynamically assembles appropriate multi-agent structures based on the complexity of the medical problem. This framework can configure specialized teams, such as a Primary Care Clinician, a Multidisciplinary Team, or an Integrated Care Team, and selectively employs reflection, iterative optimization, and collaborative methods to improve response accuracy. In order to further address limitations like potential single points of failure, TAO [134] proposes a tiered agentic operator framework to bolster system security and reliability. In this hierarchical structure, an "Agent Recruiter" selects medical agents based on safety benchmarks, while an "Agent Router" assesses and assigns them to different security tiers. The recruited agents then engage in hierarchical cooperation under strict safety protocols, enabling effective end-to-end supervision, which also incorporates possibilities for human oversight. This trend towards structured and secure workflows is underscored by the push for systems that can navigate the complexities of sensitive data and automated decisions while adhering to regulatory standards [199].

4.2.3 Environmental Simulation. Given that the healthcare domain is characterized by heterogeneous data which could be hard to collect, a significant line of research focuses on simulating realistic medical environments to enable agents to continuously optimize their performance through interactive learning. As a representative work, AIME [260] simulates a diagnostic environment with physician, patient, and referee roles. By leveraging role-playing and CoT [282] strategies, it allows an agent to self-tune its diagnostic capabilities using dialogue data within an automated feedback loop, as illustrated in Fig. 9.

Moreover, AgentClinic [235] constructs a more complex multi-modal clinical simulation, focusing on patient interaction, data collection with incomplete information, and medical tool usage. The paradigm takes a major leap forward with Agent Hospital [152], which creates a simulated hospital where agents can evolve autonomously based on task resolution, learning from both successful treatments and failed cases without reliance on manually annotated data.

Beyond agent training, these simulated environments also serve as sophisticated testbeds for evaluation and as platforms for medical education. For instance, AI Hospital [64] establishes a dynamic evaluation environment by simulating four distinct roles (physician, patient, inspector, and director) and introduces the corresponding MVME benchmark. Its decentralized, role-based setup allows for the assessment of an agent's capabilities in symptom collection, examination recommendation, diagnosis, and dispute resolution. As for medical education, Medco [281] builds a collaborative learning system by simulating multi-disciplinary scenarios where student agents can interact with patient agents, expert physicians, and radiologists to proactively gather information and refine their diagnostic decisions. Besides, MedAgentSim [5] presents a comprehensive simulation that requires agents to engage in multi-round, multi-modal interactions. It incorporates a self-improvement mechanism based on historical context and, notably, supports direct human interaction with the agents within the simulated environment.

4.3 Software Engineering

In contrast to Large Language Models specialized in code generation, agentic systems leverage a rich ecosystem of external tools to address a broader spectrum of software engineering tasks. This subsection examines the application of these agents in three key areas: *code generation and testing, program repair,* and *full-lifecycle software development.*

4.3.1 Code Generation and Testing. In the domain of code generation and testing, agentic systems significantly amplify the capabilities of LLMs beyond simple fine-tuning. These systems introduce structured collaboration and external tools, enhancing both the efficiency and reliability of code generation. Table.6 demonstrates the performance of the selected agentic coding frameworks on popular benchmarks. A primary approach involves decomposing the coding process using multi-agent frameworks that emulate human programmer workflows. Mapcoder [117] decouples code generation into four collaborating LLM agents for retrieval, planning, coding, and debugging. The framework features a dynamic agent traversal model that adapts based on confidence scores from the planning phase, alongside plan-guided debugging and autonomous retrieval. Similarly, Almorsi et al. [6] implements a deliberately structured and fine-grained approach, utilizing LLMs as fuzzy searchers and approximate information retrievers. This multi-agent system effectively compensates for the inherent limitations of LLMs in long-sequence reasoning and long-context understanding. Moreover, Agentcoder [110] pioneers a test-driven development (TDD) approach. It employs three distinct agents responsible for initial code generation, test case creation, and test execution with feedback. This test-driven iterative refinement loop enables the generation of higher-quality code with more efficient token usage.

Another major focuses is augmenting LLMs with external tools to improve code quality and mitigate hallucination. CoCoST [97] introduces a framework where a task planner decomposes complex problems, an online search module acquires external knowledge, and a code generator iterates with a correctness tester to fix latent bugs. Moreover, CodeAgent [314] integrates five distinct programming tools for repository-level code generation. Through rule-based tool usage, the system

Table 6. Performance Comparison of Different Code Generation Methods on Popular Benchmarks with Pass@1. We first demonstrates the performance of popular foundation LLMs, then collect several popular baseline prompt methods with GPT-3.5 and GPT-4, respectively. After that, we illustrate the performance of several work mentioned in this survey, with GPT-3.5 and GPT-4, respectively.

Method	Model	HumanEval[38]	HumanEval-ET [38]	MBPP[10]	MBPP-ET [56]	DS-1000[141]	EvalPlus[169]
-	GPT-3.5	57.3	42.7	52.2	36.8	-	66.5
-	GPT-4	67.6	50.6	68.3	52.2	-	-
-	GPT-4o	90.2	-	-	-	-	-
-	Claude-3.5 Sonnet	92.0	-	-	-	-	-
CoT[282]	GPT-3.5	44.6	37.2	46.1	34.8	-	65.2
ReAct[304]		56.9	49.4	67.0	45.9	-	66.5
Reflexion[241]		68.1	50.6	70.0	47.5	-	62.2
Self-planning[119]		65.2	48.8	58.6	41.5	-	-
Self-debugging[42]		61.6	45.8	60.1	52.3	-	-
CoT[282]	GPT-4	89.0	73.8	81.1	54.7	-	81.7
Reflexion[241]		91.0	78.7	78.3	51.9	-	81.7
Self-debugging[42]		-	-	80.6	-	-	-
AgentCoder[110]	GPT-3.5	79.9	77.4	89.9	84.1	-	-
Dong et al. [57]		74.4	56.1	68.2	49.5	-	-
INTERVENOR[261]		-	-	-	-	39.7	-
Mapcoder[117]		80.5	70.1	78.3	54.4	-	71.3
CoCoST[97]	GPT-4	-	-	-	-	68.0	-
Mapcoder[117]		93.9	82.9	83.1	57.7	-	83.5
MetaGPT[102]		85.9	-	87.7	-	-	-
AgentVerse[39]		89.0	-	73.5	-	-	-
ChatDev[213]		84.1	-	79.8	-	-	-
Dong et al.[57]		90.2	70.7	78.9	62.1	-	-
AgentCoder[110]		96.3	86.0	91.8	91.8	-	-

interacts with various software artifacts, iteratively performing information retrieval, code symbol navigation, and code testing. The authors also introduced CodeBench, a comprehensive benchmark for repository-level code generation, featuring code repositories across multiple domains.

4.3.2 Program Repair. Complementary to code generation, automated program repair (APR) is another cornerstone of Agent System in software engineering. During the multi-step reasoning process, they could systematically understanding code, localizing faults, generating patches, and validating fixes, often through sophisticated tool use and collaborative strategies. Table 7 shows the performance of selected APR agentic frameworks on popular software repair benchmarks.

A common strategy is to decompose the complex repair process into a structured workflow. RepairAgent [23] formalizes this into four stages: defect information collection, fault localization, patch generation, and validation, using a state machine to dynamically select tools and adapt its strategy. Similarly, AgentFL [216] employs a multi-agent system to infer defect causes, search for relevant context using program instrumentation, and validate fixes, leveraging multi-turn dialogue to manage context length. Inventor [261] introduced an interactive "repair chain" concept, where a "code teacher" agent analyzes compiler errors to generate natural language suggestions, guiding a "code learner" agent in an iterative repair process. Agentless [289] simplifies the process into three phases – locate, fix, and validate – using a hierarchical strategy that combines semantic understanding with code embedding retrieval to rapidly identify suspicious code snippets.

A critical sub-task within this workflow is precise fault localization. Several specialized agents have been developed for this purpose. OrcaLoca [308] focuses on efficient localization by using dynamic priority scheduling and relevance scoring to prioritize actions, along with distance-aware context pruning to filter irrelevant code. LocAgent [48] introduces a novel graph-based approach, parsing the codebase into a directed heterogeneous graph that captures structural dependencies, enabling an LLM agent to perform effective multi-hop reasoning for entity localization. Meanwhile, VulDebugger [177] utilizes both static and dynamic program analysis, continuously comparing the actual program state (observed via a debugger) with the expected state (inferred from constraints) to identify and rectify errors.

Building on these principles, recent frameworks aim to provide end-to-end solutions for real-world scenarios like resolving GitHub issues. MAGIS [256] employs a centralized architecture with four agents orchestrated by a central controller to manage task decomposition, file retrieval, code modification, and review. Autocoderover [325] further tackles GitHub issues by integrating structured code search with spectrum-based fault localization (SBFL) to pinpoint buggy methods. It then iteratively retrieves context via API calls and refines the issue description to synthesize a patch, demonstrating a robust solution in a practical setting. Besides, Rondon et al. [223] explored the viability of agent-based program repair in an enterprise environment, contributing a valuable dataset of both human and machine reported bugs and offering insights into real-world applicability.

Table 7. Performance Comparison on Popular Software Repair Benchmarks with Pass@1. We first present the performance of foundation LLMs on these benchmarks, then provide the performance of different methods mentioned in this survey with GPT-3.5, GPT-4, GPT-4o, Claude-3.5 Sonnet, respectively.
†AutoCoderRover-v2 is mentioned in Agentless[289], with the given reference*

Method	Model	Defects4J(Top1) [127]	SWE-bench[120]	SWE-bench-lite [120]
-	GPT-3.5	121/395	0.84%	-
-	GPT-4	-	1.74%	-
AgentFL[216]	GPT-3.5	157/395	-	-
RepairAgent[23]		90/440	-	-
MAGIS[256]	GPT-4	-	13.94%	-
AutoCodeRover[325]		-	12.42%	19.0%
SWE-Agent[300]		-	12.74%	18.0%
Agentless[289]	GPT-4o	-	-	32%
SWE-Agent[300]		-	-	18.33%
AutoCodeRover-v2†		-	-	30.67%
OrcaLoca[308]	Claude-3.5 Sonnet	-	-	41.00%

*https://www.autocoderover.net/

4.3.3 *Full-cycle Development.* Beyond discrete tasks like generation and repair, agentic systems are increasingly engineered to automate the entire software development lifecycle (SDLC), from initial requirements analysis to final testing and documentation. These systems often simulate human software teams and adopt established development methodologies. Early explorations constructed virtual teams composed of agents with distinct roles. Dong et al. [57] organize a analyst, a coder, and a tester agents managed by a waterfall model, communicating via a shared blackboard. Moreover, CHatdev [213] formalizes the process into design, coding, and testing phases, enabling role-specific agents to collaborate through natural language. It introduces a *chat chain* for task refinement and a *communicative de-hallucination* mechanism to ensure requirement clarity before coding. A significant advancement came with MetaGPT [102], which mimics the Standard Operating Procedures (SOPs) of human software companies. By assigning roles and enforcing structured workflows, MetaGPT facilitates hierarchical multi-agent collaboration across the full development pipeline, from requirements analysis to system design and testing, ensuring accurate information flow and reducing communication overhead.

Furthermore, RepoAgent [179] specializes in automated code documentation. It analyzes project-level hierarchy and code dependencies to enrich LLM prompts and integrates with Git to maintain consistency between code and documentation in real-time. Similarly, DocAgent [297] employs a multi-agent team – comprising a reader, searcher, writer, validator, and coordinator, and uses topological code processing to automate the generation of comprehensive software documentation
.

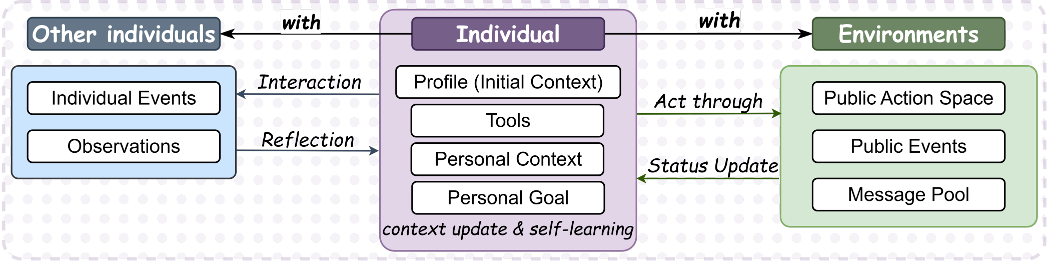


Fig. 10. The fundamental paradigm for social simulation based on an Agentic Reasoning Framework. In this framework, each Individual is powered by LLM-based Agent. *a)* ***Individual*** *(purple at the middle)*: Each agent possesses its unique initial profile, personal goal, available tools, and a dynamically updated personal context, engaging in dynamic learning and decision-making based on continuous context updates. *b)* ***Individual with Others*** *(blue at the left)*: Individual continuously updates its cognition, goals, and context by observing the behaviors of other individuals and reflecting on the Individual Events that arise from interactions. *c)* ***Individual with Environments*** *(green at the right)*: When individual executes an action in the Public Action Space, it can trigger Public Events or leave messages in the Message Pool. Correspondingly, the environment feeds these changes back to the agent via a Status Update, thus forming a complete interaction loop that influences its subsequent decisions. *In summary*, through the dual loops of social and environmental interaction, each agent maintains the independence of its context and goals, executes distinct asynchronous social behaviors, and thereby contributes to the emergence of complex and realistic social dynamics at the group level.

A more fundamental innovation focus on how could agents interact with their developing environment. SWE-Agent [300] pioneers an Agent-Computer Interface (ACI) that allows an agent to perform operations directly on a code repository, such as creating and editing files, navigating the filesystem, and executing test suites. Concurrently, Openhands [275] imitates human developer interactions by providing agents with a sandboxed environment where they can write code, use a command-line interface, browse the web, and coordinate complex tasks, granting them a higher degree of autonomy. Besides, at the level of software system, MAAD [322] tackles the complex process of software architecture design by creating a multi-agent system that learns from existing design knowledge, academic literature, and expert experience to generate and optimize architectures. Meanwhile SyncMind [91] identifies and addresses core challenges within these systems, such as the "belief inconsistency" problem in collaborative software engineering, and proposes SyncBench benchmarks to validate the solutions.

4.4 Social and Economic Simulation

Apart from previous setions, the advent and rapid advancement of Large Language Models have also established a revolutionary paradigm for simulating social and economic behaviors. LLM-based agents, endowed with sophisticated, human-like capacities for reasoning, perception, and action, serve as the foundational elements of this new approach. Ranging from single-agent decision-making models to complex multi-agent systems, these frameworks are designed to construct and simulate critical socioeconomic dynamics at various scales. Consequently, they provide a powerful and versatile methodology for researchers to explore complex phenomena within the social and economic sciences.

4.4.1 Social Simulation. LLM-based social simulation leverages the autonomy of agents to model a wide spectrum of emergent social behaviors. They could interact with different individuals which are also powered by LLMs, or dynamically act, learn and improve themselves from the simulated environments, as demonstrated in Fig. 10, Early and influential research in this area often operates within text-based sandbox environments, where agents are assigned specific profiles and action spaces to drive interaction. One of the pioneer work is Generative Agents [208], which established this paradigm by creating an interactive sandbox with 25 agents, each possessing the capacity to observe, plan, and reflect, thus exhibiting distinct personalities. This setup allowed for the qualitative assessment of individual and collective behaviors through natural language interaction. Following this paradigms, subsequent studies have explored critical social dynamics. For instance, MetaAgents [157] assessed coordination skills in a simulated job fair, while GOVSIM [211] investigated whether agents could negotiate sustainable equilibria in a commons dilemma. Other research has used similar frameworks to study the formation of information cocoons, conformity [267], and voting behaviors in election scenarios [319].

Moreover, the focus expanded from specific simulations to the development of generalizable platforms and the modeling of online interactions. To enhance reusability and extensibility, frameworks like GenSim [252] and Sotopia-S4 [338] were introduced. These platforms provide configurable systems where researchers can define roles, contextual information, and action spaces to customize diverse multi-agent scenarios and test hypotheses with tailored evaluation metrics. Besides, research began to address the complexities of online social networks. BASES [221] modeled the emergent web search behaviors of diverse user profiles, while Mou et al. [195] simulated the propagation of influence in social movements by differentiating between core agent-driven users and peripheral users. The challenge of simulating malicious online environments was tackled by BotSim [214], which created a mixed network of agents and human users. Similarly, Y-Social [225] employed digital twin technology to replicate the dynamics of user interactions on social media platforms.

A primary challenge in social simulation is achieving realism at a large scale, while Recent systems have made solid progress in this direction. Oasis [303] implements large-scale user simulation by integrating dynamic context updates and an interest-based recommendation system modeled after real-world platforms. SocioVerse [318] further enhances scalability by introducing and restructuring real-world information to create distinct contextual environments that drive divergent individual behaviors. In addition, S3 [72] simplifies the simulation of large-scale social networks by employing group agents. It follows a hierarchical architecture where each group agent represents a demographic population rather than a single individual. By assigning characteristics, emotions, and attitudes based on role distributions, the system can simulate macro-level network interactions, offering a computationally efficient abstraction for massive-scale social dynamics. As a conclusion, Table.8 summarize these simulation work with their different simulation focus, data source and scale.

Table 8. A Collection of Different Social Simulating Methods.

Simulation Focus	Data Source	Related Work	#Agent / User
Sustainable Cooperation	Simulated	GoverSim[211]	-
Job Fair	Simulated	MetaAgents[157]	-
Interactive Human Behavior	Simulated	Generative Agents[208]	25
Recommendation System	Simulated	RecAgent[267]	20
Malicious Social Botnet	Simulated	BotSim[214]	~3k
Population-level Interaction	Simulated	S3[72]	~10k
Massive Population Election	Real-world	ElectionSim[319]	10k
Web Search User Simulation	Real-world	BASES[221]	200k
World Model	Real-world	SocialVerse[318]	10M
General Social Simulation Platform	Simulated	Sotopia-S4[338]	150
		GenSim[252]	~100k
Social Media	Real-world	Y Social[225]	-
		Mou et al.[195]	1k
		OASIS[303]	100k

4.4.2 Economic Perception and Simulation. Agentic frameworks are increasingly being leveraged to perceive, analyze, and simulate complex economic markets, a domain with immense real-world value [26]. The evolution of these frameworks can be understood through three advancing frontiers: enhancing agent-native cognitive abilities, optimizing collaborative structures, and creating large-scale market simulations.

Initially, efforts focused on equipping individual agents with sophisticated cognitive modules. Early work like Finmem [306] established a foundational agent system with analysis, memory, and decision modules, where the memory component utilized reflection and interactive learning to process historical context. Similarly, EconAgent [153] situated an LLM-based agent within a simulated macroeconomic environment, employing perception, memory, and reflection modules to analyze its behavior. This focus on reflection as a core cognitive faculty was further advanced by subsequent systems. Finvision [67] places systematic reflection at its center, using a multi-agent system to analyze historical trading signals and generate feedback to improve future decisions. More recently, Zhang et al. [317] introduced a sophisticated two-level reflection module. This module processes multimodal market information to establish distinct causal links between market data and price movements, while also reflecting on historical trading performance. Its memory system separately stores parsed market information and the insights from both levels of reflection to inform decision-making.

Building upon individual agent capabilities, another line of research explores the optimization of collaborative structures. As a pioneer work, Fincon [307] addresses stock trading and portfolio management using a hierarchical manager-analyst multi-agent system. This structure facilitates synchronous collaboration and employs self-criticism to monitor market risks and update investment theses, thereby achieving robust risk control. Notably, conceptualized beliefs are only selectively communicated among agents, effectively reducing the communication overhead typical in multi-agent systems. The importance of structured collaboration is also highlighted in TradingAgents [290], which enhance automated trading performance through explicit role and objective allocation combined with streamlined information integration.

The most ambitious application in this domain involves creating comprehensive, closed-loop platforms and simulating entire economic environments. FinRobot [299] implements a full-cycle financial analysis platform. At its agent layer, it uses CoT [282] to deconstruct complex financial problems, dynamically selecting or fine-tuning different LLMs and applying varied algorithms based on the task, thus enabling rapid market response. Pushing the boundaries further, large-scale simulations now aim to replicate real-world market dynamics. StockAgent [312] deploys a massive multi-agent system to simulate a stock trading environment, allowing users to assess how external factors influence investor behavior and profitability. Similarly, Gao et al. [74] simulates the stock market by creating agents equipped with unique profiles, observational capabilities, and tool-based learning. By integrating these agents with a realistic order-matching system, these frameworks achieve a high-fidelity simulation of actual stock market operations, opening new avenues for economic research and policy testing.

4.5 Others

Beyond the applications in mainstream scenarios, *Agentic Reasoning Frameworks* also demonstrate distinct potential in several other significant domains. Although agent-based approaches have not yet become mainstream in these fields, a growing body of pioneering work has begun to explore the frontiers of their capabilities in complex tasks like *Embodied Interaction*, *GUI Operation*, and *Strategic Reasoning*.

In *Embodied Interaction*, an agent's reasoning process typically follows a perceive-plan-execute-memory cycle. The primary goal in this field is to empower agents to learn and complete tasks autonomously through continuous interaction with a virtual or physical environment [175]. For example, based on training-free framework, Voyager [277] enables an agent to achieve lifelong learning in the game *Minecraft* by designing a specialized action space and a reasoning framework that integrates iterative feedback and continuous evolution. However, existing general foundational Multimodal Large Language Models (MLLMs) still show deficiencies in handling complex, multi-level task planning. Consequently, a prevailing research direction is to fine-tune base models for specific embodied scenarios to enhance the agent's perception and planning capabilities in a tailored manner [175].

GUI (Graphical User Interface) Operation aims to enable agents to operate applications on phones, computers, and the web with human-like proficiency [332]. Researchers usually build the framework base on state-of-the-art vision-language models like GPT-4V [295]. They enhance reasoning frameworks by integrating visual memory and knowledge bases and introducing multi-agent collaboration mechanisms [146, 263, 280, 313]. These are combined with conventional components like reflection and context updating to effectively decompose complex GUI navigation tasks and enable continuous learning. However, as task complexity increases, the research focus is shifting from relying on the zero-shot capabilities of models towards more intensive, specialized training via Supervised Fine-Tuning (SFT) or Reinforcement Learning (RL) [271].

Strategic Reasoning requires an agent to understand and predict the behavior of opponents and dynamically adapt its own strategy, with the core challenge being the management of dynamics and uncertainty in multi-agent interactions. Currently, the research focus in this area is not on designing novel reasoning frameworks but rather on creating diverse test environments to accurately evaluate and enhance the strategic capabilities of LLMs [324]. These environments span a wide range, as evidenced by the development of benchmarks based on real-time strategy (RTS) games like *StarCraft II* [182]; the use of LLMs to generate expert-level decision explanations for board games [131]; and the systematic analysis of their behavioral rationality through classic game theory models [63].

5 Future Prospects

As discussed in previous chapters, agentic reasoning frameworks have made significant progress in both theory and application. However, the path to achieving a truly general, trustworthy, and efficient agent system is still full of challenges. In this chapter, we propose six potential directions for future development.

5.1 Scalability and Efficiency of Reasoning

As task complexity increases, the scalability and efficiency of agent frameworks have become major bottlenecks for large-scale applications. For instance, in multi-agent systems, poor task decomposition can cause system performance to degrade sharply as the system scales up [28]. Simply increasing the number of agents or extending reasoning time is often unsustainable, leading to spiraling costs and diminishing returns. Future works could focus on innovations at the framework level. On one hand, it will be crucial to design more efficient context management mechanisms for large-scale expansion. On the other hand, frameworks should also be equipped with dynamic task allocation and adaptive resource scheduling capabilities to handle complex tasks flexibly and efficiently.

5.2 Open-ended Autonomous Learning

Achieving open-ended autonomous learning is a key vision in agent research. The goal is to evolve agents from being mere "users" of existing knowledge into "creators" of new knowledge and tools [73], breaking free from reliance on specific environments like games [277]. We have observed that current reasoning frameworks typically rely on static toolsets, fixed interaction logic, and pre-defined prompts. This rigidity constrains an agent's creativity, potentially leading to poor performance on complex, zero-shot problems. Therefore, future framework design should focus on equipping agents with the ability to dynamically generate and optimize tools. This would allow them to autonomously create and iterate on their own methods during the reasoning process [68]. Alongside this, there is also a need to establish an effective and reasonable evaluation system to assess an agent's capacity for learning and creation in an open-ended world.

5.3 Dynamic Reasoning Framework

Improving a framework's ability to adapt to complex tasks is crucial for the evolution of agents. Currently, this adaptability mainly involves making high-level adjustments for different types of tasks by setting up different collaboration architectures [134]. However, within the multi-step reasoning process of a single complex task, the collaboration pattern inside the framework often remains static. Future research should focus on a framework's ability to self-regulate *during* the reasoning process for a single task. This requires the framework to have a deep understanding of its own reasoning process, enabling it to perceive the goal of the current step and dynamically reconfigure the interaction topology and collaboration protocols between agents. The framework should then be able to select and execute the optimal reasoning path to achieve the best balance between resource efficiency and reasoning quality.

5.4 Ethics and Fairness in Reasoning

Building trustworthy and responsible agent systems is an essential prerequisite for their deployment in the real world. As these systems become more autonomous and complex, individual biases may be amplified [262], and it will become increasingly difficult to hold them accountable and correct flawed reasoning [196]. Future research should focus on enhancing the framework's ability to proactively manage bias. This means equipping it to anticipate, identify, and mitigate potential biases during the reasoning process itself. Additionally, the framework should be able to provide clear ethical justifications for every key decision, establishing a reliable pathway for external auditing and accountability.

5.5 Reliance and Safety in Reasoning

The safety challenges for agent frameworks have evolved from securing a single language model to protecting a complex, dynamic system composed of memory, planning, and tool interfaces. This shift introduces new risks: beyond traditional attacks on LLMs, every core module and external interface of an agent can become a new target. Attackers can exploit these entry points by poisoning API data to manipulate an agent's "perception" or by hijacking its reasoning chain to control its "decisions", leading to data leaks and more severe illicit operations [264]. Future work should approach the agent system as a whole at the framework level. By implementing dynamic, coordinated defenses between components, the system can quickly respond to and patch vulnerabilities, thereby enhancing its reliability and security.

5.6 Confidence Estimation and Explainable Agentic Reasoning

When an agent system becomes an automated decision-maker, it needs a precise way to evaluate and communicate the trustworthiness of its reasoning process. Future work should focus on establishing quantifiable mechanisms for uncertainty-aware confidence estimation. For instance, introspective reasoning could be conducted within an agentic framework to align internal uncertainty with inherent task ambiguity [94, 161]. When faced with high uncertainty, an agent could actively seek information to clarify ambiguity [107]. Furthermore, calibrating the agent's confidence during tool invocation is also crucial, since interactions with external environments and tools are major sources of uncertainty for agentic frameworks [168]. This would transform confidence evaluation from an agent's unreliable self-declaration into a credible, objective proof, ensuring its safe deployment in critical fields.

6 Conclusion

With the explosive growth of large language model (LLM) based agentic reasoning methods and applications, a systematic understanding of these approaches and their scenarios has become crucial. We propose a unified taxonomy that breaks down agentic systems into three progressive levels, from single-agent methods, tool-based methods, to multi-agent systems. This framework offers a clear views through which to analyze the field. Building on this, we systematically reviewed how these frameworks are put into practice across major application domains, covering their core methodologies, key focuses, and evaluation approaches. Finally, we present our insights on the future directions of agentic reasoning, aiming to promote the development of agentic frameworks for the future generation.

References

[1] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. 2024. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature* 630, 8016 (2024), 493–500.

[2] Vaibhav Adlakha, Parishad BehnamGhader, Xing Han Lu, Nicholas Meade, and Siva Reddy. 2024. Evaluating correctness and faithfulness of instruction-following models for question answering. *Transactions of the Association for Computational Linguistics* 12 (2024), 681–699.

[3] Ali AhmadiTeshnizi, Wenzhi Gao, and Madeleine Udell. 2024. OptiMUS: scalable optimization modeling with (MI) LP solvers and large language models. In *Proceedings of the 41st International Conference on Machine Learning*. 577–596.

[4] Zaid Al-Ars, Obinna Agba, Zhuoran Guo, Christiaan Boerkamp, Ziyaad Jaber, and Tareq Jaber. 2023. Nlice: Synthetic medical record generation for effective primary healthcare differential diagnosis. In *2023 IEEE 23rd International Conference on Bioinformatics and Bioengineering (BIBE)*. IEEE, 397–402.

[5] Mohammad Almansoori, Komal Kumar, and Hisham Cholakkal. 2025. Self-Evolving Multi-Agent Simulations for Realistic Clinical Interactions. *arXiv preprint arXiv:2503.22678* (2025).

[6] Amr Almorsi, Mohanned Ahmed, and Walid Gomaa. 2024. Guided code generation with llms: A multi-agent framework for complex code tasks. In *2024 12th International Japan-Africa Conference on Electronics, Communications, and Computations (JAC-ECC)*. IEEE, 215–218.

[7] Vinicius M Alves, Eugene Muratov, Denis Fourches, Judy Strickland, Nicole Kleinstreuer, Carolina H Andrade, and Alexander Tropsha. 2015. Predicting chemically-induced skin reactions. Part I: QSAR models of skin sensitization and their application to identify potentially hazardous compounds. *Toxicology and applied pharmacology* 284, 2 (2015), 262–272.

[8] Theonie Anastassiadis, Sean W Deacon, Karthik Devarajan, Haiching Ma, and Jeffrey R Peterson. 2011. Comprehensive assay of kinase catalytic activity reveals features of kinase inhibitor selectivity. *Nature biotechnology* 29, 11 (2011), 1039–1045.

[9] D Armstrong. 2024. Exercises from "introduction to Computational Chemistry"(CHM 323), University of toronto. *Personal communication* (2024).

[10] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732* (2021).

[11] Reza Averly, Frazier N Baker, and Xia Ning. 2025. Liddia: Language-based intelligent drug discovery agent. *arXiv preprint arXiv:2502.13959* (2025).

[12] Kaito Baba, Chaoran Liu, Shuhei Kurita, and Akiyoshi Sannai. 2025. Prover Agent: An Agent-based Framework for Formal Mathematical Proofs. *arXiv preprint arXiv:2506.19923* (2025).

[13] Seongsu Bae, Daeun Kyung, Jaehee Ryu, Eunbyeol Cho, Gyubok Lee, Sunjun Kweon, Jungwoo Oh, Lei Ji, Eric Chang, Tackeun Kim, et al. 2023. Ehrxqa: A multi-modal question answering dataset for electronic health records with chest x-ray images. *Advances in Neural Information Processing Systems* 36 (2023), 3867–3880.

[14] Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. 2025. ResearchAgent: Iterative Research Idea Generation over Scientific Literature with Large Language Models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*

(Volume 1: Long Papers). 6709–6738.

[15] Kinjal Basu, Ibrahim Abdelaziz, Kiran Kate, Mayank Agarwal, Maxwell Crouse, Yara Rizk, Kelsey Bradford, Asim Munawar, Sadhana Kumaravel, Saurabh Goyal, et al. 2024. Nestful: A benchmark for evaluating llms on nested sequences of api calls. *arXiv preprint arXiv:2409.03797* (2024).

[16] Suhana Bedi, Iddah Mlauzi, Daniel Shin, Sanmi Koyejo, and Nigam H Shah. 2025. The Optimization Paradox in Clinical AI Multi-Agent Systems. *arXiv preprint arXiv:2506.06574* (2025).

[17] RM Belbin and V Brown. 2012. Team roles at work. *Routledge* (2012).

[18] Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. 2025. Small Language Models are the Future of Agentic AI. *arXiv preprint arXiv:2506.02153* (2025).

[19] A Patrícia Bento, Anna Gaulton, Anne Hersey, Louisa J Bellis, Jon Chambers, Mark Davies, Felix A Krüger, Yvonne Light, Lora Mak, Shaun McGlinchey, et al. 2014. The ChEMBL bioactivity database: an update. *Nucleic acids research* 42, D1 (2014), D1083–D1090.

[20] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. 2000. The protein data bank. *Nucleic acids research* 28, 1 (2000), 235–242.

[21] G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. 2012. Quantifying the chemical beauty of drugs. *Nature chemistry* 4, 2 (2012), 90–98.

[22] Lisa Bortolotti. 2011. Does reflection lead to wise choices? *Philosophical Explorations* 14, 3 (2011), 297–313.

[23] Islem Bouzenia, Premkumar Devanbu, and Michael Pradel. 2025. RepairAgent: An Autonomous, LLM-Based Agent for Program Repair. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE Computer Society, 694–694.

[24] Andres M Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. 2023. Chemcrow: Augmenting large-language models with chemistry tools. *arXiv preprint arXiv:2304.05376* (2023).

[25] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[26] Bokai Cao, Saizhuo Wang, Xinyi Lin, Xiaojun Wu, Haohan Zhang, Lionel M Ni, and Jian Guo. 2025. From deep learning to LLMs: a survey of AI in quantitative investment. *arXiv preprint arXiv:2503.21422* (2025).

[27] Julia Carnevale, Eric Shifrut, Nupura Kale, William A Nyberg, Franziska Blaeschke, Yan Yi Chen, Zhongmei Li, Sagar P Bapat, Morgan E Diolaiti, Patrick O'Leary, et al. 2022. RASA2 ablation in T cells boosts antigen sensitivity and long-term function. *Nature* 609, 7925 (2022), 174–182.

[28] Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, et al. 2025. Why do multi-agent llm systems fail? *arXiv preprint arXiv:2503.13657* (2025).

[29] Jingyi Chai, Shuo Tang, Rui Ye, Yuwen Du, Xinyu Zhu, Mengcheng Zhou, Yanfeng Wang, Siheng Chen, et al. 2025. SciMaster: Towards General-Purpose Scientific AI Agents, Part I. X-Master as Foundation: Can We Lead on Humanity's Last Exam? *arXiv preprint arXiv:2507.05241* (2025).

[30] Jiajun Chai, Zijie Zhao, Yuanheng Zhu, and Dongbin Zhao. 2025. A Survey of Cooperative Multi-Agent Reinforcement Learning for Multi-Task Scenarios. *Artificial Intelligence Science and Engineering* 1, 2 (2025), 98–121.

[31] Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2024. ChatEval: Towards Better LLM-based Evaluators through Multi-Agent Debate. In *The Twelfth International Conference on Learning Representations*.

[32] Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, et al. 2025. MLE-bench: Evaluating Machine Learning Agents on Machine Learning Engineering. In *The Thirteenth International Conference on Learning Representations*.

[33] Crystal T Chang, Hodan Farah, Haiwen Gui, Shawheen Justin Rezaei, Charbel Bou-Khalil, Ye-Jean Park, Akshay Swaminathan, Jesutofunmi A Omiye, Akaash Kolluri, Akash Chaurasia, et al. 2024. Red teaming large language models in medicine: real-world insights on model behavior. *medRxiv* (2024), 2024–04.

[34] Guangyao Chen, Siwei Dong, Yu Shu, Ge Zhang, Jaward Sesay, Börje Karlsson, Jie Fu, and Yemin Shi. 2024. AutoAgents: a framework for automatic agent generation. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*. 22–30.

[35] Hanjie Chen, Zhouxiang Fang, Yash Singla, and Mark Dredze. 2025. Benchmarking large language models on answering and explaining challenging medical questions. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 3563–3599.

[36] Kexin Chen, Junyou Li, Kunyi Wang, Yuyang Du, Jiahui Yu, Jiamin Lu, Lanqing Li, Jiezhong Qiu, Jianzhang Pan, Yi Huang, et al. 2023. Chemist-X: Large language model-empowered agent for reaction condition recommendation in chemical synthesis. *arXiv preprint arXiv:2311.10776* (2023).

[37] Kai Chen, Xinfeng Li, Tianpei Yang, Hewei Wang, Wei Dong, and Yang Gao. 2025. MDTeamGPT: A Self-Evolving LLM-based Multi-Agent Framework for Multi-Disciplinary Team Medical Consultation. *arXiv preprint arXiv:2503.13856* (2025).

[38] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374* (2021).

[39] Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, et al. 2024. AgentVerse: Facilitating Multi-Agent Collaboration and Exploring Emergent Behaviors. In *ICLR*.

[40] Weize Chen, Ziming You, Ran Li, Chen Qian, Chenyang Zhao, Cheng Yang, Ruobing Xie, Zhiyuan Liu, Maosong Sun, et al. 2025. Internet of Agents: Weaving a Web of Heterogeneous Agents for Collaborative Intelligence. In *The Thirteenth International Conference on Learning Representations*.

[41] Xuanzhong Chen, Ye Jin, Xiaohao Mao, Lun Wang, Shuyang Zhang, and Ting Chen. 2024. RareAgents: Autonomous Multi-disciplinary Team for Rare Disease Diagnosis and Treatment. *arXiv e-prints* (2024), arXiv–2412.

[42] Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2024. Teaching Large Language Models to Self-Debug. In *The Twelfth International Conference on Learning Representations*.

[43] Xuanzhong Chen, Xiaohao Mao, Qihan Guo, Lun Wang, Shuyang Zhang, and Ting Chen. 2024. RareBench: can LLMs serve as rare diseases specialists?. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*. 4850–4861.

[44] Xi Chen, Huahui Yi, Mingke You, WeiZhi Liu, Li Wang, Hairui Li, Xue Zhang, Yingman Guo, Lei Fan, Gang Chen, et al. 2025. Enhancing diagnostic capability with multi-agents conversational large language models. *NPJ digital medicine* 8, 1 (2025), 159.

[45] Yuxuan Chen, Xu Zhu, Hua Zhou, and Zhuyin Ren. 2024. MetaOpenFOAM: an LLM-based multi-agent framework for CFD. *arXiv preprint arXiv:2407.21320* (2024).

[46] Yuxuan Chen, Xu Zhu, Hua Zhou, and Zhuyin Ren. 2025. Metaopenfoam 2.0: Large language model driven chain of thought for automating cfd simulation and post-processing. *arXiv preprint arXiv:2502.00498* (2025).

[47] Zehui Chen, Kuikun Liu, Qiuchen Wang, Jiangning Liu, Wenwei Zhang, Kai Chen, and Feng Zhao. 2025. MindSearch: Mimicking Human Minds Elicits Deep AI Searcher. In *The Thirteenth International Conference on Learning Representations*.

[48] Zhaoling Chen, Xiangru Tang, Gangda Deng, Fang Wu, Jialong Wu, Zhiwei Jiang, Viktor Prasanna, Arman Cohan, and Xingyao Wang. 2025. Locagent: Graph-guided llm agents for code localization. *arXiv preprint arXiv:2503.09089* (2025).

[49] Yuan Chiang, Elvis Hsieh, Chia-Hong Chou, and Janosh Riebesell. 2025. LLaMP: Large Language Model Made Powerful for High-fidelity Materials Knowledge Retrieval. In *AI for Accelerated Materials Design-ICLR*.

[50] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168* (2021).

[51] 1000 Genomes Project Consortium et al. 2012. An integrated map of genetic variation from 1,092 human genomes. *Nature* 491, 7422 (2012), 56.

[52] Steven M Corsello, Joshua A Bittker, Zihan Liu, Joshua Gould, Patrick McCarren, Jodi E Hirschman, Stephen E Johnston, Anita Vrcic, Bang Wong, Mariya Khan, et al. 2017. The Drug Repurposing Hub: a next-generation drug library and information resource. *Nature medicine* 23, 4 (2017), 405–408.

[53] Debrup Das, Debopriyo Banerjee, Somak Aditya, and Ashish Kulkarni. 2024. MATHSENSEI: A Tool-Augmented Large Language Model for Mathematical Reasoning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 942–966.

[54] Tom DeMarco and Tim Lister. 2013. *Peopleware: productive projects and teams.* Addison-Wesley.

[55] Han Ding, Yinheng Li, Junhao Wang, and Hang Chen. 2024. Large language model agent in financial trading: A survey. *arXiv preprint arXiv:2408.06361* (2024).

[56] Yihong Dong, Jiazheng Ding, Xue Jiang, Ge Li, Zhuo Li, and Zhi Jin. 2025. Codescore: Evaluating code generation by learning code execution. *ACM Transactions on Software Engineering and Methodology* 34, 3 (2025), 1–22.

[57] Yihong Dong, Xue Jiang, Zhi Jin, and Ge Li. 2024. Self-collaboration code generation via chatgpt. *ACM Transactions on Software Engineering and Methodology* 33, 7 (2024), 1–38.

[58] Mengge Du, Yuntian Chen, Zhongzheng Wang, Longfeng Nie, and Dongxiao Zhang. 2024. LLM4ED: Large Language Models for Automatic Equation Discovery. *CoRR* (2024).

[59] Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*. 11733–11763.

[60] Zane Durante, Qiuyuan Huang, Naoki Wake, Ran Gong, Jae Sung Park, Bidipta Sarkar, Rohan Taori, Yusuke Noda, Demetri Terzopoulos, Yejin Choi, et al. 2024. Agent ai: Surveying the horizons of multimodal interaction. *arXiv preprint arXiv:2401.03568* (2024).

[61] Abul Ehtesham, Aditi Singh, Gaurav Kumar Gupta, and Saket Kumar. 2025. A survey of agent interoperability protocols: Model context protocol (mcp), agent communication protocol (acp), agent-to-agent protocol (a2a), and agent network protocol (anp). *arXiv preprint arXiv:2505.02279* (2025).

[62] Peter Ertl and Ansgar Schuffenhauer. 2009. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics* 1, 1 (2009), 8.

[63] Caoyun Fan, Jindou Chen, Yaohui Jin, and Hao He. 2024. Can large language models serve as rational players in game theory? a systematic analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 17960–17967.

[64] Zhihao Fan, Lai Wei, Jialong Tang, Wei Chen, Wang Siyuan, Zhongyu Wei, and Fei Huang. 2025. AI Hospital: Benchmarking Large Language Models in a Multi-agent Medical Interaction Simulator. In *Proceedings of the 31st International Conference on Computational Linguistics*. 10183–10213.

[65] Huihui Fang, Fei Li, Junde Wu, Huazhu Fu, Xu Sun, Jaemin Son, Shuang Yu, Menglu Zhang, Chenglang Yuan, Cheng Bian, et al. 2022. Refuge2 challenge: A treasure trove for multi-dimension analysis and evaluation in glaucoma screening. *arXiv preprint arXiv:2202.08994* (2022).

[66] Arsene Fansi Tchango, Rishab Goel, Zhi Wen, Julien Martel, and Joumana Ghosn. 2022. Ddxplus: A new dataset for automatic medical diagnosis. *Advances in neural information processing systems* 35 (2022), 31306–31318.

[67] Sorouralsadat Fatemi and Yuheng Hu. 2024. FinVision: A multi-agent framework for stock market prediction. In *Proceedings of the 5th ACM International Conference on AI in Finance*. 582–590.

[68] Xiang Fei, Xiawu Zheng, and Hao Feng. 2025. MCP-Zero: Proactive Toolchain Construction for LLM Agents from Scratch. *arXiv preprint arXiv:2506.01056* (2025).

[69] Helen V Firth, Shola M Richards, A Paul Bevan, Stephen Clayton, Manuel Corpas, Diana Rajan, Steven Van Vooren, Yves Moreau, Roger M Pettett, and Nigel P Carter. 2009. DECIPHER: database of chromosomal imbalance and phenotype in humans using ensembl resources. *The American Journal of Human Genetics* 84, 4 (2009), 524–533.

[70] Paul G Francoeur, Tomohide Masuda, Jocelyn Sunseri, Andrew Jia, Richard B Iovanisci, Ian Snyder, and David R Koes. 2020. Three-dimensional convolutional neural networks and a cross-docked data set for structure-based drug design. *Journal of chemical information and modeling* 60, 9 (2020), 4200–4215.

[71] Bowen Gao, Yanwen Huang, Yiqiao Liu, Wenxuan Xie, Wei-Ying Ma, Ya-Qin Zhang, and Yanyan Lan. 2025. Pharmagents: Building a virtual pharma with large language model agents. *arXiv preprint arXiv:2503.22164* (2025).

[72] Chen Gao, Xiaochong Lan, Zhihong Lu, Jinzhu Mao, Jinghua Piao, Huandong Wang, Depeng Jin, and Yong Li. 2023. S3: Social-network Simulation System with Large Language Model-Empowered Agents. *CoRR* (2023).

[73] Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Yiran Wu, et al. 2025. A survey of self-evolving agents: On path to artificial super intelligence. *arXiv preprint arXiv:2507.21046* (2025).

[74] Shen Gao, Yuntao Wen, Minghang Zhu, Jianing Wei, Yuhan Cheng, Qunzi Zhang, and Shuo Shang. 2024. Simulating financial market via large language model based agents. *arXiv preprint arXiv:2406.19966* (2024).

[75] Shanghua Gao, Richard Zhu, Zhenglun Kong, Ayush Noori, Xiao-Rui Su, Curtis Ginder, Theodoros Tsiligkaridis, and Marinka Zitnik. 2025. TxAgent: An AI Agent for Therapeutic Reasoning Across a Universe of Tools. *CoRR* (2025).

[76] Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling Large Language Models to Generate Text with Citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 6465–6488.

[77] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997* 2, 1 (2023).

[78] Alireza Ghafarollahi and Markus Buehler. 2024. ProtAgents: Protein discovery via large language model multi-agent collaborations combining physics and machine learning. In *ICLR Workshop on Large Language Model (LLM) Agents*.

[79] Alireza Ghafarollahi and Markus J Buehler. 2025. Automating alloy design and discovery with physics-aware multimodal multiagent AI. *Proceedings of the National Academy of Sciences* 122, 4 (2025), e2414074122.

[80] Alireza Ghafarollahi and Markus J Buehler. 2025. SciAgents: automating scientific discovery through bioinspired multi-agent intelligent graph reasoning. *Advanced Materials* 37, 22 (2025), 2413523.

[81] Ali Essam Ghareeb, Benjamin Chang, Ludovico Mitchener, Angela Yiu, Caralyn J Szostkiewicz, Jon M Laurent, Muhammed T Razzak, Andrew D White, Michaela M Hinks, and Samuel G Rodriques. 2025. Robin: A multi-agent system for automating scientific discovery. *arXiv preprint arXiv:2505.13400* (2025).

[82] Majid Ghasemi, Amir Hossein Moosavi, and Dariush Ebrahimi. 2024. A comprehensive survey of reinforcement learning: From algorithms to practical challenges. *arXiv preprint arXiv:2411.18892* (2024).

[83] Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, et al. 2025. Towards an AI co-scientist. *arXiv preprint arXiv:2502.18864*

(2025).

[84] Zhiming Gou, Zili Li, Zili Wang, Ming Li, Zhen Wang, and Enhong Chen. 2024. OLVERA: A Framework for Open-ended Code Snippet Verification and Rectification using LLMs. In *International Conference on Learning Representations (ICLR)*.

[85] Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Nan Duan, Weizhu Chen, et al. 2024. CRITIC: Large Language Models Can Self-Correct with Tool-Interactive Critiquing. In *The Twelfth International Conference on Learning Representations*.

[86] Mourad Gridach, Jay Nanavati, Khaldoun Zine El Abidine, Lenon Mendes, and Christina Mack. 2025. Agentic ai for scientific discovery: A survey of progress, challenges, and future directions. *arXiv preprint arXiv:2503.08979* (2025).

[87] Sven Gronauer and Klaus Diepold. 2022. Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review* 55, 2 (2022), 895–943.

[88] Yu Gu, Yiheng Shu, Hao Yu, Xiao Liu, Yuxiao Dong, Jie Tang, Jayanth Srinivasa, Hugo Latapie, and Yu Su. 2024. Middleware for LLMs: Tools Are Instrumental for Language Agents in Complex Environments. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 7646–7663.

[89] T Guo, X Chen, Y Wang, R Chang, S Pei, NV Chawla, O Wiest, and X Zhang. 2024. Large Language Model based Multi-Agents: A Survey of Progress and Challenges.. In *33rd International Joint Conference on Artificial Intelligence (IJCAI 2024)*. IJCAI; Cornell arxiv.

[90] Taicheng Guo, Bozhao Nan, Zhenwen Liang, Zhichun Guo, Nitesh Chawla, Olaf Wiest, Xiangliang Zhang, et al. 2023. What can large language models do in chemistry? a comprehensive benchmark on eight tasks. *Advances in Neural Information Processing Systems* 36 (2023), 59662–59688.

[91] Xuehang Guo, Xingyao Wang, Yangyi Chen, Sha Li, Chi Han, Manling Li, and Heng Ji. 2025. SyncMind: Measuring Agent Out-of-Sync Recovery in Collaborative Software Engineering. In *Forty-second International Conference on Machine Learning*.

[92] Zikang Guo, Benfeng Xu, Xiaorui Wang, and Zhendong Mao. 2025. MIRROR: Multi-agent Intra-and Inter-Reflection for Optimized Reasoning in Tool Learning. *arXiv preprint arXiv:2505.20670* (2025).

[93] Deepak Gupta, Kush Attal, and Dina Demner-Fushman. 2023. A dataset for medical instructional video classification and question answering. *Scientific Data* 10, 1 (2023), 158.

[94] Jiuzhou Han, Wray Buntine, and Ehsan Shareghi. 2024. Towards Uncertainty-Aware Language Agent. In *Findings of the Association for Computational Linguistics ACL 2024*. 6662–6685.

[95] Tessa Han, Aounon Kumar, Chirag Agarwal, and Himabindu Lakkaraju. 2024. Medsafetybench: Evaluating and improving the medical safety of large language models. *Advances in Neural Information Processing Systems* 37 (2024), 33423–33454.

[96] Xuehai He, Yichen Zhang, Luntian Mou, Eric Xing, and Pengtao Xie. 2020. Pathvqa: 30000+ questions for medical visual question answering. *arXiv preprint arXiv:2003.10286* (2020).

[97] Xinyi He, Jiaru Zou, Yun Lin, Mengyu Zhou, Shi Han, Zejian Yuan, and Dongmei Zhang. 2024. CoCoST: Automatic Complex Code Generation with Online Searching and Correctness Testing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 19433–19451.

[98] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*.

[99] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

[100] Daniel Scott Himmelstein, Antoine Lizee, Christine Hessler, Leo Brueggeman, Sabrina L Chen, Dexter Hadley, Ari Green, Pouya Khankhanian, and Sergio E Baranzini. 2017. Systematic integration of biomedical knowledge prioritizes drugs for repurposing. *elife* 6 (2017), e26726.

[101] Sebastian Hofstätter, Jiecao Chen, Karthik Raman, and Hamed Zamani. 2023. Fid-light: Efficient and effective retrieval-augmented text generation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1437–1447.

[102] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. 2024. MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework. In *The Twelfth International Conference on Learning Representations*.

[103] SU Hongjin, Ruoxi Sun, Jinsung Yoon, Pengcheng Yin, Tao Yu, and Sercan O Arik. 2025. Learn-by-interact: A Data-Centric Framework For Self-Adaptive Agents in Realistic Environments. In *The Thirteenth International Conference on Learning Representations*.

[104] Max A Horlbeck, Albert Xu, Min Wang, Neal K Bennett, Chong Y Park, Derek Bogdanoff, Britt Adamson, Eric D Chow, Martin Kampmann, Tim R Peterson, et al. 2018. Mapping the genetic landscape of human cells. *Cell* 174, 4 (2018), 953–967.

[105] Xinyi Hou, Yanjie Zhao, Shenao Wang, and Haoyu Wang. 2025. Model context protocol (mcp): Landscape, security threats, and future research directions. *arXiv preprint arXiv:2503.23278* (2025).

[106] Brian Hu, Bill Ray, Alice Leung, Amy Summerville, David Joy, Christopher Funk, and Arslan Basharat. 2024. Language Models are Alignable Decision-Makers: Dataset and Application to the Medical Triage Domain. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*. 213–227.

[107] Zhiyuan Hu, Chumin Liu, Xidong Feng, Yilun Zhao, See-Kiong Ng, Anh Tuan Luu, Junxian He, Pang Wei Koh, and Bryan Hooi. 2024. Uncertainty of Thoughts: Uncertainty-Aware Planning Enhances Information Seeking in Large Language Models. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

[108] Chenghua Huang, Shisong Chen, Zhixu Li, Jianfeng Qu, Yanghua Xiao, Jiaxin Liu, and Zhigang Chen. 2024. Geoagent: To empower llms using geospatial tools for address standardization. In *Findings of the Association for Computational Linguistics ACL 2024*. 6048–6063.

[109] Daoyi Huang, Jianping Jiang, Tingting Zhao, Shengnan Wu, Pin Li, Yongfen Lyu, Jincai Feng, Mingyue Wei, Zhixing Zhu, Jianlei Gu, et al. 2023. diseaseGPS: auxiliary diagnostic system for genetic disorders based on genotype and phenotype. *Bioinformatics* 39, 9 (2023), btad517.

[110] Dong Huang, Jie M Zhang, Michael Luck, Qingwen Bu, Yuhao Qing, and Heming Cui. 2023. Agentcoder: Multi-agent-based code generation with iterative testing and optimisation. *arXiv preprint arXiv:2312.13010* (2023).

[111] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems* 43, 2 (2025), 1–55.

[112] Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. 2024. Understanding the planning of LLM agents: A survey. *arXiv preprint arXiv:2402.02716* (2024).

[113] Yangyu Huang, Tianyi Gao, Haoran Xu, Qihao Zhao, Yang Song, Zhipeng Gui, Tengchao Lv, Hao Chen, Lei Cui, Scarlett Li, et al. 2025. Peace: Empowering geologic map holistic understanding with mllms. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 3899–3908.

[114] Yoshitaka Inoue, Tianci Song, Xinling Wang, Augustin Luna, and Tianfan Fu. 2025. DrugAgent: Multi-Agent Large Language Model-Based Reasoning for Drug-Target Interaction Prediction. In *ICLR Workshop on Machine Learning for Genomics Explorations*.

[115] John J Irwin, Teague Sterling, Michael M Mysinger, Erin S Bolstad, and Ryan G Coleman. 2012. ZINC: a free tool to discover chemistry for biology. *Journal of chemical information and modeling* 52, 7 (2012), 1757–1768.

[116] Shoichi Ishida, Tomohiro Sato, Teruki Honma, and Kei Terayama. 2025. Large language models open new way of AI-assisted molecule design for chemists. *Journal of Cheminformatics* 17, 1 (2025), 36.

[117] Md Ashraful Islam, Mohammed Eunus Ali, and Md Rizwan Parvez. 2024. MapCoder: Multi-Agent Code Generation for Competitive Problem Solving. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 4912–4944.

[118] Juyong Jiang, Fan Wang, Jiasi Shen, Sungju Kim, and Sunghun Kim. 2024. A survey on large language models for code generation. *arXiv preprint arXiv:2406.00515* (2024).

[119] Xue Jiang, Yihong Dong, Lecheng Wang, Zheng Fang, Qiwei Shang, Ge Li, Zhi Jin, and Wenpin Jiao. 2024. Self-planning code generation with large language models. *ACM Transactions on Software Engineering and Methodology* 33, 7 (2024), 1–30.

[120] Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. 2024. SWE-bench: Can Language Models Resolve Real-world Github Issues?. In *The Twelfth International Conference on Learning Representations*.

[121] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences* 11, 14 (2021), 6421.

[122] Haolin Jin, Linghan Huang, Haipeng Cai, Jun Yan, Bo Li, and Huaming Chen. 2024. From llms to llm-based agents for software engineering: A survey of current, challenges and future. *arXiv preprint arXiv:2408.02479* (2024).

[123] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A Dataset for Biomedical Research Question Answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2567–2577.

[124] Wengong Jin, Connor Coley, Regina Barzilay, and Tommi Jaakkola. 2017. Predicting organic reaction outcomes with weisfeiler-lehman network. *Advances in neural information processing systems* 30 (2017).

[125] Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, et al. 2023. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific data* 10, 1 (2023), 1.

[126] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1601–1611.

[127] René Just, Darioush Jalali, and Michael D Ernst. 2014. Defects4J: A database of existing faults to enable controlled testing studies for Java programs. In *Proceedings of the 2014 international symposium on software testing and analysis*. 437–440.

[128] Yeonghun Kang and Jihan Kim. 2024. ChatMOF: an artificial intelligence system for predicting and generating metal-organic frameworks using large language models. *Nature communications* 15, 1 (2024), 4705.

[129] Zixuan Ke, Fangkai Jiao, Yifei Ming, Xuan-Phi Nguyen, Austin Xu, Do Xuan Long, Minzhi Li, Chengwei Qin, Peifeng Wang, Silvio Savarese, et al. 2025. A survey of frontiers in llm reasoning: Inference scaling, learning to reason, and agentic systems. *arXiv preprint arXiv:2504.09037* (2025).

[130] M Keestra et al. 2017. Metacognition and Reflection by Interdisciplinary Experts: Insights from Cognitive Science and Philosophy. *Issues in Interdisciplinary Studies* 35 (2017).

[131] Jaechang Kim, Jinmin Goh, Inseok Hwang, Jaewoong Cho, and Jungseul Ok. 2025. Bridging the Gap between Expert and Language Models: Concept-guided Chess Commentary Generation and Evaluation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 9497–9516.

[132] Junseok Kim, Nakyeong Yang, and Kyomin Jung. 2024. Persona is a Double-edged Sword: Mitigating the Negative Impact of Role-playing Prompts in Zero-shot Reasoning Tasks. *arXiv preprint arxiv:2408.08631* (2024).

[133] Sehoon Kim, Suhong Moon, Ryan Tabrizi, Nicholas Lee, Michael W Mahoney, Kurt Keutzer, and Amir Gholami. 2024. An LLM compiler for parallel function calling. In *Proceedings of the 41st International Conference on Machine Learning*. 24370–24391.

[134] Yubin Kim, Hyewon Jeong, Chanwoo Park, Eugene Park, Haipeng Zhang, Xin Liu, Hyeonhoon Lee, Daniel McDuff, Marzyeh Ghassemi, Cynthia Breazeal, et al. 2025. Tiered Agentic Oversight: A Hierarchical Multi-Agent System for AI Safety in Healthcare. *arXiv preprint arXiv:2506.12482* (2025).

[135] Yubin Kim, Chanwoo Park, Hyewon Jeong, Yik S Chan, Xuhai Xu, Daniel McDuff, Hyeonhoon Lee, Marzyeh Ghassemi, Cynthia Breazeal, and Hae W Park. 2024. Mdagents: An adaptive collaboration of llms for medical decision-making. *Advances in Neural Information Processing Systems* 37 (2024), 79410–79452.

[136] Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, Xin Zhou, Enzhi Wang, and Xiaohang Dong. 2024. Better Zero-Shot Reasoning with Role-Play Prompting. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 4099–4113.

[137] Yuqing Kong, Yunqi Li, Yubo Zhang, Zhihuan Huang, and Jinzhao Wu. 2022. Eliciting thinking hierarchy without a prior. *Advances in Neural Information Processing Systems* 35 (2022), 13329–13341.

[138] Adarsh Kumarappan, Mo Tiwari, Peiyang Song, Robert Joseph George, Chaowei Xiao, and Anima Anandkumar. 2025. LeanAgent: Lifelong Learning for Formal Theorem Proving. In *The Thirteenth International Conference on Learning Representations*.

[139] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics* 7 (2019), 453–466.

[140] Alexey Lagunin, Dmitrii Filimonov, Alexey Zakharov, Wei Xie, Ying Huang, Fucheng Zhu, Tianxiang Shen, Jianhua Yao, and Vladimir Poroikov. 2009. Computer-aided prediction of rodent carcinogenicity by PASS and CISOC-PSCT. *QSAR & Combinatorial Science* 28, 8 (2009), 806–810.

[141] Yuhang Lai, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, Wen-tau Yih, Daniel Fried, Sida Wang, and Tao Yu. 2023. DS-1000: A natural and reliable benchmark for data science code generation. In *International Conference on Machine Learning*. PMLR, 18319–18345.

[142] Jason J Lau, Soumya Gayen, Asma Ben Abacha, and Dina Demner-Fushman. 2018. A dataset of clinically generated visual questions and answers about radiology images. *Scientific data* 5, 1 (2018), 1–10.

[143] Andrew Laverick, Kristen Surrao, Inigo Zubeldia, Boris Bolliet, Miles Cranmer, Antony Lewis, Blake Sherwin, and Julien Lesgourgues. 2024. Multi-Agent System for Cosmological Parameter Analysis. *arXiv preprint arXiv:2412.00431* (2024).

[144] Chaehong Lee, Varatheepan Paramanayakam, Andreas Karatzas, Yanan Jian, Michael Fore, Heming Liao, Fuxun Yu, Ruopu Li, Iraklis Anagnostopoulos, and Dimitrios Stamoulis. 2025. Multi-Agent Geospatial Copilots for Remote Sensing Workflows. *arXiv preprint arXiv:2501.16254* (2025).

[145] Namkyeong Lee, Edward De Brouwer, Ehsan Hajiramezanali, Tommaso Biancalani, Chanyoung Park, and Gabriele Scalia. 2025. RAG-Enhanced Collaborative LLM Agents for Drug Discovery. In *ICLR Workshop on Machine Learning for Genomics Explorations*.

[146] Sunjae Lee, Junyoung Choi, Jungjae Lee, Munim Hasan Wasi, Hojun Choi, Steve Ko, Sangeun Oh, and Insik Shin. 2024. Mobilegpt: Augmenting llm with human-like app memory for mobile task automation. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*. 1119–1133.

[147] Mosh Levy, Alon Jacoby, and Yoav Goldberg. 2024. Same Task, More Tokens: the Impact of Input Length on the Reasoning Performance of Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 15339–15353.

[148] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems* 33 (2020), 9459–9474.

[149] Binxu Li, Tiankai Yan, Yuanting Pan, Jie Luo, Ruiyang Ji, Jiayuan Ding, Zhe Xu, Shilong Liu, Haoyu Dong, Zihao Lin, et al. 2024. MMedAgent: Learning to Use Medical Tools with Multi-modal Agent. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. 8745–8760.

[150] Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. 2023. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *Advances in Neural Information Processing Systems* 36 (2023), 28541–28564.

[151] Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems* 36 (2023), 51991–52008.

[152] Junkai Li, Yunghwei Lai, Weitao Li, Jingyi Ren, Meng Zhang, Xinhui Kang, Siyu Wang, Peng Li, Ya-Qin Zhang, Weizhi Ma, et al. 2024. Agent hospital: A simulacrum of hospital with evolvable medical agents. *arXiv preprint arXiv:2405.02957* (2024).

[153] Nian Li, Chen Gao, Mingyu Li, Yong Li, and Qingmin Liao. 2024. EconAgent: Large Language Model-Empowered Agents for Simulating Macroeconomic Activities. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 15523–15536.

[154] Xinzhe Li. 2025. A review of prominent paradigms for llm-based agents: Tool use, planning (including rag), and feedback learning. In *Proceedings of the 31st International Conference on Computational Linguistics*. 9760–9779.

[155] Xinyue Li, Zhenpeng Chen, Jie M. Zhang, Yiling Lou, Tianlin Li, Weisong Sun, Yang Liu, and Xuanzhe Liu. 2024. Benchmarking Bias in Large Language Models during Role-Playing. *arXiv preprint arxiv:2411.00585* (2024).

[156] Xiaonan Li and Xipeng Qiu. 2023. Finding Support Examples for In-Context Learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 6219–6235.

[157] Yuan Li, Yixuan Zhang, and Lichao Sun. 2023. Metaagents: Simulating interactions of human behaviors for llm-based task-oriented coordination via collaborative generative agents. *arXiv preprint arXiv:2310.06500* (2023).

[158] Zhenlong Li and Huan Ning. 2023. Autonomous GIS: the next-generation AI-powered GIS. *International Journal of Digital Earth* 16, 2 (2023), 4668–4686.

[159] Zhenlong Li, Huan Ning, Song Gao, Krzysztof Janowicz, Wenwen Li, Samantha T Arundel, Chaowei Yang, Budhendra Bhaduri, Shaowen Wang, A Zhu, et al. 2025. Giscience in the era of artificial intelligence: A research agenda towards autonomous gis. *arXiv preprint arXiv:2503.23633* (2025).

[160] Zhucong Li, Jin Xiao, Bowei Zhang, Zhijian Zhou, Qianyu He, Fenglei Cao, Jiaqing Liang, and Yuan Qi. 2025. ChemHTS: Hierarchical Tool Stacking for Enhancing Chemical Agents. *arXiv preprint arXiv:2502.14327* (2025).

[161] Kaiqu Liang, Zixu Zhang, and Jaime F Fisac. 2024. Introspective Planning: Aligning Robots' Uncertainty with Inherent Task Ambiguity. *Advances in Neural Information Processing Systems* 37 (2024), 71998–72031.

[162] Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging Divergent Thinking in Large Language Models through Multi-Agent Debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 17889–17904.

[163] Xun Liang, Jiawei Yang, Yezhaohui Wang, Chen Tang, Zifan Zheng, Shichao Song, Zehao Lin, Yebin Yang, Simin Niu, Hanyu Wang, et al. 2025. Surveyx: Academic survey automation via large language models. *arXiv preprint arXiv:2502.14776* (2025).

[164] Zhehui Liao, Maria Antoniak, Inyoung Cheong, Evie Yu-Yen Cheng, Ai-Heng Lee, Kyle Lo, Joseph Chee Chang, and Amy X Zhang. 2024. LLMs as Research Tools: A Large Scale Survey of Researchers' Usage and Perceptions. *arXiv preprint arXiv:2411.05025* (2024).

[165] Christopher A Lipinski, Franco Lombardo, Beryl W Dominy, and Paul J Feeney. 1997. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced drug delivery reviews* 23, 1-3 (1997), 3–25.

[166] Bang Liu, Xinfeng Li, Jiayi Zhang, Jinlin Wang, Tanjin He, Sirui Hong, Hongzhang Liu, Shaokun Zhang, Kaitao Song, Kunlun Zhu, et al. 2025. Advances and challenges in foundation agents: From brain-inspired intelligence to evolutionary, collaborative, and safe systems. *arXiv preprint arXiv:2504.01990* (2025).

[167] Bo Liu, Li-Ming Zhan, Li Xu, Lin Ma, Yan Yang, and Xiao-Ming Wu. 2021. Slake: A semantically-labeled knowledge-enhanced dataset for medical visual question answering. In *2021 IEEE 18th international symposium on biomedical imaging (ISBI)*. IEEE, 1650–1654.

[168] Hao Liu, Zi-Yi Dou, Yixin Wang, Nanyun Peng, and Yisong Yue. 2024. Uncertainty Calibration for Tool-Using Language Agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. 16781–16805.

[169] Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. 2023. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in Neural Information Processing Systems* 36 (2023), 21558–21572.

[170] Pengfei Liu, Jun Tao, and Zhixiang Ren. 2025. A quantitative analysis of knowledge-learning preferences in large language models in molecular science. *Nature Machine Intelligence* 7, 2 (2025), 315–327.

[171] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Comput. Surveys* 55, 9 (2023), 1–35.

[172] Shengchao Liu, Jiongxiao Wang, Yijin Yang, Chengpeng Wang, Ling Liu, Hongyu Guo, and Chaowei Xiao. 2024. Conversational drug editing using retrieval and domain feedback. In *The twelfth international conference on learning representations*.

[173] Wei Liu, Jun Li, Yitao Tang, Yining Zhao, Chaozhong Liu, Meiyi Song, Zhenlin Ju, Shwetha V Kumar, Yiling Lu, Rehan Akbani, et al. 2025. DrBioRight 2.0: an LLM-powered bioinformatics chatbot for large-scale cancer functional proteomics analysis. *Nature communications* 16, 1 (2025), 2256.

[174] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. 2024. AgentBench: Evaluating LLMs as Agents. In *ICLR*.

[175] Yang Liu, Weixing Chen, Yongjie Bai, Xiaodan Liang, Guanbin Li, Wen Gao, and Liang Lin. 2025. Aligning cyber space with physical world: A comprehensive survey on embodied ai. *IEEE/ASME Transactions on Mechatronics* (2025).

[176] Yungeng Liu, Zan Chen, Yu Guang Wang, and Yiqing Shen. 2024. Toursynbio-search: A large language model driven agent framework for unified search method for protein engineering. In *2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 5395–5400.

[177] Zhengyao Liu, Yunlong Ma, Jingxuan Xu, Junchen Ai, Xiang Gao, Hailong Sun, and Abhik Roychoudhury. 2025. Agent That Debugs: Dynamic State-Guided Vulnerability Repair. *arXiv preprint arXiv:2504.07634* (2025).

[178] Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292* (2024).

[179] Qinyu Luo, Yining Ye, Shihao Liang, Zhong Zhang, Yujia Qin, Yaxi Lu, Yesai Wu, Xin Cong, Yankai Lin, Yingli Zhang, et al. 2024. RepoAgent: An LLM-Powered Open-Source Framework for Repository-level Code Documentation Generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 436–464.

[180] Yi Luo, Linghang Shi, Yihao Li, Aobo Zhuang, Yeyun Gong, Ling Liu, and Chen Lin. 2025. From intention to implementation: automating biomedical research via LLMs. *Science China Information Sciences* 68, 7 (2025), 1–18.

[181] Bohan Lyu, Xin Cong, Heyang Yu, Pan Yang, Yujia Qin, Yining Ye, Yaxi Lu, Zhong Zhang, Yukun Yan, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2023. GitAgent: Facilitating Autonomous Agent with GitHub by Tool Extension. *arXiv preprint arxiv:2312.17294* (2023).

[182] Weiyu Ma, Qirui Mi, Yongcheng Zeng, Xue Yan, Runji Lin, Yuqiao Wu, Jun Wang, and Haifeng Zhang. 2024. Large language models play starcraft ii: Benchmarks and a chain of summarization approach. *Advances in Neural Information Processing Systems* 37 (2024), 133386–133442.

[183] Aman Madaan, Niket Tandon, Prakhar Gupta, Kevin Hall, Luyu Gao, Rohan Majumder, Julian McAuley, Srijan Narayan, and Sean Welleck. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, Vol. 36.

[184] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems* 36 (2023), 46534–46594.

[185] Agile Manifesto. 2001. Manifesto for Agile Software Development. *http://www. agilemanifesto. org/* (2001).

[186] Tula Masterman, Sandi Besen, Mason Sawtell, and Alex Chao. 2024. The landscape of emerging ai agent architectures for reasoning, planning, and tool calling: A survey. *arXiv preprint arXiv:2404.11584* (2024).

[187] Andrew D McNaughton, Gautham Krishna Sankar Ramalaxmi, Agustin Kruel, Carter R Knutson, Rohith A Varikoti, and Neeraj Kumar. 2024. Cactus: Chemistry agent connecting tool usage to science. *ACS omega* 9, 46 (2024), 46563–46573.

[188] Nikita Mehandru, Amanda K Hall, Olesya Melnichenko, Yulia Dubinina, Daniel Tsirulnikov, David Bamman, Ahmed Alaa, Scott Saponas, and Venkat S Malladi. 2025. BioAgents: Democratizing bioinformatics analysis with multi-agent systems. *arXiv preprint arXiv:2501.06314* (2025).

[189] Shikib Mehri and Maxine Eskenazi. 2020. Unsupervised Evaluation of Interactive Dialog with DialoGPT. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. 225–235.

[190] Lingrui Mei, Jiayu Yao, Yuyao Ge, Yiwei Wang, Baolong Bi, Yujun Cai, Jiazhi Liu, Mingyu Li, Zhong-Zhi Li, Duzhen Zhang, et al. 2025. A Survey of Context Engineering for Large Language Models. *arXiv preprint arXiv:2507.13334* (2025).

[191] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 11048–11064.

[192] Marvin Minsky. 1986. *Society of mind.* Simon and Schuster.

[193] Lluis Morey, Luigi Aloia, Luca Cozzuto, Salvador Aznar Benitah, and Luciano Di Croce. 2013. RYBP and Cbx7 define specific biological functions of polycomb complexes in mouse embryonic stem cells. *Cell reports* 3, 1 (2013), 60–69.

[194] Adam Moss. 2025. The AI Cosmologist I: An Agentic System for Automated Data Analysis. *arXiv preprint arXiv:2504.03424* (2025).

[195] Xinyi Mou, Zhongyu Wei, and Xuan-Jing Huang. 2024. Unveiling the Truth and Facilitating Change: Towards Agent-based Large-scale Social Movement Simulation. In *Findings of the Association for Computational Linguistics ACL 2024*. 4789–4809.

[196] Chunyan Mu, Muhammad Najib, and Nir Oren. 2025. Responsibility-aware Strategic Reasoning in Probabilistic Multi-Agent Systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 23258–23266.

[197] Sriraam Natarajan, Saurabh Mathur, Sahil Sidheekh, Wolfgang Stammer, and Kristian Kersting. 2025. Human-in-the-loop or AI-in-the-loop? Automate or Collaborate?. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 28594–28600.

[198] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2023. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology* (2023).

[199] Subash Neupane, Sudip Mittal, and Shahram Rahimi. 2025. Towards a hipaa compliant agentic ai system in healthcare. *arXiv preprint arXiv:2504.17669* (2025).

[200] Huan Ning, Zhenlong Li, Temitope Akinboyewa, and M Naser Lessani. 2025. An autonomous GIS agent framework for geospatial data retrieval. *International Journal of Digital Earth* 18, 1 (2025), 2458688.

[201] David Ochoa, Andrew Hercules, Miguel Carmona, Daniel Suveges, Jarrod Baker, Cinzia Malangone, Irene Lopez, Alfredo Miranda, Carlos Cruz-Castillo, Luca Fumis, et al. 2023. The next-generation Open Targets Platform: reimagined, redesigned, rebuilt. *Nucleic acids research* 51, D1 (2023), D1353–D1359.

[202] Timothy J O'Donnell, Alex Rubinsteyn, and Uri Laserson. 2020. MHCflurry 2.0: improved pan-allele prediction of MHC class I-presented peptides by incorporating antigen processing. *Cell systems* 11, 1 (2020), 42–48.

[203] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*. PMLR, 248–260.

[204] Aosong Pan, Sameen Al-Azani, Yifei An, Zhipeng Jiang, Wen-Bin Wang, Xipeng Wan, and Man Lan. 2023. LogicLM: Empowering Large Language Models with Tool-Enhanced Logic-Evolving Reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 8500–8518.

[205] Melissa Z Pan, Mert Cemri, Lakshya A Agrawal, Shuyi Yang, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Kannan Ramchandran, Dan Klein, et al. 2025. Why do multiagent systems fail?. In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*.

[206] Himanshu Gautam Pandey, Akhil Amod, and Shivang Kumar. 2024. Advancing Healthcare Automation: Multi-Agent System for Medical Necessity Justification. In *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*. 39–49.

[207] Dmitrii Pantiukhin, Boris Shapkin, Ivan Kuznetsov, Antonia Anna Jost, and Nikolay Koldunov. 2025. Accelerating Earth Science Discovery via Multi-Agent LLM Systems. *arXiv preprint arXiv:2503.05854* (2025).

[208] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*. 1–22.

[209] Anthony D Pellegrini. 2009. *The role of play in human development.* Oxford University Press.

[210] Jean Piaget. 2013. *The construction of reality in the child.* Routledge.

[211] Giorgio Piatti, Zhijing Jin, Max Kleiman-Weiner, Bernhard Schölkopf, Mrinmaya Sachan, and Rada Mihalcea. 2024. Cooperate or collapse: Emergence of sustainable cooperation in a society of llm agents. *Advances in Neural Information Processing Systems* 37 (2024), 111715–111759.

[212] Kevin Pu, KJ Kevin Feng, Tovi Grossman, Tom Hope, Bhavana Dalvi Mishra, Matt Latzke, Jonathan Bragg, Joseph Chee Chang, and Pao Siangliulue. 2025. Ideasynth: Iterative research idea development through evolving and composing

idea facets with literature-grounded feedback. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–31.

[213] Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, et al. 2024. ChatDev: Communicative Agents for Software Development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 15174–15186.

[214] Boyu Qiao, Kun Li, Wei Zhou, Shilong Li, Qianqian Lu, and Songlin Hu. 2025. BotSim: LLM-Powered Malicious Social Botnet Simulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 14377–14385.

[215] Yujia Qin, Shengding Hu, Yankai Lin, Weize Chen, Ning Ding, Ganqu Cui, Zheni Zeng, Xuanhe Zhou, Yufei Huang, Chaojun Xiao, et al. 2024. Tool learning with foundation models. *Comput. Surveys* 57, 4 (2024), 1–40.

[216] Yihao Qin, Shangwen Wang, Yiling Lou, Jinhao Dong, Kaixin Wang, Xiaoling Li, and Xiaoguang Mao. 2024. AgentFL: Scaling LLM-based Fault Localization to Project-Level Context. *CoRR* (2024).

[217] Haoxuan Qu, Xiaofei Hui, Yujun Cai, and Jun Liu. 2023. LMC: large model collaboration with cross-assessment for training-free open-set object recognition. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*. Red Hook, NY, USA, Article 2016, 14 pages.

[218] Yuanhao Qu, Kaixuan Huang, Ming Yin, Kanghong Zhan, Dyllan Liu, Di Yin, Henry C Cousins, William A Johnson, Xiaotong Wang, Mihir Shah, et al. 2025. CRISPR-GPT for agentic automation of gene-editing experiments. *Nature Biomedical Engineering* (2025), 1–14.

[219] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. 2019. Evaluating protein transfer learning with TAPE. *Advances in neural information processing systems* 32 (2019).

[220] Suhas SP Rao, Miriam H Huntley, Neva C Durand, Elena K Stamenova, Ivan D Bochkov, James T Robinson, Adrian L Sanborn, Ido Machol, Arina D Omer, Eric S Lander, et al. 2014. A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell* 159, 7 (2014), 1665–1680.

[221] Ruiyang Ren, Peng Qiu, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2024. BASES: Large-scale Web Search User Simulation with Large Language Model based Agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. 902–917.

[222] Shuo Ren, Pu Jian, Zhenjiang Ren, Chunlin Leng, Can Xie, and Jiajun Zhang. 2025. Towards scientific intelligence: A survey of llm-based scientific agents. *arXiv preprint arXiv:2503.24047* (2025).

[223] Pat Rondon, Renyao Wei, José Cambronero, Jürgen Cito, Aaron Sun, Siddhant Sanyam, Michele Tufano, and Satish Chandra. 2025. Evaluating Agent-based Program Repair at Google. *CoRR* (2025).

[224] Yusuf H Roohani, Andrew H Lee, Qian Huang, Jian Vora, Zachary Steinhart, Kexin Huang, Alexander Marson, Percy Liang, and Jure Leskovec. 2025. BioDiscoveryAgent: An AI Agent for Designing Genetic Perturbation Experiments. In *The Thirteenth International Conference on Learning Representations*.

[225] Giulio Rossetti, Massimo Stella, Rémy Cazabet, Katherine Abramski, Erica Cau, Salvatore Citraro, Andrea Failla, Riccardo Improta, Virginia Morini, and Valentina Pansanella. 2024. Y social: an llm-powered social media digital twin. *arXiv preprint arXiv:2408.00818* (2024).

[226] Yixiang Ruan, Chenyin Lu, Ning Xu, Yuchen He, Yixin Chen, Jian Zhang, Jun Xuan, Jianzhang Pan, Qun Fang, Hanyu Gao, et al. 2024. An automatic end-to-end chemical synthesis development platform powered by large language models. *Nature communications* 15, 1 (2024), 10160.

[227] Stuart J Russell and Peter Norvig. 2016. *Artificial intelligence: a modern approach*. pearson.

[228] Daniel Saeedi, Denise K Buckner, Jose C Aponte, and Amirali Aghazadeh. 2025. AstroAgents: A Multi-Agent AI for Hypothesis Generation from Mass Spectrometry Data. In *Towards Agentic AI for Science: Hypothesis Generation, Comprehension, Quantification, and Validation*.

[229] Liane Salewski, Arian Safavi, and R. Groh. 2024. Can LLMs Learn to Reason from Role-Playing?. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

[230] Carlos G Sanchez, Christopher M Acker, Audrey Gray, Malini Varadarajan, Cheng Song, Nadire R Cochran, Steven Paula, Alicia Lindeman, Shaojian An, Gregory McAllister, et al. 2021. Genome-wide CRISPR screen identifies protein pathways modulating tau protein levels in neurons. *Communications biology* 4, 1 (2021), 736.

[231] Samantha G Scharenberg, Wentao Dong, Ali Ghoochani, Kwamina Nyame, Roni Levin-Konigsberg, Aswini R Krishnan, Eshaan S Rawat, Kaitlyn Spees, Michael C Bassik, and Monther Abu-Remaileh. 2023. An SPNS1-dependent lysosomal lipid transport pathway that enables cell survival under choline limitation. *Science Advances* 9, 16 (2023), eadf8966.

[232] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems* 36 (2023), 68539–68551.

[233] Samuel Schmidgall and Michael Moor. 2025. Agentrxiv: Towards collaborative autonomous research. *arXiv preprint arXiv:2503.18102* (2025).

[234] Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Michael Moor, Zicheng Liu, and Emad Barsoum. 2025. Agent laboratory: Using llm agents as research assistants. *arXiv preprint arXiv:2501.04227* (2025).

[235] Samuel Schmidgall, Rojin Ziaei, Carl Harris, Eduardo Pontes Reis, Jeffrey Jopling, and Michael Moor. 2024. AgentClinic: a multimodal agent benchmark to evaluate AI in simulated clinical environments. *CoRR* (2024).

[236] Ralf Schmidt, Zachary Steinhart, Madeline Layeghi, Jacob W Freimer, Raymund Bueno, Vinh Q Nguyen, Franziska Blaeschke, Chun Jimmie Ye, and Alexander Marson. 2022. CRISPR activation and interference screens decode stimulation responses in primary human T cells. *Science* 375, 6580 (2022), eabj4008.

[237] Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, et al. 2024. The prompt report: a systematic survey of prompt engineering techniques. *arXiv preprint arXiv:2406.06608* (2024).

[238] Minju Seo, Jinheon Baek, Seongyun Lee, and Sung Ju Hwang. 2025. Paper2code: Automating code generation from scientific papers in machine learning. *arXiv preprint arXiv:2504.17192* (2025).

[239] Murray Shanahan, Kyle McDonell, and Laria Reynolds. 2023. Role play with large language models. *Nature* 623, 7987 (2023), 493–498.

[240] Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, et al. 2023. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations*.

[241] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems* 36 (2023), 8634–8652.

[242] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *International Conference on Learning Representations*.

[243] Simranjit Singh, Michael Fore, and Dimitrios Stamoulis. 2024. Geollm-engine: A realistic environment for building geospatial copilots. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 585–594.

[244] Simranjit Singh, Andreas Karatzas, Michael Fore, Iraklis Anagnostopoulos, and Dimitrios Stamoulis. 2024. An llm-tool compiler for fused parallel function calling. *arXiv preprint arXiv:2405.17438* (2024).

[245] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adri Garriga-Alonso, et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on machine learning research* (2023).

[246] Isabella Stewart and Markus J Buehler. 2025. Molecular analysis and design using generative artificial intelligence via multi-agent modeling. *Molecular Systems Design & Engineering* 10, 4 (2025), 314–337.

[247] Buxin Su, Jiayao Zhang, Natalie Collina, Yuling Yan, Didong Li, Kyunghyun Cho, Jianqing Fan, Aaron Roth, and Weijie Su. 2025. The ICML 2023 ranking experiment: Examining author self-assessment in ML/AI peer review. *J. Amer. Statist. Assoc.* just-accepted (2025), 1–16.

[248] Haoyang Su, Renqi Chen, Shixiang Tang, Zhenfei Yin, Xinzhe Zheng, Jinzhe Li, Biqing Qi, Qi Wu, Hui Li, Wanli Ouyang, Philip Torr, Bowen Zhou, and Nanqing Dong. 2025. Many Heads Are Better Than One: Improved Scientific Idea Generation by A LLM-Based Multi-Agent System. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 28201–28240.

[249] Houcheng Su, Weicai Long, and Yanlin Zhang. 2025. BioMaster: Multi-agent System for Automated Bioinformatics Analysis Workflow. *bioRxiv* (2025), 2025–01.

[250] Jiankai Sun, Chuanyang Zheng, Enze Xie, Zhengying Liu, Ruihang Chu, Jianing Qiu, Jiaqi Xu, Mingyu Ding, Hongyang Li, Mengzhe Geng, et al. 2025. A survey of reasoning with foundation models: Concepts, methodologies, and outlook. *Comput. Surveys* 57, 11 (2025), 1–43.

[251] Damian Szklarczyk, Alberto Santos, Christian Von Mering, Lars Juhl Jensen, Peer Bork, and Michael Kuhn. 2016. STITCH 5: augmenting protein–chemical interaction networks with tissue and affinity data. *Nucleic acids research* 44, D1 (2016), D380–D384.

[252] Jiakai Tang, Heyang Gao, Xuchen Pan, Lei Wang, Haoran Tan, Dawei Gao, Yushuo Chen, Xu Chen, Yankai Lin, Yaliang Li, et al. 2025. GenSim: A General Social Simulation Platform with Large Language Model based Agents. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (System Demonstrations)*. 143–150.

[253] Jiabin Tang, Lianghao Xia, Zhonghang Li, and Chao Huang. 2025. AI-Researcher: Autonomous Scientific Innovation. *arXiv preprint arXiv:2505.18705* (2025).

[254] Xiangru Tang, Anni Zou, Zhuosheng Zhang, Ziming Li, Yilun Zhao, Xingyao Zhang, Arman Cohan, and Mark Gerstein. 2024. MedAgents: Large Language Models as Collaborators for Zero-shot Medical Reasoning. In *Findings of the Association for Computational Linguistics ACL 2024*. 599–621.

[255] Yang Tang, Chaoqiang Zhao, Jianrui Wang, Chongzhen Zhang, Qiyu Sun, Wei Xing Zheng, Wenli Du, Feng Qian, and Juergen Kurths. 2022. Perception and navigation in autonomous systems in the era of learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems* 34, 12 (2022), 9604–9624.

[256] Wei Tao, Yucheng Zhou, Yanlin Wang, Wenqiang Zhang, Hongyu Zhang, and Yu Cheng. 2024. Magis: Llm-based multi-agent framework for github issue resolution. *Advances in Neural Information Processing Systems* 37 (2024), 51963–51993.

[257] Raghav Thind, Youran Sun, Ling Liang, and Haizhao Yang. 2025. OptimAI: Optimization from Natural Language Using LLM-Powered AI Agents. *arXiv preprint arXiv:2504.16918* (2025).

[258] Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D Nguyen. 2025. Multi-agent collaboration mechanisms: A survey of llms. *arXiv preprint arXiv:2501.06322* (2025).

[259] Oleg Trott and Arthur J Olson. 2010. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry* 31, 2 (2010), 455–461.

[260] Tao Tu, Mike Schaekermann, Anil Palepu, Khaled Saab, Jan Freyberg, Ryutaro Tanno, Amy Wang, Brenna Li, Mohamed Amin, Yong Cheng, et al. 2025. Towards conversational diagnostic artificial intelligence. *Nature* (2025), 1–9.

[261] Hanbin Wang, Zhenghao Liu, Shuo Wang, Ganqu Cui, Ning Ding, Zhiyuan Liu, and Ge Yu. 2024. INTERVENOR: Prompting the Coding Ability of Large Language Models with the Interactive Chain of Repair. In *Findings of the Association for Computational Linguistics ACL 2024*. 2081–2107.

[262] Han Wang, An Zhang, Nguyen Duy Tai, Jun Sun, Tat-Seng Chua, et al. 2024. Ali-agent: Assessing llms' alignment with human values via agent-based evaluation. *Advances in Neural Information Processing Systems* 37 (2024), 99040–99088.

[263] Junyang Wang, Haiyang Xu, Haitao Jia, Xi Zhang, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. 2024. Mobile-agent-v2: Mobile device operation assistant with effective navigation via multi-agent collaboration. *Advances in Neural Information Processing Systems* 37 (2024), 2686–2710.

[264] Kun Wang, Guibin Zhang, Zhenhong Zhou, Jiahao Wu, Miao Yu, Shiqian Zhao, Chenlong Yin, Jinhu Fu, Yibo Yan, Hanjun Luo, et al. 2025. A comprehensive survey in llm (-agent) full stack safety: Data, training and deployment. *arXiv preprint arXiv:2504.15585* (2025).

[265] Luoqi Wang, Haipeng Li, Linshu Hu, Jiarui Cai, and Zhenhong Du. 2024. Mitigating Interpretation Bias in Rock Records with Large Language Models: Insights from Paleoenvironmental Analysis. *arXiv preprint arXiv:2407.09977* (2024).

[266] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science* 18, 6 (2024), 186345.

[267] Lei Wang, Jingsen Zhang, Hao Yang, Zhi-Yuan Chen, Jiakai Tang, Zeyu Zhang, Xu Chen, Yankai Lin, Hao Sun, Ruihua Song, et al. 2025. User behavior simulation with large language model-based agents. *ACM Transactions on Information Systems* 43, 2 (2025), 1–37.

[268] Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, et al. 2024. Large Language Models are not Fair Evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 9440–9450.

[269] Ruida Wang, Rui Pan, Yuxin Li, Jipeng Zhang, Yizhen Jia, Shizhe Diao, Renjie Pi, Junjie Hu, and Tong Zhang. 2025. MA-LoT: Multi-Agent Lean-based Long Chain-of-Thought Reasoning enhances Formal Theorem Proving. *arXiv e-prints* (2025), arXiv–2503.

[270] Sihan Wang, Suiyang Jiang, Yibo Gao, Boming Wang, Shangqi Gao, and Xiahai Zhuang. 2025. Empowering Medical Multi-Agents with Clinical Consultation Flow for Dynamic Diagnosis. *arXiv preprint arXiv:2503.16547* (2025).

[271] Shuai Wang, Weiwen Liu, Jingxuan Chen, Yuqi Zhou, Weinan Gan, Xingshan Zeng, Yuhan Che, Shuai Yu, Xinlong Hao, Kun Shao, et al. 2024. Gui agents with foundation models: A comprehensive survey. *arXiv preprint arXiv:2411.04890* (2024).

[272] Shuangquan Wang, Huiyong Sun, Hui Liu, Dan Li, Youyong Li, and Tingjun Hou. 2016. ADMET evaluation in drug discovery. 16. Predicting hERG blockers by combining multiple pharmacophores and machine learning approaches. *Molecular pharmaceutics* 13, 8 (2016), 2855–2866.

[273] Wenxuan Wang, Xiaoyuan Liu, Kuiyi Gao, Jen-tse Huang, Youliang Yuan, Pinjia He, Shuai Wang, and Zhaopeng Tu. 2025. Can't See the Forest for the Trees: Benchmarking Multimodal Safety Awareness for Multimodal LLMs. *CoRR* (2025).

[274] Wenxuan Wang, Zizhan Ma, Zheng Wang, Chenghan Wu, Jiaming Ji, Wenting Chen, Xiang Li, and Yixuan Yuan. 2025. A survey of llm-based agents in medicine: How far are we from baymax? *arXiv preprint arXiv:2502.11211* (2025).

[275] Xingyao Wang, Boxuan Li, Yufan Song, Frank F Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, et al. 2025. OpenHands: An Open Platform for AI Software Developers as Generalist Agents. In *The Thirteenth International Conference on Learning Representations*.

[276] Yidong Wang, Qi Guo, Wenjin Yao, Hongbo Zhang, Xin Zhang, Zhen Wu, Meishan Zhang, Xinyu Dai, Qingsong Wen, Wei Ye, et al. 2024. Autosurvey: Large language models can automatically write surveys. *Advances in neural information processing systems* 37 (2024), 115119–115145.

[277] Yuqi Xie Yunfan Jiang Ajay Mandlekar Chaowei Xiao Yuke Zhu Linxi Fan Wang, Guanzhi and Anima Anandkumar. 2023. Voyager: An Open-Ended Embodied Agent with Large Language Models. *Intrinsically-Motivated and Open-Ended Learning Workshop @NeurIPS2023* (2023).

[278] Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. 2024. Unleashing the Emergent Cognitive Synergy in Large Language Models: A Task-Solving Agent through Multi-Persona Self-Collaboration. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 257–279.

[279] Ziyue Wang, Junde Wu, Linghan Cai, Chang Han Low, Xihong Yang, Qiaxuan Li, and Yueming Jin. 2025. MedAgent-Pro: Towards Evidence-Based Multi-Modal Medical Diagnosis via Reasoning Agentic Workflow. *arXiv preprint arXiv:2503.18968* (2025).

[280] Zhenhailong Wang, Haiyang Xu, Junyang Wang, Xi Zhang, Ming Yan, Ji Zhang, Fei Huang, and Heng Ji. 2025. Mobile-agent-e: Self-evolving mobile assistant for complex tasks. *arXiv preprint arXiv:2501.11733* (2025).

[281] Hao Wei, Jianing Qiu, Haibao Yu, and Wu Yuan. 2024. Medco: Medical education copilots based on a multi-agent framework. In *European Conference on Computer Vision*. Springer, 119–135.

[282] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[283] Jinjie Wei, Dingkang Yang, Yanshu Li, Qingyao Xu, Zhaoyu Chen, Mingcheng Li, Yue Jiang, Xiaolu Hou, and Lihua Zhang. 2024. Medaide: Towards an omni medical aide via specialized llm-based multi-agent collaboration. *arXiv preprint arXiv:2410.12532* (2024).

[284] Yixuan Weng, Minjun Zhu, Guangsheng Bao, Hongbo Zhang, Jindong Wang, Yue Zhang, and Linyi Yang. 2025. CycleResearcher: Improving Automated Research via Automated Review. In *The Thirteenth International Conference on Learning Representations*.

[285] David S Wishart, Yannick D Feunang, An C Guo, Elvis J Lo, Ana Marcu, Jason R Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, et al. 2018. DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic acids research* 46, D1 (2018), D1074–D1082.

[286] Michael Wooldridge and Nicholas R Jennings. 1998. Pitfalls of agent-oriented development. In *Proceedings of the second international conference on Autonomous agents*. 385–391.

[287] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. 2024. Autogen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*.

[288] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. 2018. MoleculeNet: a benchmark for molecular machine learning. *Chemical science* 9, 2 (2018), 513–530.

[289] Chunqiu Steven Xia, Yinlin Deng, Soren Dunn, and Lingming Zhang. 2024. Agentless: Demystifying LLM-based Software Engineering Agents. *CoRR* (2024).

[290] Yijia Xiao, Edward Sun, Di Luo, and Wei Wang. 2025. TradingAgents: Multi-Agents LLM Financial Trading Framework. In *The First MARW: Multi-Agent AI in the Real World Workshop at AAAI*.

[291] Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. 2024. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems* 37 (2024), 52040–52094.

[292] Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, et al. 2025. Towards large reasoning models: A survey of reinforced reasoning with large language models. *arXiv preprint arXiv:2501.09686* (2025).

[293] Youjun Xu, Ziwei Dai, Fangjin Chen, Shuaishi Gao, Jianfeng Pei, and Luhua Lai. 2015. Deep learning for drug-induced liver injury. *Journal of chemical information and modeling* 55, 10 (2015), 2085–2093.

[294] Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. 2025. The ai scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *arXiv preprint arXiv:2504.08066* (2025).

[295] An Yan, Zhengyuan Yang, Wanrong Zhu, Kevin Lin, Linjie Li, Jianfeng Wang, Jianwei Yang, Yiwu Zhong, Julian McAuley, Jianfeng Gao, et al. 2023. Gpt-4v in wonderland: Large multimodal models for zero-shot smartphone gui navigation. *arXiv preprint arXiv:2311.07562* (2023).

[296] Xiangchao Yan, Shiyang Feng, Jiakang Yuan, Renqiu Xia, Bin Wang, Bo Zhang, and Lei Bai. 2025. Surveyforge: On the outline heuristics, memory-driven generation, and multi-dimensional evaluation for automated survey writing.

arXiv preprint arXiv:2503.04629 (2025).

[297] Dayu Yang, Antoine Simoulin, Xin Qian, Xiaoyi Liu, Yuwei Cao, Zhaopu Teng, and Grey Yang. 2025. DocAgent: A Multi-Agent System for Automated Code Documentation Generation. *arXiv preprint arXiv:2504.08725* (2025).

[298] Ei-Wen Yang and Enrique Velazquez-Villarreal. 2025. AI-HOPE: An AI-Driven conversational agent for enhanced clinical and genomic data integration in precision medicine research. *Bioinformatics* 41, 7 (2025), btaf359.

[299] Hongyang Yang, Boyu Zhang, Neng Wang, Cheng Guo, Xiaoli Zhang, Likun Lin, Junlin Wang, Tianyu Zhou, Mao Guan, Runjia Zhang, et al. 2024. Finrobot: An open-source ai agent platform for financial applications using large language models. *arXiv preprint arXiv:2405.14767* (2024).

[300] John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. 2024. Swe-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems* 37 (2024), 50528–50652.

[301] Yingxuan Yang, Huacan Chai, Shuai Shao, Yuanyi Song, Siyuan Qi, Renting Rui, and Weinan Zhang. 2025. Agentnet: Decentralized evolutionary coordination for llm-based multi-agent systems. *arXiv preprint arXiv:2504.00587* (2025).

[302] Zonglin Yang, Wanhao Liu, Ben Gao, Tong Xie, Yuqiang Li, Wanli Ouyang, Soujanya Poria, Erik Cambria, and Dongzhan Zhou. 2025. MOOSE-Chem: Large Language Models for Rediscovering Unseen Chemistry Scientific Hypotheses. In *The Thirteenth International Conference on Learning Representations*.

[303] Ziyi Yang, Zaibin Zhang, Zirui Zheng, Yuxian Jiang, Ziyue Gan, Zhiyu Wang, Zijian Ling, Martin Ma, Bowen Dong, Prateek Gupta, et al. 2024. OASIS: Open Agents Social Interaction Simulations on One Million Agents. In *NeurIPS Workshop on Open-World Agents*.

[304] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

[305] Asaf Yehudai, Lilach Eden, Alan Li, Guy Uziel, Yilun Zhao, Roy Bar-Haim, Arman Cohan, and Michal Shmueli-Scheuer. 2025. Survey on evaluation of llm-based agents. *arXiv preprint arXiv:2503.16416* (2025).

[306] Yangyang Yu, Haohang Li, Zhi Chen, Yuechen Jiang, Yang Li, Denghui Zhang, Rong Liu, Jordan W Suchow, and Khaldoun Khashanah. 2024. Finmem: A performance-enhanced llm trading agent with layered memory and character design. In *Proceedings of the AAAI Symposium Series*, Vol. 3. 595–597.

[307] Yangyang Yu, Zhiyuan Yao, Haohang Li, Zhiyang Deng, Yuechen Jiang, Yupeng Cao, Zhi Chen, Jordan Suchow, Zhenyu Cui, Rong Liu, et al. 2024. Fincon: A synthesized llm multi-agent system with conceptual verbal reinforcement for enhanced financial decision making. *Advances in Neural Information Processing Systems* 37 (2024), 137010–137045.

[308] Zhongming Yu, Hejia Zhang, Yujie Zhao, Hanxian Huang, Matrix Yao, Ke Ding, and Jishen Zhao. 2025. OrcaLoca: An LLM Agent Framework for Software Issue Localization. In *Forty-second International Conference on Machine Learning*.

[309] Ling Yue, Sixue Xing, Jintai Chen, and Tianfan Fu. 2024. Clinicalagent: Clinical trial multi-agent system with large language model-based reasoning. In *Proceedings of the 15th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*. 1–10.

[310] Murong Yue. 2025. A survey of large language model agents for question answering. *arXiv preprint arXiv:2503.19213* (2025).

[311] Chaoyun Zhang, Shilin He, Liqun Li, Si Qin, Yu Kang, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. 2025. Api agents vs. gui agents: Divergence and convergence. *arXiv preprint arXiv:2503.11069* (2025).

[312] Chong Zhang, Xinyi Liu, Mingyu Jin, Zhongmou Zhang, Lingyao Li, Zhenting Wang, Wenyue Hua, Dong Shu, Suiyuan Zhu, Xiaobo Jin, et al. 2024. When AI Meets Finance (StockAgent): Large Language Model-based Stock Trading in Simulated Real-world Environments. *CoRR* (2024).

[313] Chi Zhang, Zhao Yang, Jiaxuan Liu, Yanda Li, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. 2025. Appagent: Multimodal agents as smartphone users. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–20.

[314] Kechi Zhang, Jia Li, Ge Li, Xianjie Shi, and Zhi Jin. 2024. CodeAgent: Enhancing Code Generation with Tool-Integrated Agent Systems for Real-World Repo-level Coding Challenges. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 13643–13658.

[315] Ruiqi Zhang, Jing Hou, Florian Walter, Shangding Gu, Jiayi Guan, Florian Röhrbein, Yali Du, Panpan Cai, Guang Chen, and Alois Knoll. 2024. Multi-agent reinforcement learning for autonomous driving: A survey. *arXiv preprint arXiv:2408.09675* (2024).

[316] S Zhang. 2018. Compendium of China's First List of Rare Disease. *People's Medical Publishing House, Beijing, China* (2018), 6–503.

[317] Wentao Zhang, Lingxuan Zhao, Haochong Xia, Shuo Sun, Jiaze Sun, Molei Qin, Xinyi Li, Yuqing Zhao, Yilei Zhao, Xinyu Cai, et al. 2024. A multimodal foundation agent for financial trading: Tool-augmented, diversified, and generalist. In *Proceedings of the 30th acm sigkdd conference on knowledge discovery and data mining*. 4314–4325.

[318] Xinnong Zhang, Jiayu Lin, Xinyi Mou, Shiyue Yang, Xiawei Liu, Libo Sun, Hanjia Lyu, Yihang Yang, Weihong Qi, Yue Chen, et al. 2025. Socioverse: A world model for social simulation powered by llm agents and a pool of 10 million

real-world users. *arXiv preprint arXiv:2504.10157* (2025).

[319] Xinnong Zhang, Jiayu Lin, Libo Sun, Weihong Qi, Yihang Yang, Yue Chen, Hanjia Lyu, Xinyi Mou, Siming Chen, Jiebo Luo, et al. 2024. ElectionSim: Massive Population Election Simulation Powered by Large Language Model Driven Agents. *CoRR* (2024).

[320] Xiaoman Zhang, Chaoyi Wu, Ziheng Zhao, Weixiong Lin, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023. Pmc-vqa: Visual instruction tuning for medical visual question answering. *arXiv preprint arXiv:2305.10415* (2023).

[321] Yifan Zhang, Zhengting He, Jingxuan Li, Jianfeng Lin, Qingfeng Guan, and Wenhao Yu. 2024. MapGPT: an autonomous framework for mapping by integrating large language model and cartographic tools. *Cartography and Geographic Information Science* 51, 6 (2024), 717–743.

[322] Yiran Zhang, Ruiyin Li, Peng Liang, Weisong Sun, and Yang Liu. 2025. Knowledge-Based Multi-Agent Framework for Automated Software Architecture Design. In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*. 530–534.

[323] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2025. Siren's song in the ai ocean: A survey on hallucination in large language models. *Computational Linguistics* (2025), 1–45.

[324] Yadong Zhang, Shaoguang Mao, Tao Ge, Xun Wang, Yan Xia, Wenshan Wu, Ting Song, Man Lan, and Furu Wei. 2024. LLM as a Mastermind: A Survey of Strategic Reasoning with Large Language Models. In *First Conference on Language Modeling*.

[325] Yuntong Zhang, Haifeng Ruan, Zhiyu Fan, and Abhik Roychoudhury. 2024. Autocoderover: Autonomous program improvement. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*. 1592–1604.

[326] Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2024. A survey on the memory mechanism of large language model based agents. *ACM Transactions on Information Systems* (2024).

[327] Zhexin Zhang, Leqi Lei, Lindong Wu, Rui Sun, Yongkang Huang, Chong Long, Xiao Liu, Xuanyu Lei, Jie Tang, and Minlie Huang. 2024. SafetyBench: Evaluating the Safety of Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 15537–15553.

[328] Zhuosheng Zhang, Yao Yao, Aston Zhang, Xiangru Tang, Xinbei Ma, Zhiwei He, Yiming Wang, Mark Gerstein, Rui Wang, Gongshen Liu, et al. 2025. Igniting language intelligence: The hitchhiker's guide from chain-of-thought reasoning to language agents. *Comput. Surveys* 57, 8 (2025), 1–39.

[329] Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 19632–19642.

[330] Debbie Zhao, Edward Ferdian, Gonzalo D Maso Talou, Gina M Quill, Kathleen Gilbert, Vicky Y Wang, Thiranja P Babarenda Gamage, João Pedrosa, Jan D'hooge, Timothy M Sutton, et al. 2023. MITEA: A dataset for machine learning segmentation of the left ventricle in 3D echocardiography using subject-specific labels from cardiac magnetic resonance imaging. *Frontiers in Cardiovascular Medicine* 9 (2023), 1016703.

[331] Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, Jie Jiang, and Bin Cui. 2024. Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473* (2024).

[332] Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2024. GPT-4V (ision) is a generalist web agent, if grounded. In *Proceedings of the 41st International Conference on Machine Learning*. 61349–61385.

[333] Junhao Zheng, Chengming Shi, Xidi Cai, Qiuke Li, Duzhen Zhang, Chenxing Li, Dong Yu, and Qianli Ma. 2025. Lifelong learning of large language model based agents: A roadmap. *arXiv preprint arXiv:2501.07278* (2025).

[334] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems* 36 (2023), 46595–46623.

[335] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2023. Least-to-most prompting enables complex reasoning in large language models. In *International Conference on Learning Representations*.

[336] Juexiao Zhou, Bin Zhang, Guowei Li, Xiuying Chen, Haoyang Li, Xiaopeng Xu, Siyuan Chen, Wenjia He, Chencheng Xu, Liwei Liu, et al. 2024. An AI Agent for Fully Automated Multi-Omic Analyses. *Advanced Science* 11, 44 (2024), 2407094.

[337] Xinyang Zhou, Yongyong Ren, Qianqian Zhao, Daoyi Huang, Xinbo Wang, Tingting Zhao, Zhixing Zhu, Wenyuan He, Shuyuan Li, Yan Xu, et al. 2025. An LLM-Driven Multi-Agent Debate System for Mendelian Diseases. *arXiv preprint arXiv:2504.07881* (2025).

[338] Xuhui Zhou, Zhe Su, Sophie Feng, Jiaxu Zhou, Jen-tse Huang, Hsien-Te Kao, Spencer Lynch, Svitlana Volkova, Tongshuang Wu, Anita Woolley, et al. 2025. SOTOPIA-S4: a user-friendly system for flexible, customizable, and

large-scale social simulation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (System Demonstrations)*. 350–360.

[339] Zhehua Zhou, Jiayang Song, Kunpeng Yao, Zhan Shu, and Lei Ma. 2024. Isr-llm: Iterative self-refined large language model for long-horizon sequential task planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2081–2088.

[340] Dongsheng Zhu, Weixian Shi, Zhengliang Shi, Zhaochun Ren, Shuaiqiang Wang, Lingyong Yan, and Dawei Yin. 2025. Divide-Then-Aggregate: An Efficient Tool Learning Method via Parallel Tool Invocation. *arXiv preprint arxiv:2501.12432* (2025).

[341] Yunheng Zou, Austin H Cheng, Abdulrahman Aldossary, Jiaru Bai, Shi Xuan Leong, Jorge Arturo Campos-Gonzalez-Angulo, Changhyeok Choi, Cher Tian Ser, Gary Tom, Andrew Wang, et al. 2025. El Agente: An autonomous agent for quantum chemistry. *Matter* 8, 7 (2025).

[342] Kaiwen Zuo, Yirui Jiang, Fan Mo, and Pietro Lio. 2025. Kg4diagnosis: A hierarchical multi-agent llm framework with knowledge graph enhancement for medical diagnosis. In *AAAI Bridge Program on AI for Medicine and Healthcare*. PMLR, 195–204.

Received 24 August 2025