Agentic Reasoning for Large Language Models: A Survey

Anonymous authors
Paper under double-blind review

Abstract

Reasoning is a fundamental cognitive process underlying inference, problem-solving, and decision-making. While large language models (LLMs) demonstrate strong reasoning capabilities in closed-world settings, exemplified by standard benchmarks in mathematics and code, they struggle in open-ended and dynamic environments. The emergence of *agentic reasoning* marks a paradigm shift, bridging thought and action by reframing LLMs as autonomous agents that plan, act, and learn through continual interaction. In this survey, we provide a systematic roadmap by organizing agentic reasoning along three complementary dimensions. First, we characterize environmental dynamics through three layers: *foundational agentic reasoning* establishes core single-agent capabilities, including planning, tool use, and search, that operate in stable environments; *self-evolving agentic reasoning* examines how agents refine these capabilities through feedback, memory, and adaptation in evolving settings; and *collective multi-agent reasoning* extends intelligence to collaborative scenarios where multiple agents coordinate roles, share knowledge, and pursue shared goals. Across all layers, we analyze system constraints and optimization settings by distinguishing *in-context reasoning*, which scales test-time interaction through structured orchestration and adaptive workflow design, from *post-training reasoning*, which optimizes behaviors through reinforcement learning and supervised fine-tuning. We further review and contextualize agentic reasoning frameworks in real-world applications and benchmarks spanning science, robotics, healthcare, autonomous research, and math, illustrating how different reasoning mechanisms are instantiated and evaluated across domains. This survey synthesizes agentic reasoning methods into a unified roadmap that bridges thoughts and actions, offering actionable guidance for agentic systems across environmental dynamics, optimization settings, and agent interaction settings. Finally, we outline open challenges and future directions, situating how agentic reasoning has developed while identifying what remains ahead: personalization, long-horizon interaction, world modeling, scalable multi-agent training, and governance frameworks for real-world deployment.

Keywords: Agentic AI, LLM Agent, Agentic Reasoning, Self-evolving

1 Introduction

Reasoning lies at the core of intelligence, enabling logical inference, problem-solving, and decision-making across interactive and dynamic settings. Large language models (LLMs) have achieved remarkable gains in closed-world domains such as mathematical problem solving and code generation. Empirically, techniques that explicitize intermediate reasoning, such as Chain-of-Thought prompting, decomposition, and program-aided solving, have significantly bolstered inference performance (Wei et al., 2022; Zhou et al., 2022; Gao et al., 2023; Yao et al., 2023b). Yet, these approaches often assume static contexts and short-horizon reasoning. Conventional LLMs lack mechanisms to act, adapt, or improve in open-ended environments where information evolves over time.

In this survey, we systematize this evolution under the framework of *Agentic Reasoning*: rather than passively generating sequences, LLMs are reframed as autonomous reasoning agents that plan, act, and learn through continual interaction with their environment. This reframing unifies *reasoning* with *acting*, positioning reasoning as the organizing principle for perception, planning, decision, and verification. Systems such

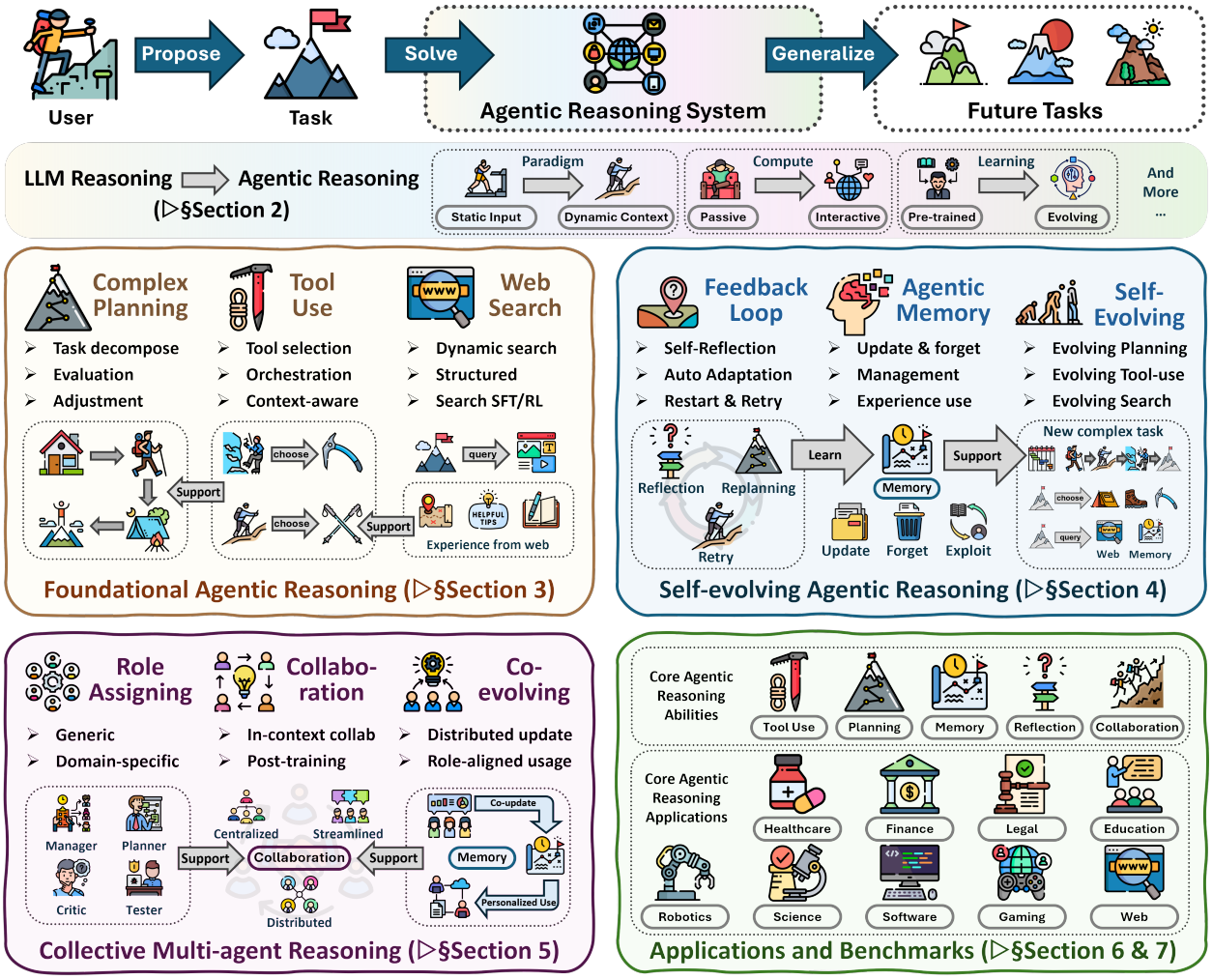

Figure 1: An overview of agentic reasoning.

Definition of Agentic Reasoning

Agentic reasoning positions reasoning as the central mechanism of intelligent agents, spanning *foundational capabilities* (planning, tool use, and search), *self-evolving adaptation* (feedback, and memory-driven adaptation), and *collective coordination* (multi-agent collaboration), realizable through either *in-context* orchestration or *post-training* optimization.

as ReAct (Yao et al., 2023c) interleave deliberation with environment interaction, tool-use frameworks enable self-directed API calling, and workflow-based agents dynamically orchestrate sub-tasks and verifiable actions (Yao et al., 2023c; Schick et al., 2023; Shen et al., 2023). Conceptually, this parallels the shift from static, one-shot inference to sequential decision-making under uncertainty. Unlike simple input-output mapping, this paradigm requires agents to plan over long horizons, navigate partial observability, and actively improve through feedback (Wang et al., 2024f; Singh et al., 2025a; Huang & Huang, 2024).

To systematically characterize the environmental dynamics, we structure our survey around three complementary scopes of agentic reasoning: foundational capabilities, self-evolution, and collective intelligence, spanning diverse interactive and dynamic settings. ***Foundational Agentic Reasoning*** establishes the bedrock of core single-agent capabilities, including planning, tool use, and search, that enable operations within stable, albeit complex, environments. Here, agents act by decomposing goals, invoking external tools, and verifying

Survey Scope

This survey reviews *reasoning-empowered agentic systems* where reasoning drives adaptive behavior. We analyze these systems through two complementary optimization modes:

- **In-context Reasoning**: scales inference-time interaction through structured orchestration and planning without parameter updates.
- **Post-training Reasoning**: internalizes reasoning strategies into model parameters via reinforcement learning and fine-tuning.

Our scope covers methodologies embedding these modes into planning, memory, and self-improvement across single-agent and multi-agent contexts.

results through executable actions. For instance, program-aided reasoning (Gao et al., 2023) grounds logical derivations in code execution; repository-level systems such as OpenHands (Wang et al., 2024i) integrate reasoning, planning, and testing into unified loops; and structured memory modules (Chhikara et al., 2025; Li et al., 2025m) transform factual recall into procedural competence by persisting intermediate reasoning traces for reuse.

Building upon these foundations, ***Self-Evolving Agentic Reasoning*** enables agents to improve continually through cumulative experience. Encompassing task-specific *self-improvement* (e.g., via iterative critique), this paradigm extends adaptation to include persistent updates of internal states like memory and policy. Rather than following fixed reasoning paths, agents develop mechanisms for feedback integration and memory-driven adaptation to navigate evolving environments. Reflection-based frameworks such as Reflexion (Shinn et al., 2023) allow agents to critique and refine their own reasoning processes, while reinforcement formulations such as RL-for-memory (Yan et al., 2025) formalize memory writing and retrieval as policy optimization. Through these mechanisms, agents dynamically integrate inference-time reasoning with learning, progressively updating internal representations and decision policies without full retraining. This continual adaptation links reasoning with learning, enabling models to accumulate competence, and generalize across tasks.

Finally, ***Collective Multi-Agent Reasoning*** scales intelligence from isolated solvers to collaborative ecosystems. Rather than operating in isolation, multiple agents coordinate to achieve shared goals through explicit role assignment (e.g., manager–worker–critic), communication protocols, and shared memory systems (Chen et al., 2023a; Hong et al., 2024). As agents specialize in subtasks and refine each other's outputs, collaboration amplifies reasoning diversity, enabling systems to debate, resolve disagreements, and achieve consistency through natural language-based multi-turn interactions (Wang et al., 2024k;g). However, this complexity also introduces challenges in stability, communication efficiency, and trustworthiness, necessitating structured coordination frameworks and rigorous evaluation standards (Liu et al., 2023c; Zhu et al., 2025b).

Across all layers, we analyze system constraints and optimization settings by distinguishing two complementary modes, corresponding to inference-time orchestration (Yao et al., 2023c; Shinn et al., 2023; Ni et al., 2025b; Li et al., 2025i; Xu et al., 2025h; Wei et al., 2025c) and training-based capability optimization (Ma et al., 2024b; Jin et al., 2025a; Wei et al., 2025e; Yan et al., 2025). ***In-context Reasoning*** focuses on scaling inference-time compute: through structured orchestration, search-based planning, and adaptive workflow design, it enables agents to navigate complex problem spaces dynamically without modifying model parameters. Conversely, ***Post-training Reasoning*** targets capability internalization: it consolidates successful reasoning patterns or tool-use strategies into the model's weights via reinforcement learning and fine-tuning. Together, they provide an actionable roadmap for designing agents.

Building on the three-layer taxonomy, agentic reasoning has begun to underpin a wide range of practical applications, from mathematical exploration (Trinh et al., 2024; Romera-Paredes et al., 2024) and vibe coding (Wang et al., 2024i; Sapkota et al., 2025; Karpathy, 2025) to scientific discovery (Bran et al., 2023; Bousetouane, 2025; Ding et al., 2024), embodied robotics (Wang et al., 2023a; Meghan et al., 2024; Zhao et al., 2025a), healthcare (Li et al., 2024b; Huang et al., 2025b), and autonomous web exploration (Li et al., 2025f; Zheng et al., 2025a). These applications expose distinct reasoning demands shaped by domain-specific

data modalities, interaction constraints, and feedback loops, motivating diverse system designs (Sapkota et al., 2026; Liu et al., 2024h) that integrate planning, tool use, search, reflection, memory mechanisms, and multi-agent coordination. On the other hand, the benchmark landscape has emerged to evaluate agentic reasoning, ranging from targeted tests that isolate individual agentic capabilities to application-specific benchmarks that assess end-to-end behavior in domain-specific environments and scenarios (Zhou et al., 2023b; Koh et al., 2024a; Jang et al., 2024; Shridhar et al., 2020; Liu et al., 2023c; Zhu et al., 2025b; Deng et al., 2023; Gou et al., 2025).

Together, this survey synthesizes agentic reasoning methods into a unified roadmap that bridges reasoning and acting. We systematically characterize these methods across the complementary scopes of foundational, self-evolving, and collective reasoning, while distinguishing between in-context and post-training optimization modes. We further contextualize this roadmap through representative applications and evaluation benchmarks, illustrating how different agentic reasoning mechanisms are instantiated and assessed across realistic domains and task settings. Finally, we outline open challenges and future directions, identifying key frontiers such as personalization, long-horizon interaction, world modeling, scalable multi-agent training, and governance frameworks for real-world deployment.

Contributions

This survey makes the following contributions:

- **Conceptual framing**: We formalize the paradigm of *Agentic Reasoning*, spanning foundational, self-evolving, and collective reasoning layers.
- **Systematic review**: We analyze single-agent, adaptive, and multi-agent systems, emphasizing reasoning-centered workflow orchestration across in-context and post-training dimensions.
- **Applications and evaluation**: We review real-world applications and benchmarks to illustrate the instantiation and evaluation of agentic reasoning mechanisms.
- **Future agenda**: We identify emerging challenges in robustness, trustworthiness, and efficiency, outlining directions for the next generation of adaptive and collaborative agents.

Contents

1 Introduction **1**

2 From LLM Reasoning to Agentic Reasoning **7**

2.1 Positioning Our Survey . . . 7

2.2 Preliminaries . . . 8

3 Foundational Agentic Reasoning **10**

3.1 Planning Reasoning . . . 10

3.1.1 In-context Planning . . . 10

3.1.2 Post-training Planning . . . 13

3.2 Tool-Use Optimization . . . 13

3.2.1 In-Context Tool-integration . . . 14

3.2.2 Post-training Tool-integration . . . 14

3.2.3 Orchestration-based Tool-integration . . . 16

3.3 Agentic Search . . . 16

3.3.1 In-Context Search . . . 18

3.3.2 Post-Training Search . . . 18

4 Self-evolving Agentic Reasoning **19**

4.1 Agentic Feedback Mechanisms . . . 19

4.1.1 Reflective Feedback . . . 19

4.1.2 Parametric Adaptation . . . 20

4.1.3 Validator-Driven Feedback . . . 21

4.2 Agentic Memory . . . 22

4.2.1 Agentic Use of Flat Memory . . . 23

4.2.2 Structured Use of Memory . . . 24

4.2.3 Post-training Memory Control . . . 25

4.3 Evolving Foundational Agentic Capabilities . . . 25

4.3.1 Self-evolving Planning . . . 25

4.3.2 Self-evolving Tool-use . . . 26

4.3.3 Self-evolving Search . . . 26

5 Collective Multi-agent Reasoning **27**

5.1 Role Taxonomy of Multi-Agent Systems (MAS) . . . 28

5.1.1 Generic Roles . . . 28

5.1.2 Domain-Specific Roles . . . 29

5.2 Collaboration and Division of Labor . . . 31

5.2.1 In-context Collaboration . . . 32

5.2.2 Post-training Collaboration . . . 33

5.3 Multi-Agent Evolution . . . 35

5.3.1 From Single-Agent Evolution to Multi-Agent Evolution . . . 35

5.3.2 Multi-agent Memory Management for Evolution . . . 36

5.3.3 Training Multi-agent to Evolve . . . 38

6 Applications **40**

6.1 Math Exploration & Vibe Coding Agents . . . 40

6.2 Scientific Discovery Agents . . . 44
6.3 Embodied Agents . . . 47
6.4 Healthcare & Medicine Agents . . . 50
6.5 Autonomous Web Exploration & Research Agents . . . 52
7 Benchmarks **59**
7.1 Core Mechanisms of Agentic Reasoning . . . 59
7.1.1 Tool Use . . . 59
7.1.2 Search . . . 60
7.1.3 Memory and Planning . . . 61
7.1.4 Multi-Agent System . . . 63
7.2 Applications of Agentic Reasoning . . . 64
7.2.1 Embodied Agents . . . 64
7.2.2 Scientific Discovery Agents . . . 65
7.2.3 Autonomous Research Agents . . . 65
7.2.4 Medical and Clinical Agents . . . 66
7.2.5 Web Agents . . . 66
7.2.6 General Tool-Use Agents . . . 67
8 Open Problems **67**
8.1 User-centric Agentic Reasoning and Personalization . . . 67
8.2 Long-horizon Agentic Reasoning from Extended Interaction . . . 68
8.3 Agentic Reasoning with World Models . . . 68
8.4 Multi-agent Collaborative Reasoning and Training . . . 68
8.5 Latent Agentic Reasoning . . . 68
8.6 Governance of Agentic Reasoning . . . 69

Survey Structure

This survey is organized as follows:

- **Sec. 2:** ***Preliminaries.*** Key background on LLM and Agentic reasoning.
- **Sec. 3:** ***Foundational Agentic Reasoning.*** Core single-agent capabilities including planning, tool use, and search.
- **Sec. 4:** ***Self-evolving Reasoning.*** Feedback, memory, and continual adaptation mechanisms that enhance reasoning over time.
- **Sec. 5:** ***Collective Multi-agent Reasoning.*** Coordination, communication, and shared-memory strategies for collaboration.
- **Sec. 6:** ***Applications.*** Reasoning-empowered applications across science, robotics, healthcare, autonomous research and math/code.
- **Sec. 7:** ***Benchmarks.*** Datasets, metrics, and evaluation protocols for assessing reasoning and agentic abilities.
- **Sec. 8:** ***Open Problems.*** Challenges and future directions for AI Agent reasoning.

2 From LLM Reasoning to Agentic Reasoning

Traditional reasoning with large language models (LLMs) is typically formulated as a one-shot or few-shot prediction task over static inputs. These models rely on scaling **test-time computation**, improving accuracy by increasing model size or inference budget, but without the ability to interact, remember, or adapt to changing goals. Methods such as prompt engineering, in-context learning, and chain-of-thought prompting have made reasoning more explicit, yet conventional LLMs remain passive sequence predictors that operate within fixed prompts.

Agentic reasoning, in contrast, emphasizes **scaling test-time interaction**. Instead of depending solely on internal parameters, agentic systems reason through action: invoking tools, exploring alternatives, updating memory, and integrating feedback. This transforms inference into an iterative process that includes decision steps, reflection, and learning from experience. Reasoning becomes a dynamic loop that connects the model, memory, and environment.

Table 1: Contrasting capabilities of **LLM reasoning** and **agentic reasoning**.

Dimension	LLM Reasoning	↔	Agentic Reasoning
Paradigm	passive	↔	interactive
	static input	↔	dynamic context
Computation	single pass	↔	multi step
	internal compute	↔	with feedback
Statefulness	context window	↔	external memory
	no persistence	↔	state tracking
Learning	offline pretraining	↔	continual improvement
	fixed knowledge	↔	self evolving
Goal Orientation	prompt based	↔	explicit goal
	reactive	↔	planning

This transition marks a conceptual shift: reasoning no longer scales through static capacity, but through structured interaction that enables planning, adaptation, and collaboration across time and tasks.

2.1 Positioning Our Survey

While several recent surveys have examined LLM reasoning or agent architectures (Huang & Chang, 2022; Chen et al., 2025d; Xu et al., 2025a; Ke et al., 2025a; Zhang et al., 2025l;f; Lin et al., 2025c; Fang et al., 2025a; Gao et al., 2025b), our work focuses specifically on **agentic reasoning** as a unified paradigm for understanding reasoning as interaction. We position this survey at the intersection of model-centric reasoning and system-level intelligence, aiming to bridge prior discussions on reasoning mechanisms and agent architectures.

Relation to LLM Reasoning Surveys. Existing surveys on LLM reasoning mainly investigate how to elicit or enhance reasoning within a model's internal computation process. For example, Huang & Chang (2022); Chen et al. (2025d); Xu et al. (2025a); Ke et al. (2025a) summarize prompting and scaling techniques such as chain-of-thought, reinforcement post-training, and long-context reasoning, emphasizing how LLMs can learn to reason better through inference-time supervision or post-training alignment. These works improve the internal expressiveness of reasoning traces but typically remain within static inference settings, where reasoning unfolds in a single forward pass without external interaction. In contrast, our survey examines how reasoning extends *beyond* text generation, encompassing dynamic planning, adaptive memory, and feedback-driven behavior during deployment.

Relation to AI Agent Surveys. Several contemporary surveys have begun to explore LLM-based agents from architectural or system perspectives (Zhang et al., 2025f; Lin et al., 2025c; Fang et al., 2025a; Gao et al., 2025b). These works analyze how agents employ reinforcement learning, planning, and tool-use modules to operate in complex environments. For instance, Zhang et al. (2025f); Lin et al. (2025c) focus on reinforcement learning for agentic search and decision-making, while Fang et al. (2025a); Gao et al. (2025b) emphasize self-evolving and lifelong agentic systems that continuously learn from interaction. Our focus complements these perspectives by centering on the **reasoning process** that these architectures enable, specifically how interaction, feedback, and collaboration transform static inference into adaptive reasoning. Rather than viewing reasoning as an implicit by-product of architectural design, we treat it as the unifying mechanism that links single-agent reinforcement, multi-agent coordination, and self-evolving intelligence.

In summary, our survey provides a reasoning-centric lens on intelligent agency. We examine how foundational reasoning mechanisms, post-training adaptation, and long-term self-evolution jointly constitute the basis of *agentic reasoning*, illustrating the transition from static prediction to interactive, adaptive, and continually improving intelligence.

2.2 Preliminaries

This subsection formalizes the transition from static language modeling to agentic reasoning. To align with the **three-layered dimensions** (Foundational, Self-Evolving, Collaboration) outlined in the introduction, we unify these capabilities under a single control-theoretic framework.

Formalizing Agentic Reasoning: A Latent-Space View. Standard approaches often conflate the agent's context with the environment state. We model the environment as a **Partially Observable Markov Decision Process (POMDP)** and introduce an internal *reasoning variable* to expose the "think–act" structure of agentic policies. Concretely, we consider the tuple $\langle \mathcal{X}, \mathcal{O}, \mathcal{A}, \mathcal{Z}, \mathcal{M}, \mathcal{T}, \Omega, \mathcal{R}, \gamma \rangle$, where $\mathcal{X}$ is the latent *environment state* space (unobservable to the agent), $\mathcal{O}$ is the observation space (e.g., user queries, API returns), $\mathcal{A}$ is the external action space (e.g., tool invocation, final answer), $\mathcal{Z}$ is a *reasoning trace* space (e.g., latent plans, optionally verbalized as chain-of-thought), and $\mathcal{M}$ is the agent's *internal memory/context* space (e.g., a sufficient statistic of interaction history). $\mathcal{T}$ and Ω denote the transition and observation kernels, $\mathcal{R}$ the reward, and $\gamma \in (0, 1)$ the discount factor.

At timestep t, the agent conditions on a history $h_t = (o_{\leq t}, z_{<t}, a_{<t})$ (i.e., o_t is observed before generating z_t and then a_t). Equivalently, the history can be summarized by an internal memory state $m_t \in \mathcal{M}$. Crucially, we distinguish external actions from internal reasoning. We factorize the policy as

$$\pi_\theta(z_t, a_t \mid h_t) = \underbrace{\pi_{\text{reason}}(z_t \mid h_t)}_{\text{Internal Thought}} \cdot \underbrace{\pi_{\text{exec}}(a_t \mid h_t, z_t)}_{\text{External Action}}. \tag{1}$$

This decomposition highlights the core shift in agentic systems: performing computation in $\mathcal{Z}$ (thinking) before committing to $\mathcal{A}$ (acting). The objective remains maximizing the expected return $J(\theta) = \mathbb{E}_\tau \left[\sum_{t \geq 0} \gamma^t r_t \right]$.

In-Context Reasoning: Inference-Time Search. In this regime, model parameters θ are frozen. The agent optimizes the reasoning trajectory by searching over $\mathcal{Z}$ to maximize a heuristic value function $\hat{v}(h_t, z)$. We model inference as selecting a trajectory $\tau = (h_0, z_0, a_0, h_1, z_1, a_1, \ldots)$. Methods like ReAct (Yao et al., 2023c) perform greedy decoding over alternating thoughts z and actions a. Tree-of-Thoughts (ToT (Yao et al., 2023b)) and related MCTS-style approaches treat partial thoughts as nodes $u \in \mathcal{U}$ (e.g., a representation derived from (h_t, z_t)) and search for an optimal path:

$$\tau^\star \in \arg\max_\tau \sum_t \hat{v}_\phi(u_t), \tag{2}$$

where $\hat{v}_\phi$ is a heuristic evaluator or verifier. This corresponds to planning in $\mathcal{Z}$ without updating the policy parameters.

Post-Training: Policy Optimization. This paradigm optimizes θ to align the policy with long-horizon rewards r_t (e.g., correctness, safety), including reasoning models (e.g., DeepSeek-R1 (Guo et al., 2025a)) and learning-to-search systems (e.g., Search-R1 (Jin et al., 2025a), DeepRetrieval (Jiang et al., 2025a)) that train multi-turn reasoning or tool use with RL. While PPO (Schulman et al., 2017) is standard, **Group Relative Policy Optimization (GRPO)** (Shao et al., 2024)-based methods are widely used for reasoning tasks. GRPO eliminates the value network by constructing advantages from group-relative rewards. For a group of G sampled outputs $\{y_i\}_{i=1}^{G}$ from the same prompt q, a common GRPO objective is:

$$\mathcal{L}^{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim P(Q)}\left[\frac{1}{G}\sum_{i=1}^{G}\left(\min\Big(\rho_i\hat{A}_i,\ \text{clip}(\rho_i, 1-\epsilon, 1+\epsilon)\hat{A}_i\Big) - \beta\,\mathbb{D}_{KL}(\pi_\theta \,\|\, \pi_{\text{ref}})\right)\right], \tag{3}$$

where $\rho_i = \frac{\pi_\theta(y_i|q)}{\pi_{\theta_{\text{old}}}(y_i|q)}$ and the group-normalized advantage is

$$\hat{A}_i = \frac{r_i - \mu}{\sigma + \delta}, \quad \mu = \frac{1}{G}\sum_{j=1}^{G} r_j, \quad \sigma = \sqrt{\frac{1}{G}\sum_{j=1}^{G}(r_j - \mu)^2}, \tag{4}$$

with $\delta > 0$ a small constant for numerical stability. Advanced methods such as ARPO (Lu et al., 2025b) and DAPO (Yu et al., 2025c) extend this framework to handle sparse rewards and improve stability in complex tool-use environments (e.g., via replay/rollout strategies and decoupled clipping).

Collective Intelligence: Multi-Agent Reasoning. We extend the single-agent formulation to a *decentralized* partially observable multi-agent setting, commonly formalized as a *Dec-POMDP*. The core distinction lies in expanding each agent's observation to include a **communication channel** $\mathcal{C}$. For a system of N agents, the joint policy $\boldsymbol{\pi}$ is composed of individual policies π^i, where agent i's observation o_t^i explicitly includes communicative messages c_{t-1}^{-i} generated by peers. Crucially, in agentic MARL, communication is not merely signal transmission but an extension of the reasoning process: one agent's external action can act as a prompt that triggers another agent's internal reasoning chain. Existing frameworks like AutoGen (Wu et al., 2024b) and CAMEL (Li et al., 2023a) represent static role-playing with fixed policies. Recent agentic RL advances (e.g., GPTSwarm (Zhuge et al., 2024a), MaAS, agents trained via PPO/GRPO (Hong et al., 2025)) aim to *optimize* this joint reasoning distribution. The challenge shifts from single-agent planning to **mechanism design**: optimizing the communication topology and incentive structures to align decentralized reasoning processes π^i_{reason} toward a coherent global objective, often utilizing Centralized-Training/Decentralized-Execution (CTDE) paradigms to stabilize the emergence of cooperative behaviors.

Self-Evolving Agents: The Meta-Learning Loop. While foundational agents optimize reasoning z within an episode, self-evolving agents optimize the agent system itself across episodes $k = 1, \ldots, K$. Let $\mathcal{S}_k$ denote the evolvable system state (e.g., explicit memories, tool libraries, or code). A generic meta-update rule is

$$\mathcal{S}_{k+1} \leftarrow U(\mathcal{S}_k, \tau_k, \mathcal{F}_k), \tag{5}$$

where $\mathcal{F}_k$ represents environmental feedback (rewards, execution errors) and $\mathcal{S}_k$ represents the evolvable state. We categorize self-evolution by the nature of $\mathcal{S}$:

- **Verbal Evolution:** $\mathcal{S}$ consists of textual reflections or guidelines. Methods like Reflexion (Shinn et al., 2023) update $\mathcal{S}$ by synthesizing error logs into linguistic cues that condition future reasoning policies.
- **Procedural Evolution:** $\mathcal{S}$ consists of a library of executable tools or skills. Agents like Voyager (Wang et al., 2023a) evolve by synthesizing new code-based skills, expanding the action space $\mathcal{A}$ permanently.
- **Structural Evolution:** $\mathcal{S}$ consists of the agent's source code or architecture itself. Advanced methods like AlphaEvolve (Novikov et al., 2025) treat the agent's code as a hypothesis space, using an LLM as a mutation operator to search for superior reasoning algorithms.

This framework unifies these diverse approaches as gradient-free or gradient-based optimization steps over the agent's explicit memories and artifacts (and optionally parameters), closing the loop between experience and competence.

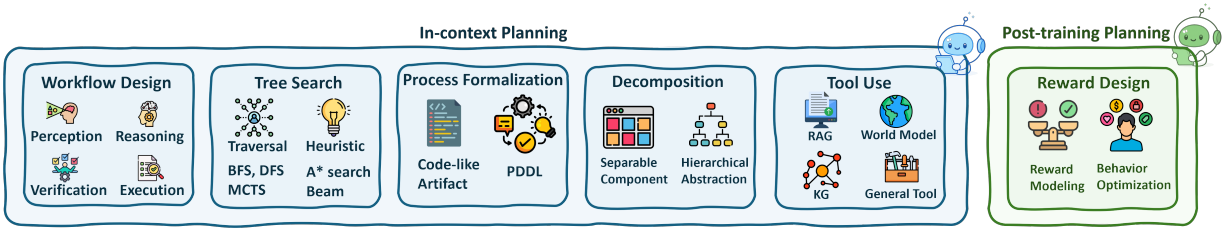

Figure 2: Overview of **Planning Reasoning** in LLM agents, categorized into in-context planning and post-training planning.

3 Foundational Agentic Reasoning

Agentic reasoning originates from the behavior of a single agent. Before discussing adaptation and collaboration, we focus on how an individual agent translates reasoning into structured action through three core components: *planning*, *search*, and *tool use*. In this setting, the agent is not a passive text generator but an autonomous problem solver that formulates plans, explores alternatives through retrieval or environment search, and leverages tools to execute grounded operations. Together, these mechanisms establish the foundation of agentic reasoning, linking abstract deliberation with verifiable action.

A canonical foundational workflow can be viewed as an iterative cycle that interleaves **planning** (goal decomposition and task formulation), **tool use** (invoking external systems or APIs to act on the world) and **search** (retrieval and exploration for decision support), Reasoning serves as the organizing principle across these stages, determining when to plan, what to retrieve, and how to act, transforming static inference into interactive decision-making.

By analyzing these components, we clarify how structured reasoning elevates a static LLM into an autonomous, goal-driven agent. The next section introduces **self-evolving reasoning**, where *feedback* and *memory* enable continual adaptation and extension of these foundational capabilities. Subsequently, we examine **collective reasoning**, in which multiple agents coordinate through roles, communication, and shared memory to achieve objectives beyond individuals.

3.1 Planning Reasoning

Planning is a central component of intelligent behavior, enabling agents to decompose problems, sequence decisions, and navigate complex environments with foresight. Recent research has increasingly explored planning in the context of large language models (LLMs), either as autonomous agents or as components in broader systems. In this section, we categorize existing work in agent planning for reasoning into six methodological styles, where each category highlights a distinct planning strategy that supports complex agentic reasoning.

3.1.1 In-context Planning

Workflow Design. Workflow-based approaches often emphasize structuring the overall planning process into distinct stages (e.g., perception, reasoning, execution, verification), which are either explicitly scaffolded or learned implicitly. For example, Liu et al. (2023a); Valmeekam et al. (2023b); Xu et al. (2023a); Hao et al. (2024a) design planning pipelines that decompose task solving into subtasks, often leveraging a deliberate plan-and-act framework. Similarly, Zhou et al. (2022); Wang et al. (2023b); Sel et al. (2023); Shen et al. (2023) rely on structured prompting to sequentialize tasks and guide reasoning progression. Methods like Liu et al. (2023b) use structured transitions between diverse "X-of-Thought" strategies. PERIA (Ni et al., 2024) combines perception, imagination, and action in a unified multimodal workflow. Others such as Erdogan et al. (2025) explicitly target long-horizon planning through structured sequencing, while Wen et al. (2025b) build workflows for code-related planning.

Table 2: Representative **Agentic Planning** systems categorized by *Modality*, *Structure*, *Format*, and *Tool*.

Method	Structure	Format	Tool
Modality I: Language Agents (e.g., Search Agents, Code Agents)			
ReWOO (Xu et al., 2023a)	Decomposed	Natural Language	None
Reflexion (Shinn et al., 2023)	Sequential	Natural Language	None
LLM+P (Liu et al., 2023a)	Sequential	Formal Language	None
IPC (Valmeekam et al., 2023b)	Sequential	Formal Language	None
ToT (Yao et al., 2023b)	Tree	Natural Language	None
GoT (Besta et al., 2024)	Graph	Natural Language	None
AoT (Sel et al., 2023)	Graph	Natural Language	None
HTP (Gui et al., 2025)	Hypertree	Natural Language	Retrieval
RefPlan (Jeong et al., 2025)	Tree	Constrained Space	None
Gorilla (Patil et al., 2024)	Sequential	Programming Language	Retrieval, API
CodeNav (Gupta et al., 2024)	Sequential	Programming Language	Code Indexer, Code Search
PoG (Chen et al., 2024b)	Graph	Natural Language	Knowledge Graph
Tool-Planner (Liu et al., 2025g)	Sequential	Natural Language	Tool Cluster
Modality II: Visual/Multimodal Agents (e.g., GUI Agents, Embodied Agents)			
VisualPredictor (Liang et al., 2025b)	Tree	Formal Language	None
LLM-Planner (Song et al., 2023a)	Sequential	Formal Language	Object Detector, KNN
Agent-E (Abuelsaad et al., 2024)	Sequential	Formal Language	DOM Grounder, Screenshot
Agent S (Agashe et al., 2024)	Hierarchical	Natural Language	API, Search, Memory
ExRAP (Yoo et al., 2024)	Sequential	Natural Language	Memory
AESOP (Sinha et al., 2024)	Reactive	Natural Language	Anomaly Detector
HRV (Cornelio et al., 2025)	Hierarchical	Formal Language	Symbolic Verifier
BehaviorGPT (Zhou et al., 2024c)	Sequential	Visual Features	World Model
Dino-WM (Zhou et al., 2024a)	Tree	Visual Features	World Model
FLIP (Gao et al., 2024)	Sequential	Visual Features	Language Model

These workflows are then grounded by a reactive controller that iteratively consumes the current state and interleaves reasoning with actions: in web automation, agents follow inspect-reason-act-observe loops (Yao et al., 2023c; Deng et al., 2023), with robustness improved by dynamically adapting in-context examples (Lutz et al., 2024); in code, agents decide immediate executions/API calls, read outputs or errors, and refine step-by-step (Wang et al., 2024h; Patil et al., 2024; Shinn et al., 2023; Gupta et al., 2024; Rahman et al., 2025; He et al., 2024a; Rawat et al., 2025; Aksitov et al., 2023; Jiang et al., 2024c); in robotics, language models can also serve as plan teachers that generate, eliminate, and track candidate plans before execution, while monitors trigger on-the-fly safety interventions and VLM-guided subgoal execution with real-time adjustment (Sinha et al., 2024; Shah et al., 2023; Wu et al., 2023; Hu et al., 2024a). This *reactive workflow* view unifies scripted stage design with online adaptation: the workflow provides interpretable structure and interfaces (what is done when), while the reactive loop supplies closed-loop grounding and error recovery (how it is done in context). The approach is broadly effective yet can accumulate errors over long horizons, motivating incremental verification and memory within the workflow to stabilize execution.

Tree Search / Algorithm Simulation. Tree-based search strategies, especially BFS, DFS, A*, MCTS, and beam search, have become prominent as interpretable and effective planning scaffolds. Several works simulate tree traversal algorithms to mimic deliberative processes: Yao et al. (2023b); Markowitz et al. (2024); Long (2023); Koh et al. (2024b) apply breadth- or depth-first strategies to explore structured thought trees. A*-like guided expansions appear in (Wang et al., 2024a; Meng et al., 2024a; Liu et al., 2024a; Zhang et al., 2023c), providing heuristic-driven planning with state evaluation. Besides that, MCTS is heavily explored in agentic research: Hao et al. (2023a); Putta et al. (2024); Sprueill et al. (2023); Yu et al. (2023b);

Zhao et al. (2023b); Ding et al. (2023); Chen et al. (2024e); Kong et al. (2024); Feng et al. (2023); Yoon et al. (2025); Schultz et al. (2024); Chen et al. (2025i); Xiong et al. (2026b) use MCTS or its variations for controlled exploration and improved reasoning fidelity. Beam search is leveraged in (Xie et al., 2023; Golovneva et al., 2023; Qian et al., 2025d) to prune and prioritize reasoning trajectories efficiently. Other tree-search-inspired works include Gandhi et al. (2024) which uses learned search policies and Saha et al. (2025) which differentiates between fast (reactive) and slow (deliberative) planning. These methods mirror traditional algorithmic planning, grounding LLMs' search processes in classical decision-making frameworks.

This search-over-hierarchy view maps cleanly onto domain systems. In the web setting, planner-executor architectures generate high-level subtask trees in natural language and bind leaves to DOM-grounded actions, often with memory to persist context (Abuelsaad et al., 2024; Guan et al., 2023b; Agashe et al., 2024). For code agents, hierarchical task trees and pseudo-code plans recursively break problems into compilable/editable units, with Tree-of-Code (Ni et al., 2025b) and related structured pipelines embedding hierarchical RL or MCTS within the tree to choose promising edits and verification paths (Gui et al., 2025; Ni et al., 2025b; Chen et al., 2025b; Hu et al., 2025f; Antoniades et al., 2024). In robotics, behavior trees and high-level goal decomposition translate language instructions into subgoal sequences executed by low-level controllers and skills (Lykov & Tsetserukou, 2024; Cao & Lee, 2023; Izzo et al., 2024; Ahn et al., 2022; Huang et al., 2022).

Taken together, hierarchical tree-search couples *plan synthesis* (node expansion, heuristic/evidence-based selection) with *plan realization* (leaf grounding and feedback), yielding interpretable, long-horizon agents that can backtrack, refine, and verify before committing to irreversible actions, while remaining flexible enough to incorporate learned policies and memory for efficiency and robustness.

Process Formalization. Formalizing planning through symbolic representations, programming languages, or logic frameworks ensures compositionality, interpretability, and generalization. Several works encode plans as code-like artifacts or PDDL programs: Guan et al. (2023a); Mahdavi et al. (2024); Katz et al. (2024); Wen et al. (2025b); Hao et al. (2024c); Vyas et al. (2024) incorporate symbolic logic or procedural programming into LLM prompting or output generation. These representations enable downstream tool execution and interface more cleanly with classical planners or robot controllers. PDDL-based formulations explicitly bridge LLM planning with well-established planning ecosystems, as in (Mahdavi et al., 2024; Katz et al., 2024). CodePlan (Wen et al., 2025b) highlights the use of program synthesis to scaffold long-horizon reasoning. Such formalization provides structural scaffolds for agent behavior and often enhances explainability and robustness of the generated plans.

Decoupling / Decomposition. Decoupling strategies aim to modularize complex planning into separable components such as goal recognition, memory retrieval, and plan refinement. Notably, ReWOO (Xu et al., 2023a) explicitly separates observation and reasoning modules to optimize for efficiency. Similarly, works like Zhang et al. (2025u); Dong et al. (2024b); Lo et al. (2024); Li et al. (2024a); Wang et al. (2023c); Vyas et al. (2024); Kang et al. (2025a) break reasoning into reusable or hierarchical abstractions. Gui et al. (2025) promotes hierarchical thinking through hypertrees, while Liang et al. (2025b) abstracts the world with symbolic predicates to reduce planning burden. Others, such as Ye et al. (2024) and Kong et al. (2024), decompose via latent variables or state spaces. These decompositions not only enhance tractability, but also align with neural-symbolic hybrid frameworks. They are especially common in long-horizon or multi-agent planning scenarios, such as (Zheng et al., 2024c; Nayak et al., 2024).

External Aid / Tool Use. Many systems leverage external structures or tools to aid planning, including retrieval-augmented generation (RAG), knowledge graphs, world models, and general-purpose tool use. Knowledge-augmented frameworks like Chen et al. (2024b); Cornelio et al. (2025); Meng & Fan (2025); Zhong et al. (2024); Zhang et al. (2025u) inject structured representations (e.g., graphs, scene layouts) into the LLM context. RAG-style systems (Yoo et al., 2024; Li et al., 2024h; Zou et al., 2025b) retrieve relevant knowledge to support continual instruction planning. World model-based agents such as Hao et al. (2023a); Guan et al. (2023a); Qiao et al. (2024); Zhou et al. (2024c;a); Gao et al. (2024); Liu et al. (2025j); Wang et al. (2025c) learn or leverage environment models for model-based planning. Tool-oriented frameworks like HuggingGPT (Shen et al., 2023), Tool-Planner (Liu et al., 2025g), and RetroInText (Kang et al., 2025a) use external

APIs or modular toolchains to support planning execution. These systems often reflect agent-environment interaction and capitalize on external resources to scaffold or augment LLM capabilities.

3.1.2 Post-training Planning

Reward Design / Optimal Control. Finally, planning as optimization entails designing suitable reward structures and solving for optimal behavior using RL or control-theoretic tools. Reflexion (Shinn et al., 2023), Reflect-then-Plan (Jeong et al., 2025), and Rational Decision Agents (Ye et al., 2023) incorporate utility-based learning to guide planning behavior. Reward modeling appears in works such as Chen et al. (2025h), while others like Luyten et al. (2025) emphasize reward shaping. Planner-level supervision has also emerged for long-horizon agents, where a global planner is trained to provide efficient high-level decompositions before local execution (Si et al., 2026). Optimal control is tackled explicitly in (Ma et al., 2024a; Ni et al., 2025a; Matada et al., 2024; Su et al., 2025a), and trajectory optimization via diffusion models is seen in (Xie et al., 2025a; Xiao et al., 2023; Shan et al., 2025). Offline RL methods like Kong et al. (2024); Ruoss et al. (2024); Wang et al. (2023c) leverage pretrained dynamics or cost models. The control-theoretic orientation in these works complements symbolic or heuristic approaches by optimizing over continuous, structured, or learned reward spaces.

3.2 Tool-Use Optimization

Table 3: Representative **Tool-Use Optimization** systems categorized by *Integration Stage*, *Learning Type*, and *Tool Strategy*.

Method	Stage	Learning	Tool Strategy
Modality I: In-Context Integration			
ReAct (Yao et al., 2023c)	Inference	Prompting	Interleaved reasoning–action
ART (Paranjape et al., 2023)	Inference	Few-shot	Retrieved multi-step demos
ChatCoT (Chen et al., 2023e)	Inference	Prompting	CoT with tool calls
GEAR (Lu et al., 2024b)	Inference	Delegation	Light model for tool selection
AVATAR (Wu et al., 2024d)	Inference	Contrastive	In-context tool reasoning
Modality II: Post-Training Integration			
Toolformer (Schick et al., 2023)	Post-train	Self-sup. + SFT	Self-generated API calls
ToolLLM (Qin et al., 2024b)	Post-train	SFT	Large-scale API demos
ToolAlpaca (Tang et al., 2023a)	Post-train	SFT	Simulated dialogues
ReSearch (Chen et al., 2026c)	Post-train	RL + Reflec.	Adaptive retrieval reasoning
ReTool (Dong et al., 2025c)	Post-train	RL	Reinforced code execution
ToolRL (Qian et al., 2025a)	Post-train	RL	Multi-tool policy learning
Modality III: Orchestration-based Integration			
HuggingGPT (Shen et al., 2023)	System	Planner–Exec.	Multi-tool coordination
TaskMatrix.AI (Liang et al., 2023)	System	Planner	Massive API ecosystem
ToolPlanner (Liu et al., 2025g)	System	RL	Plan-before-act framework
OctoTools (Lu et al., 2025c)	System	Rule-based	Hierarchical orchestration
ToolExpNet (Zhang et al., 2025y)	System	Embedding	Experience-based selection
ToolChain* (Zhuang et al., 2024)	System	Search	A* decision over tools

Tool use optimization is the capacity of an agent to augment its intrinsic capabilities by intelligently invoking external modules. This allows agents to overcome limitations such as outdated knowledge, inability to perform precise calculations, or lack of access to private information. The core challenge lies in the agent's ability to reason about **when** to use a tool, **which** tool to select from a library, and **how** to generate a valid call. In this section, we examine existing approaches to tool use optimization, which can be broadly classified into three styles: *in-context tool-integration*, *post-training tool-integration*, and *orchestration-based tool-integration*.

3.2.1 In-Context Tool-integration

The in-context demonstration paradigm is a training-free approach to empowering LLMs with new capabilities at inference time. This method leverages the remarkable in-context learning ability of modern LLMs, guiding a frozen, off-the-shelf model to perform complex tasks by providing carefully crafted instructions, examples, and contextual information directly in the prompt.

Interleaving Reasoning and Tool Use. The foundation of in-context agentic reasoning lies in augmenting the Chain-of-Thought (CoT) process with the ability to take action.(Wei et al., 2022). ChatCoT (Chen et al., 2023e) formalizes this paradigm by structuring reasoning traces as alternating "thought-tool-observation" steps in natural language, allowing LLMs to reflect on intermediate outputs and dynamically plan the next tool query. While CoT enables LLMs to break down problems into intermediate reasoning steps, it operates in a closed world, limited by the model's internal knowledge. The key innovation in agentic tool use is to interleave these reasoning steps with actions (tool calls), creating a dynamic loop that allows the agent to interact with external environments to gather information and execute tasks (Inaba et al., 2023; Trivedi et al., 2022; Xu et al., 2025d). ReAct (Yao et al., 2023c) introduced the "Reasoning+Acting" synergy. This approach enables the model to use reasoning to create, track, and adjust its action plans, while the actions allow it to interface with and gather information from external environments like knowledge bases or the web. Similarly, ART (Paranjape et al., 2023) provides a structured approach by maintaining a library of successful task demonstrations. For a new task, ART retrieves a relevant multi-step exemplar and uses it as a few-shot prompt, guiding the LLM to follow a proven reasoning and tool-use path. Formal planning tools extend this interleaving pattern further: PDDL Planning Copilot (Benyamin et al., 2025) connects natural-language reasoning with PDDL artifacts so that agents can formulate, revise, and invoke symbolic planning routines during tool-augmented problem solving.

Optimizing Context for Tool Interaction. While the foundational interleaved loop is powerful, its performance degrades when agents must handle large or complex toolsets. A significant branch of research addresses this by optimizing the in-context information provided to the agent. Recent studies demonstrate that well-written tool documentation enables LLMs to utilize new tools in a zero-shot manner (Hsieh et al., 2023a; Yuan et al., 2024b). This finding aligns with the key insight that LLMs, much like humans, benefit from clear and concise instructions. Alternatively, GEAR (Lu et al., 2024b) introduces a computationally efficient, training-free algorithm that delegates the tool selection process to a small language model while reserving the more powerful LLM for the final reasoning step to reduce costs. AVATAR (Wu et al., 2024d) enhances the robustness of this choice by prompting the agent to perform in-context "contrastive reasoning" before acting.

While these in-context methods are flexible, their performance is ultimately bounded by the inherent capabilities of the frozen LLM and the length of its context window. Consequently, subsequent research has focused on post-training methods.

3.2.2 Post-training Tool-integration

Tool integration (Yao et al., 2023c; Qu et al., 2025a; Shi et al., 2025c) with post-training techniques has emerged as a key strategy for addressing the inherent limitations of LLMs or LRMs, such as outdated knowledge, limited computational precision, and shallow multi-step reasoning. By *learning how to interact with external tools*, reasoning models can dynamically access up-to-date information, execute precise symbolic or numerical computations, and decompose complex tasks into grounded, tool-assisted reasoning steps (Wang et al., 2024c; Yang et al., 2024e; Singh et al., 2025a; Cheng et al., 2024; Zou et al., 2026). With tools as intermediaries, models are enriched and augmented by external capabilities, enabling the generation of more accurate and generalizable agentic reasoning trajectories (Nam et al., 2024; Yuan et al., 2024b; Wu et al., 2025e).

Bootstrapping of Tool Use via SFT. Early works on tool-integration (Yao et al., 2023c; Schick et al., 2023; Qin et al., 2024b; Tang et al., 2023a; Lu et al., 2023b; Song et al., 2023b; Prasad et al., 2023; Yin et al., 2023; Shi et al., 2024b; Meng et al., 2024b; Li et al., 2025c) primarily apply supervised fine-tuning (SFT) over

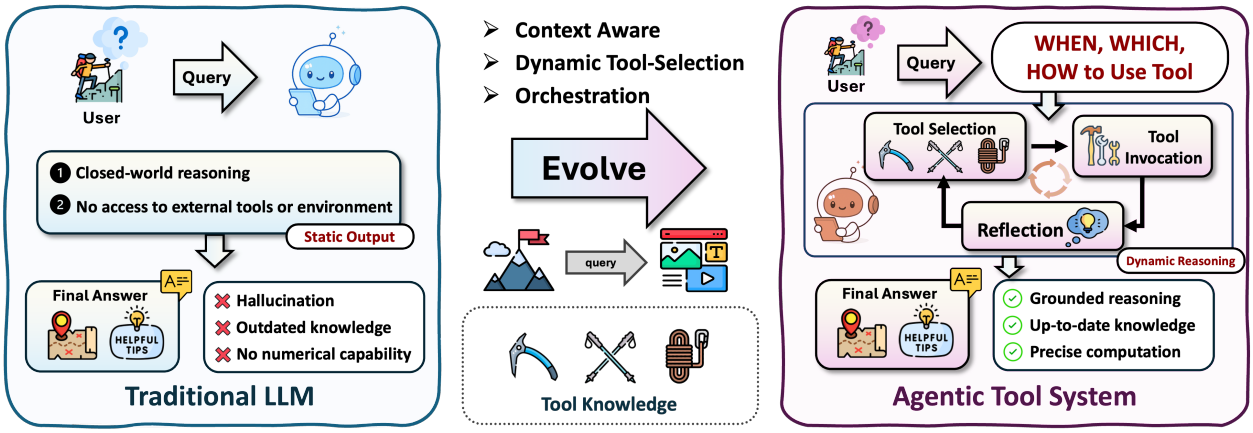

Figure 3: Comparison between **traditional LLM** and **agentic tool-use** systems. While traditional models operate in a closed world with fixed reasoning, agentic tool-use systems enable dynamic selection, orchestration, and integration of external tools, allowing agents to extend reasoning, improve precision, and dynamically adapt across domains.

curated tool-use reasoning steps, where models were trained to imitate demonstrations of search queries, code executions, or API calls. The SFT stage provided an initial competency in invoking tools, interpreting tool outputs, and integrating the results into coherent reasoning chains (Song et al., 2023b; Shinn et al., 2023). For example, Toolformer (Schick et al., 2023) introduces a self-supervised framework in which large language models generate, validate, and retain useful API calls within unlabeled text, followed by fine-tuning on the filtered data to enhance factual accuracy and practical utility. ToolLLM (Qin et al., 2024b) further scales SFT training to over 16,000 real-world APIs, applying supervised fine-tuning on massive curated demonstrations to endow models with robust planning and invocation abilities. ToolAlpaca (Tang et al., 2023a) extends the idea to compact LLMs by automatically constructing a diverse toolset and generating multi-turn tool-use dialogues via multi-agent simulation, followed by fine-tuning to enable generalized tool-use even for previously unseen tools. While effective at bootstrapping tool-awareness, applying SFT alone suffers from overfitting to the specific patterns in the training data (Kirk et al., 2023; Li et al., 2024k; O'Mahony et al., 2024; Zou et al., 2025a), leading to brittle tool-selection strategies and limited adaptability in unseen downstream application scenarios (Zeng et al., 2025b; Qian et al., 2025a; Yu et al., 2025e).

Mastery of Tool Use via RL. Recent studies (Zhou et al., 2025c; Qian et al., 2025a; Wei et al., 2025d; Chen et al., 2026c; Zhang et al., 2025z; Jin et al., 2025a; Zou et al., 2025d; Dong et al., 2025c; Ma et al., 2025b;a; Dong et al., 2025b;a) leverage reinforcement learning (RL) during model post-training to go beyond imitation and achieve mastery in tool-integrated reasoning. With the integration of RL, models refine their tool-use strategies through outcome-driven rewards, learning *when*, *how*, and *which* tools to invoke via trial and error (Chen et al., 2026c; Feng et al., 2025a; Dong et al., 2025c; Sun et al., 2025a; Chen et al., 2025f). For instance, SWE-RL (Wei et al., 2025d) optimizes code-editing policies on large-scale software evolution data, improving not only software issue resolution but also general reasoning skills. ReSearch (Chen et al., 2026c) embeds search operations into multi-hop reasoning chains, enabling adaptive retrieval during complex QA. ReTool integrates real-time code execution into reasoning rollouts, leading to optimal performance on advanced math reasoning benchmarks. ToolRL (Qian et al., 2025a) generalizes this paradigm to diverse toolsets by introducing principled reward designs for stable and scalable multi-tool learning. Newer agent-scale systems broaden the target from individual tool calls to long-horizon, multi-tool policies: DeepAgent (Li et al., 2026a) scales general reasoning over large toolsets, while pattern-aware and mid-level supervision methods teach models not only when to call tools but also how to follow reusable tool-use structures (Xu et al., 2026a; Jiang & Ferraro, 2026). Across these settings, RL has been shown to yield more robust, adaptive, and generalizable tool-use policies than SFT alone, often transferring effectively to out-of-domain tasks (Team et al., 2025d; Comanici et al., 2025; Team et al., 2025c; Zeng et al., 2025a; Zou et al., 2025c).

3.2.3 Orchestration-based Tool-integration

In real-world applications, tool use within complex systems often extends beyond the single-model, single-tool setting, requiring orchestration among multiple tools to complete complex tasks. This orchestration typically involves planning, sequencing, and managing dependencies across tools, i.e., ensuring that intermediate outputs are passed and transformed appropriately. Several early works (Shen et al., 2023; Liang et al., 2023; Hao et al., 2023b) explore this direction by devising strategies for the coordinated use of multiple tools, enabling systems to solve multi-stage tasks that no single tool can handle in isolation. Specifically, HuggingGPT (Shen et al., 2023) employs a centralized agent that leverages a language interface to plan which tools to invoke and when, enabling the solution of complex tasks requiring multiple tools in sequence. TaskMatrix.AI (Liang et al., 2023) connects foundation models with millions of APIs, using the models to generate task-solution outlines and automatically matching certain sub-tasks to off-the-shelf models and systems with specialized functionalities. ToolkenGPT (Lu et al., 2025c) augments frozen language models with massive tool sets by encoding each tool as a special token during next-token prediction.

Agentic Pipelines for Tool Orchestration. There are many frameworks designed to enable LLMs to call and orchestrate tools effectively. Most of the current agentic paradigm follows a "plan before action" strategy, where the model first generates a structured plan for tool use and then executes it. ToolPlanner (Liu et al., 2025g) introduces a two-stage reinforcement learning framework with path planning and feedback, supported by MGToolBench, to bridge the gap between API-heavy training data and real-world user instructions. A related ToolPlanner variant focuses on multi-granularity user instructions, using path planning and feedback to make tool-augmented solutions more explicit and easier to revise (Wu et al., 2024c). Tool-MVR (Ma et al., 2025c) enhances reliability and reflection through meta-verification of tool calls and exploration-based reflection learning, achieving strong gains over GPT-4 and other baselines. More recently, OctoTools (Lu et al., 2025c) provides a training-free, extensible framework with standardized tool cards, a hierarchical planner, and an executor, showing broad improvements across multi-domain reasoning tasks. Chain-of-Tools (Wu et al., 2025g) leverages frozen LLMs' semantic representations to dynamically compose unseen tools in chain-of-thought reasoning, enabling generalization to massive tool pools without fine-tuning. PyVision (Zhao et al., 2025e) introduces an interactive, multi-turn framework that enables MLLMs to dynamically generate, execute, and refine Python-based tools, moving beyond static toolsets in visual reasoning. ConAgents (Shi et al., 2024b) makes an initial extension of tool use frameworks for interactive multi-agent settings. We are also glad to see emerging applications of such agentic tool orchestration frameworks in the chemistry domain (Zou et al., 2025f).

Tool Representations for Orchestration. Beyond designing orchestration pipelines, another line of research focuses on optimizing the tools themselves to facilitate more accurate selection, composition, and coordination during orchestration. ToolExpNet (Zhang et al., 2025y) models tools and their usage experiences as a network that encodes semantic similarity and dependency relations, allowing LLMs to distinguish between similar tools and account for interdependencies during selection. T2Agent (Cui et al., 2026) addresses multimodal misinformation detection by representing tools with standardized templates and using Bayesian optimization to select a task-relevant subset. Coupled with Monte Carlo Tree Search over this reduced action space, T2Agent enables efficient multi-source verification. ToolChain* (Zhuang et al., 2024) frames the entire tool action space as a decision tree and applies A* search with task-specific cost functions to guide navigation. This representation allows efficient pruning of high-cost branches and identification of optimal tool-use paths. ToolRerank (Zheng et al., 2024b) refines tool retrieval by introducing adaptive truncation for seen vs. unseen tools and hierarchy-aware reranking to balance concentration (for single-tool queries) and diversity (for multi-tool queries).

3.3 Agentic Search

Single-agent Agentic Retrieval-Augmented Generation (RAG) systems embed reasoning and control into a centralized agent that governs the entire retrieval-generation loop. Unlike traditional RAG pipelines (Lewis et al., 2020; Huang & Huang, 2024; Yang et al., 2024f) that perform fixed, one-shot retrieval before generation, agentic RAG agents dynamically control *when*, *what*, and *how* to retrieve based on real-time reasoning needs.

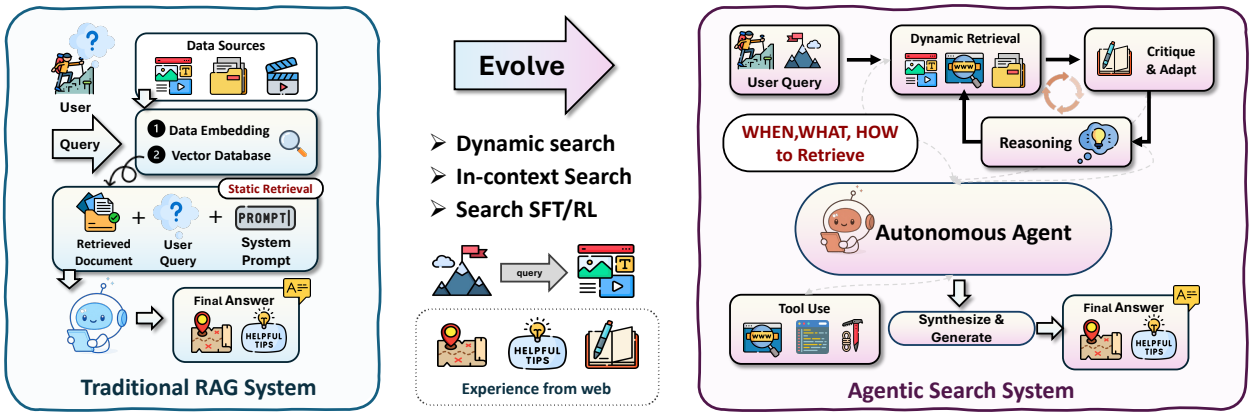

Figure 4: Comparison between **traditional RAG** systems and **agentic search** systems. Traditional RAG relies on static retrieval over a vector database, while agentic search introduces autonomous decision-making for when, what, and how to retrieve, enabling dynamic search, in-context retrieval, critique-and-adapt loops, and tool use.

This enables the model to adapt retrieval strategies mid-inference, refine its queries, and better integrate evidence from multiple sources. Based on how the agent selects, refines, and integrates retrieved content during reasoning, we categorize single-agent Agentic RAG systems into three distinct architectural styles: *in-context*, *post-training*, and *structure-enhanced* agentic RAG.

Table 4: Representative **Agentic Search** systems categorized by *Reasoning Structure*, *Format*, and *Tool Use*. NL denotes natural language traces used during reasoning, Ops refers to symbolic or graph operations, and KG stands for knowledge graph. Tool use includes search APIs, browser actions, or KG-based retrieval.

Method	Structure	Format	Tool
Modality I: In-Context Agentic Search			
ReAct (Yao et al., 2023c)	Interleaved	NL + Actions	Search API
Self-Ask (Press et al., 2022)	Decomposed	NL Queries	Search API
IRCoT (Trivedi et al., 2022)	Sequential	NL + CoT	Search API
Self-RAG (Asai et al., 2023)	Reflective	NL Self-check	Conditional Search
DeepRAG (Guan et al., 2025)	Iterative	NL Feedback	Search API
Modality II: Post-Training Agentic Search			
Toolformer (Schick et al., 2023)	Sequential	Tool Tokens	APIs, Search
INTERS (Zhu et al., 2024c)	Sequential	Instructions	Search API
WebGPT (Nakano et al., 2021)	Sequential	NL + Browser	Web Search
RAG-RL (Huang et al., 2025a)	Decision	NL Policy	Evidence API
Search-R1 (Jin et al., 2025a)	Iterative	NL + Tokens	Live Web
Deep-Researcher (Zheng et al., 2025c)	Multi-step	NL Trajectories	Browser Tools
ReSearch (Chen et al., 2026c)	Step-wise	NL Steps	Search + Verifier
ReARTeR (Sun et al., 2025d)	Reflective	NL Policy	Tool Cluster
Modality III: Structure-Enhanced Agentic Search			
Agent-G (Lee et al., 2024c)	Modular	NL + Graph Ops	KG Query
MC-Search (Ning et al., 2025)	Multi-step	NL	Multimodal Search
GeAR (Shen et al., 2025)	Graph	Graph Ops	KG Expansion
ARG (Zhang et al., 2025h)	Reflective	NL + Symbols	KG Traversal

3.3.1 In-Context Search

Interleaving Reasoning and Search. In-context agentic RAG systems embed retrieval behavior directly into the inference process of language models through carefully designed prompting strategies. Rather than training the model to learn retrieval behavior, these methods guide it to alternate between reasoning and search within a single forward pass, typically via few-shot exemplars or special tokens. A representative example is ReAct (Yao et al., 2023c), which interleaves Chain-of-Thought reasoning with tool-use commands such as `<Search>` to dynamically invoke external APIs or knowledge sources. Extensions such as Self-Ask (Press et al., 2022) and IRCoT (Trivedi et al., 2022) go beyond sequential reasoning by prompting the model to recursively decompose questions and retrieve sub-evidence accordingly. More recent methods (Asai et al., 2023; Li et al., 2024h; Guan et al., 2025; Ning et al., 2025) introduce reflective retrieval, where the model explicitly assesses whether it needs additional information at each step, deciding to retrieve only when necessary. These approaches require no additional training, making them highly flexible and deployable, but often rely on prompt engineering and may struggle with stability across diverse domains.

Structure-Enhanced Search. Structure-enhanced agentic RAG systems enhance retrieval-augmented generation by enabling a single agent to reason over symbolic knowledge sources such as knowledge graphs through dynamic querying, tool invocation, and reflective self-monitoring. Unlike static KG retrievers or query executors, these agents decide when to access structured knowledge, how to formulate graph-based queries, and whether retrieved information suffices for continuing the reasoning trajectory. Agent-G (Lee et al., 2024c) introduces a modular agentic architecture that integrates unstructured document retrieval with structured graph reasoning, using feedback loops and specialized retriever modules to ensure accurate multi-hop responses. MC-Search (Ning et al., 2025) introduces five canonical reasoning topologies to model multimodal search-enhanced reasoning process, and proposes a end-to-end agentic RAG and step-wise evaluation pipeline to evaluate model's planning and retrieval fidelity across heterogeneous sources. Similarly, GeAR (Shen et al., 2025) incorporates graph expansion operations into an agentic controller to address challenges in complex multi-hop queries, enhancing coherence across structured and unstructured sources. Beyond retrieval orchestration, ARG (Zhang et al., 2025h) proposes a fully end-to-end agentic framework for reasoning over knowledge graphs via active self-reflection. The model autonomously determines when to retrieve, performs iterative critique based on symbolic inputs, and exhibits interpretable, step-wise reasoning behavior over graphs. Together, these systems represent a shift from passive graph access to active, feedback-driven symbolic reasoning, highlighting the potential of structured agentic RAG to achieve both factual reliability and interpretability.

3.3.2 Post-Training Search

Post-training agentic RAG methods endow language models with retrieval-aware capabilities by fine-tuning them to make informed decisions throughout multi-step reasoning. Unlike in-context prompting, these approaches train models, either via supervised fine-tuning (SFT) or reinforcement learning (RL), to determine when retrieval is necessary, how to formulate queries, and how to incorporate retrieved evidence.

SFT-Based Agentic Search. These methods construct curated or synthetic datasets that interleave retrieval operations with natural language reasoning, and subsequently apply supervised fine-tuning to instill retrieval-aware capabilities into the model. Toolformer (Schick et al., 2023) introduces a self-supervised approach to annotate tool-use behaviors within model-generated text, enabling LLMs to learn when and how to invoke tools such as web search or calculators. INTERS (Zhu et al., 2024c) extends this direction by performing instruction-based fine-tuning over a diverse, multi-task dataset compiled from over 40 sources, capturing a wide spectrum of retrieval-reasoning patterns. This class of methods benefits from scalable data generation pipelines (Mao et al., 2024; Zhang et al., 2024i; Li et al., 2025i), which minimize the need for human annotation. Instructional reformulation techniques (Lin et al., 2024; Zhu et al., 2024c; Nguyen et al., 2024) further enhance generalization by aligning tasks with human-preferred formats and reasoning.

RL-Based Agentic Search. These methods optimize retrieval-aware behaviors through reward signals that reflect answer quality, factuality, or user preferences. WebGPT (Nakano et al., 2021) introduces reward modeling to supervise search-augmented chains aligned with human judgment, while RAG-RL (Huang et al.,

2025a) formulates retrieval as a sequential decision-making task over evidence access. More recent efforts such as Search-R1 (Jin et al., 2025a) and Deep-Researcher (Zheng et al., 2025c) go further by training agents to dynamically issue retrieval actions (e.g., generating `<Search>` tokens mid-reasoning) and operate in open-ended environments such as the live web. These agents exhibit emergent capabilities such as iterative decomposition, re-verification, and evidence planning. In structured knowledge settings, KBQA-R1 (Sun et al., 2025c) applies reinforcement learning to knowledge-base question answering, encouraging models to learn retrieval and reasoning decisions over symbolic evidence chains. Finally, systems like ReSearch (Chen et al., 2026c) and ReARTeR (Sun et al., 2025d) pursue not only accurate answers but also interpretable and faithful reasoning trajectories, highlighting the potential of reinforcement-learned retrievers to act as controllable and reflective agents.

4 Self-evolving Agentic Reasoning

Self-evolving agentic reasoning refers to an agent's capacity to *improve its own reasoning process through experience*. At the core of this evolution lie two fundamental mechanisms: **feedback** and **memory**. **Feedback** provides evaluative signals for self-correction and refinement, allowing the agent to revise its reasoning strategies based on outcomes or environmental responses. **Memory**, in turn, acts as a persistent substrate for storing, organizing, and synthesizing past interactions, enabling knowledge accumulation and reuse across tasks. Together, these mechanisms transform reasoning from a static process into a dynamic, adaptive loop capable of continual improvement.

Building upon foundational capabilities such as *planning*, *search*, and *tool use*, self-evolving agents integrate feedback and memory to refine their internal reasoning policies, adjust decision-making strategies, and generalize across diverse contexts, often without explicit external supervision. This continual adaptation marks a critical step toward lifelong reasoning and lays the groundwork for the collective intelligence explored in the next section.

4.1 Agentic Feedback Mechanisms

Agentic feedback mechanisms enable models to iteratively refine their reasoning and actions rather than relying on one-shot responses. By incorporating self-critique, verifier guidance, or validator-based resampling, these methods emulate human trial-and-error learning and form the foundation for autonomous self-improvement. Broadly, they operate through three distinct feedback regimes: (1) reflective feedback, where models revise their reasoning through self-critique or verification; (2) parametric adaptation, where feedback is consolidated into updated model parameters; and (3) validator-driven feedback, where binary outcome signals guide resampling without introspection.

These regimes define a continuum between dynamic, inference-time adaptability, durable learning through parameter updates, and efficient correction through external signals. Together, they highlight how modern agents leverage feedback to balance flexibility, reliability, and efficiency.

4.1.1 Reflective Feedback

Reflective feedback methods improve model reliability by modifying the reasoning process during inference, without updating model parameters. These approaches expose intermediate reasoning outputs, such as chains of thought or partial solutions, and introduce additional assessment steps that directly influence how the model continues its generation.

Early self-critique and rationale-refinement methods (Shinn et al., 2023; Madaan et al., 2023) implement reflection through an explicit generate–critique–revise loop. A model first produces an answer together with its reasoning. The same model, or a separately prompted critic role, then analyzes this output to identify logical errors, unsupported assumptions, or missing steps. The critique is appended as context for a revised generation, and this process may be repeated multiple times or augmented with external evidence such as retrieval. More recent self-improvement frameworks (Wang et al., 2024n) extend reflective feedback beyond a single inference episode by accumulating critiques or failure cases across interactions. Instead of correcting only one response, these methods reuse past feedback to guide future generations through prompt

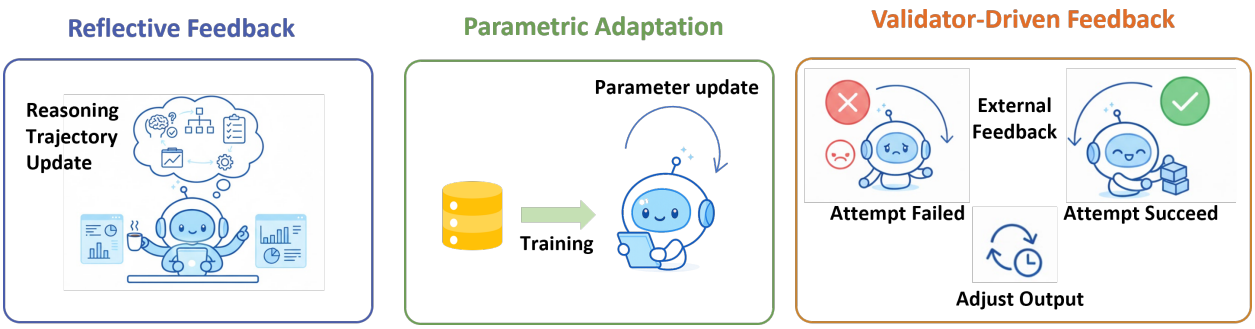

Figure 5: **Illustration of three forms of agentic feedback mechanisms.** *Inference-time reflection* enables real-time self-critique and revision during reasoning; *offline adaptation* consolidates feedback into model parameters for long-term improvement; and *outcome-based feedback* relies on validator signals (success or failure) to refine behavior through retry. Together, they represent a continuum from adaptive reflection to stable learning and efficient validation.

refinement or curated supervision signals, while still operating without direct parameter updates at inference time. Search-based reasoning strategies (Wang et al., 2023d; Yao et al., 2023b; Besta et al., 2024) improve reliability by generating and comparing multiple candidate reasoning paths. These methods explore the solution space through stochastic sampling or structured search, then select or aggregate outputs using voting schemes, heuristic scores, or learned evaluators. Improvement arises from comparison across alternatives rather than explicit revision of a single reasoning trajectory. Decomposition-based prompting methods (Zhou et al., 2022; Chen et al., 2023c) reformulate complex problems into ordered sequences of simpler subproblems. Intermediate results are reused in later steps, allowing partial inspection of reasoning progress and reducing error propagation, even when no explicit critique step is introduced.

Overall, reflective feedback alters inference-time reasoning trajectories by introducing additional reasoning or comparison steps. Feedback is used to guide generation within an episode, while the model's parameters remain unchanged.

4.1.2 Parametric Adaptation

Parametric adaptation incorporates feedback into a model's parameters through additional training, producing persistent behavioral changes that generalize beyond individual inference episodes. Unlike reflective feedback, these methods transform feedback signals into supervised or preference-based training objectives that update the model's weights.

Trajectory-level supervised fine-tuning approaches (Zeng et al., 2024; Aksitov et al., 2023) attach feedback to intermediate reasoning traces rather than only final answers. Models first generate multi-step trajectories, which are then reviewed by humans, auxiliary models, or automated verifiers. Incorrect steps are corrected or replaced, and the resulting feedback-enriched trajectories are used as supervised training data, encouraging the model to internalize improved reasoning patterns. Distillation-based methods (Hsieh et al., 2023b) further leverage improved reasoning traces by training student models on high-quality chains of thought or self-corrected solutions generated by stronger teachers. This process transfers structured reasoning behaviors into more stable or efficient models, removing the need for explicit reflection at inference time. Preference-alignment approaches (Christiano et al., 2017; Rafailov et al., 2023; Bai et al., 2022) incorporate feedback in the form of comparative judgments that distinguish preferred from dispreferred outputs. Training objectives such as reward modeling or direct preference optimization adjust the model's parameters so that preferred behaviors become more likely. Although feedback is often defined over final outputs, it implicitly shapes the internal reasoning strategies that produce them. Recent work shows that verification-augmented training data can further improve reasoning robustness across domains (Li et al., 2025e; Zheng & Lee, 2025). In these settings, trajectories are filtered or revised based on correctness or consistency signals before training, yielding datasets that emphasize reliable reasoning patterns.

Table 5: Representative **Agentic Feedback Mechanisms** categorized by *Feedback Stage*, *Feedback Source*, and *Update Target*.

Method / System	Feedback Stage	Feedback Source	Update Target
I. Reflective Feedback			
Reflexion (Shinn et al., 2023)	Inference	Self-generated critique	Trajectory
Self-Refine (Madaan et al., 2023)	Inference	Self-evaluation	Trajectory
Constitutional AI (Bai et al., 2022)	Inference	Normative rules	Trajectory
RLAIF (Lee et al., 2024a)	Inference	AI verifier	Trajectory
SelfCheckGPT (Manakul et al., 2023)	Inference	Cross-sample divergence	Trajectory
COT STEP (Chowdhury & Caragea, 2025)	Inference	External verifier	Trajectory
ASCoT (Zhang et al., 2025d)	Inference	Vulnerability detection	Trajectory
MM-Verify (Sun et al., 2025b)	Inference	Multimodal verifier	Trajectory
ReAct (Yao et al., 2023c)	Inference	Action outcomes	Trajectory
PAL (Gao et al., 2023)	Inference	Code execution	Trajectory
WebGPT (Nakano et al., 2021)	Inference	Web evidence	Trajectory
MemGPT (Packer et al., 2023)	Inference	Retrieved memory	Trajectory
Voyager (Wang et al., 2023a)	Inference	Environment + memory	Trajectory
II. Parametric Adaptation			
AgentTuning (Zeng et al., 2024)	Training	High-quality trajectories	Model parameters
ReST (Aksitov et al., 2023)	Training	Critique–revision pairs	Model parameters
ReFT (Dou et al., 2024)	Training	Reflection-augmented data	Model parameters
Distill-CoT (Hsieh et al., 2023b)	Training	Expert CoT	Model parameters
ReflectEvo (Li et al., 2025e)	Training	Reflection traces	Model parameters
Reasoning-CV (Zheng & Lee, 2025)	Training	Verification signals	Model parameters
III. Validator-Driven Feedback			
ReZero (Dao & Le, 2025)	Inference	Binary validator	Output only
Retrials (Potamitis & Arora, 2025)	Inference	Acceptance signal	Output only
CodeRL (Le et al., 2022)	Inference	Unit tests	Output only
LEVER (Ni et al., 2023)	Inference	Execution results	Output only
SWE-bench (Jimenez et al., 2024)	Inference	Test suite	Output only
SayCan (Ahn et al., 2022)	Inference	Environment state	Output only
PaLM-E (Driess et al., 2023)	Inference	Environment feedback	Output only
Reflect–Retry–Reward (Bensal et al., 2025)	Inference	Validator + reflection signal	Output only

In summary, parametric adaptation embeds feedback directly into the model's parameters, yielding durable improvements across tasks. This durability comes at the cost of additional training and reduced flexibility compared to inference-time methods.

4.1.3 Validator-Driven Feedback

Validator-driven feedback improves model outputs using external success or failure signals, without modifying the model's reasoning process or parameters. A validator, such as a unit test, constraint checker, simulator, or environment signal, evaluates candidate outputs and determines whether they satisfy predefined correctness criteria.

Retry-based systems (Dao & Le, 2025; Potamitis & Arora, 2025) implement this paradigm by repeatedly sampling candidate outputs until one passes validation. The model generates a complete solution, submits it to the validator, and discards it if validation fails. Subsequent attempts are generated independently, without conditioning on explicit information about previous failures. This strategy is particularly effective in domains

with reliable and inexpensive validation, such as program synthesis and software engineering (Le et al., 2022; Ni et al., 2023; Jimenez et al., 2024). Generated code can be executed against unit tests, providing an unambiguous correctness signal. The model iterates until a solution satisfies all tests, even in the absence of explicit reasoning correction. Similar mechanisms appear in embodied and interactive agents (Ahn et al., 2022; Driess et al., 2023), where action sequences are repeatedly executed until the environment signals task completion. Failed sequences are abandoned and new ones are attempted, based solely on external success signals. Some hybrid methods introduce lightweight guidance within the retry loop, for example by assigning higher reward to behaviors that eventually lead to successful outcomes (Bensal et al., 2025). However, the dominant mechanism remains selection through external validation rather than revision of reasoning steps or parameter updates.

Overall, validator-driven feedback offers an efficient and scalable way to improve output correctness when reliable validators are available. Its limitation is that feedback is non-diagnostic, correcting individual outputs without explaining failures or altering the model's reasoning behavior.

4.2 Agentic Memory

Recent advances in memory-augmented LLM agents have shifted the focus from static memory storage to more dynamic, interactive mechanisms that directly support agentic reasoning. Rather than merely extending the context window or storing historical inputs, memory is increasingly treated as an integral component of the reasoning loop, used for reflecting on past experiences, guiding future actions, and dynamically adapting to complex, long-horizon tasks. Formally, an agent maintains a memory module where each memory entry may represent a raw observation, summarized trajectory, subgoal, tool invocation trace, or other structured element depending on the system design.

The agent's reasoning process then operates not only on its immediate context but also on this persistent memory, enabling reflection, generalization, and long-term goal tracking. In this section, we organize prior work along four emerging trends in the use of memory to support and enable agentic reasoning. Figure 6 summarizes how agentic memory progresses from contextual recall to adaptive control. In-context memory captures textual and semantic information from prior interactions; structured memory integrates these into graph and multimodal representations; post-training control enables agents to evolve, update, and retrieve memory through learned reward-based mechanisms.

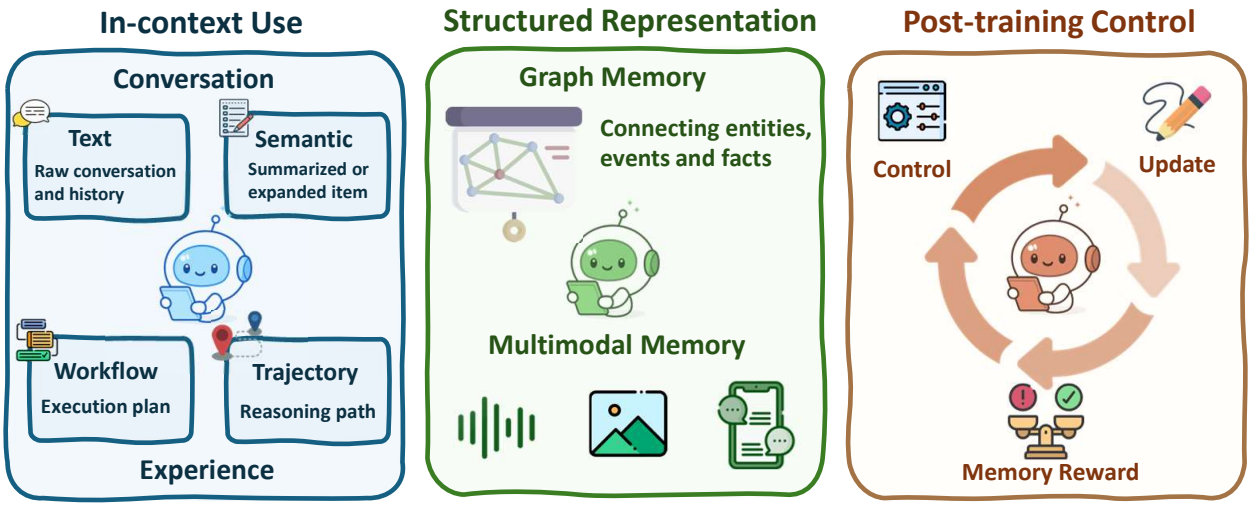

Figure 6: Overview of **Agentic Memory** in LLM agents, showing three parallel dimensions: in-context use (text and experience), structured representation (graph and multimodal memory), and post-training control (reward-guided memory management).

Table 6: Representative **Agentic Memory** systems categorized by *Setting*, *Format*, and *Memory Type*.

Method / System	Setting	Format	Memory Type
I. Agentic Use of Flat Memory (In-Context)			
LangMem (AI, 2023)	In-Context	Text	Factual
LlamaIndex (Liu, 2022)	In-Context	Text	Factual
MemGPT (Packer et al., 2023)	In-Context	Text	Factual
MemoryBank (Zhong et al., 2023)	In-Context	Semantic	Factual
Amem (Xu et al., 2025h)	In-Context	Semantic	Factual
Workflow Memory (Wang et al., 2024o)	In-Context	Workflow	Experience
MemOS (Li et al., 2025m)	In-Context	Semantic	Factual
LightMem (Fang et al., 2025b)	In-Context	Semantic	Factual
Nemori (Nan et al., 2025)	In-Context	Semantic	Factual
ACE (Zhang et al., 2025n)	In-Context	Workflow	Experience
Reasoning Bank (Ouyang et al., 2025a)	In-Context	Workflow	Experience
Dynamic Cheatsheet (Suzgun et al., 2025)	In-Context	Trajectory	Experience
Sleep-time Compute (Lin et al., 2025b)	In-Context	Trajectory	Experience
Evo-Memory (Wei et al., 2025c)	In-Context	Semantic	Experience
II. Structured Memory Representations			
GraphRAG (Edge et al., 2024)	In-Context	Graph	Factual
MEM0 (Chhikara et al., 2025)	In-Context	Graph	Factual
Zep (Rasmussen et al., 2025)	In-Context	Graph	Factual
Optimus-1 (Li et al., 2024i)	In-Context	Multimodal	Experience
RAP (Kagaya et al., 2024)	In-Context	Multimodal	Experience
M3-Agent (Long et al., 2025)	In-Context	Multimodal	Factual
Mem-Gallery (Bei et al., 2026)	In-Context	Multimodal	Factual
Agent-ScanKit (Cheng et al., 2025a)	In-Context	Multimodal	Experience
III. Post-training Memory Control			
Mem1 (Zhou et al., 2025g)	Post-training	Semantic	Factual
Memory-as-Action (Zhang et al., 2025v)	Post-training	Semantic	Factual
MemAgent (Yu et al., 2025a)	Post-training	Semantic	Factual
Mem-α (Wang et al., 2025k)	Post-training	Semantic	Factual
Memory-R1 (Yan et al., 2025)	Post-training	Semantic	Factual
Agent Early Experience (Zhang et al., 2025k)	Post-training	Implicit	Experience
Agentic Memory (Yu et al., 2026)	Post-training	Semantic	Experience
MemRL (Zhang et al., 2026b)	Post-training	Semantic	Experience

4.2.1 Agentic Use of Flat Memory

Factual Memory. Traditional memory systems for LLM agents typically treat memory as a passive buffer, mainly used to store dialogue histories or recent observations to address the limited context window of transformer models. Examples include dense retrieval methods (Lewis et al., 2020; Asai et al., 2024; Zhong et al., 2023), pre-defined modules in LangChain and LLamaIndex (Liu, 2022), and cache-inspired designs like MemGPT (Packer et al., 2023). These approaches usually retrieve semantically similar past content to augment prompts, without influencing the agent's internal reasoning. Enhancements such as RET-LLM with differentiable memory (Modarressi et al., 2023), SCM with controller-based mechanisms (Wang et al., 2025a), as well as LOCOMO and LongMemEval benchmarks for long-term retention (Maharana et al.,

2024; Wu et al., 2024a) further improve recall but remain largely static. These systems often rely on fixed heuristics and unstructured token lists (Zhong et al., 2023), limiting adaptability for tasks involving goal decomposition (Yang et al., 2025d; Zhang et al., 2025u), long-term planning (Zheng et al., 2024c), or iterative self-improvement (Patel et al., 2024). In contrast, emerging agentic memory treats memory as part of the reasoning loop, supporting reflection (Bo et al., 2024), and decision-making (Yu et al., 2023c). Amem (Xu et al., 2025h) enables LLM agents to autonomously generate contextual memory descriptions, build dynamic links between related experiences, and evolve memory content in response to new information. Reflective memory managers further introduce prospective and retrospective updates for long-term personalized dialogue, deciding what should be retained before future interactions and what should be revised after observing outcomes (Tan et al., 2025c). Similarly, Zep (Rasmussen et al., 2025), Mirix (Wang & Chen, 2025), MemOS (Li et al., 2025m), LightMem (Fang et al., 2025b), and Nemori (Nan et al., 2025) leverage LLMs to automatically produce context-aware memory representations. Beyond LLM-driven approaches, recent work has explored reinforcement learning to explicitly train agents to acquire and organize factual memory, such as Mem-α (Wang et al., 2025k) and Memory-R1 (Yan et al., 2025), which we discuss in detail in later sections.

Experience Memory. Workflow Memory (Wang et al., 2024o) tracks procedural traces to enable plan recovery and consistent reasoning. Sleep-time Compute enables LLM agents to **pre-compute** and store anticipated reasoning steps before user interaction, effectively "thinking offline" using memory as a preparatory resource (Lin et al., 2025b). Dynamic Cheatsheet (DC) (Suzgun et al., 2025) equips black-box models with external memory to store reusable strategies, reducing redundant reasoning. Other efforts explore complementary paradigms of agentic memory. In parallel, workflow memory has emerged as another structured approach, particularly suited for procedural and tool-augmented tasks. It explicitly tracks procedural traces during execution, supporting plan recovery, long-term consistency, and interpretable chaining of actions. Atomic reasoning (Zhang et al., 2025u) proposes a structured trace over a finite set of reusable atomic skills in a streamlined generation space to reduce spurious reasoning patterns. Context evolution (ACE) (Zhang et al., 2025n) treats contexts as evolving playbooks rather than building a static structured store, whereas Reasoning Bank (Ouyang et al., 2025a) focuses on reusing failed reasoning traces to enhance future task performance. Evo-Memory (Wei et al., 2025c) synthesizes these ideas by benchmarking self-evolving memory under streaming task settings, highlighting experience reuse as a central capability for stateful, long-horizon agentic reasoning. In addition to factual memory, Mirix (Wang & Chen, 2025) further introduces a procedural memory component to capture reusable action patterns, while Agentic Memory (Yu et al., 2026) and MemRL (Zhang et al., 2026b) adopt reinforcement learning to optimize the acquisition and management of experiential memory.

This marks a shift from static buffers toward structured, reasoning-centric memory architectures. In these agentic memory systems, memory serves as a dynamically growing context: agents not only record past actions but actively **reflect, edit, and refine** their strategy over time.

4.2.2 Structured Use of Memory

Beyond flat memory usage and control, the structure of memory plays a critical role in enabling complex reasoning. Recent work increasingly explores structured representations, such as semantic graphs, workflows, and hierarchical trees, often extended to multimodal settings, to better capture dependencies, and contextual relationships.

Graph-based representations provide a flexible substrate for organizing relational knowledge in agents (Ouyang et al., 2025b). GraphRAG (Edge et al., 2024) serves as a foundational technique that augments retrieval with graph-structured reasoning, enabling more contextually coherent and multi-hop information integration. Building on this foundation, agent systems such as MEM0 (Chhikara et al., 2025) and Zep (Rasmussen et al., 2025) organize memory explicitly as dynamic knowledge graphs, allowing agents to store, retrieve, and reason over entities, attributes, and their relations with improved efficiency and semantic grounding. Beyond graphs, structured memory has also been explored through alternative organizational forms. MemTree (Rezazadeh et al., 2024) leverages a dynamic tree-structured representation to hierarchically organize and integrate information, while workflow-oriented systems such as AutoFlow (Li et al., 2024j), AFLOW (Zhang et al.,

2024f), and FlowMind (Zeng et al., 2023) represent reasoning workflows explicitly in memory, capturing sequences of subgoals, tool invocations, and decision points.

New benchmarks have pushed reasoning memory into multimodal domains, where agents are required to ground, retrieve, and reuse information across heterogeneous modalities. M3-Agent (Long et al., 2025) evaluates visual–audio–text reasoning through "see, listen, and reason," while Agent-Scankit (Cheng et al., 2025a) proposes multimodal agents with integrated memory modules for adaptive retrieval and grounding. Optimus-1 (Li et al., 2024i) proposes a hybrid multimodal memory architecture that represents world knowledge as a hierarchical directed knowledge graph and abstracts past interactions into a multimodal experience pool. RAP (Kagaya et al., 2024) retrieves relevant experiences based on contextual similarity, enabling adaptive reuse of multimodal memory.

These structured memory formats align task semantics, temporal dependencies, and multimodal signals, enabling agents to reason compositionally and maintain coherent behavior over extended interactions. As task complexity increases, the abstraction and organization of memory become increasingly critical for building robust and generalist agents.

4.2.3 Post-training Memory Control

Conversely, memory systems can also be controlled by the agent's reasoning process itself. Rather than relying on fixed heuristics for reading and writing memory, recent work has explored agent-controllable memory operations, where the agent explicitly decides what to store, when to retrieve, and how to interact with memory. This reframes memory as a policy target, no longer a passive buffer, but a resource that is actively shaped by reasoning.

MemAgent (Yu et al., 2025a) formulates memory overwrite as a reinforcement learning problem: the agent is rewarded for preserving information that proves useful and for discarding irrelevant content. By using a newly proposed DAPO algorithm, the model learns to maintain a constant-sized memory across conversations while maximizing future utility. Mem1 (Zhou et al., 2025g) presents an end-to-end reinforcement learning framework where agents maintain a compact, shared internal state across turns, jointly supporting reasoning and memory consolidation. Memory-R1 (Yan et al., 2025) further advances this line by introducing a dual-agent design: a Memory Manager that dynamically decides when to add, update, or delete entries in the memory store, and an Answer Agent that distills the most relevant retrieved memories to guide response generation. Recent work such as Mem-α (Wang et al., 2025k) also explores RL-based control of multi-component memory construction in agents, providing a unified perspective on adaptive memory construction and reasoning control. Memory-as-Action (Zhang et al., 2025v) integrates memory editing including insertions, deletions, and modifications directly into the reasoning policy, proposing a Dynamic Context Policy Optimization algorithm to handle non-prefix trajectory changes caused by memory operations. Agent Learning via Early Experience (Zhang et al., 2025k) further relaxes reward dependence by enabling agents to learn from their own interaction traces through self-prediction and reflection, bridging imitation and reinforcement learning. Moreover, Agentic Memory (Yu et al., 2026) and MemRL (Zhang et al., 2026b) adopt reinforcement learning to optimize the acquisition and management of experiential memory.

Together, these systems mark a shift toward **learning-based memory control**, where memory usage is optimized through reinforcement or imitation learning. By integrating memory management into the reasoning policy, agents become more adaptive, scalable, and capable of long-horizon decision-making in dynamic environments.

4.3 Evolving Foundational Agentic Capabilities

4.3.1 Self-evolving Planning

Recent advances view planning not as a fixed reasoning routine but as an evolving capability. Instead of relying on static datasets or human-designed curricula, agents can autonomously generate tasks, learn from their own feedback, and adapt strategies through iterative interaction with the environment. This enables continuous improvement without external supervision.

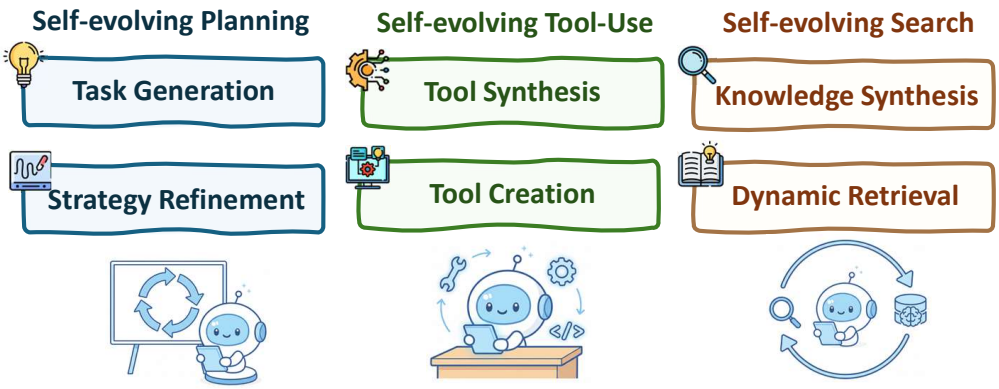

Figure 7: An overview of evolving foundational agentic capabilities along three key dimensions: *planning* (task generation and strategy refinement), *tool-use* (tool creation and synthesis), and *search* (dynamic retrieval and knowledge synthesis). These dimensions reflect how agentic systems autonomously enhance their reasoning and problem-solving capacity over time.

A representative direction is self-generated task construction. For example, SCA enables agents to alternate between generating problems and solving them, reusing successful trajectories for fine-tuning (Zhou et al., 2025d). Self-rewarding frameworks further allow agents to assess their own outputs, producing high-quality training signals without human labels (Yuan et al., 2024d; Simonds et al., 2025). Other works directly leverage execution feedback for online adaptation, such as SELF, SCoRe, PAG, TextGrad, and AutoRule, which transform natural-language critiques or traces into training rewards, enabling continual policy refinement (Lu et al., 2023a; Kumar et al., 2024; Yuksekgonul et al., 2024; Wang & Xiong, 2025).

Beyond internal feedback, agents can also evolve through environment shaping. AgentGen constructs adaptive environments to induce curriculum learning (Hu et al., 2025b), while Reflexion and AdaPlanner use self-reflective or adaptive strategies to refine plans at runtime (Shinn et al., 2023; Sun et al., 2023). Self-Refine iteratively critiques and improves outputs (Madaan et al., 2023), and SICA allows self-modification of code and reasoning tools (Robeyns et al., 2025). From an RL perspective, RAGEN and DYSTIL model planning as a Markov Decision Process and optimize strategies with dense feedback (Wang et al., 2025n;b).

Together, these methods establish a self-improving planning loop, where agents generate their own tasks, shape their environments, and refine strategies, laying the groundwork for autonomous, open-ended planning evolution.

4.3.2 Self-evolving Tool-use

Creating and Synthesizing Tools. The culmination of in-context reasoning is the emergent capability of agents to autonomously create new tools. This is achieved not through training, but by prompting a frozen LLM to act as a programmer when it encounters a problem that its existing toolset cannot solve. The LATM framework (Cai et al., 2024) uses a powerful model as a one-time "tool maker" and a cheaper, lightweight model as a frequent "tool user," thus amortizing the cost of creation. To enable specialization beyond the limits of general-purpose APIs, frameworks like CRAFT (Yuan et al., 2024a) and CREATOR (Qian et al., 2023) generate custom tools tailored for specific domains. Taking this a step further, ToolMaker (Wölflein et al., 2025) can convert entire public code repositories into usable tools, allowing agents to leverage complex, human-written codebases on the fly.

4.3.3 Self-evolving Search

Search plays a central role in agentic reasoning, enabling models to retrieve, select, and synthesize relevant knowledge across large and evolving memory spaces. In early systems, search was typically static—built on fixed retrieval heuristics or similarity-based dense retrievers (Lewis et al., 2020; Asai et al., 2023; Zhong et al., 2023; Packer et al., 2023). These methods augmented prompts with retrieved information but lacked adaptive control over how memory evolves or how search strategies are improved over time.

Recent research increasingly links search and memory in a **co-evolutionary loop**: agents continuously update their *memory base* during task execution, while dynamically adjusting how search is performed over this evolving knowledge. Agentic memory systems such as MemGPT (Packer et al., 2023), MemoryBank (Zhong et al., 2023), and Workflow Memory (Wang et al., 2024o) already highlight how retrieved information can be synthesized and re-inserted into memory, gradually improving retrieval quality. Dynamic Cheatsheet (DC) (Suzgun et al., 2025) further demonstrates how reusable strategies can be accumulated and leveraged across queries, effectively transforming static search into a *living retrieval substrate* that evolves with agent experience.

Evolving Memory Bases. Unlike static index-based retrieval, self-evolving agents actively refine their memory base through reflection and post-execution updates. Reflexion (Shinn et al., 2023) allows agents to critique their own reasoning traces and store distilled insights, improving future search relevance. Reasoning Bank (Ouyang et al., 2025a) and context evolution methods (Zhang et al., 2025n) explicitly restructure memory representations to align retrieval results with evolving problem-solving strategies, effectively making the retrieval target itself adaptive over time.

Dynamic Search and Synthesis. Beyond memory updates, search strategies themselves can evolve through dynamic prioritization and synthesis. Structured memory representations—such as workflows (Li et al., 2024j; Zhang et al., 2024f; Zeng et al., 2023) and knowledge graphs (Ouyang et al., 2025b; Edge et al., 2024; Chhikara et al., 2025; Rasmussen et al., 2025)—provide semantic scaffolding that enables multi-hop and compositional search, supporting richer reasoning over longer horizons. Systems like MemOS (Li et al., 2025m) and Memory-as-Action (Zhang et al., 2025v) take this further by integrating search decisions directly into the reasoning policy, allowing retrieval targets, strategies, and sources to co-adapt as agents accumulate experience.

Overall, self-evolving search transforms retrieval from a static utility into a continuously adapting component of the reasoning loop. By evolving memory bases, dynamically adjusting search strategies, and synthesizing retrieval results into structured knowledge, agents can maintain more relevant, structured, and actionable information over extended time horizons.

5 Collective Multi-agent Reasoning

Building upon the single-agent foundation, where reasoning supports planning, search, and tool use within a unified perception–action loop, **multi-agent reasoning** extends these principles to collaborative settings. In a multi-agent system (MAS), multiple reasoning agents interact to jointly solve complex tasks. Rather than identical problem solvers, agents assume *complementary roles*, such as *Manager* for task decomposition, *Worker* for execution, and *Verifier* for evaluation, enabling specialization and division of cognitive labor. This role differentiation marks the first step toward collective intelligence, where reasoning is distributed and coordinated across multiple agents.

Beyond role assignment, the essence of multi-agent reasoning lies in how these agents *collaborate, communicate, and co-evolve*. Collaboration schemas define how reasoning traces are exchanged, conflicts are resolved, and shared memory is maintained to achieve alignment. Through such interaction, reasoning transitions from an individual process into a distributed, iterative loop, in which agents refine each other's outputs and collectively converge toward better solutions.

Compared with single-agent systems, multi-agent reasoning introduces new challenges that require rethinking reasoning at the system level:

- **Role differentiation:** how to design static or adaptive roles that align with task structure and expertise distribution;
- **Collaboration and communication:** how agents exchange intermediate reasoning, negotiate consensus, and divide labor efficiently;

- **Collective memory and evolution:** how shared or distributed state supports long-term coordination and continual adaptation.

These challenges motivate the following structure of our analysis. Section **5.1** examines the *role taxonomy* of multi-agent systems, from generic organizational roles to domain-specific specializations. Section **5.2** focuses on *collaboration and division of labor*, including in-context and post-training coordination strategies. Finally, Section **5.3** explores how *memory* enables multi-agent systems to evolve over time and maintain collective consistency. Together, these perspectives provide a unified view of how reasoning scales from individual agents to adaptive, collaborative intelligence.

5.1 Role Taxonomy of Multi-Agent Systems (MAS)

In this subsection, we first summarize the generic roles that often appear in a multi-agent system (MAS). Then, we introduce the specific functions of different roles when an MAS is applied in different domains, such as software engineering, finance, legal activities, education, healthcare, biomedicine, and music applications.

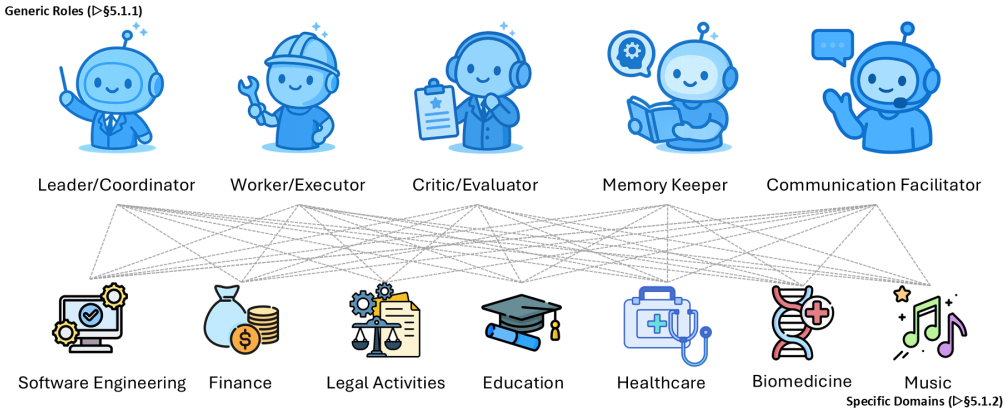

Figure 8: An overview of generic roles of agent and their specific domain adaptations in Section 5.1.

5.1.1 Generic Roles

- **Leader/Coordinator**: The leader, or coordinator, is responsible for maintaining high-level coherence within the system. This role involves setting global objectives, decomposing tasks into manageable subgoals, and assigning them to appropriate agents. In addition, the leader arbitrates conflicts that emerge between agents with overlapping or contradictory outputs. In practice, this role often manifests itself as a meta-controller that monitors the progress of other agents and ensures that execution adheres to an overarching plan.

- **Worker/Executor**: Executors, often called workers, are the operational backbone of MAS. They engage in concrete actions such as invoking external tools, writing or executing code, retrieving documents, or interfacing with the environment. Although they typically act under the directives of a leader, well-designed systems allow for adaptive autonomy, where executors can refine or optimize their assigned tasks when new local information becomes available.

- **Critic/Evaluator**: The critic/evaluator role centers on quality assurance. This role includes verifying correctness, testing hypotheses, red-teaming responses, and surfacing potential risks. In LLM-based systems, this often corresponds to *LLM-as-a-judge* setups, where dedicated evaluators assess the factuality, safety, or stylistic alignment of output. Critic roles help introduce checks and balances into otherwise generative workflows, thereby mitigating error propagation.

- **Memory Keeper**: Effective MAS requires persistent memory to accumulate context, prevent repetitive failures, and enable learning across episodes. The memory keeper curates and maintains long-term knowledge structures such as episodic logs, semantic embeddings, retrieval indices, or knowledge graphs.

By abstracting memory management into a dedicated role, the system can better balance short-term reactivity with long-term continuity and adaptation.

- **Communication Facilitator**: Communication overhead can easily undermine MAS efficiency. This role governs protocols for inter-agent exchange, including defining message schemas, managing communication bandwidth, enforcing gating mechanisms, and orchestrating consensus-building. By reducing ambiguity and ensuring structured information flow, the communication facilitator prevents bottlenecks and coordination failures in large-scale or heterogeneous agent populations.

5.1.2 Domain-Specific Roles

Beyond generic agent roles, domain-specific tasks often require specialized functions. These roles reflect professional practices in particular industries and map naturally onto MAS architectures.

Software Engineering: In software engineering, MAS generally maps onto roles that mirror the software development lifecycle: *architects*, *developers*, *code reviewers/testers*, *CI orchestrators*, and *release managers* (Hong et al., 2024; Qian et al., 2024a). The rationale is to distribute the responsibilities in a way that balances creativity, verification, automation, and governance, just as in industrial software practice.

- Architects define system-level design principles and establish structural blueprints.
- Developers translate these abstractions into concrete implementations.
- Code reviewers and testers safeguard reliability, checking correctness, maintainability, and functional coverage.
- CI orchestrators automate builds, testing, and artifact pipelines, reducing integration frictions.
- Finally, release managers oversee deployment, aligning new versions with milestones and safety protocols.

Previous work has demonstrated similar mappings, such as MetaGPT (Hong et al., 2024), which decomposes development into Product Manager, Architect, and Engineer agents. ChatDev (Qian et al., 2024a) further emphasizes communicative collaboration among specialized agents to support requirement analysis, coding, and testing. More recently, self-evolving collaboration networks have expanded this paradigm by enabling MAS to dynamically reorganize and optimize their roles throughout the software lifecycle (Hu et al., 2024c). A variant of MAS is also applied to the High-Performance Computing (HPC) domain (Kamatar et al., 2025) By structuring MAS around these stages, the architecture gains the same robustness and scalability as professional engineering workflows.

Finance: The financial domain can be roughly decomposed into four archetypal roles: *analysts*, *risk managers*, *traders/execution agents*, and *compliance officers* (LeBaron, 2006; Hull, 2018). This division reflects the established institutional design of financial organizations, where the responsibilities are segmented to balance profit generation with systemic stability.

- Analysts operate at different levels (e.g., fundamental, sentiment, or technical), each extracting distinct signals from raw market or textual data.
- Risk Managers then monitor portfolio exposure, apply stress tests, and enforce safeguards to prevent cascading vulnerabilities.
- Traders take responsibility for market interaction, while Execution agents ensure that orders are placed with speed and efficiency under liquidity constraints.
- Finally, Compliance roles ensure that activities remain aligned with regulatory requirements, enabling traceable decision-making and proper oversight.

Together, this layered ecology mirrors real-world financial institutions, where specialization and checks-and-balances are indispensable. Recent advances in MAS for finance mirror this layered ecology. R&D-Agent-Quant (Li et al., 2025l) demonstrates how agents can specialize in factor discovery and joint optimization for quantitative strategies. FinRobot (Yang et al., 2024a) provides an open source multi-agent platform tailored to financial applications, reflecting the practical need for modularity and scalability. PEER (Wang et al., 2024j) introduces expertization and tuning methods to adapt MAS to domain-specific responsibilities, while FinCon (Yu et al., 2024c) highlights the role of conceptual verbal reinforcement to enhance decision-making and compliance. Together, these works underscore how MAS can replicate the specialization, checks, and balances of real-world financial institutions.

Legal Activities: Multi-agent systems are also designed to model the collaborative and adversarial processes inherent in legal practice, with roles assigned to manage consultation, reasoning, and argumentation.

- For legal consultation, frameworks often simulate a law firm's structure with a *receptionist agent* for client intake, specialized *lawyer agents* for providing advice, a *secretary agent* for documentation, and a *boss agent* for quality control. In a consultation model, the *receptionist agent* first clarifies a user's query before routing it to the appropriate *lawyer agent*. After the multi-turn consultation, the *secretary agent* summarizes the interaction, and the *boss agent* provides an evaluation, ensuring a comprehensive and high-quality service (Sun et al., 2024a).
- For statutory reasoning, tasks are decomposed between *knowledge acquisition agents* that interpret legal texts and *knowledge application agents* that apply formalized rules to case facts. To be specific, in reasoning systems, the *knowledge acquisition agent* first builds a reusable ontology from legal statutes; then, the *knowledge application agent* uses this formal structure to analyze the specifics of a new case, ensuring consistent and transparent logic (Sadowski & Chudziak, 2025).
- To simulate courtroom dynamics, roles such as *judge*, *plaintiff*, *defendant*, and adversarial *lawyer agents* are created (Chen et al., 2024a). In courtroom simulations, adversarial *lawyer agents* engage in debate before a *judge agent*, reflecting on their performance after each trial to iteratively improve their argumentation strategies by updating their internal knowledge bases (Chen et al., 2024a).

Education: In education, MAS is being developed to provide personalized and adaptive learning experiences by distributing pedagogical functions among specialized agents.

- For personalized tutoring, a central *tutor agent* might engage a student using Socratic dialogue, while a *memory dispatcher* agent tracks the student's progress and misconceptions to adapt the difficulty and focus of the lesson in real-time (Chudziak & Kostka, 2025).
- For curriculum design, a pipeline of agents collaborates: a *research agent* gathers relevant information, a *planning agent* structures it into a coherent course, and other agents generate specific learning activities or assessments. Also, it can be modeled by an adversarial process, where *evaluator agent* critiques a lesson plan created by *generator agent*, and *optimizer agent* refines it based on feedback (Zhang et al., 2025p).

These systems demonstrate a shift towards creating intelligent, adaptive platforms that can support educators and provide students with more effective, engaging, and individualized learning journeys.

Healthcare: In the healthcare domain, multi-agent systems are structured to mirror clinical and research workflows, distributing complex tasks among specialized AI agents.

- For clinical diagnostics and consultation, these roles often include a *triage agent* (or *moderator*) for initial case assessment, various *specialist agents* (e.g., pathologists, neurologists), a *doctor agent* for patient interaction, and a *measurement agent* to provide test results (Kim et al., 2024b; Ghezloo et al., 2025). To be more specific, in the diagnostic setting, a *triage agent* first assesses the complexity of a case and routes it to the appropriate *specialist agents* for analysis. These specialists may then

engage in multiround discussions, with a *lead physician* agent synthesizing their opinions to reach a consensus. In addition to that, a *doctor agent* conducts a multi-turn dialogue with a *patient agent*, requesting specific data from a *measurement agent* to gather information dynamically.

- For autonomous research, roles are modeled after the scientific process, featuring a *meta agent* for strategic planning, an *executor* for running analyses, an *evaluator* for assessing outcomes, and a *reflector* for synthesizing knowledge (Zhu et al., 2025c). This division of labor allows for a systematic and comprehensive approach to multifaceted health challenges. Especially, the *meta agent* plans an experiment, the *executor* carries it out, the *evaluator* provides immediate feedback, and the *reflector* distills successful strategies into a persistent knowledge base, creating a self-improving cycle that enhances future planning.
- For public health events, ShortageSim (Cui et al., 2025) models FDA regulators, manufacturers, and healthcare buyers interacting under information asymmetry, enables counterfactual policy testing and evaluates how announcements and disruptions shape investment, stockpiling, and resolution timing against historical trajectories.

Other frameworks such as MMedAgent and MedAgent-Pro focus on orchestrating specialized medical tools, using a central agent to plan actions and aggregate results from various tool-based agents to handle multimodal data (Li et al., 2024b; Wang et al., 2025r).

Biomedicine: In biomedicine, particularly in drug and material discovery, MAS is designed to automate and accelerate the scientific process by assigning roles that reflect the iterative cycle of design, testing, and refinement.

For de novo molecule design, key roles include the *actor* (or *reasoner*) for generating novel structures, the *evaluator* for assessing chemical properties, and the *self-reflector* for refining future hypotheses based on results. To be specific, the *actor* agent proposes new candidates, which are then passed to the *evaluator* agent. The evaluator uses computational chemistry tools to calculate properties like binding affinity and synthetic accessibility, providing quantitative feedback (Averly et al., 2025). This feedback is then analyzed by the *self-reflector* agent to update the system's strategy for the next generation cycle, creating a feedback-driven process of optimization (Hu et al., 2025e).

Similarly, LIDDIA acts as a "digital chemist" with a *Reasoner*, *Executor*, *Evaluator*, and *Memory* component to navigate the drug discovery process and balance the exploration of new chemical spaces with the exploitation of promising candidates (Averly et al., 2025). To streamline the creation of machine learning workflows, DrugAgent uses an *LLM Planner* and an *LLM Instructor* to automate programming for tasks like ADMET prediction (Liu et al., 2024f). In genomics, GenoMAS orchestrates six specialized agents through a guided-planning framework to analyze complex gene expression data, integrating the reliability of structured workflows with the adaptability of autonomous agents (Liu et al., 2025c).

Music: In the creative domain of music composition, MAS is being explored to decompose the intricate process of creating music into collaborative, specialized roles. A system like ComposerX might feature a *conductor agent* that interprets a high-level user prompt and oversees the project, a *melody agent* that generates primary musical themes, a *harmony agent* that creates supporting chord progressions, and a *rhythm agent* that lays down the percussive and temporal foundation. These agents would interact iteratively, with the conductor agent synthesizing their outputs and providing feedback to ensure the different musical layers are coherent and aligned with the initial creative vision. This mirrors the collaborative process of a human orchestra or band, distributing creative responsibilities to achieve a complex and harmonious final product (Deng et al., 2024).

5.2 Collaboration and Division of Labor

Collaboration and division of labor constitute a central organizing principle in modern multi-agent systems. Instead of treating agents as homogeneous components, recent work emphasizes how responsibilities are decomposed and coordinated across specialized agents to improve efficiency and robustness. From this perspective, existing approaches can be broadly organized along two dimensions. **In-context collaboration**

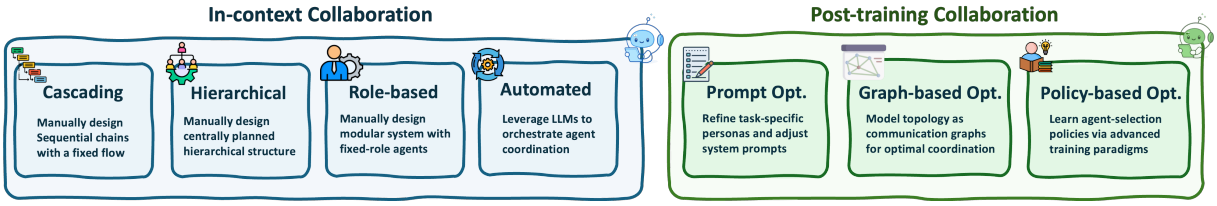

Figure 9: An overview of **Agentic Collaboration** in the multi-agent system, containing two parallel dimensions: in-context collaboration (training-free task-specific coordination design) and post-training collaboration (optimization-based automated workflow generation).

focuses on coordination strategies that are specified or induced at inference time without additional training. **Post-training collaboration** instead optimizes agent roles, interaction structures, or routing policies through learning or search. In addition, **agentic routing** can be viewed as a special case of this division of labor, where routing decisions explicitly offload cognition and computation to different agents based on task demands.

5.2.1 In-context Collaboration

In the design of multi-agent systems, several studies have observed that leveraging task-specific in-context information is often sufficient to build highly effective systems without the need for explicit training. Among these works, one line of research relies on manually crafted pipelines, where researchers design the agent interactions and workflows tailored to the target task. In contrast, another line explores LLM-driven automatic pipeline generation, allowing the model itself to construct and adapt the system's structure dynamically based on the task context.

Manually Crafted Pipelines. These approaches rely on predefined hierarchies or fixed collaboration workflows, where agent roles, execution order, and communication rules are determined before execution. Hierarchical systems such as AgentOrchestra (Zhang et al., 2026c), MetaGPT (Hong et al., 2024), and SurgRAW (Low et al., 2025) feature a central planner or conductor directing subordinate agents through structured subgoals. Cascading pipelines like Collab-RAG (Xu et al., 2025f), MA-RAG (Nguyen et al., 2025), Chain of Agents (Zhang et al., 2024l), and AutoAgents (Chen et al., 2023a) process information sequentially, passing intermediate outputs downstream with limited revision. Modular role-decomposed frameworks such as RAG-KG-IL (Yu & McQuade, 2025), SMoA (Li et al., 2024d), and MDocAgent (Han et al., 2025b) define fixed functional roles (e.g., retriever, reasoner, or vision agent) but allow minimal dynamic coordination. While these manually designed pipelines offer interpretability, modularity, and low execution complexity, their rigidity restricts adaptability to ambiguous or evolving reasoning tasks, motivating more flexible, reasoning-driven coordination mechanisms.

LLM-Driven Pipelines. This category leverages LLMs as orchestrators that decompose high-level goals into subgoals, route them to role-specialized agents or tools, and iteratively refine workflows based on intermediate feedback until completion. AutoML-Agent (Trirat et al., 2024) proposes a full-pipeline, orchestrator-led agent team that plans, assigns, and coordinates web/API/code tools through role-specialized micro-agents (e.g., coder/tester/runner), enabling end-to-end software development workflows. Magentic-One (Fourney et al., 2024) introduces a generalist multi-agent system where a central Orchestrator plans, tracks progress, and performs ledger-based routing over specialized agents (WebSurfer, FileSurfer, Coder, ComputerTerminal), achieving competitive results on GAIA (Mialon et al., 2023), AssistantBench (Yoran et al., 2024), and WebArena (Zhou et al., 2023b). MAS-GPT (Ye et al., 2025b) trains an LLM to emit executable MAS code conditioned on a user query, so a single forward pass generates a query-specific multi-agent workflow. MetaAgent (Zhang et al., 2025r) presents a finite-state-machine (FSM) abstraction to declare states, transitions, and tools, from which a LLM designer automatically constructs the MAS pipeline. AOP (Li et al., 2024a) formalizes orchestrator responsibilities and introduces three design principles, i.e., solvability, completeness, non-redundancy, and then operationalizes them with fast decomposition/assignment plus a reward-model evaluator.

Agent Routing. Closely related to LLM-driven orchestration, a line of work explicitly models **agent routing** as a decision layer that selects appropriate specialists for each query or subtask. For example, AgentRouter (Zhang et al., 2025w) proposes a knowledge-graph-guided router that leverages structured task semantics to dispatch questions to relevant agents, enabling effective collaborative question answering without modifying individual agents. Similarly, Talk to Right Specialists (Wu et al., 2025a) frames routing and planning as a unified inference-time process, where a controller dynamically assigns subtasks to domain-specialized agents based on intermediate reasoning states. These approaches highlight that agentic routing itself can be viewed as an inference-time realization of division of labor, where cognition is selectively offloaded to specialized agents.

Theory-of-Mind-Augmented Collaboration. Another interesting line of research is Theory of Mind (ToM), which refers to the ability of an agent to infer and reason about the beliefs, intentions, and mental states of other agents. Li et al. (2023b) first showed that equipping LLM agents with explicit belief-state representations in a cooperative text game improves both collaboration performance and the accuracy over ToM-free LLM baselines. Building on this, Hypothetical Minds (Cross et al., 2024) scaffolds ToM as a modular hypothesis-generation and refinement loop for other agents' strategies, while MindForge (Lică et al., 2024) extends ToM-aware reasoning to embodied collaborative learning. In parallel, Wu et al. (2025j) provides a mechanistic account of how LLMs encode ToM, identifying sparse parameter patterns whose perturbation selectively disrupts social reasoning. Pushing toward, ToM-agent (Yang et al., 2025a) augments LLM generative agents with counterfactual reflection over counterparts' beliefs and BeliefNet (Sagara et al., 2025) offers a ToM-centric joint-action simulator where embodied agents act based on nested belief states.

5.2.2 Post-training Collaboration

In multi-agent systems, the design of agent prompts (or personas) and the interaction topology plays a critical role in determining the system's ability to solve complex tasks. Recently, optimizing these components during the post-training phase has emerged as an important research direction. Based on the optimization objective, existing studies can be broadly categorized into two lines of work: prompt optimization and topology optimization.

Multi-agent Prompt Optimization. Prompt optimization in multi-agent systems focuses on how agent roles, workflows, and feedback are encoded in prompts to yield reliable coordination and stronger task performance. For example, AutoAgents (Chen et al., 2023a) extends prompt optimization from single-agent contexts to multi-agent teams, refining role specialization and execution plans through structured dialogue among meta-agents. SPP (Wang et al., 2024k) introduces a cognitive synergist that dynamically selects multiple personas during multi-agent collaboration for knowledge-intensive and reasoning-intensive tasks, enabling complementary expertise to emerge. DSPy Assertions (Singhvi et al., 2023) introduces LM Assertions that can be either hard (Assert) or soft (Suggest). When violated, these assertions trigger backtracking and prompt revision using erroneous outputs and error traces. During compilation, the mechanism bootstraps examples and counterexamples to reinforce few-shot prompts, which improves both recall and accuracy. MASS (Zhou et al., 2025a) demonstrates that prompts are often the dominant factor in MAS performance, and further applies automatic prompt optimization (Pryzant et al., 2023) by incorporating local and global topology information to refine each agent's prompt in a fine-grained manner.

As for topology optimization, two categories of research have emerged, each pursuing relatively independent optimization pathways. The first category of work treats the multi-agent topology as a communication graph, leveraging graph-based methods to identify an optimal structure that achieves strong performance under constrained communication costs (i.e., limited graph size). The second category adopts a policy-based perspective, where variable training paradigms are employed to learn an agent-selection policy with specially designed rewards or supervision signals. Through iterative, policy-based selection of subsequent agents, these approaches aim to progressively construct topologies that yield optimal overall performance. We discuss these two categories of approaches in greater detail in the following paragraphs.

Graph-based Topology Generation. A large body of work models multi-agent systems (MAS) as graphs where agents are nodes, and inter-agent communication forms edges. Then MAS design becomes a problem of

learning the communication/coordination topology. These works could be roughly divided into three groups as follows.

Graph generation. These methods aim to construct communication topologies from scratch by adaptively generating task-conditioned graphs. GommFormer (Hu et al., 2024b) uses an encoder-decoder framework to learn the communication graph via continuous relaxation of the graph representation, optimizing topology end-to-end under bandwidth constraints. G-designer (Zhang et al., 2024b) starts from a task-anchored network with a virtual task node, then uses a variational graph auto-encoder to decode a query-adaptive communication graph. MCGD (Zeng et al.) builds a sparse coordination graph with continuous node and discrete edge attributes, and performs categorical diffusion on edges and anisotropic diffusion on actions to capture structure diversity.

Graph pruning. These works start from dense collaboration graphs and aim to prune them into compact, task-appropriate pipelines while preserving utility and lowering token and compute costs. For example, AgentPrune (Zhang et al., 2024a) first formulates the MAS problem as a spatial-temporal graph sparsification problem, and then applies one-shot magnitude pruning to learn a sparse and effective pipeline. AGP (Li et al., 2025a) learns a dual-pruning policy, i.e., soft-pruning on edges and hard-pruning on nodes, to acquire a per-query topology. G-Safeguard (Wang et al., 2025i) introduces pruning as a security mechanism. It treats communication edges as the search space, employs a graph neural network to identify risky nodes, and applies deterministic rules to prune their outward edges based on a model-driven threshold, thereby defending the system against adversarial attacks.

Topology search. This line of research explores the graph space by searching over agentic operators and communication edges to identify effective pipelines. Specifically, AFlow (Zhang et al., 2024f) automates multi-agent workflow design with Monte-Carlo Tree Search over a fixed library of operators. MASS pre-defines some influential graph motifs, such as debating and tool-using, and then implements topology search inside this pruned motif subset. Then MASS (Zhou et al., 2025a) performs a prompt search on that topology to maximize performance. MaAS (Zhang et al., 2025g) replaces single-graph search with a probabilistic "agentic supernet" over layered operator choices and uses a controller to sample a query-conditioned subgraph. DynaSwarm (Leong & Wu, 2025) broadens the design space from a single optimized communication graph to a portfolio of candidate structures. It employs Actor–Critic (A2C) optimization to refine this portfolio and introduces a lightweight graph selector that chooses the most suitable topology for each instance. GPTSwarm (Zhuge et al., 2024a) formulates the search space as inter-agent connections within a computational graph. It relaxes the discrete topology into continuous edge probabilities and leverages reinforcement learning to optimize the resulting connection schemes, thereby enabling flexible and adaptive graph structures.

Policy-based Topology Generation. A growing line of research strengthens multi-agent pipeline generation by learning the policy of selecting subsequent agents with advanced training paradigms such as supervised fine-tuning (SFT), and reinforcement learning (RL). These approaches embed auxiliary signals into the optimization process, enabling agents to acquire stronger reasoning skills and more reliable coordination. Routing can be viewed as a special case of collaboration, in which a router conditions on task state and system context to learn a policy for selecting agents that maximize efficiency and performance (Yue et al., 2025; Liu et al., 2025d; Qian et al., 2025b; Wang et al., 2025d). Broadly, these methods can be grouped into three categories based on the signal type they inject into learning.

Relative-advantage policy learning. Several approaches rely on critic-free objectives to form advantages, thereby avoiding centralized value models and providing effective guidance to optimize policy. For example, MAGRPO (Liu et al., 2026c) proposes a Dec-POMDP formulation for LLM collaboration and replaces centralized critics with a group-relative advantage signal, enabling decentralized training/execution at dialog-turn granularity. MHGPO (Chen et al., 2025a) extends GRPO-style signals to heterogeneous groups: it jointly optimizes different agent roles via a shared group-relative objective, and introduces practical sampling/optimization tweaks. COPY (Ma et al., 2024b) utilizes two-agent co-training framework with shared rewards and KL regularization (to a frozen ref and cross-agent policies), improving stability and transfer between pioneer/observer roles on reasoning tasks.

LLM-generated prior guidance. Other methods leverage LLMs to generate rewards or priors for learning. Specifically, LGC-MARL (Jia et al., 2025b) uses an LLM to propose a Reward Function Generator (RFG) that turns natural-language objectives into structured reward terms. LAMARL (Zhu et al., 2025a) lets an LLM synthesize a prior policy and a task-specific reward function, then fine-tunes agents with RL. MAPoRL (Park et al., 2025) defines rewards as weighted sums of LLM verifier scores on current and future turns, then updates policies with multi-agent PPO. COPPER (Bo et al., 2024) learns a shared reflector with a counterfactual-PPO pipeline in which a learned reward model scores each agent's reflection by its marginal contribution to task improvement. SIRIUS (Zhao et al., 2025f) builds an experience library by retaining trajectories that lead to successful outcomes and augmenting failures, while a Judgment–Critic–Actor triad supplies LLM-generated correctness signals that filter and supervise subsequent fine-tuning across reasoning tasks. Multiagent Finetuning (Subramaniam et al., 2025) bootstraps reasoning by running multi-agent debates among generator LLMs and using LLM critics plus majority voting to produce self-generated supervisory signals, then fine-tunes role-specialized agents on critic-selected trajectories to improve both accuracy and diversity.

Human preference signals. This line of research replaces or augments environment rewards with human-derived feedback to align behavior with human intent, in both online and offline regimes. For instance, M3HF (Wang et al., 2025q) organizes human input into multi-phase feedback (e.g., scalar ratings, pairwise comparisons, and natural-language rationales) processed by LLMs into reward shaping signals. O-MAPL (Bui et al., 2025) introduces an end-to-end preference-based learning framework and directly learns Q-values from offline preference data, bypassing the two-stage reward-model-then-RL pipeline.

5.3 Multi-Agent Evolution

While self-evolving agents enable individual models to continuously improve through interaction and feedback, many real-world applications require collective intelligence supported by cooperation among multiple agents. Therefore, recent studies extend self-evolution from single-agent settings including planning, tool-use, and search evolution (Shinn et al., 2023; Sun et al., 2023; Madaan et al., 2023; Wang et al., 2025n; 2023a) to multi-agent co-evolution, where adaptation emerges across distributed agents (Zhang et al., 2024f; Ben Abdessalem et al., 2016; Wan et al., 2025; Feng et al., 2025b; Ke et al., 2025b). Beyond evolving model parameters, memory, prompts, and tools (Chhikara et al., 2025; Zhao et al., 2024a; Khattab et al., 2023; Qiu et al., 2025), multi-agent evolution further targets shared memory, communication mechanisms, and collaboration protocols (Zhou et al., 2025a; Ke et al., 2025b; Wang et al., 2024o).

As a result, multi-agent memory must jointly evolve along architecture, topology, content, and management dimensions, supported by hierarchy-structured, role-aware architectures (Zhang et al., 2025e), governed and distributed storage topologies (Xu et al., 2025b; Rezazadeh et al., 2025), modular and task-structured memory contents (Wang & Chen, 2025; Han et al., 2025a), and active management mechanisms for compression, verification, and continual updating (Kaiya et al., 2023; Tang et al., 2025b) to ensure coherent and scalable collaboration.

The goal thus shifts from optimizing a single agent's capability to improving the collective performance of multiple agents on complex, long-horizon tasks (Zhang et al., 2024f; Ben Abdessalem et al., 2016; Wan et al., 2025; Novikov et al., 2025).

5.3.1 From Single-Agent Evolution to Multi-Agent Evolution

While the shift from single-agent evolution to multi-agent co-evolution broadens the spatial dimension of adaptation from an individual model to a collective, the temporal dimension of evolution remains equally crucial. Beyond determining who evolves (a single agent or a population), recent studies also investigate when and how fast agents should adapt during interaction. This perspective leads to a complementary axis of analysis that distinguishes short-horizon, within-episode updates from long-term, cross-episode improvements, commonly referred to as intra-test-time evolution and inter-test-time evolution. We summarize these temporal modes of self-evolving behavior.

Intra-test-time evolution refers to the ability of agents to adapt and improve during task execution, enabling them to correct failures and refine strategies on the fly when facing unseen states or unexpected feedback. Unlike static inference pipelines, this paradigm embeds self-reflection, dynamic planning, memory rewriting, or even localized fine-tuning into the execution loop. Representative works leverage natural-language self-critique (Shinn et al., 2023; Madaan et al., 2023) and runtime adaptive planning (Sun et al., 2023; Hua et al., 2024) to generate corrective signals without external supervision. Reflexion (Shinn et al., 2023) allows agents to store distilled reflective feedback for immediate behavior improvement, while AdaPlanner (Sun et al., 2023) dynamically revises and replans mid-trajectory based on environmental mismatch detection. Beyond contextual adaptation, methods such as test-time supervised updating (Zweiger et al., 2025) and test-time reinforcement learning (TTRL) (Zuo et al., 2025; Simonds & Yoshiyama, 2025) directly modify model behavior when encountering difficult cases, often through problem-variant generation and targeted optimization. These approaches demonstrate that performance at inference time can improve within a single episode, forming short-horizon adaptation loops where the model learns while solving, rather than merely executing a fixed policy.

Inter-test-time evolution extends the self-improving process to across-task learning, where adaptations made in one task can be consolidated and transferred to future tasks. This enables the accumulation of persistent, generalizable capabilities over a lifelong interaction stream. A prominent paradigm involves offline self-distillation, where the agent generates responses and then refines them via self-evaluation before using them for supervised fine-tuning-such as in SELF (Lu et al., 2023a), STaR (Zelikman et al., 2022), and Quiet-STaR (Ferrag et al., 2025). These methods turn incorrect initial reasoning into high-quality labeled data for future performance gains. Additionally, online reinforcement learning frameworks such as RAGEN (Wang et al., 2025n) and DYSTIL (Wang et al., 2025b) continuously update policies based on dense interaction feedback, allowing agents to gradually internalize complex decision-making strategies over long horizons. Inter-test-time evolution can also incorporate curriculum mechanisms that automatically adjust task difficulty and environment complexity (Qi et al., 2024; Lee et al., 2024b), as well as experience structuring via memory evolution to preserve accumulated reasoning heuristics (Liang et al., 2025a; Salama et al., 2025; Wang et al., 2024o). This temporal mode focuses on stable long-term improvement, transforming short-lived corrections from individual tasks into continual competence growth across diverse task distributions.

To support these new capabilities, mechanisms evolve from individual reward-based or reflective adaptation (Lu et al., 2023a; Kumar et al., 2024; Yuksekgonul et al., 2024) to multi-agent reinforcement learning and game-theoretic co-optimization (Wan et al., 2025; Feng et al., 2025b), enabling collaborative structures to self-organize under evolving task requirements. Moreover, memory-driven multi-agent evolution (e.g., shared workflow memory or knowledge graphs) helps maintain accumulative group intelligence across episodes (Wang et al., 2024o; Li et al., 2025m). Overall, multi-agent evolution transforms isolated self-improvement loops into adaptive intelligent ecosystems capable of self-correction, self-organization, and social learning. This transition marks a critical step toward artificial collective intelligence, where cooperative dynamics drive continuous progress beyond the capabilities of any individual agent (Zhou et al., 2025a; Zhang et al., 2024f; Ben Abdessalem et al., 2016; Zou et al., 2025e).

5.3.2 Multi-agent Memory Management for Evolution

Multi-agent LLM systems pose unique challenges for memory design compared with single-agent settings. Beyond maintaining an individual agent's local context, they must capture inter-agent interactions, track roles and dependencies over time, and preserve both shared and private knowledge coherently. Memory must also remain scalable as collaboration grows and interactions accumulate. To provide a clearer understanding of this landscape, **we categorize existing approaches along four key dimensions**: (1) *architecture*, how memory is organized within and across agents; (2) *topology*, whether it is centralized, distributed, or hybrid; (3) *content*, the type and structure of stored knowledge; and (4) *management*, how memory is written, retrieved, and updated over time. Illustrations are shown in Figure 10.

Architecture Dimension: Hierarchical and Heterogeneous Designs. Recent work highlighted that prevailing multi-agent memory mechanisms were overly simplistic and lacked per-agent customization (Zhang et al., 2025e). To address this, G-Memory constructs a three-tier graph hierarchy (insight, query, interaction

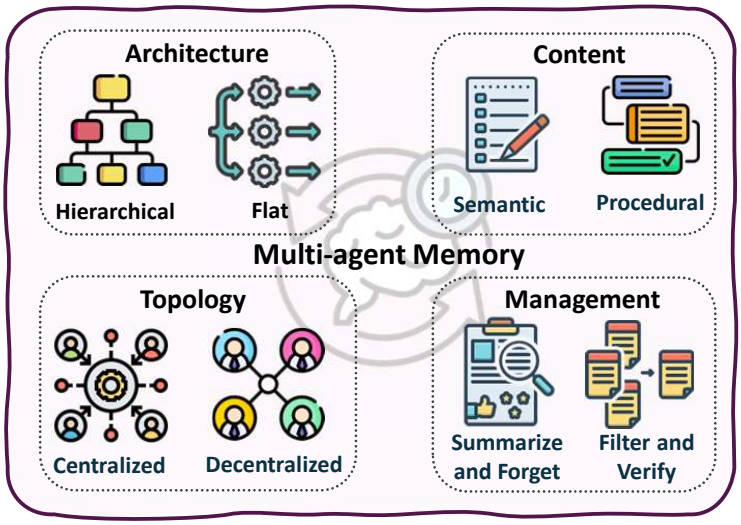

Figure 10: **Four dimensions of multi-agent memory design.** The framework includes (1) **Architecture**, how memory is structured; (2) **Topology**, where it is stored and shared; (3) **Content**, what type of knowledge is stored; and (4) **Management**, how it is maintained and updated.

graphs) that separates high-level generalizable insights from fine-grained execution traces. This hierarchical approach enables bi-directional memory traversal for retrieving both abstract lessons and concrete precedents across episodes. However, instead of global aggregation, Intrinsic Memory Agents adopts an opposing strategy by maintaining dedicated role-aligned memory templates for each agent (Yuen et al., 2025). This heterogeneous approach preserves specialized perspectives on collaborative planning benchmarks by reducing irrelevant information per agent. Recent work further explores hybrid strategies, with some systems employing adaptive hierarchical knowledge graphs in decentralized architectures that allow agents to reason over past interactions and share only relevant information rather than raw experiences (Yang et al., 2025c). These contrasting approaches reveal a fundamental trade-off: hierarchical designs optimize for global coherence and cross-episode learning, while heterogeneous designs optimize for role fidelity and computational efficiency.

Storage Topology and Memory Governance. Systems employ different topologies to balance scalability, privacy, and coherence, each reflecting different assumptions about trust and coordination. SEDM (Self-Evolving Distributed Memory) (Xu et al., 2025b) tackles memory management by turning memory into an active, self-optimizing component through verifiable write admission (via reproducible replay) and utility-based consolidation. This centralized approach with verification gates ensures that only factual or useful information enters the repository and performs cross-domain knowledge diffusion to enable transfer across heterogeneous tasks. In contrast, when privacy and organizational boundaries matter, Collaborative Memory (Rezazadeh et al., 2025) distinguishes private versus shared memory fragments using bipartite graph policies. Every entry carries immutable provenance (source agent, accessed resources, timestamp), enabling compliance auditing and safe cross-agent knowledge transfer in federated systems. At the other end of the spectrum, some systems like Memory Sharing (Gao & Zhang, 2026) adopt uncontrolled pooling where all agents freely exchange experiences in a shared memory pool. Research shows that memory sharing among LLM agents leads to a more diverse collective memory pool, which improved performance on open-ended tasks by creating emergent collective intelligence. These three topologies represent increasing levels of formality and control, reflecting different priorities for managing the trade-off between knowledge diversity and verification rigor.

Memory Content: Semantic, Task, and Cognitive-Phase Decomposition. Different content decomposition strategies suit different task characteristics, and the choice of content structure fundamentally shapes how agents interact with memory. MIRIX (Wang & Chen, 2025) pioneered *semantic decomposition* by defining six specialized memory types (Core, Episodic, Semantic, Procedural, Resource, Knowledge Vault) managed by distinct agents, achieving a 35% accuracy gain on multimodal QA tasks while reducing storage through flexible routing. Building on this modular principle, LEGOMem (Han et al., 2025a) instead

employs *task-based decomposition*, breaking execution traces into reusable memory units flexibly assigned to either central planners or specialist task agents. This design shows that orchestrator memory improves task decomposition and delegation, while agent memory enhances subtask execution, effectively narrowing performance gaps between small and large LLM teams. Recently, MAPLE introduced Cognitive-phase Decomposition (Bai et al., 2025), using specialized agents (Solver, Checker, Reflector, Archiver) to enable systematic error detection and plan repair cycles. The Reflector diagnoses errors after each episode, and the Archiver stores refined plans to avoid repeated mistakes, supporting feedback-driven learning. These three content decomposition strategies reveal that memory design should align with task structure: semantic content for heterogeneous information, task-based for workflow automation, and cognitive-phase for error-sensitive reasoning.

Memory Management Strategies. Effective long-term memory requires active management balancing relevance, efficiency, and coherence through different approaches that trade off simplicity against sophistication. Lyfe Agents (Kaiya et al., 2023) pioneered the forgetting-based approach using Summarize-and-Forget mechanisms to regularly compress memory, retaining only critical context. This strategy is suitable when storage is severely constrained, though it risks losing nuanced details for edge cases. To improve upon simple forgetting, AGENT-KB (Tang et al., 2025b) introduced more sophisticated management by organizing procedural traces into structured (entity, action, observation) triples and learning pattern abstractions reusable across tasks. Agents collaborate to retrieve, update, and reason over memory segments, enabling generalization without explicit retraining while central coordination ensures long-term consistency for scalable embodied planning. The choice among these strategies depends on system priorities: forgetting prioritizes storage efficiency, verification prioritizes reliability, and learning-based approaches prioritize adaptability. Production systems typically combine strategies, e.g., verification for critical memories and forgetting for low-utility peripheral information, to balance multiple objectives.

Discussions. Despite substantial progress, multi-agent memory systems remain largely unexplored with respect to post-training and model adaptation. Current approaches focus primarily on memory organization and retrieval for pre-trained models, with little investigation into how multiple agents can jointly optimize their memories through post-training procedures such as reinforcement learning or supervised fine-tuning. This represents a notable gap: while post-training techniques have been actively explored for single-agent memory systems, extending them to enable multi-agent teams to co-evolve their memory structures and management policies remains an open problem.

5.3.3 Training Multi-agent to Evolve

Recent advancements have shifted multi-agent systems from fixed, hand-designed coordination toward training paradigms that enable agents to evolve over time (Ma et al., 2024b; Chen et al., 2025g; Zhao et al., 2025f). Training multi-agent systems to evolve represents a critical step toward realizing adaptive, long-horizon intelligence beyond static coordination. In this emerging paradigm, agents improve collectively through interaction, feedback, and shared memory, rather than isolated or independently optimized behaviors. By embedding reasoning into the learning loop, via reinforcement learning (Liao et al., 2025), self-play (Zhao et al., 2025g), curriculum evolution (Park et al., 2025), and verifier-driven feedback (Zhang et al., 2025m), multi-agent systems can internalize coordination strategies, address inter-agent credit assignment, and progressively refine divisions of labor. This evolution transforms multi-agent reasoning from a static ensemble of cooperating LLMs into a self-improving organization that adapts its structure, communication patterns, and policies in response to task complexity and environmental change (Zhao et al., 2023a).

Co-evolution via Interaction and Intrinsic Feedback. A growing body of work has operationalized multi-agent evolution through explicit training objectives that couple interaction, feedback, and role specialization. For instance, Multi-Agent Evolve (Chen et al., 2025g) instantiates a closed-loop co-evolution framework containing three interacting roles (*Proposer*, *Solver*, and *Judge*), all of which are derived from a shared LLM backbone and jointly optimized via reinforcement learning. This forms a self-improving curriculum that enables collective skill growth without external supervision. In a related spirit, CoMAS (Xue et al., 2025b) emphasizes intrinsic interaction rewards, extracting learning signals directly from multi-agent

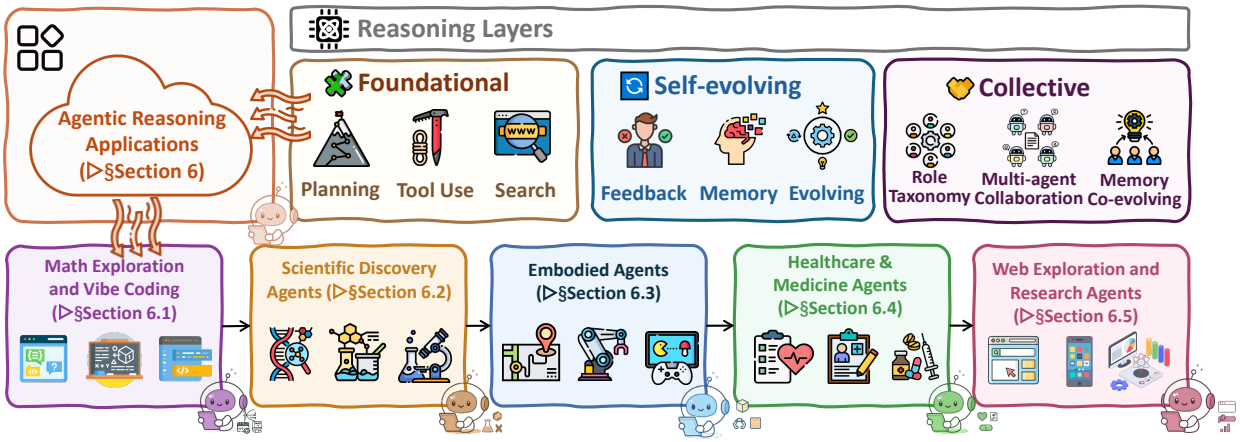

Figure 11: An overview of the applications of agentic reasoning.

discussion dynamics through an LLM-based judge, thereby enabling decentralized co-evolution driven purely by collaborative interaction.

Multi-Agent Reinforcement Fine-Tuning for Collective Adaptation. Additional works have focused on principled reinforcement fine-tuning frameworks tailored to LLM-absed multi-agent systems. For example, MARFT (Liao et al., 2025) formalizes multi-agent reinforcement fine-tuning by highlighting key mismatches between classical MARL assumptions and LLM-based agent organizations, such as role heterogeneity, dynamic coordination, and long-horizon dialogue, and provides a systematic framework for stabilizing collective post-training. Stronger-MAS (Zhao et al., 2025g) further adapts on-policy reinforcement learning to multi-role, multi-turn settings by introducing agent- and turn-wise grouping strategies that extend GRPO-style optimization, enabling more effective coordination learning across complex agent workflows. Similarly, MAPoRL (Park et al., 2025) proposes multi-agent post-co-training, where multiple LLMs are jointly optimized using a collaboration-aware verifier that rewards not only final outcomes but also the quality of intermediate discussions, encouraging the emergence of transferable communication strategies.

Role Specialization and Joint Credit Assignment. Other approaches have explored structured role specialization and joint credit assignment. MALT (Motwani et al., 2025) trains sequential pipelines of heterogeneous agents using trajectory expansion and outcome-based reinforcement signals, allowing each agent to improve its specialized function while optimizing end-to-end collaborative performance. MARS (Chen et al., 2026a) extends this idea to long-horizon research settings by jointly training complementary System 1 (fast, intuitive) and System 2 (deliberate, tool-using) agents via multi-agent reinforcement learning, enabling adaptive division of labor under complex tool interactions.

Preference- and Alignment-Driven Multi-Agent Evolution. Finally, another line of work has studied evolution under preference- and alignment-driven objectives. Preference-based multi-agent reinforcement learning (Zhang et al., 2025m) studies how collective policies and equilibria can be learned from preference-only feedback, addressing data coverage and stability challenges inherent in multi-agent settings. From a safety perspective, Alignment Waltz (Zhang et al., 2025j) frames alignment as a cooperative co-evolution process between a generation agent and a feedback agent, where evolving guidance enables the system to iteratively refine unsafe or unhelpful behaviors. Collectively, these methods demonstrate how embedding reinforcement learning, co-evolution, and verifier-driven feedback into multi-agent training enables LLM-based systems to evolve from static collaborations into adaptive, self-improving organizations.

6 Applications

Building on the established three-layer taxonomy (i.e. foundational, self-evolving and collective reasoning) mentioned in previous sections, we now examine how these capabilities manifest across real-world applications. This section surveys representative reasoning-empowered agentic systems across several key domains, as illustrated in Figure 11, including math exploration and vibe coding (Section 6.1), scientific discovery (Section 6.2), robotics (Section 6.3), healthcare (Section 6.4), and autonomous web exploration and research (Section 6.5). Specifically, each domain exhibits distinctive forms of reasoning, influenced by its data modalities and environmental constraints. Accordingly, our discussion in each subsection is organized around three layers: (1) **core abilities** such as planning, tool use and search that span scientific hypothesis generation, embodied control, medical reasoning, automated experimentation and symbolic problem solving, for example; (2) **self-evolving abilities** that integrate feedback, reflection and memory modules which refine domain-specific competence through iterative experiment loops, lifelong skill learning and clinical adaptation; and (3) **collective multi-agent reasoning** that enables collaboration and specialization from cooperative scientific assistants to coordinated robotic teams, diagnostic ensembles or multi-aspect experts. This section highlights how agentic reasoning frameworks adapt to domain-specific knowledge structures and tasks, illustrating the transition from traditional LLM reasoning to goal-directed, domain-aware and active agentic intelligence.

6.1 Math Exploration & Vibe Coding Agents

Mathematics and code have traditionally served as two of the most widely used domains for evaluating reasoning in artificial intelligence, as both require structured symbolic manipulation and precise multi-step deduction. Traditional benchmark-driven evaluation in these domains is showing clear limitations. Widely used math datasets such as GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), and AIME (Mathematical Association of America) are increasingly saturated, which makes it difficult to distinguish among modern high-performing models. The problems in these datasets often rely on a small set of recurring techniques and do not require the sustained and exploratory reasoning needed to assess more advanced mathematical capabilities. Even recent evaluations such as FrontierMath (Glazer et al., 2024) continue to emphasize final-answer accuracy, which offers only a partial view of an agent's reasoning process and its ability to adjust strategies during problem solving.

Under the agentic reasoning paradigm, however, both areas are undergoing a substantial shift from static problem solving to dynamic processes that emphasize exploration, adaptation, and collaboration. In mathematics, recent systems (Novikov et al., 2025; Luong et al., 2025; Trinh et al., 2024; Romera-Paredes et al., 2024) demonstrate that agents can engage in competition-level reasoning, building on the success of LLMs in coding tasks. Work in foundational mathematics (Swirszcz et al., 2025; Georgiev et al., 2025) further shows that agents can search for new problems, propose conjectures, construct auxiliary lemmas, and explore deeper structures in mathematical concepts. These developments position mathematics not merely as an evaluation benchmark but as a domain of active *mathematical exploration*.

Large Language Models have also reshaped coding through the emerging workflow known as agentic coding and *vibe coding* (Karpathy, 2025; Willison, 2025). In this paradigm, the model acts as an interactive collaborator that engages in multi-turn natural-language dialogue. Users iteratively design and refine programs while the agent maintains context, adapts to evolving requirements, and continuously self-corrects. Modern tools such as Copilot[1] and Cursor[2] have further popularized this collaborative workflow, making interactive programming a common practice in real-world software development.

In this section, we organize our discussion according to the three-layer framework introduced earlier. The foundational layer (Section 6.1.1) concerns the core reasoning and execution skills: mathematical agents perform symbolic manipulations and step-by-step derivations across arithmetic, algebra, geometry, and calculus, while code agents carry out syntax-aware generation, implement functions, and verify correctness through interpreter or compiler feedback. The self-evolving layer (Section 6.1.2) introduces mechanisms for reflection and adaptation. Mathematical agents learn from intermediate reasoning traces to correct missteps

[1] https://github.com/features/copilot
[2] https://cursor.com

or explore alternative solution paths, and code agents iteratively debug, refine, and optimize implementations based on runtime feedback or test results. The collective layer (Section 6.1.3) focuses on collaboration, where agents exchange intermediate results, share reusable modules, and jointly develop complex proofs or codebases. Taken together, these layers reveal how mathematics and coding are becoming domains in which agentic reasoning enables increasingly creative and adaptive problem solving.

6.1.1 Foundational agentic reasoning

Planning. Explicit planning is widely recognized as a core mechanism for enhancing the structured reasoning capabilities of LLMs. In the domain of mathematical discovery, several systems exhibit structures that can be interpreted as forms of planning. In representation theory and knot theory, the system of Davies et al. (2021) guides human mathematicians by proposing intermediate objects and promising avenues of exploration, which function as high-level suggestions for organizing problem-solving workflows. In geometric reasoning, Trinh et al. (2024) solves Olympiad-level geometry problems by decomposing them into sequential stages of construction, lemma generation, and verification, yielding a structured multi-step process that resembles a planned reasoning trajectory. Program-search approaches (Romera-Paredes et al., 2024) iteratively refine candidate programs and mathematical structures, a procedure that naturally forms a coarse-to-fine exploration path. Large-scale exploration frameworks (Georgiev et al., 2025; Swirszcz et al., 2025) also operate through cycles of proposing, testing, and modifying conjectures or geometric objects, which collectively create a procedural structure aligned with planning. Efforts toward more robust mathematical reasoning (Luong et al., 2025) similarly rely on stepwise reasoning patterns, further reinforcing the presence of implicit planning dynamics. of implicit planning dynamics across mathematical agents.

In code agents, planning has likewise emerged as an essential component for organizing multi-step reasoning and enabling more structured decision-making. Early systems such as CodeChain (Le et al., 2024) and CodeAct (Wang et al., 2024h) introduce explicit planning or action spaces to support modular code construction, while KareCoder (Huang et al., 2024b) integrate external knowledge sources or domain-specific information into the planning process. Subsequent works explore more structured planning organizations, including multi-stage control flows (Bairi et al., 2024; Han & Lyu, 2025), tree-shaped planning structures (Li et al., 2024e; Ni et al., 2025b), and adaptive refinement mechanisms (Aggarwal et al., 2025). Planning has also been linked to improved exploration breadth: GIF-MCTS (Dainese et al., 2024) incorporates Monte Carlo Tree Search to explore multiple code-generation trajectories. Recent extensions demonstrate applicability in specialized domains such as hardware design, where VerilogCoder (Ho et al., 2025) employs graph-structured planning and waveform-based verification. To address environments where state serialization is difficult, Guided Search (Zainullina et al., 2025) introduces lookahead and trajectory selection strategies for evaluating candidate actions without full environment access.

Tool-Use. Integrating external computational tools with LLMs has become a central mechanism for extending the reasoning and generation capabilities of single-agent systems. A defining characteristic of many mathematical reasoning systems is their integration with external computational tools. Formal theorem-proving agents such as Thakur et al. (2024) operate directly within the Lean proof assistant, selecting tactics and interacting with the underlying prover through in-context guidance. Position papers on formal mathematical reasoning (Yang et al., 2024c) emphasize that progress in mathematical AI will depend on systems that can call theorem provers, satisfiability solvers, and computer algebra systems as part of a broader reasoning loop. Program-search frameworks for discovery (Romera-Paredes et al., 2024) rely on executing generated programs and employing symbolic routines for verification. Generative modelling approaches (Ellenberg et al., 2025) make use of computational number-theoretic tools to check and filter generated candidates. Geometry-focused systems (Trinh et al., 2024; AlphaProof and AlphaGeometry teams, 2024) integrate automated geometric solvers and checkers to validate constructions and derived relations. Recent mathematical reasoning agents also couple decomposition with code execution: DotaMath (Li et al., 2024c) uses code assistance and self-correction to make decomposed mathematical thoughts executable, while CoRT (Li et al., 2025b) embeds code-integrated reasoning inside the thinking process. Across these systems, external computational resources play a central role in enabling correct and scalable mathematical reasoning.

In code agents, external tools have similarly become crucial for extending the capabilities of LLM-based agents beyond pure text generation. Early work such as Toolformer (Schick et al., 2023) and ToolCoder (Zhang et al., 2023b) explored how models can learn to invoke APIs or search tools to obtain missing information during generation. Subsequent systems integrate increasingly rich toolchains: ToolGen (Wang et al., 2025h) leverages automatic completion tools to resolve undefined dependencies, while CodeAgent (Zhang et al., 2024h) incorporates multiple programming utilities including search, documentation reading, symbol navigation, and code execution to support more realistic software workflows. Several methods focus on improving tool-feedback loops, such as ROCODE (Jiang et al., 2025b), which combines real-time error detection with adaptive backtracking, and CodeTool (Lu et al., 2025e), which introduces process-level supervision to improve the reliability of tool invocation. Collectively, these systems show that tool integration provides essential external signals, via search results, documentation, static analysis, or execution feedback, that extend the reasoning and generation capabilities of single-agent LLMs.

Search and Retrieval. Search and retrieval has emerged as a complementary mechanism that enriches model contexts through external information sources. Search is a recurring mechanism in mathematical discovery. Program-search based systems (Romera-Paredes et al., 2024) treat mathematical discovery as navigating a program space in which candidate programs encode conjectures or structural hypotheses, with iterative filtering based on symbolic or numerical checks. Generative modelling approaches (Ellenberg et al., 2025) explore families of mathematical objects by sampling from flexible distributions that capture structural regularities. Geometric systems such as Trinh et al. (2024) and Swirszcz et al. (2025) search over constructions, configurations, and high-dimensional polytopes, guided by learned heuristics or structural constraints. Large-scale discovery frameworks (Georgiev et al., 2025) operate through repeated propose–test–refine cycles across conjectures, supporting wide exploration over mathematical landscapes. All these systems rely on systematic search procedures that structure the exploration of mathematical ideas.

In code generation, repository-level retrieval systems such as RepoHyper (Phan et al., 2024) locate reusable code segments from large-scale code bases to provide more informative contexts for generation. CodeNav (Gupta et al., 2024) dynamically indexes real repositories during generation, retrieving relevant functions and adjusting based on execution feedback. AUTOPATCH (Acharya et al., 2025) applies retrieval to performance optimization, combining historical code examples with control flow graph analysis for context-aware improvements. Structure-aware retrieval has also been explored: knowledge-graph-based repository representations (Athale & Vaddina, 2025) improve retrieval quality by capturing symbolic and relational structure, while cAST (Zhang et al., 2025s) introduces AST-based chunking to enhance syntactic coherence and retrieval granularity. These retrieval methods demonstrate how external knowledge sources can augment single-agent LLMs by providing high-quality, structured contexts that guide both understanding and generation.

6.1.2 Self-evolving agentic reasoning

Agentic Feedback and Reflection. Across mathematical and code reasoning tasks, feedback operates as an external signal that highlights discrepancies, confirms correct inferences, and directs the agent toward more reliable subsequent computations. Feedback mechanisms appear prominently across mathematical discovery systems. In program-search based discovery (Romera-Paredes et al., 2024), executing candidate programs and evaluating their outputs against constraints yields counterexamples or confirmations, enabling iterative refinement of conjectures. In geometry, automated checkers validate constructions and derived relationships (Trinh et al., 2024; AlphaProof and AlphaGeometry teams, 2024), providing correctness signals that guide subsequent revisions. Interactive evaluation frameworks (Collins et al., 2024) show that human clarifications and follow-up prompts expose reasoning errors and improve model responses. Position work on formal reasoning (Yang et al., 2024c) highlights verification, proof checking, and model checking as essential sources of structured feedback. Curriculum-based signals can also shape what an agent learns to inspect first; for example, CLCL (Zhou et al., 2023a) combines contrastive and curriculum learning to detect non-compositional expressions, illustrating how adaptive training schedules can expose brittle reasoning compositions. In several systems involving multiple candidate hypotheses (Romera-Paredes et al., 2024; Ellenberg et al., 2025; Georgiev et al., 2025; Ren et al., 2025), the use of verification signals to retain promising candidates functions analogously to a fitness-based evaluation step, since these signals determine which

hypotheses survive and which are discarded, thereby shaping the direction of subsequent exploration without introducing an explicit learning signal.

For code agents, feedback and reflection are central to improving reliability over multi-step reasoning. Fault-aware editing methods such as Self-Edit (Zhang et al., 2023a) incorporate execution-based signals to refine erroneous code, while Self-Repair (Olausson et al., 2024) integrates code and feedback models to diagnose test failures and propose targeted corrections. More structured systems like LeDeX (Jiang et al., 2024b) combine stepwise annotation, execution-driven verification, and automated repair into a closed-loop pipeline in which feedback continually informs the next revision. Reflection also functions as a form of memory: iterative self-improvement frameworks such as Self-Refine (Madaan et al., 2023), Self-Iteration (Chang et al., 2023), and Self-Debug (Chen et al., 2023d) reuse earlier drafts, analyses, and explanations to guide subsequent revisions, while artifact-level mechanisms such as CodeChain (Le et al., 2024) and LeDeX (Jiang et al., 2024b) retain reusable components, corrected snippets, and execution traces as persistent representations. Together, these approaches demonstrate how feedback—whether symbolic, execution-based, or self-generated—interacts with iterative memory to support structured refinement and long-horizon improvement in code-oriented agentic systems.

Memory. Memory provides agents with a mechanism for retaining and leveraging information from earlier reasoning steps, allowing them to maintain consistency, improve intermediate states, and improve their performance over extended problem-solving horizons. While few systems introduce an explicit memory module, many mathematical agents rely on forms of persistent state that can be viewed as implicit memory. Interactive evaluation frameworks (Collins et al., 2024) maintain conversational and problem-state context across multiple turns, allowing models to build upon earlier partial derivations. Formal-theorem-proving agents (Thakur et al., 2024) operate over evolving proof states in Lean, which accumulate tactics, subgoals, and intermediate lemmas, functioning as structured persistent information. Program-search and discovery systems (Romera-Paredes et al., 2024; Georgiev et al., 2025) retain conjecture histories, counterexamples, and successful constructions as part of their iterative refinement processes. Their role in preserving and reusing information across reasoning steps aligns with the broader notion of memory in agentic systems.

In code agents, memory increasingly takes the form of explicit structures that maintain coherence over long-horizon generation. Several systems construct shared or structured workspaces: Self-Collaboration (Dong et al., 2024a) introduces a blackboard memory for storing task descriptions, intermediate drafts, and revision records, enabling agents to coordinate through a common representation. Architectural approaches such as L2MAC (Holt et al., 2025) and Cogito (Li et al., 2025k) extend this idea by organizing context into dedicated registers, hierarchical memory units, or long-term knowledge stores, overcoming context-window limits and supporting multi-file or large-function reasoning. Across these designs, the underlying insight is consistent: effective code agents require persistent, structured, and often domain-aware memory that preserves intermediate reasoning and enables self-improvement across extended development trajectories.

6.1.3 Collective multi-agent reasoning

To address the growing complexity of tasks in mathematical discovery and code generation, recent systems increasingly rely on multi-agent or modular designs that decompose problems into cooperating specialized components. Mathematical discovery frameworks often organize reasoning into explicitly defined multi-agent or multi-component workflows that collaborate to explore and validate mathematical ideas. The polytope-generation system (Swirszcz et al., 2025) uses multiple specialized components that generate, evaluate, and refine geometric objects, forming a genuine collaborative workflow. Large-scale exploration frameworks (Georgiev et al., 2025) often divide discovery into modules for proposing conjectures, identifying counterexamples, and refining statements, which, although implemented within a unified system, mirror multi-agent role specialization. Early work on AI-assisted mathematical research (Davies et al., 2021) and Olympiad-level systems (AlphaProof and AlphaGeometry teams, 2024) also involve human–AI collaboration, where human mathematicians interact with AI systems in a complementary manner. These developments indicate that mathematical discovery is an inherently collaborative process, and multi-agent architectures provide a natural vehicle for expressing such collaboration in agentic systems.

Multi-agent systems for code generation have progressed from simple role-based pipelines to adaptive, collaborative frameworks capable of handling long-horizon software development. Early approaches such as Self-Collaboration (Dong et al., 2024a) and AgentCoder (Huang et al., 2023a) decompose tasks into sequential roles, while hierarchical designs like PairCoder (Zhang et al., 2024d) and FlowGen (Lin et al., 2025a) introduce an architecture in which high-level agents handle planning and lower-level agents carry out concrete implementation. Flexible systems such as SoA (Ishibashi & Nishimura, 2024) further adjust the number and specialization of agents in response to task complexity. Other frameworks, including MapCoder (Islam et al., 2024), AutoSafeCoder (Nunez et al., 2024), and QualityFlow (Hu et al., 2025c), rely on repeated cycles in which multiple agents generate, test, analyze, and repair code. Recent work explores self-evolving system structures, as in SEW (Liu et al., 2025f), which reorganizes collaboration pathways based on runtime feedback, and EvoMAC (Hu et al., 2024c), which adjusts agent strategies through an iterative text-based update mechanism. Collaborative optimization methods such as Lingma SWE-GPT (Ma et al., 2024d), CodeCoR (Pan et al., 2025), SyncMind (Guo et al., 2025b), and CANDOR (Xu et al., 2026b) explicitly improve cross-agent coordination. SWE-Chain (Lam et al., 2026) evaluates coding agents in a continuous software maintenance setting, where agents must perform sequential package upgrades and each release transition depends on the correctness of previously applied code changes. PBT-Bench (Jing et al., 2026) focuses on the testing capability of coding agents, assessing whether they can generate meaningful property-based tests that capture software invariants and expose potential implementation bugs. TEBench (Shang et al., 2026) evaluates coding agents on test evolution scenarios where tests may be broken, stale, or missing. VISTA (Guo et al., 2026a) evaluates coding agents on generating functional and visually coherent web apps from visual specifications. Multi-SWE-bench (Zan et al., 2025) extends repository-level issue resolution to multiple programming languages, providing a more diverse benchmark for evaluating coding agents beyond Python-centric SWE-bench. SWE-Lancer (Miserendino et al., 2025) evaluates coding agents on real freelance software engineering tasks from Upwork, measuring whether frontier models can solve economically valuable development work. Together, these systems show a clear shift toward multi-agent code generators that rely not only on role decomposition, but also on reflection, distributed evaluation, adaptive restructuring, and team-level optimization, transforming code generation into an increasingly coordinated and resilient problem-solving process.

6.2 Scientific Discovery Agents

Scientific-discovery agents aim to accelerate the entire life cycle for scientific research, from hypothesis generation through experimental execution, by coupling LLMs with domain-specific simulators, laboratory automation and up-to-date literature. These systems ground decision in verifiable processes while handling heterogeneous data, safety constraints and long-horizon goals.

In this subsection, we begin with the foundational layer (Section 6.2.1), which encompasses planning under scientific context, tool-augmented interaction with scientific resources, search and retrieval mechanisms including RAG-based systems and execution-time integration with laboratory hardware. Building upon these capabilities, the self-evolving layer (Section 6.2.2) introduces agentic memory, feedback and reflection, which enable scientific agents to refine hypotheses, adapt protocols and learn from experimental outcomes. Finally, the collective layer (Section 6.2.3) explores multi-agent collaboration, where agents coordinate roles, share intermediate knowledge and jointly reason toward complex scientific goals.

6.2.1 Foundational agentic reasoning

Planning. Scientific agents utilize reasoning-enhanced planning ability to decompose a research goal into steps, decides which tool or simulator to call next, then revises the plan as evidence arrives. In short, the *chain of thought* emerges from LLM reasoning that compiles instructions into rigorous executable plans (Wei et al., 2022). For example, ProtAgents (Ghafarollahi & Buehler, 2024b) materializes a planner agent that utilizes LLM reasoning capability to formulate a concrete plan for protein analysis and keep modifying it with feedback from another critic agent, and Eunomia (Ansari & Moosavi, 2024) uses ReAct-style (Yao et al., 2023c) workflow to make in-context reasoning: after retrieving a top-k evidence set, the backbone LLM quote a warranting sentence, and that citation drives the next action choice. Other examples include

MatExpert (Ding et al., 2024), which deploys a chain-of-thought LLM to author a stepwise transition pathway and then emits a structured crystal candidate from a feedback loop.

Planning can also act as reasoning constraint. For instance, Curie (Kon et al., 2025) utilizes a rigor engine to align, setup and do reproducibility check within planning steps proposed by the Architect LLM. Thus, the Architect's free-form reasoning cannot advance unless these rigor gates are satisfied, which transforms planning into both a guide and a regulator of the reasoning process. In addition, a general purpose biomedical agent, Biomni (Huang et al., 2025b) constrains its reasoning within a dynamically constructed biomedical action space of comprehensive tools, software packages and databases, requiring each hypothesis to be operationalized as executable code.

Tool-Use. Tool use is an important part of the reasoning loop for scientific agents nowadays. Specifically, rather than following rigid rules, these agents can decide which tool and when to call, how to fill parameters and verify or revise based on evidence. For example, SciAgent (Ma et al., 2024e) formalizes *tool-augmented reasoning* as a four-step procedure: planning, retrieval, tool-based action and execution. Agents are trained to decide when to call a tool, which one, and how to integrate it into solving scientific tasks. Through domain-specific tools, ChemCrow (Bran et al., 2023) chains various expert chemistry tools so intermediate calculations become premises in the next reasoning step, which enables end-to-end planning and autonomous syntheses. CACTUS (McNaughton et al., 2024) similarly grounds explanations in cheminformatics outputs, reducing reliance on free-form reasoning by language models alone.

Other notable examples include ChemToolAgent (Yu et al., 2024a) and CheMatAgent (Wu et al., 2025f). In particular, ChemToolAgent (Yu et al., 2024a) employs a ReAct-like (Yao et al., 2023c) architecture with multiple specialized chemistry tools, allowing the LLM to choose and parameterize tool calls while CheMatAgent (Wu et al., 2025f) pushes further by learning tool use: it integrates over 100 chemistry/materials tools, curates a tool-specific benchmark, and uses Monte Carlo Tree Search with step-level fine-tuning to learn both which tool to pick and how to fill arguments.

For biomedical agents, TxAgent (Gao et al., 2025d) scales therapeutic reasoning across 211 vetted tools and it carries out multi-step reasoning that reconciles drug labels, interactions, and patient context—turning clinical justification into an executable trace. On the other hand, AgentMD (Jin et al., 2025b) builds a two-stage tool memory: it first mines thousands of clinical calculators from literature (i.e. making tools), then selects and applies the right ones at inference (i.e. using tools), pinning predictions to concrete computations. Other recent systems (Lála et al., 2023; Skarlinski et al., 2024; Chiang et al., 2024; Zhang et al., 2024e; Qu et al., 2025b; Gao et al., 2025a; Zhao et al., 2025c; Du et al., 2026; Qu et al., 2026; Li et al., 2026b) reinforce similar design: co-design tool-use with reasoning so each claim is computable and auditable.

Another notable category of tool-use is the ability of agentic execution, which includes but not limited to run codes and simulate environments. Execution layers bridge high-level plans to physical infrastructure, which enables scientific agents to autonomously operate laboratory hardware, orchestrate simulation pipelines, and manage large-scale data workflows. Recent works such as Organa (Darvish et al., 2025) ties LLM reasoning to task-and-motion planning plus scheduling and perception, executing multi-step experiments with autonomous robots; AtomAgents (Ghafarollahi & Buehler, 2024a) exemplifies the simulation side of execution: a physics-aware system that plans and runs atomistic workflows, coordinating tools for code execution, analysis, and hypothesis checking; and Chemist-x (Chen et al., 2025c) shows wet-lab execution beyond a digital-only scenario, where agents generate control scripts and drive an automated platform to validate conditions without human intervention.

Several other platforms couple execution with optimization or team-based autonomy. For instance, SGA (Ma et al., 2024c) formalizes the workflow of *LLM-as-proposer and simulator-as-optimizer* while MatExpert (Ding et al., 2024) operates a *retrieval, transition and generation* workflow for material discovery tasks, and CellAgent (Xiao et al., 2024b) coordinates planner, executor and evaluator roles to run full single-cell analysis pipelines.

Search and retrieval. Beyond simple context stuffing, recent scientific agentic systems elevate retrieval into a deliberate reasoning step: agents decide when and what to fetch, and how to use the evidence before

committing to a hypothesis. With retrieval ability, BioDiscoveryAgent (Roohani et al., 2024) pulls literature and interim assay results inside a closed loop so the model's next gene-perturbation choices are conditioned on what was read and measured; while DrugAgent (Inoue et al., 2025) coordinates knowledge graph queries, targeted literature search through web API and machine learning predictors. Its planner selects retrieval actions and then reconciles heterogeneous evidence into an explainable rationale. To facilitate scientific research, ARIA (Ramirez-Medina et al., 2025) operationalizes a *search, filter then synthesis* workflow as role-bound steps that carry citations forward, turning literature into actionable procedures. Similarly, AI Scientist-v2 (Yamada et al., 2025) employs an agentic tree-search framework in which the agent actively queries scientific literature database during hypothesis formulation and manuscript drafting, ensuring that analyses and writing are grounded in existing evidence. For research idea generation, another recent work (Qi et al., 2023) constrains the process with curated background packets, using retrieval as an experimental control.

Building on these developments, retrieval-augmented generation (RAG) frameworks position external sources not merely as supporting references but as active components of the reasoning process. Specifically, RAG-enhanced scientific agents make external sources as primary inputs to LLM context and reasoning material, mostly with explicit planning, passage extraction, citation and contradiction checks. For example, PaperQA (Lála et al., 2023) and PaperQA2 (Skarlinski et al., 2024) treat retrieval as the main loop. By deciding which documents to read, attributing every claim, and detecting conflicts to steer synthesis, these works can yield expert-level literature reviews that are inherently verifiable. In material science, LLaMP (Chiang et al., 2024) extends RAG beyond text. Specifically, it utilizes hierarchical ReAct (Yao et al., 2023c) agents call material-specific APIs to fetch band gaps or elastic tensors, edit structures and then reason with computed properties.

6.2.2 Self-evolving agentic reasoning

Scientific discovery agents can go beyond static reasoning and acquire the ability to self-evolve, which is to learn from experience, refine their internal representations and improve decision quality over successive interactions. This self-evolving layer equips agents with mechanisms to monitor and revise their own reasoning, retain and reuse intermediate hypotheses and adjust future plans based on external feedback or environmental signals. In the following paragraphs, we discuss how memory modules enable the accumulation of scientific knowledge and how feedback and reflection mechanisms support continual adaptation and reasoning consistency throughout long-horizon scientific workflows.

Memory. ChemAgent (Tang et al., 2025a) implements a self-updating library. It decomposes chemistry problems into sub-tasks and writes reusable *skills* (ex: procedures, patterns, solutions) that later prompts can retrieve and adapt, stabilizing long multi-step reasoning without re-deriving everything from scratch. On the other hand, MatAgent (Takahara et al., 2025) emphasizes interpretable generation for inorganic materials, where *short-term memory* recalls recent compositions and feedback, *long-term memory* preserves successful designs together with their reasoning traces, and both are reused across iterations to guide proposal refinement and enable transparent audit.

Agentic Feedback and Reflection. Firstly, Scientific Generative Agent (Ma et al., 2024c) ties discrete LLM proposals to inner-loop simulations that optimize continuous parameters, advancing only when evidence improves. The reflection ability is driven by measurable loss reductions. Next, ChemReasoner (Sprueill et al., 2024) performs heuristic search over the LLM's idea space but scores and steers candidates with quantum-chemical feedback, turning electronic-structure signals into a principled critique of linguistic hypotheses. Complementing these physics-based signals, Curie (Kon et al., 2025) embeds rigor check directly into control flow via intra-agent checks, inter-agent gates and an experiment-knowledge module. In parallel, LLMatDesign (Jia et al., 2024) builds explicit self-reflection into materials workflows, prompting the agent to surface and repair inconsistencies before they propagate to tool calls. Moreover, NovelSeek (Team et al., 2025b) utilizes reflection as a closed loop, updating code and plans with human-interactive feedback after each round. Finally, a recent study (Kumbhar et al., 2025) regularizes the process up front with explicit goals & constraints and afterwards with standardized scoring to provides an objective standard that makes reflection repeatable.

6.2.3 Collective multi-agent reasoning

Multi-agent frameworks for scientific discovery distribute labor across specialized LLM-driven roles, where advanced LLM reasoning not only orchestrates coordination between scientific agents but also adjudicates conflicting evidence to maintain coherence in the process.

To illustrate, we introduce some important multi-agent frameworks as follows. Firstly, ProtAgents (Ghafarollahi & Buehler, 2024b) exemplifies this pattern in protein design. The framework involve agents for literature retrieval, structure analysis, physics simulation, and results analysis. Specifically, the backbone LLM directs reasoning over multi-modal outputs, choosing when to iterate or convergence-check based on feedback signals. PiFlow (Pu et al., 2025), on the other hand, instantiates reasoning as principle-aware uncertainty reduction with a multi-agent loop in which a Planner agent relays strategy to a Hypothesis agent and a validation loop, explicitly tying multi-agent communication to hypothesis–evidence alignment. AtomAgents (Ghafarollahi & Buehler, 2024a) also brings similar role specialization to alloy discovery. In particular, the agent uses LLM-guided reasoning to control over when to trigger simulations and how to evaluate multi-modal results, letting reasoning allocate computational resources and prune alloy candidates.

With a similar *planner, executor and evaluator* framework, CellAgent (Xiao et al., 2024b) instantiates researches on single-cell analysis, where the planner LLM reasoning selects tools or hyper-parameters and the evaluator LLM triggers self-iterative re-runs when quality checks fail. Some other notable works include ARIA (Ramirez-Medina et al., 2025) that introduces a four-agent framework (scout, filter, synthesizer and procedure-drafter), Curie (Kon et al., 2025) that embeds rigor into multi-agent planning, Team of AI-made Scientists (TAIS) (Liu et al., 2024b) for gene-expression discovery and the Virtual Lab (Swanson et al., 2024) for nano-body design with role agents.

6.3 Embodied Agents

Embodied agents extend reasoning beyond text, anchoring language in robotic perception, manipulation and navigation. By embedding LLMs within robotic and simulated bodies, these embodied agents tackle real-world generalization, continual adaptation and multi-modal grounding.

In this subsection, we begin with the foundational layer (Section 6.3.1), which covers long-horizon embodied planning, tool-assisted perception, manipulation and execution. Building upon these capabilities, the self-evolving layer (Section 6.3.2) introduces agentic memory, feedback and self-reflection capabilities enabling robots to refine control policies, adapt to novel environments and improve performance through continual interaction. Finally, the collective reasoning layer explores multi-robot collaboration (Section 6.3.3), where agents coordinate perception, share learned representations and jointly reason about tasks to achieve complex embodied goals.

6.3.1 Foundational agentic reasoning

Planning. Early work such as SayCan (Ahn et al., 2022) established the template by mapping linguistic descriptions to skill affordance estimates and SayPlan (Rana et al., 2023) refined this grounding by leveraging 3D scene graphs to align goal references with object-centric representations and spatial models. Beyond symbolic representations, EmbodiedGPT (Mu et al., 2023) use curated video CoT annotations of sub-goals to train models taht map multi-model input to structured sequences for embodied planning, while context-aware planning system (Kim et al., 2023) adds semantic spatial map and object location information to the planning pipeline, enabling dynamical planning during execution. In addition, DEPS (Wang et al., 2024l) introduces an interactive planning loop (i.e. describe, explain, plan and select) for open-world multi-task agents.

Embodied agents also rely on multi-modal reasoning traces that explicitly align perception with action. For example, Embodied CoT (Zawalski et al., 2024) trains vision-language-action models to generate reasoning steps incorporating visual features before executing an action. Fast ECoT (Duan et al., 2025) accelerates this by caching and re-using reasoning segments across time-steps, reducing inference latency while preserving task success. More recently, Cosmos-Reason1 (Azzolini et al., 2025) establishes an ontology of space, time and dynamics that lets CoT sequences encode structured physical priors. CoT-VLA (Zhao et al., 2025d) builds a visual chain-of-thought by predicting future image frames as intermediate sub-goals prior to action generation.

Finally, Emma-X (Sun et al., 2024b) integrates grounded chain-of-thought with look-ahead spatial reasoning, improving long-horizon embodied task performance.

Another line of works strengthen embodied planning through reinforcement learning, considering planning not only as static decomposition but as a self-evolving process that adapts to environment feedback. Robot-R1 (Kim et al., 2025) trains large VLMs to predict keypoint transitions under visual context, turning RL into a mechanism for learning physically grounded forward models. ManipLVM-R1 (Song et al., 2025b) exploits verifiable physical reward signals (e.g., trajectory match and affordance correctness) to reduce reliance on dense expert annotation. Embodied-R (Zhao et al., 2025a) presents a collaborative framework where VLMs handle perception and smaller LMs handle reasoning, and the whole is trained via RL for embodied spatial reasoning. VIKI-R (Kang et al., 2025b) further extends this direction into heterogeneous multi-agent cooperation with a two-stage design, employing a two-stage pipeline of chain-of-thought fine-tuning followed by hierarchical RL across agents coordinating activation and planning.

Tool-use. Embodied agents can also be strengthened to interact with external tools to enhance perception and compensate for incomplete observations. GSCE (Wang et al., 2025j), for example, provides a prompt-framework that binds skill APIs and constraints for safe LLM-driven drone control. MineDojo (Fan et al., 2022) links agents to internet-scale corpora and thus enabling richer affordance grounding. Physical AI Agents (Bousetouane, 2025) further introduces a modular architecture and a retrieval augmented generation design pattern for embedding real-world physical interaction into LLM-driven agents. Beyond offline tool use, some systems treat the environment itself as an API. For example, Matcha agent (Zhao et al., 2023a) uses an LLM to issue queries about objects and scenes and thereby acquire perceptual information needed for task completion.

On the other hand, execution module is one of the most important tool type. It translates high-level language instructions into continuous motor commands, enabling embodied agents to act reliably in physical environments. Early systems such as SayCan (Ahn et al., 2022) uses language to invoke robot pick-and-place skills; while LEO (Huang et al., 2023b) broaden execution to more general manipulation settings and Hi Robot (Shi et al., 2025a) uses a VLM reasoner to process complex prompts and a low-level action policy executes the chosen step. More recent efforts broaden the execution space: Gemini Robotics (Team et al., 2025a) introduces a large-scale vision-language-action model for real-world robot control and Octopus (Yang et al., 2024b) generates executable code in simulated environments that bridges planning and manipulation.

Beyond single-agent control, hybrid pipelines couple reactive reflexes with language-guided policies to support complex domains. For example, CaPo (Liu et al., 2024c) incorporates an execution phase where agents carry out decomposed sub-tasks and adapt their meta-plan based on progress; COHERENT (Liu et al., 2024e) embeds a robot executor module within its PEFA (i.e. proposal, execution, feedback and adjustment) loop, which ensures each assigned sub-task is acted and refined appropriately; and MP5 (Qin et al., 2024a) integrates multi-modal perception to generate executable plans in open-ended Minecraft. At the perception–action interface, LLM-Planner (Song et al., 2023a) generates sub-goals and maps them into action sequences via a low-level controller and EmbodiedGPT (Mu et al., 2023) illustrate how LLM-generated plans can be translated into control policy for embodied control in physical environments.

Search and retrieval. Embodied agents can also use search and retrieval ability to ground language in spatial structure and past experience. Early navigation systems such as L3MVN (Yu et al., 2023a) use LLMs to query a semantic map and select promising frontiers as long-term goals during visual target navigation, while SayNav (Rajvanshi et al., 2024) and SayPlan (Rana et al., 2023) build 3D scene graphs and then search task-relevant subgraphs so language instructions can be translated into grounded waypoints and sub-tasks in large environments. Long-horizon navigation works like ReMEmbR (Anwar et al., 2025) maintain a structured spatio-temporal memory that can be queried to answer "where" and "when" questions about past robot experience. Additionally, RAG-style systems make retrieval a first-class part of the planning loop: Embodied-RAG (Xie et al., 2024b) and EmbodiedRAG (Meghan et al., 2024) treat an agent's experience and 3D scene graphs as non-parametric memories from which task-relevant episodes or subgraphs are retrieved for navigation and task planning; Retrieval-Augmented Embodied Agents (Zhu et al., 2024b) retrieve policies from a shared memory bank and condition action generation on them; and MLLM-as-Retriever (Yue et al.,

2024a) trains a multi-modal LLM retriever to rank past trajectories so each decision step can condition on the most useful prior experience rather than only the current observation.

6.3.2 Self-evolving agentic reasoning

Embodied agents reliably achieve long-horizon autonomy when they can self-evolve over time: monitor their own internal states, store and update task-relevant knowledge and adjust behaviors when plans deviate. In the following paragraphs, we examine how memory modules, feedback signals and agentic reflection enable embodied agents to turn planning from a one-shot process into a continually improving cycle of behavior.

Memory. Effective memory mechanisms enable agents to reuse past experiences and maintain coherent task execution over extended interactions. Many systems cache recent observations in episodic buffers while summarizing long-term semantics in structured graphs, as in household planning (Glocker et al., 2025) and long-horizon agents with hybrid multi-modal memory (Li et al., 2024i). Skills and routines can be shared across tasks via indexed memory stores. For example, HELPER-X (Sarch et al., 2023) indexes discovered skills and action scripts, which aid future dialogue and can be shared across domains. Spatial navigation methods such as BrainNav (Bai et al., 2026) maintain biologically inspired dual-map memories linked by a hippocampal hub to reduce hallucinations and drift. Broader contexts also benefit: CAPEAM (Kim et al., 2023) incorporates environment-aware memory modules that track object states and spatial changes. Finally, lifelong episodic systems such as Ella (Zhang et al., 2025i) maintains long-term multi-modal memory system to support social-robot interaction.

Agentic Feedback and Reflection. Dialogue-based critique, calibrated uncertainty and environment-aware reward shaping refine policies beyond binary success signals. For example, Matcha agent (Zhao et al., 2023a) treats objects and scenes as interactive information sources before acting and FAMER (Wang et al., 2025l) uses lightweight preference feedback to adapt embodied agents to user intentions in real time. Uncertainty-aware planners such as KnowNo (Ren et al., 2023), which proactively solicit guidance when confidence falls below guarantees, and Octopus (Yang et al., 2024b), which exploits environmental feedback to improve generated executable programs over time. At the multi-agent level, MindForge (Lică et al., 2024) introduces theory-of-mind style perspective feedback so heterogeneous robots adapt to each other's reasoning strategies; while ReAd (Zhang et al., 2024j) introduces a advantage-based feedback loop that enables an LLM planner to self-refine its collaboration strategies across embodied multi-agent tasks.

Robust reflection mechanisms help agents anticipate failures by monitoring their own reasoning and actions and then adjusting plans. Optimus-1 (Li et al., 2024i) couples a *Knowledge-guided Planner* with an *Experience-Driven Reflector* to revise decisions using stored experience, while another recent study (Moncada-Ramirez et al., 2025) defines structured agentic workflows (including self-Reflection, multi-Agent reflection and LLM Ensemble) that enable robots to reflect on and refine LLM-generated object-centered plans, thus reducing reasoning errors. Systems such as EMAC+ (Ao et al., 2025) interleave perception, planning and verification steps to perform online plan refinement and earlier works such as Voyager (Wang et al., 2023a) also embeds an iterative prompting loop that uses environment feedback and execution errors to refine its skill library over time.

6.3.3 Collective multi-agent reasoning

Multi-agent collaboration enables embodied systems to divide labor and coordinate complex tasks more efficiently, with language often serving as the primary medium for negotiation and role allocation. For instance, SMART-LLM (Kannan et al., 2024) decomposes high-level instructions and allocates sub-tasks across multiple robots, while CaPo (Liu et al., 2024c) optimizes cooperative plans to avoid redundant exploration. For heterogeneity and coordination mechanisms, COHERENT (Liu et al., 2024e) deploys a propose-execution-feedback-adjust loop across diverse robot types to enable seamless joint operation. In addition, Theory of Mind (ToM), which refers to an embodied agent's ability to infer and reason about others' beliefs and mental states, is also highly related to embodied multi-agent systems (Li et al., 2023b; Wu et al., 2025j; Cross et al., 2024). For example, MindForge (Lică et al., 2024) equips agents with explicit theory-of-mind representations and natural inter-agent communication to coordinate collaboratively.

For multi-modal frameworks, EMAC+ (Ao et al., 2025) integrate vision and language modules and continuously refine plans via visual feedback, COMBO (Zhang et al., 2024c) integrates vision and language modules and continuously refine plans via visual feedback, and VIKI-R (Kang et al., 2025b) demonstrates reinforcement learning as a scalable coordination mechanism among embodied agents. At larger scales, studies such as RoCo (Zhao et al., 2024b) show how role negotiation and flexible protocols support adaptable teamwork in dynamic environments.

6.4 Healthcare & Medicine Agents

Healthcare and medical agents seek to support the full clinical decision pipeline, from initial symptom triage to treatment planning and integrating LLMs with structured patient records, medical ontologies and expert guidelines. Unlike general assistants, these systems must operate under strict safety constraints, multi-modal evidence and legal justification.

In this subsection, we begin with the foundational layer (Section 6.4.1), which includes medical and diagnostic reasoning and tool-augmented access to various biomedical knowledge bases and APIs. Building on these primitives, the self-evolving layer (Section 6.4.2) examines memory, feedback and reflective modules that allow these agents to accumulate patient-specific context, adapt to longitudinal trajectories and revise clinical plans over time. Finally, the collective layer (Section 6.4.3) highlights multi-agent collaboration, which includes doctor–agent co-planning, human–AI shared autonomy and specialist model ensembles.

6.4.1 Foundational agentic reasoning

Planning. Planning is a core capability for healthcare agents, which enables them to structure long-horizon clinical pathways into diagnostic and treatment phases, refine workflows dynamically as patient conditions evolve, and coordinate across teams and tools toward cohesive care delivery. We discuss several recent advancements as follows. For instance, a recent agentic clinical system (Ferber et al., 2024) orchestrates specialized tools and guideline citations to support oncology decision-making, EHRAgent (Shi et al., 2024a) decomposes multi-table EHR inference into code-execution steps with feedback learning, and PathFinder (Ghezloo et al., 2025) presents a multi-agent, multi-modal histopathology workflow for diagnostic reasoning.

Other frameworks model planning as an explicit orchestration layer across levels of abstraction. For example, MedAgent-Pro (Wang et al., 2025r) proposes a hierarchical workflow which first generates disease-level diagnostic plans from guideline criteria and then dispatches tool-agent modules for execution. MedOrch (He et al., 2025d) treats tool invocation itself as a planning primitive across modalities, orchestrating reasoning agents for multi-step diagnostic execution. On the other hand, ClinicalAgent (Yue et al., 2024b) coordinates multi-agent workflows for clinical planning, leveraging LLM reasoning to allocate tools and synthesize evidence. In addition, planning in healthcare agents is increasingly adaptive, responding to new information and evolving contexts. For example, DoctorAgent-RL (Feng et al., 2026) models clinical consultation as a dynamic decision-making process under uncertainty, optimizing questioning strategies and diagnostic paths via reinforcement learning; while DynamiCare (Shang et al., 2025) adjusts specialist-agent teams across multi-round interactions as new patient information emerges.

Tool-use. Tool integration significantly expands a healthcare agent's action space, enabling precise calculations, medical image interpretation and access to specialized databases. Recent studies are summarized as follows. Several systems explicitly foreground extensibility. MedOrch (He et al., 2025d) introduces a modular architecture that allows new diagnostic APIs to be incorporated without retraining, while TxAgent (Gao et al., 2025d) integrates over two hundreds pharmacological tools to support therapeutic decision-making across drug–disease–treatment relationships. AgentMD (Jin et al., 2025b) similarly curates and leverages over two thousands executable clinical calculators to learn risk-prediction pipelines.

Other approaches focus on structured function calling for safe execution. For example, LLM-based agents can reliably invoke bedside calculators when provided with explicit function signatures, ensuring arithmetic correctness in dosing and risk scoring (Goodell et al., 2025). MeNTi (Zhu et al., 2024a) goes further by enabling nested tool calls across multi-step medical calculators. Complementing these text-based integrations, MMedAgent (Li et al., 2024b) demonstrates that agents can learn to select among multi-modal tools.

In addition, execution is crucial for translating high-level clinical plans into concrete actions such as code operations, database queries or robotic procedures. VoxelPrompt (Hoopes et al., 2025) embed 3-D volumetric priors so that language instructions drive spatial segmentation and analysis of medical image volumes. On the other hand, embodied ultrasound-robot controllers (Xu et al., 2024) translate LLM-generated plans into closed-loop robotic scanning via a "think-observe-execute" loop. Adaptive reasoning-and-acting systems (Dutta & Hsiao, 2024) further refine both the reasoning and actions over time in simulated clinical environments. In medical imaging, systems like MedRAX (Fallahpour et al., 2025) materializes multi-step reasoning by integrating specialized chest-X-ray tools and LLM reasoning into an end-to-end diagnostic agent. PathFinder (Ghezloo et al., 2025) similarly executes multi-agent, multi-modal diagnostic workflows in histopathology.

Another class of healthcare agents deploys code-level workflows. For example, Conversational Health Agents (Abbasian et al., 2023) compile dialogue actions into function calls and code execution for downstream processing, while EHRAgent (Shi et al., 2024a) materializes EHR operations via executable code. MedAgentGym (Xu et al., 2025g) trains agents to produce code that is directly executed and graded, enforcing reliability of reasoning traces. DoctorAgent-RL (Feng et al., 2026) validates multi-turn dialogue acts by executing reinforcement-learned strategies in simulated consultations, while AIPatient (Yu et al., 2025b) materializes realistic patient scenarios for execution-based evaluation and another recent study (Almansoori et al., 2025) demonstrates how self-evolving multi-agent simulations allow execution behaviors themselves to improve over time.

Search and retrieval. Search-based agents enhance clinical decision-making by linking LLM reasoning with external biomedical knowledge sources. For instance, MeNTi (Zhu et al., 2024a) supplements therapeutic reasoning by bridging LLM calls into multi-step medical calculators while EHRAgent (Shi et al., 2024a) dynamically executes code operations over multi-table EHR data to support complex tabular inference. Conversational Health Agents (Abbasian et al., 2023) enrich personalized dialogue by integrating developer-defined external sources and orchestrating action flows. Another line of work explicitly embeds retrieval-augmented generation (RAG) into healthcare agents. For example, CLADD (Lee et al., 2025b) retrieves molecular graphs and prior assay results before proposing compound hypotheses and MedReason (Wu et al., 2025d) issues targeted knowledge-graph sub-queries to anchor each reasoning step for clinical QA.

6.4.2 Self-evolving agentic reasoning

Self-evolving capabilities enable healthcare agents to maintain longitudinal clinical coherence. Representative use cases include accumulating relevant medical context across encounters, updating beliefs as new evidence arrives and revising decisions when inconsistencies surface. In the following paragraphs, we examine how memory, feedback and reflective mechanisms collectively turn clinical reasoning from a one-shot prediction into a continually improving care process.

Memory. Persistent memory is essential for tracking medical or patient history and maintaining context across interactions. For instance, epidemic-modeling agents (Williams et al., 2023) maintain temporal contact histories to trace infection chains over time; while MedAgentSim (Almansoori et al., 2025) stores experience histories and refine diagnostic strategies over time. In structured data settings, EHRAgent (Shi et al., 2024a) records intermediate computations over tabular EHRs so subsequent steps can reference prior results. EvoPatient (Du et al., 2024b) interleaves memory with coevolution maintains evolving clinical state across dialogue phases while AIPatient (Yu et al., 2025b) persists longitudinal EHR-derived variables to drive consistent responses. Multi-agent systems such as MedOrch (He et al., 2025d) contain clinical knowledge graph agent which can be considered as external memory that can be queried to retrieve known relationships or diagnostic patterns.

Agentic Feedback and Reflection. Agentic feedback and self-reflection are complementary mechanisms that improve reliability and adaptability of healthcare agents. Feedback converts execution outcomes into learning signals: DynamiCare (Shang et al., 2025) updates multi-agent treatment strategies when newly observed patient state contradicts prior plans; DoctorAgent-RL (Feng et al., 2026) optimizes questioning policies from consultation rewards; and MedAgentGym (Xu et al., 2025g) enforces correctness by executing

and grading generated code. Tool-use pipelines also propagate execution feedback. For example, the success/failure of table queries in EHRAgent (Shi et al., 2024a) or calculator calls in MeNTi (Zhu et al., 2024a) and clinical-calculation agents (Goodell et al., 2025) to refine subsequent actions.

6.4.3 Collective multi-agent reasoning

Multi-agent collaboration is central to healthcare AI, since clinical decision-making often depends on consensus among specialists, negotiation of competing hypotheses and coordination across roles such as physicians, patients and trial designers. In the following, we discuss several strands of research centered around multi-agent capabilities.

For collaborative decision-making, notable frameworks include MDAgents (Kim et al., 2024b), which automatically assigns tailored collaboration structures to teams of LLMs depending on medical task complexity, and DoctorAgent-RL (Feng et al., 2026), which uses a multi-agent reinforcement-learning framework to optimize multi-turn doctor-patient consultation dialogues. In addition, Agent-derived Multi-Specialist Consultation (AMSC) (Wang et al., 2024b) explores staged multi-specialist dialogues for differential diagnosis that mimics the medical scene of a patient consulting with multiple specialists. Other notable works include ClinicalAgent (Yue et al., 2024b), which organizes clinical trial workflows via role-based agent collaboration / LLM reasoning and PathFinder (Ghezloo et al., 2025), which integrates a diverse set of agents that can gather evidence and provide comprehensive diagnoses with natural language explanations.

On the other hand, there are studies focusing on simulation-driven collaboration. These works highlight how multi-agent setups enrich training and evaluation. MedAgentSim (Almansoori et al., 2025) co-evolves doctor and patient agents to simulate real-world multi-turn clinical interactions, and EvoPatient (Du et al., 2024b) uses co-evolution of patient and doctor agents to generate diagnostic dialogue data and therefore gathers experience to improve the quality of both questions and answers to enable accurate human doctor training. In addition, DynamiCare (Shang et al., 2025) initiates a team of specialist agents that iteratively queries the patient system to integrate new information and adapts the composition and strategy. Finally, medical agents can also collaborative to aid medical reasoning process. For example, MedAgents (Tang et al., 2023b) demonstrates zero-shot cooperation among domain-specialist agents in medical reasoning tasks, CLADD (Lee et al., 2025b) uses retrieval-augmented generation to support drug-discovery workflows across agents and GMAI-VL-R1 (Su et al., 2025b) combines multi-modal reasoning and reinforcement learning in a multi-agent framework to support large-scale medical decision-making.

6.5 Autonomous Web Exploration & Research Agents

Web agents, GUI agents and autonomous research agents constitute three interlinked but distinct trajectories of agentic reasoning systems. Firstly, web agents specialize in navigating online resources, issuing web API calls or browser actions to retrieve dynamic evidence and steer research direction. GUI agents go further by manipulating software interfaces and multi-modal dashboards directly (i.e. clicking, typing, navigating) to execute experiments, data workflows and interface-based tasks. Autonomous research agents sit at the top of this hierarchy, pairing LLM reasoners with scientific workflows, tool-chains and meta-loops to drive hypothesis generation, data synthesis and paper writing. The core connection is a progression of autonomy: first web agents retrieve evidence from online resources, then GUI agents operationalize actions inside software interfaces, and finally autonomous research agents orchestrate full scientific workflows end-to-end.

In this subsection we begin with the foundational layer (Section 6.5.1), which captures the core capabilities that any autonomous agent must support: perceiving its environment, reasoning about goals, planning actions and grounding those into tool-augmented workflows. Building on these primitives, the self-evolving layer (Section 6.5.2) examines how agents incorporate feedback, memory and reflection to iteratively refine their behaviors and improve methods over time. Finally, the collective layer (Section 6.5.3) highlights how agents move beyond individual competence into coordination, specialization and emergent collaboration. While web agents, GUI agents and autonomous agents share common themes of goal-directed autonomy, tool-use and iterative improvement, they differ in where they act on, how they manipulate their environment and what goal they aim to achieve.

6.5.1 Foundational agentic reasoning

Planning. Planning is essential for web agents because they must decompose long-horizon tasks into manageable steps, adapt to dynamic pages and coordinate tool/invocation strategies. Early work such as WebGPT (Nakano et al., 2021) fine-tuned GPT-3 (Brown et al., 2020) to answer open-ended questions via a text-based web-browser interface, building on the broader observation that LLMs can act as zero-shot reasoners when prompted to externalize intermediate steps (Kojima et al., 2023). Then, various web-based methods deepened the planning paradigm: for example, SEEACT (Zheng et al., 2024a) explored large multi-modal models as generalists that integrating visual and HTML grounding for web-based tasks, and AutoWebGLM (Lai et al., 2024) introduced HTML simplification and various learning techniques for open-domain web task decomposition and navigation. These works paved the way for recent systems such as Agent Q (Putta et al., 2024) that integrate guided MCTS, self-critique and off-policy preference optimization on web-task benchmarks, and set the stage for even more advanced long-horizon web planners such as WebExplorer (Liu et al., 2025e) and WebSailor (Li et al., 2025f).

In addition, reinforcement learning has become a core tool for improving the decision-making and planning behavior of web-based LLM agents. WebRL (Qi et al., 2024) introduces a self-evolving online curriculum that generates new tasks from unsuccessful attempts and trains an outcome-supervised reward model to guide policy optimization. WebAgent-R1 (Wei et al., 2025e) performs end-to-end multi-turn RL, learning web interaction policies directly from online rollouts with binary success rewards. DeepResearcher (Zheng et al., 2025c) scales RL to real-world web environments, using a multi-agent browsing architecture and exhibiting emergent behaviors such as plan formulation, cross-source corroboration, and self-reflection. Hybrid pipelines like AutoWebGLM (Lai et al., 2024) combine supervised training with RL fine-tuning to strengthen task decomposition and structured navigation, while Navigating WebAI (Thil et al., 2024) combines supervised learning and RL techniques to improve web navigation performance. Methods such as Pangu DeepDiver (Shi et al., 2025b), EvolveSearch (Zhang et al., 2025c), ZeroSearch (Sun et al., 2026), StepSearch (Wang et al., 2025o) and WebEvolver (Fang et al., 2025c) use RL-based self-improvement, for example by adaptively scaling search depth or jointly training an agent and a world-model-like simulator to improve long-horizon web decision-making. Hierarchical approaches like ArCHer (Zhou et al., 2024b) optimize high-level and low-level policies with a multi-turn hierarchical RL framework, while PAE (Zhou et al., 2025e) combines a task proposer, an acting agent and an evaluator to support autonomous skill discovery via RL in internet environments.

Planning is a core capability for GUI agents, enabling them to coordinate long, multi-step interactions across applications and operating environments. OS-Copilot (Wu et al., 2024e) approaches this by treating the desktop as a unified control space in which a generalist agent continually refines its multi-step workflows. Agent S (Agashe et al., 2024) builds an experience-augmented planning stack that decomposes tasks into sub-goals while retrieving past trajectories and external knowledge to guide action sequencing. InfiGUIAgent (Liu et al., 2025h) strengthens planning by integrating hierarchical task structuring into a multi-modal backbone, allowing agents to organize GUI procedures at multiple levels of abstraction. MobA (Zhu et al., 2025d) and PC Agent (Liu et al., 2025b) employ hierarchical architectures that separate high-level planning from low-level execution—on mobile and desktop respectively. GUI foundation models such as OS-ATLAS (Wu et al., 2024f), OSCAR (Wang & Liu, 2024) and UItron (Zeng et al., 2025c) further emphasize robust cross-application planning: OS-ATLAS offers a platform-agnostic action model for consistent control, OSCAR maintains state-aware plans that adapt as execution unfolds and UItron unifies offline and online planning within a single general-purpose GUI agent.

Likewise, reinforcement learning has become a central way to endow GUI agents with planning over long action sequences. End-to-end frameworks such as ARPO (Lu et al., 2025a) and ComputerRL (Lai et al., 2025) directly optimize multi-step GUI trajectories with replay buffers or large-scale online interaction, replacing hand-crafted scripts with learned policies for general desktop control. R1-style and semi-online methods, including UI-R1 (Lu et al., 2025f), GUI-R1 (Luo et al., 2025b), InfiGUI-R1 (Liu et al., 2025i) and UI-S1 (Lu et al., 2025g), start from strong vision–language backbones and then use RL to sharpen action prediction and long-horizon reasoning. A complementary line focuses on where to act by improving visual grounding: GUI-Bee (Fan et al., 2025), SE-GUI (Yuan et al., 2025b), UIShift/GUI-Shift (Gao et al., 2025c), spatial-reasoning feedback for GUI grounding (Zhao et al., 2026b), and UI-AGILE (Lian et al., 2025) develop

RL-based grounding frameworks to help agents reliably localize target elements before executing actions. ZeroGUI (Yang et al., 2025b) pushes toward fully automated online RL loops, where the agent generates its own tasks and trajectories and improves with zero human annotation, while ComputerRL (Lai et al., 2025) scales end-to-end online RL in large distributed desktop environments. AgentCPM-GUI (Zhang et al., 2025x) couples supervised pre-training with reinforcement fine-tuning to strengthen decision quality on mobile apps. Finally, foundation-style GUI agents such as AutoGLM (Liu et al., 2024g) and Mobile-Agent-v3 (Ye et al., 2025a) serve as general backbones that unify perception, grounding and action, and are trained or fine-tuned with scalable RL frameworks to align long-horizon GUI planning with real-world success signals.

For autonomous research agents, planning modules translate abstract goals into actionable research itineraries. For example, Agent Laboratory (Schmidgall et al., 2025) organizes work into three structured stages, namely literature review, experimentation and report writing, and supports the workflow with tool-hooks that automate code execution, experiment runs and documentation. GPT Researcher (Elovic, 2023) uses a plan → research → write cycle, where a dedicated planner drafts the outline, retrieval/analysis agents gather evidence and a writer compiles the final report. Chain of Ideas (Li et al., 2024f) retrieves literature into a chain structure to reflect domain progression and support ideation via experiment design, whereas IRIS (Garikaparthi et al., 2025) performs hypothesis exploration via Monte Carlo Tree Search to expand promising branches before committing to downstream tasks. Broader variants include ARIA (Ramirez-Medina et al., 2025) and NovelSeek (Team et al., 2025b) that automate the research workflow with a complete literature search, hypothesis generation and experiment planning cycle.

Tool-Use. For web agents, tool-use abilities underpins execute plans in realistic, dynamic environments. For example, WebVoyager (He et al., 2024b) systematizes multi-modal execution by building an end-to-end agent that operates on real websites. On the interaction side, BrowserAgent (Yu et al., 2025d) makes the action space more human-like, defining a compact set of browser primitives (e.g., click, scroll, type) and coupling them with an explicit memory mechanism to maintain key conclusions across steps, yielding strong gains on multi-hop QA benchmarks. Finally, methods like WALT (Prabhu et al., 2025) and pipeline-oriented systems such as WebDancer (Wu et al., 2025b) and WebShaper (Tao et al., 2025b) push tool use from mere execution toward tool discovery and data-centric interaction. Specifically, WALT teaches agents to reverse-engineer reusable tools from website functionality, while WebDancer and WebShaper embed web actions inside multi-turn information-seeking and dataset-synthesis loops, respectively.

Tool use is another core capability for GUI agents, enabling them to invoke system functions and application features as structured tools. As pioneering systems, AutoDroid (Wen et al., 2024) automatically analyzes Android apps to construct functionality-aware UI abstractions that LLM agents can reason over as capabilities rather than raw layouts, while its successor AutoDroid-V2 (Wen et al., 2025a) re-frames mobile UI automation as LLM-driven code generation, with an on-device small language model emitting executable scripts for a local interpreter. MobileExperts (Zhang et al., 2024g) models each expert as a tool-capable specialist and uses a dual-layer controller to select which expert and its associated tool-set to invoke at different stages of a mobile workflow. AgentStore (Jia et al., 2025a) pushes this idea to the platform level by treating heterogeneous agents themselves as tools: a MetaAgent uses AgentTokens to route operating-system subtasks to the most suitable specialized "tool-agent" through a unified interface. OS-Copilot (Wu et al., 2024e) and OSCAR (Wang & Liu, 2024) integrate rich system-level tools into unified computer-control frameworks, so that complex desktop tasks are expressed as sequences of tool calls. OS-ATLAS (Wu et al., 2024f) complements these systems with a foundation action model that offers robust cross-platform GUI grounding, serving as a reliable actuator layer for downstream tool-using agents. Finally, SeeClick (Cheng et al., 2024) strengthens the execution stack by pre-training a visual GUI agent for GUI grounding, improving the ability to locate the correct on-screen elements from instructions.

Specialized tools can expand an autonomous research agent's capabilities beyond pure text, allowing more fine-grained ability. For instance, Agentic Reasoning (Wu et al., 2025e) automatically routes queries to appropriate tool modules like code execution, web search and structured memory agents when the main LLM detects a gap in reasoning; while Webthinker (Li et al., 2025j) empowers autonomous web exploration and page navigation during long-horizon investigations, by interleaving reasoning, search and draft-writing with a web explorer module. PaperQA (Lála et al., 2023) and its follow-up synthesis agent (Skarlinski et al., 2024)

integrate PDF parsing and citation-level grounding to produce verifiable answers and literature syntheses, while Scideator (Radensky et al., 2026) provides an IDE-style tool-chain that combines paper facets with novelty checks for real-time brainstorming. In addition, DeepResearcher (Zheng et al., 2025c) shows that reinforcement learning over real-web interactions improves deep-research efficiency and quality, with emergent behaviors such as plan refinement and cross-source corroboration.

Execution components ground high-level reasoning in code, simulations or laboratory protocols to produce verifiable scientific outcomes. Agent Laboratory (Schmidgall et al., 2025) executes experiments specified in declarative configuration files by orchestrating external toolchains, while Agentic Reasoning (Wu et al., 2025e) integrates a coding agent that executes Python alongside web search and structured memory, feeding the results back into the reasoning process. MLR-Copilot (Li et al., 2024g) turns research plans into runnable implementations via an ExperimentAgent that leverages retrieved prototype code, runs experiments, and iteratively debugs implementations. Dolphin (Yuan et al., 2025a) closes the loop by generating ideas, implementing them through code templates with traceback-guided debugging, executing experiments, and using the analyzed results to steer the next research cycle. Early autonomous-science systems such as Coscientist (Boiko et al., 2023a) already showed that LLMs can plan, execute, and analyze scientific procedures with tool access. The AI Scientist (Lu et al., 2024a) automates end-to-end ML experiments, i.e. generating ideas, writing code, executing experiments and visualizing results, so that observed outcomes can guide subsequent runs, while The AI Scientist-v2 (Yamada et al., 2025) adds a dedicated experiment manager and progressive agentic tree search to prioritize and schedule experiment branches. Most recently, NovelSeek (Team et al., 2025b) introduces a unified closed-loop multi-agent framework that spans hypothesis generation, idea-to-methodology construction, and multi-round automated experiment execution with feedback across diverse scientific domains.

Search and Retrieval. Search and retrieval lie at the heart of what differentiates web agents from static language models: they must locate, synthesize and refactor web-scale information in dynamic environments. WebExplorer (Liu et al., 2025e) tackles this by generating challenging information-seeking trajectories and training agents to interleave a search tool and a browse tool over many turns, resulting in improved multi-step retrieval policies on complex benchmarks. WebSailor (Li et al., 2025f) likewise focuses on information-seeking under extreme uncertainty, constructing high-uncertainty search tasks and using a two-stage post-training pipeline to instill uncertainty-reducing search strategies for long-horizon web tasks. INFOGENT (Reddy et al., 2025b) also performs multi-query search across diverse web sources, enabling comprehensive information retrieval beyond task completion. Video-Browser (Liang et al., 2026b) extends this open-web search setting to video evidence, where agents must browse, retrieve, and validate visual-temporal information across sources. For retrieval-augmented generation applications, RaDA (Kim et al., 2024a) explicitly disentangles web-agent planning into Retrieval-augmented Task Decomposition and Retrieval-augmented Action Generation, so that each high-level subgoal and concrete action is conditioned on fresh search results while respecting context limits. In addition, GeAR (Shen et al., 2025) advances retrieval itself by augmenting a base retriever with graph expansion and an agent framework, enabling multi-hop passage retrieval along graph-structured evidence chains. Finally, WebRAGent (Zhang et al., 2026d) exemplifies retrieval-augmented generation for web agents by retrieving past trajectories and external knowledge into a multi-modal RAG policy.

Several GUI agents use retrieval capability to inject external experience or knowledge at inference time. Synapse (Zheng et al., 2023) maintains an exemplar memory of abstracted trajectories and, for each new task, retrieves similar past trajectories as in-context plans, substantially improving multi-step decision-making. LearnAct (Liu et al., 2025a) builds a three-agent pipeline that mines human demonstrations into a knowledge store and retrieves the most relevant instructions to guide mobile GUI execution on unseen and diverse tasks. MobileGPT's Explore–Select–Derive–Recall framework (Lee et al., 2023) equips a phone agent with human-like app memory, storing modular procedures that can be recalled and recomposed when similar tasks reappear. TongUI (Zhang et al., 2025a) turns large-scale multi-modal web tutorials into the GUI-Net trajectory corpus, effectively giving agents a large offline memory of how humans operate hundreds of apps across multiple operating systems. RAG-GUI (Xu et al., 2025e) makes retrieval explicit at inference time by querying web tutorials and generating textual guidelines that are fed into any VLM-based GUI agent as step-by-step hints. WebRAGent (Zhang et al., 2026d) shows a related pattern in web automation, combining a multi-modal retriever with a web agent so that each action is conditioned on retrieved guidance.

Search modules probe the research landscape to surface relevant papers and passages, enriching context and grounding subsequent reasoning. WebThinker (Li et al., 2025j) equips large reasoning models with a Deep Web Explorer module for autonomous web search and page navigation, and uses an Autonomous Think–Search–and–Draft strategy with RL-based training to decide when to browse and what to extract during long-horizon tasks. DeepResearcher (Zheng et al., 2025c) scales end-to-end training on the real web via reinforcement learning in live search environments, optimizing the iterative think–search loop and exhibiting emergent behaviors such as plan formulation, cross-source corroboration, and self-correction over multi-step research trajectories. Retrieval-centric agents like PaperQA (Lála et al., 2023) and its successor PaperQA2 (Skarlinski et al., 2024) demonstrate that tightly coupling full-text retrieval with generation can substantially improve scientific QA accuracy while preserving cited provenance for literature synthesis.

In research settings, retrieval-augmented generation (RAG) grounds idea generation and analysis in freshly retrieved and citable passages. For example, GPT Researcher (Elovic, 2023) is an autonomous research-agent that retrieves sources and generates reports with citations, enabling traceability of claims to evidence. Chain of Ideas (Li et al., 2024f) organizes relevant literature into a chain-structured scaffold that mirrors a field's progressive development, thereby guiding retrieval and ideation toward subsequent links in the argument. Meanwhile, Scideator (Radensky et al., 2026) extracts key paper facets (e.g., purpose, mechanism, evaluation) and leverages them to drive targeted retrieval and recombination of ideas for identifying methodological or evidentiary gaps.

6.5.2 Self-evolving agentic reasoning

Effective self-evolving abilities enable these autonomous agents to adapt their behavior over time, retain crucial task context across interaction cycles and incrementally refine planning and execution strategies. The following paragraphs review how memory, feedback and self-reflection mechanisms support this continual improvement across these agent families, turning interaction from a one-shot pipeline into an iterative learning loop.

Memory. Memory modules transform brittle, single-pass web interactions into reusable experience. For example, Agent Workflow Memory (AWM) (Wang et al., 2024o) induces reusable workflows from successful trajectories and retrieves them to guide future tasks, while ICAL (Sarch et al., 2025) distills noisy trajectories into high-level verbal and visual abstractions that are stored as a memory of multimodal experience and later injected into prompts. Control-oriented designs such as BrowserAgent (Yu et al., 2025d) maintain explicit histories of past actions and intermediate conclusions in the agent's context, instead of only re-encoding the current page view. GLM-based agents like AutoWebGLM (Lai et al., 2024) and AgentOccam (Yang et al., 2024d) emphasize compressed page representations, using HTML simplification and carefully tuned observation spaces so that the agent's prompt contains a shorter, more informative view of the state, with past steps preserved through the usual action–observation history. More integrated frameworks like LiteWebAgent (Zhang et al., 2025b) expose planning, memory and tree search as modular components, and can plug in workflow memories together with search traces for long-horizon reuse.

Recent GUI agents adopt explicit memory modules that store and retrieve task-relevant information during long-horizon execution. Earlier work such as MobileGPT (Lee et al., 2024d) equips a mobile assistant with human-like app memory: it decomposes procedures into modular sub-tasks that are explored, selected, derived, and then stored so they can be recalled and reused instead of being re-discovered from scratch. Chain-of-Memory (CoM) (Gao et al., 2025e) incorporates short- and long-term memory by recording action descriptions and task-relevant screen information in a dedicated memory module, enabling cross-application navigation to track task state. More recent systems build increasingly structured memories: MobA's multifaceted memory module (Zhu et al., 2025d) maintains environment- and user-level traces that an adaptive planner retrieves when refining mobile task plans, while MGA (Cheng et al., 2025b) represents each step as a triad of current screenshot, spatial layout, and a dynamically updated structured memory that summarizes past transitions, mitigating error accumulation in long chains of actions. Mobile-Agent-E (Wang et al., 2025m) adds a persistent long-term store of tips and shortcuts distilled from prior trajectories, so later plans can call reusable guidance and subroutines instead of relearning them. Mirage-1 (Xie et al., 2025b) similarly organizes

experience into a hierarchical skill memory that a planner can retrieve as reusable building blocks for new GUI tasks.

Long-term memory is crucial for autonomous research agents because it enables accumulation and reuse of prior knowledge, fostering continuity across research cycles. For example, Agent Laboratory (Schmidgall et al., 2025) retains prior experiment code, results, and interpretation across its multi-phase workflow, enabling later stages to build on earlier work. GPT Researcher (Elovic, 2023) generates reports with embedded citations and provides context for planning and extension of research topics. Chain of Ideas (Li et al., 2024f) structures relevant literature into a chain scaffold that reflects a field's progression and can be revisited as new evidence arises. The AI Scientist-v2 (Yamada et al., 2025) incorporates a progressive agentic tree-search approach that enables branching, backtracking and follow-up experimentation across iterations.

Agentic Feedback and Reflection. Modern web agents treat interaction as a continual learning process, using feedback signals and reflection modules to refine their reasoning and recover from failures over time. Agent Q (Putta et al., 2024) combines guided Monte Carlo tree search with a self-critique stage, so that rollouts provide not only action sequences but also preference-style supervision. ReAP (Azam et al., 2025) makes reflection explicit by treating it as a retrieval problem: it stores task–reflection key–value pairs summarizing what was learned from past trajectories, then, at inference time, retrieves the most relevant reflections and appends them to the agent's prompt to guide planning on new web-navigation tasks. Agent-E (Azam et al., 2024) introduces an automatic validation pipeline that detects execution errors across text and vision, and then triggers self-refinement, enabling agents to iteratively correct their own workflows. Recon-Act (He et al., 2025c) uses a dual-team architecture in which a Reconnaissance team extracts generalized tools from successful and failed trajectories, and an Action team applies these tools to re-plan tasks, forming a closed feedback loop. INFOGENT (Reddy et al., 2025b), on the other hand, leverages aggregator feedback to iteratively refine navigation and search strategies based on identified information gaps. And WINELL (Reddy et al., 2025a), as a updating web agent, relies on feedback from the aggregation process to adapt subsequent searches and update selection during continuous operation. Finally, self-reflective search agents such as WebSeer (He et al., 2025a) integrate explicit self-reflection signals into reinforcement learning, constructing reflection-annotated trajectories and a two-stage training framework so that mis-solved or uncertain cases become targeted feedback that deepens future search and reasoning.

GUI agents also integrate explicit reflection so they can critique and repair their own plans. Early computer-control systems with structured reflection, for example, a zero-shot desktop control agent with structured self-reflection loops (Li et al., 2023d), provides conceptual templates that later GUI agents adapt to visual, multi-application settings. GUI-Reflection (Wu et al., 2025i) instantiates this idea end-to-end: it builds a reflection-oriented task suite, automatically synthesizes error scenarios from existing successful trajectories, and adds an online reflection-tuning stage so multi-modal GUI models learn to detect failures, reason about causes, and generate corrective actions without human annotation. History-Aware Reasoning (HAR) (Wang et al., 2025p) treats long-horizon GUI automation as a reflective learning problem, constructing reflective learning scenarios, synthesizing tailored correction guidelines, and designing a hybrid RL reward so the agent acquires episodic reasoning knowledge from its own errors and shifts from history-agnostic to history-aware reasoning. MobileUse (Li et al., 2025g) introduces hierarchical reflection on mobile devices, where the agent self-monitors at the action, subtask, and task level and triggers reflection on demand, pausing only when needed to diagnose and recover. InfiGUIAgent (Liu et al., 2025h) integrates hierarchical and expectation–reflection reasoning in a second training stage, enabling the agent to run expectation–reflection cycles that compare expected and actual outcomes and revise multi-step plans when they diverge. Mobile-Agent-E (Wang et al., 2025m) embeds an Action Reflector and Notetaker that evaluate executed steps and write refined Tips and Shortcuts back into persistent long-term memory, forming a self-evolution loop where the agent's behavior is progressively refined from accumulated experience.

For autonomous research agents, learning from outcomes is essential to improve reasoning and experimental reliability over time. CycleResearcher (Weng et al., 2024) couples a research agent with a reviewer agent that provides automated peer-review feedback, and uses an iterative preference-training loop so the research agent can refine future drafts and decisions. MLR-Copilot (Li et al., 2024g) monitors execution results and human comments during experiment implementation and execution, using these signals to iteratively refine

code, configurations and even upstream hypotheses. Dolphin (Yuan et al., 2025a) implements a closed-loop auto-research framework in which generated code is run on benchmarks and exception-guided debugging plus outcome analysis feed back into idea generation and implementation, pruning unproductive paths. At the search–reasoning interface, DeepResearcher (Zheng et al., 2025c) optimizes query, browsing, and answering policies via reinforcement learning on real-web trajectories, with outcome rewards inducing behaviors such as planning, cross-validation, and self-reflection. Agentic Deep Research (Zhang et al., 2025o) further emphasizes reward design for reasoning-driven search, arguing that principled incentives over answer quality and reasoning traces provide structured signals that improve downstream synthesis in deep-research agents.

6.5.3 Collective multi-agent reasoning

Collective multi-agent reasoning for web agents reframes browser use as cooperation among specialized roles rather than a single monolithic policy. WebPilot (Zhang et al., 2025q) models web task execution as a multi-agent system with a global planning agent that decomposes tasks and local MCTS-based executors that solve subtasks, jointly steering search in complex web environments. INFOGENT (Reddy et al., 2025b) organizes web information aggregation into a Navigator, Extractor, and Aggregator, so exploration, evidence extraction, and synthesis are handled by distinct cooperating agents with feedback from the Aggregator to guide future navigation; WINELL (Reddy et al., 2025a) leverages agentic web search to plan and execute iterative information gathering for discovering timely factual updates relevant to a target Wikipedia article.. Recon-Act (He et al., 2025c) adopts a Reconnaissance–Action paradigm in which a Recon team analyzes successful and failed trajectories to derive generalized tools or hints, and an Action team re-plans and executes with this evolving toolset. PAE (Zhou et al., 2025e) uses three roles, namely a task proposer, an acting web agent and a VLM-based evaluator, to autonomously generate vision-based web tasks and feed success signals back into the policy via RL. Synthetic-data and multi-step RL pipelines further enrich this training loop by generating tool-use trajectories before optimizing agent behavior (Goldie et al., 2025). Hierarchical web agents such as Agent-E (Abuelsaad et al., 2024) and Plan-and-Act (Wang et al., 2023b) similarly separate a high-level planner from a browser-navigation agent, enabling structured plan–execute cooperation. At a more conceptual level, Agentic Web (Yang et al., 2025e) envisions the internet as an agentic web of interacting agents and analyzes how coordination, communication protocols and economic incentives shape such ecosystems, while Agentic Deep Research (Zhang et al., 2025o) frames information seeking as iterative feedback loops of reasoning, retrieval, and synthesis that can be instantiated by single- or multi-agent web research systems.

Multi-agent designs for GUI agents typically decompose "using a computer" into cooperating roles that plan, perceive, decide and execute. COLA (Zhao et al., 2025b) instantiates a scenario-aware task scheduler, a planner, a decision-agent pool, an executor, and a reviewer, so UI tasks are split into basic capability units and routed to domain-specialized agents rather than a single monolith. On mobile, Mobile-Agent-v2 (Wang et al., 2024e) adopts a tri-role pattern with planner, decision, and reflection agents for progress navigation, local action selection, and error correction, while Mobile-Agent-E (Wang et al., 2025m) further builds a hierarchical stack with a Manager and four subordinate agents (i.e. Perceptor, Operator, Action Reflector, Notetaker) plus a self-evolution module that learns long-term Tips and Shortcuts from experience. Mobile-Agent-V (Wang et al., 2025e) similarly employs a video agent, decision agent, and reflection agent to coordinate multi-modal perception and execution, and MobileExperts (Zhang et al., 2024g) dynamically forms teams of expert agents with a dual-layer planner that allocates subtasks to tool-specialized experts. SWIRL (Lu et al., 2025d) makes this structure explicit for RL, training a Navigator that converts language and screen context into structured plans and an Interactor that grounds those plans into atomic GUI actions within a multi-agent RL workflow. Collaborative adaptation methods such as CoLA (Zhou et al., 2025f) suggest another path for coordination, where agents or model components share low-rank updates rather than only exchanging natural-language messages. PC Agent (Liu et al., 2025b) uses separate planning and grounding agents in a two-stage pipeline for desktop automation, illustrating how multi-agent decomposition can improve long-horizon PC control.

To facilitate autonomous research agents, multi-agent collaboration enables a single model's linear workflow to become a coordinated research group: specialized agents operate in parallel, exchange intermediate artifacts through explicit interfaces, and provide adversarial or complementary feedback to improve both creativity and rigor (de Lafuente et al., 2026; Kong et al., 2026; Ghareeb et al., 2025; Aygün et al., 2026).

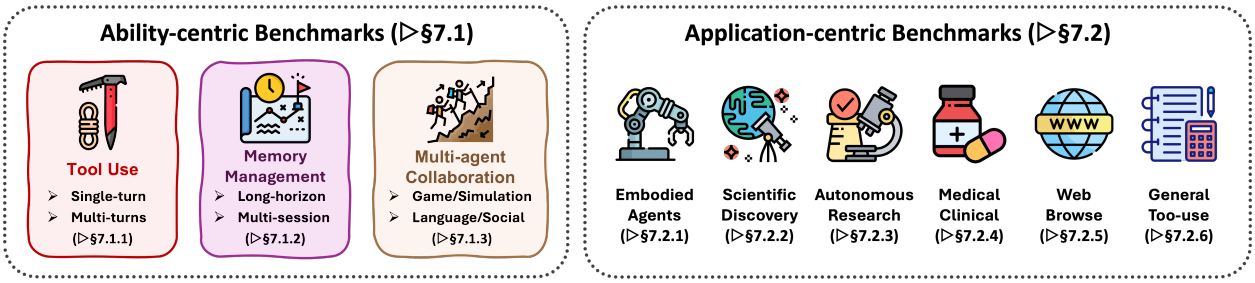

Figure 12: An overview of the benchmarks on agentic reasoning.

For example, AgentRxiv (Schmidgall & Moor, 2025) coordinates author, reviewer, and editor agents that iteratively refine manuscripts and share evolving artifacts across virtual "labs." ARIA (Ramirez-Medina et al., 2025) instantiates a role-structured multi-LLM team that searches, filters, and synthesizes scientific literature into actionable experimental procedures. Co-Scientist (Gottweis et al., 2026) introduces a Gemini-based multi-agent system that generates, critiques, and refines novel scientific hypotheses, demonstrating experimentally validated discoveries in biomedical applications such as drug repurposing, target discovery, and antimicrobial resistance. Earlier multi-agent designs such as CAMEL (Qi et al., 2023) demonstrate how cooperative role-play with tool access can enhance hypothesis generation and task decomposition. In experimental sciences, Coscientist (Boiko et al., 2023b) integrates planning, robotic instrument control, and analysis into a multi-agent closed loop that autonomously designs and executes wet-lab experiments. Finally, TAIS (Liu et al., 2024b) defines a hierarchical team, namely project manager, data engineer and domain expert, that jointly discovers disease-predictive genes from expression data through coordinated division of labor.

7 Benchmarks

Agentic reasoning has been evaluated through a rapidly growing set of benchmarks, but existing suites often differ in what they treat as the core capability, such as tool invocation accuracy, memory retention under long contexts, or coordination quality in multi-agent settings. To provide a coherent view, we organize benchmarks from two complementary perspectives. We first summarize benchmarks that isolate core mechanisms of agentic reasoning, which helps pinpoint where systems succeed or fail at the capability level. We then review application-level benchmarks that evaluate end-to-end agent behavior in realistic domains, capturing the combined effects of perception, planning, tool use, memory, and coordination.

7.1 Core Mechanisms of Agentic Reasoning

We begin with benchmarks that target mechanism-level capabilities, aiming to evaluate agentic reasoning in a more controlled and interpretable manner. Concretely, these benchmarks decompose agentic behavior into a small set of recurring primitives, including tool use, search, memory and planning, and multi-agent coordination. Such mechanism-centric evaluations make it easier to attribute performance changes to specific components, and they complement end-to-end benchmarks that may conflate multiple sources of errors.

7.1.1 Tool Use

Evaluating tool-using models remains an open challenge due to the diversity of tasks, tools, and usage scenarios involved (Qin et al., 2025). The key difficulties arise from the wide range of available tools, varying levels of scenario complexity, and the prevalence requirements specifically for the task domain.

Single-Turn Tool Use. While agentic reasoning often focuses on multi-turn or long-horizon interactions, single-turn tool use remains a foundational capability for evaluating LLMs' basic tool invocation skills. ToolQA (Zhuang et al., 2023) constructs a dataset of 1,530 dialogues involving 13 specialized tools, designed to assess LLMs' ability to interface with external knowledge sources in a question-answering context. APIBench (Patil et al., 2024) introduces a large-scale benchmark grounded in real-world APIs from HuggingFace, TorchHub,

and TensorHub, comprising 1,645 unique APIs and 16,450 instruction–API pairs. It is used to train and evaluate Gorilla, an LLM capable of invoking a broad range of APIs, emphasizing generalization across diverse tool interfaces. ToolLLM-ToolBench (Qin et al., 2024b) curates 16,464 real-world APIs across 49 categories from the RapidAPI Hub, and uses ChatGPT to generate diverse, instruction-style prompts for these APIs. The benchmark is used to train ToolLLaMA, a model that demonstrates strong tool-use capabilities and exhibits promising generalization to unseen APIs. MetaTool (Huang et al., 2024c) introduces the TOOLE dataset, containing over 20,000 entries and a benchmark comprising approximately 200 tools across diverse scenarios, including software engineering, finance, and art design. It splits tool selection tasks to tool selection with similar choices, tool selection in specific scenarios , tool selection with possible reliability issues, and multi-tool selection. T-Eval (Chen et al., 2024c) decomposes tool utilization into a series of sub-processes: instruction following, planning, reasoning, retrieval, understanding, and review, and evaluates each step individually to provide a fine-grained assessment of tool-use capabilities. The benchmark includes a total of 23,305 test cases spanning 15 different tools. GTA (General Tool Agents) (Wang et al., 2024d) targets realistic tool-use scenarios by emphasizing real user queries, real-world deployed tools, and multimodal inputs. It introduces 229 challenging tasks grounded in practical applications, spanning 14 tools across diverse domains. ToolRet (Shi et al., 2025d) focuses specifically on the task of tool retrieval, introducing a heterogeneous benchmark consisting of 7.6K diverse retrieval tasks and a corpus of 43K tools.

Multi-Turn Tool Use. Multi-turn tool use offers a more realistic simulation of real-world applications, where agents autonomously select and sequence tools to solve complex tasks. ToolAlpaca (Tang et al., 2023a) is one of the earliest efforts in this direction, using multi-agent simulations to generate 3,938 tool-use instances from over 400 real-world APIs across 50 distinct categories. SambaNova-ToolBench (Xu et al., 2023c) introduces a benchmark centered on software tool manipulation for real-world tasks, with varying levels of API complexity to test agent capabilities. API-Bank (Li et al., 2023c) provides a dataset of 1,888 tool-use dialogues from 2,138 APIs, along with a runnable evaluation system containing 73 APIs and 314 tool-use test cases. UltraTool (Huang et al., 2024a) evaluates tool-use capabilities across six dimensions: planning awareness, planning ability, creation, tool-use awareness, tool selection, and tool usage. The benchmark spans 22 domains, includes 2,032 tools, and provides 5,824 evaluation samples. ToolFlow distinguishes itself from prior benchmarks by emphasizing long-term planning. It features 224 expert-curated tasks involving 107 real-world tools, highlighting challenges in goal decomposition and multi-step decision-making. More recently, MTU-Bench (Wang et al., 2025g) presents a multi-granularity benchmark for multi-turn, multi-tool scenarios, and releases MTU-Instruct, a large-scale instruction dataset containing 54,798 dialogues involving 136 tools. m & m's introduces a benchmark with over 4,000 multi-step, multimodal tasks involving 33 tools, including multimodal models, public APIs, and image processing modules. It also provides a high-quality subset of 1,565 task plans that are human-verified and executable end-to-end. GTA-2 (Wang et al., 2026b) evaluates general tool-use agents from short atomic calls to long-horizon, open-ended workflows with real user queries, deployed tools, and multimodal contexts. AAR (Kim et al., 2026) tests whether agents can handle fork-merge tool chains over Wikipedia rather than simple linear tool-use sequences. AgentEscapeBench (Guo et al., 2026b) evaluates whether agents can infer, execute, and revise novel tool-use procedures under long-range dependency constraints. RobustBench-TC (Zhou et al., 2026) evaluates tool-use agents under realistic deployment perturbations such as noisy inputs, unreliable APIs, and tool registry errors.

7.1.2 Search

To systematically assess an agent's ability to acquire information through interaction, recent benchmarks cast search as a sequential reasoning problem and can be broadly categorized into unimodal and multimodal settings, differing in the nature of evidence sources, interaction spaces, and grounding requirements.

Unimodal Search. Recent benchmarks for single-modal agentic search increasingly frame information seeking as a sequential, decision-driven process, emphasizing planning, interaction, and evidence synthesis. For example, WebWalker (Wu et al., 2025c) emphasizes structured website traversal, explicitly modeling search as coordinated horizontal exploration and vertical drilling across interconnected pages. To reflect realistic open-world information seeking, InfoDeepSeek (Xi et al., 2025) introduces a dynamic Web setting with verifiable yet non-curated answers, highlighting robustness to noise and distributional shift. Several

benchmarks scale search along temporal and informational dimensions: Mind2Web 2 (Gou et al., 2025) focuses on long-horizon browsing and citation-grounded synthesis, whereas RAVine (Xu et al., 2025i) augments answer quality with process-level efficiency and interaction fidelity. Complementarily, WideSearch (Wong et al., 2025) and DeepWideSearch (Lan et al., 2025) distinguish between breadth-oriented large-scale fact aggregation and depth-oriented multi-hop reasoning, revealing the difficulty of jointly optimizing coverage and reasoning coherence. Domain-specific benchmarks further stress reliability under strict correctness constraints: MedBrowseComp (Chen et al., 2025e) targets clinical decision support by requiring agents to integrate heterogeneous and potentially conflicting medical evidence, while FinAgentBench (Choi et al., 2025) evaluates retrieval-centric reasoning in financial analysis through document-type selection and fine-grained passage localization. Finally, LocalSearchBench (He et al., 2025b) grounds agentic search in real-world local services, evaluating multi-constraint, multi-entity reasoning over large structured databases. Collectively, these benchmarks redefine agentic search evaluation around planning depth, interaction quality, evidence integration, and real-world fidelity, providing a more holistic assessment of search-centric reasoning in language-based agents.

Multimodal Search. Recent benchmarks on multimodal agentic search move beyond static multimodal question answering to systematically evaluate an agent's ability to actively retrieve, browse, and reason over heterogeneous information sources under realistic constraints (Paul et al., 2026). Benchmarks such as MMSearch (Jiang et al., 2024a) and its extension MMSearch-Plus (Tao et al., 2025a) frame multimodal search as an end-to-end process, where agents must interpret multimodal queries and synthesize answers by jointly leveraging textual and visual evidence, explicitly modeling different input–output modality configurations. Complementing this setting, MM-BrowseComp (Li et al., 2025h) adapts the "hard-to-find, easy-to-verify" paradigm to multimodal web environments, enforcing mandatory image dependence to prevent text-only shortcuts and to stress-test multimodal evidence grounding during open-web browsing. BEARCUBS (Song et al., 2025a) further emphasizes computer-using agents in live web scenarios, requiring explicit interaction trajectories and multimodal manipulation (e.g., videos or 3D navigation), thereby evaluating not only retrieval accuracy but also procedural competence. Moving into domain-specific and tool-augmented regimes, PaperArena (Wang et al., 2026a) evaluates multimodal agentic search in scientific workflows, where agents must coordinate PDF parsing, figure understanding, database queries, and web search to answer research-level questions. AgentSearchBench (Wu et al., 2026) introduces a large-scale benchmark for "agent search in the wild," evaluating how well systems can retrieve and rerank suitable AI agents from nearly 10,000 real-world agents across multiple providers. Finally, Video-Browser (Liang et al., 2026a) and VideoDR (Liu et al., 2026a) extend agentic search to video-centric settings, requiring agents to extract visual-temporal cues from videos and iteratively validate hypotheses via open-web evidence, with carefully designed constraints to ensure dual dependence on video and external retrieval. KARLBench (Chang et al., 2026) evaluates enterprise search agents across constraint-driven search, cross-document synthesis, tabular reasoning, exhaustive retrieval, and procedural reasoning. Together, these benchmarks delineate a clear evolution toward evaluating multimodal agents as interactive researchers, highlighting planning, tool use, and multimodal evidence integration as first-class capabilities in agentic search.

7.1.3 Memory and Planning

A distinctive advantage of agents lies in their ability to leverage memory to achieve accurate long-term performance and strong reasoning capabilities. This ability can be assessed from two complementary perspectives. The first concerns memory management, which reflects how effectively an agent integrates, organizes, and retrieves long-term memories. The second concerns memory utilization, which captures how well an agent exploits historical information to support planning and informed feedback. In this section, we separately discuss benchmarks from these two aspects.

From the perspective of memory management, existing benchmarks can be broadly categorized into *Long-Horizon Episodic Memory* and *Multi-session Recall*, depending on whether the textual context consists of a single continuous long-form input or multiple discontinuous conversational sessions.

Long-Horizon Episodic Memory. This category targets single-episode tasks with partial observability and delayed rewards, requiring agents to store and retrieve information over extended time spans. Benchmarks

in this space evaluate memory retention, retrieval, and reasoning across long contexts. ATM-Bench proposes a multimodal, multi-source long-term personalized memory QA benchmark for testing whether AI assistants can retrieve, resolve references, and reason over years of user-specific memory (Mei et al., 2026). PerLTQA (Du et al., 2024a) simulates personalized dialogue, where agents answer questions using long-term persona and event memories. It includes 8.5K QA pairs and evaluates memory classification, retrieval ranking, and synthesis fidelity. ELITR-Bench (Thonet et al., 2024) tests QA on noisy meeting transcripts, where relevant evidence may appear far earlier than the query. Models are scored via GPT-4 across various ASR noise levels and dialogue settings. In the meanwhile, Multi-IF (He et al., 2024c) and MultiChallenge (Sirdeshmukh et al., 2025) focus on multi-turn instruction following. Multi-IF (He et al., 2024c) spans 4.5K tri-turn conversations in 8 languages, with evaluation based on strict and relaxed instruction accuracy. MultiChallenge (Sirdeshmukh et al., 2025) tests four memory-intensive phenomena: retention, inference, editing, and coherence, using 273 curated dialogues with binary pass/fail evaluation. TurnBench-MS (Zhang et al., 2025t) evaluates multi-step reasoning across 540 symbolic logic games, tracking win rate, round-level accuracy, and verifier usage. StoryBench (Wan & Ma, 2025) casts memory as decision-making in interactive narratives, where agents must remember prior choices to progress. It assesses decision accuracy, retry counts, and runtime efficiency. MemBench (Tan et al., 2025a) tests factual and reflective memory across 60K episodes in participatory and observational settings, with metrics for accuracy, recall, capacity, and retrieval speed. MMRC (Xue et al., 2025a) develops a multimodal memory benchmark focused on single-round multimodal conversations. EvoMemBench (Wang et al., 2026d) systematically evaluates agent memory across in-episode/cross-episode and knowledge/execution-oriented settings. Together, these benchmarks emphasize structured memory demands, with metrics capturing not just task success but also memory precision, synthesis quality, and robustness under long-context stress.

Multi-session Recall. Multi-session Recall focuses on multi-episode tasks where agents must retain and integrate knowledge across separate sessions, supporting lifelong adaptation and mitigating catastrophic forgetting. A range of recent benchmarks systematically probe this capability under realistic, long-term interaction scenarios. LOCOMO (Maharana et al., 2024) evaluates LLM agents on sustained conversational memory across 19-session dialogues, using tasks such as multi-hop QA, event summarization, and multi-modal response generation. MemSim (Zhang et al., 2024m) introduces a simulator-based framework with over 2,900 synthetic trajectories in daily life domains, assessing fact retention across sessions via accuracy, diversity, and rationality scores. LONGMEMEVAL (Wu et al., 2024a) benchmarks assistants on five sub-tasks: information extraction, multi-session reasoning, temporal inference, knowledge updating, and abstention, over dialogue histories spanning up to 1.5M tokens, with GPT-4 judged accuracy and retrieval recall. REALTALK (Lee et al., 2025a) presents 21-day real human conversations with 17K tokens per dyad, enabling evaluation of memory probing and persona simulation through multi-hop QA and emotional grounding metrics. Furthermore, MemoryAgentBench (Hu et al., 2025d) unifies diverse memory tasks such as test-time learning, conflict resolution, and long-range understanding across multiple datasets, with task-specific metrics including classification accuracy, partial-match F1, and ROUGE. Mem-Gallery (Bei et al., 2026) introduces a multimodal long-term memory evaluation benchmark that systematically covers a wide range of memory management and utilization scenarios. Lastly, Evo-Memory (Wei et al., 2025c) introduces a benchmark and a unified evaluation protocol for measuring experience reuse in test-time learning. Collectively, these benchmarks underscore the importance of dynamic memory integration across sessions and provide comprehensive evaluations across factual recall, adaptation, and reasoning.

From the perspective of memory utilization, we provide a detailed discussion of benchmarks that evaluate an agent's ability to support planning and feedback using historical information.

Planning and Feedback. Benchmarks targeting planning and feedback primarily assess whether agents can effectively utilize memory to support multi-step planning based on environmental feedback, and maintain coherent internal state over extended interactions. First, ALFWorld (Shridhar et al., 2020) employs interactive environments to evaluate the consistency of multi-step planning, requiring agents to accumulate observations across actions and maintain latent internal states throughout execution. Moreover, formal planning benchmarks such as PlanBench (Valmeekam et al., 2023a) and ACPBench (Kokel et al., 2025) assess planning capabilities in explicitly defined dynamic environments, testing whether agents can correctly reason about action

preconditions, effects, reachability, and overall plan validity. TEXT2WORLD (Hu et al., 2025a) integrate fragmented textual descriptions into a coherent and executable world model, evaluating the capacity to continuously consolidate historical facts into structured planning representations. PlanningBench (Zhao et al., 2026c) introduces a scalable and automatically verifiable framework for generating diverse planning tasks to evaluate and train LLMs on structured multi-step planning ability. NARCBench (Rose et al., 2026) evaluates whether collusion in multi-agent systems can be detected by aggregating signals across agents. More recent benchmarks place greater emphasis on feedback integration and planning under non-stationary conditions. For example, REALM-Bench (Geng & Chang, 2025) introduces dynamic disturbances in real-world manufacturing scenarios, requiring agents to remember prior commitments and replan when underlying assumptions are violated, while TravelPlanner (Xie et al., 2024a) focuses on accurate itinerary construction under constrained and evolving information. Finally, FlowBench (Xiao et al., 2024a) and UrbanPlanBench (Zheng et al., 2025b) assess planning performance in procedural and domain-specific settings, respectively, where agents must preserve conversational or policy context and apply it consistently across decision steps. Together, these benchmarks go beyond one-shot plan generation and systematically investigate whether agents can leverage historical information to support sustained planning, adaptive feedback integration, and iterative decision revision over time.

7.1.4 Multi-Agent System

To evaluate coordination, competition, and decision making beyond isolated reasoning, recent benchmarks situate multi-agent systems in interactive environments. These works broadly span game-based evaluations, simulation-centric real-world scenarios, and language-driven social reasoning tasks.

Game-based reinforcement learning evaluation. Game-based reinforcement learning evaluation benchmarks leverage classical and novel gaming environments to systematically compare the performance of multi-agent RL algorithms under cooperative and adversarial settings. MAgent (Zheng et al., 2018) facilitates massive-scale multi-agent scenarios such as pursuit and resource competition within customizable grid-worlds, evaluating individual cumulative rewards and competitive metrics like resource occupancy rates. Pommerman (Resnick et al., 2018) adapts the classic Bomberman game for cooperative and adversarial interactions, quantifying performance through win rates, survival duration, and kill-to-suicide ratios. SMAC (Samvelyan et al., 2019) centers on decentralized micromanagement challenges in StarCraft II scenarios, evaluating team success via win rates, average damage output, and formation dispersion. MineLand (Yu et al., 2024b) utilizes Minecraft as a realistic ecological simulation for large-scale multi-agent coordination, with up to 64 agents cooperating to meet physical needs under partial observability. TeamCraft (Long et al., 2024) also employs Minecraft to benchmark embodied multi-modal agents tasked with interpreting visual, textual, and environmental prompts to collaboratively achieve 55,000 procedurally generated task instances. Melting Pot (Leibo et al., 2021) assesses agents' zero-shot generalization capabilities in diverse social dilemma environments, utilizing metrics such as per-capita return, social welfare, and inequality indices. BenchMARL (Bettini et al., 2024) provides standardized algorithm comparisons across multiple scenarios (e.g., SMACv2, VMAS, MPE), measuring convergence rates, final performance, and hyperparameter sensitivity. Finally, Arena (Song et al., 2020) encompasses a comprehensive suite of cooperative and adversarial games across various complexities, evaluating individual returns, collective social welfare, and emergent communication protocols.

Simulation-centric real-world assessment. Simulation-centric real-world benchmarks simulate realistic or pseudo-realistic environments, emphasizing scalability, partial observability, and dynamic planning. SMARTS (Zhou et al., 2020) offers a scalable multi-agent driving platform for real-world traffic scenarios like merges and intersections, with evaluation based on collision rates, task completion, and agent behavior distributions. Nocturne (Vinitsky et al., 2022) provides high-throughput, partially observable driving simulations using Waymo trajectories, testing coordination and human-like behavior in tasks such as intersections and roundabouts. MABIM (Yang et al., 2023) benchmarks multi-echelon inventory management, simulating cooperative and competitive retail dynamics, evaluated via profit metrics across diverse inventory settings. IMP-MARL (Leroy et al., 2023) addresses infrastructure inspection and maintenance scheduling, measuring risk reduction and cost efficiency in large-scale systems. POGEMA (Skrynnik et al., 2024) focuses on decentralized multi-agent pathfinding in grids, tracking success rate, path efficiency, and large-scale coordination.

INTERSECTIONZOO (Jayawardana et al., 2024) studies contextual RL for cooperative eco-driving at intersections, using traffic simulations to evaluate emissions and travel-time performance. REALM-Bench (Geng & Chang, 2025) introduces real-world planning tasks from logistics to disaster relief, with dynamic disruptions, multi-threaded dependencies, and evaluation via planning quality, adaptability, and constraint satisfaction. Together, these benchmarks reflect challenges in scaling, uncertainty, coordination, and dynamic adaptation, offering rigorous testbeds for real-world multi-agent systems.

Language, Communication, and Social Reasoning. Benchmarks in Language, Communication, and Social Reasoning explore multi-agent communication protocols, Theory-of-Mind reasoning, game-theoretic interactions, and language-driven coordination. LLM-Coordination (Agashe et al., 2023) examines collaborative reasoning and joint-planning abilities of LLM agents through cooperative gameplay (e.g., Hanabi, Overcooked-AI), measured by holistic scores and fine-grained coordination question accuracy. AVALONBENCH (Light et al., 2023) leverages the social deduction game Avalon to assess role-conditioned language-based reasoning, with datasets of thousands of five-player dialogues and metrics on win-rate, role accuracy, and voting dynamics. Welfare Diplomacy (Mukobi et al., 2023) extends the classic game Diplomacy to general-sum welfare negotiation, using 50-game datasets to quantify coalition stability and welfare-oriented strategic reasoning. MAgIC (Xu et al., 2023b) covers social deduction and classic dilemmas (e.g., Chameleon, Prisoner's Dilemma), employing handcrafted scenario datasets to benchmark reasoning, deception, coordination, and rationality. BattleAgentBench (Wang et al., 2024g) assesses language-based cooperative and competitive dynamics in strategic gameplay environments, scoring navigation accuracy, agent interactions, and exploitability across diverse map datasets. COMMA (Ossowski et al., 2025) evaluates multimodal communicative reasoning through collaborative puzzle-solving tasks involving visual-language coordination, measured by grounding accuracy, privacy compliance, and dialogue effectiveness across thousands of scenarios. IntellAgent (Levi & Kadar, 2025) introduces synthetic conversational AI tasks in retail and airline domains, generating extensive policy-constrained dialogue datasets evaluated by conversational success, mistake frequency, and policy adherence. Finally, MultiAgentBench (Zhu et al., 2025b) provides a comprehensive assessment across tasks such as Minecraft building, coding, and bargaining, employing dynamic key-performance indicators and LLM-scored communication quality across various multi-agent topologies and scenarios.

7.2 Applications of Agentic Reasoning

While mechanism-centric benchmarks help isolate individual capabilities, real-world deployments require these capabilities to work together under realistic constraints, such as partial observability, long-horizon dependencies, and safety-critical decisions. We therefore next review application-level benchmarks that evaluate end-to-end agent performance across representative environments, with tasks that jointly stress perception, reasoning, action execution, and coordination.

In this subsection, we review benchmarks designed to evaluate the application-level performance of agentic reasoning systems across various domains. These benchmarks assess agents' ability to perceive, reason, and act in realistic or high-impact task settings. We organize the discussion into six categories based on the application environment: *Embodied Agents*, *Scientific Discovery Agents*, *Autonomous Research Agents*, *Medical and Clinical Agents*, *Web Agents*, and *Tool-Use Agents*. Each subsubsection introduces representative benchmarks and describes their design motivation, task format, and evaluation metrics.

7.2.1 Embodied Agents

Benchmarks under this category evaluate agents that interact with physical or simulated environments, requiring grounding, perception, and action planning. AgentX (Ashraf et al., 2025) provides a diverse suite of vision-language embodied tasks in driving and sports, where agents must make decisions using multimodal information from videos. It emphasizes reasoning across scenes with occlusions, temporal gaps, or distractors. BALROG (Paglieri et al., 2024) builds a reinforcement learning-centric framework for benchmarking agentic planning in game environments, focusing on instruction-following, temporal abstraction, and error correction. ALFWorld (Shridhar et al., 2020) links language instructions to object interactions in a text-based 3D environment, evaluating perception-grounded execution. AndroidArena (Xing et al., 2024) targets GUI-based mobile tasks, where agents must perform actions like form-filling and app navigation using vision-language

understanding. StarDojo (Tan et al., 2025b) leverages the open-ended Stardew Valley game to study social planning and role-based coordination. MindAgent (Gong et al., 2023), repeated-game evaluations (Akata et al., 2025) and NetPlay (Jeurissen et al., 2024) create multiplayer gaming testbeds to benchmark emergent social reasoning and negotiation under uncertainty. OSWorld (Xie et al., 2024c) offers a simulated desktop environment with diverse cross-app productivity tasks, such as opening files, converting formats, and modifying documents. OSI-Bench (Wu et al., 2025h) introduces a large-scale open-world spatial reasoning benchmark, showing that current MLLMs rely heavily on linguistic priors and fail to generalize from indoor spatial tasks to real-world metric and kinematic reasoning. EnactToM (Juneja et al., 2026) tests whether embodied agents can act on implicit beliefs in partially observable multi-agent household tasks. EgoSound (Zhu et al., 2026a) introduces the first benchmark for evaluating multimodal LLMs' sound understanding in egocentric videos, covering sound perception, spatial localization, causal inference, and cross-modal reasoning. These environments challenge agents to coordinate between perception, planning, and symbolic action in dynamic and often partially observable scenarios.

7.2.2 Scientific Discovery Agents

Scientific benchmarks aim to test agents' capabilities in knowledge acquisition, hypothesis generation, and experimental automation. DISCOVERYWORLD (Jansen et al., 2024) introduces a virtual lab where agents explore scientific phenomena in biology, chemistry, and physics through simulated tools and instruments. ScienceWorld (Wang et al., 2022) focuses on elementary science experiments using textual instructions and environment interactions, requiring step-by-step hypothesis testing. SUPERChem (Zhao et al., 2025h) introduces a leakage-resistant multimodal chemistry reasoning benchmark for evaluating LLMs on expert-level chemical knowledge, reasoning processes, and problem-solving beyond simple answer accuracy. ScienceAgent-Bench (Chen et al., 2024d) builds a benchmark from real-world scientific papers, translating tasks like code implementation, figure generation, and variable extraction into executable subtasks, assessing agents' ability to automate the research process. The AI Scientist (Lu et al., 2024a) simulates a full end-to-end research pipeline, where agents perform literature review, method writing, experiment execution, and peer-review simulation. LAB-Bench (Laurent et al., 2024) evaluates biology-specific agents on tasks involving genetic sequence reasoning and experiment planning. MLAgentBench (Huang et al., 2023c) benchmarks agents' ability to autonomously train, evaluate, and tune machine learning models, offering realistic experimentation workflows. DiscoverPhysics (Wiemann et al., 2026) asks agents to interactively discover laws of motion in simulated worlds whose physics differs from our own. MolClaw (Zhang et al., 2026a) introduces MolBench to evaluate autonomous molecular agents on planning and verification across molecular discovery scenarios. LongAct (Zhu et al., 2026b) evaluates long-horizon household task execution with free-form instructions, dependency management, memory, and adaptive planning. These benchmarks collectively probe open-ended reasoning, long-horizon planning, and scientific grounding in semi-structured data settings.

7.2.3 Autonomous Research Agents

This category benchmarks agents designed for long-horizon workflows across general-purpose research, office, or planning tasks. FrontierScience (Wang et al., 2026c) introduces a benchmark of expert-level scientific tasks to evaluate whether AI systems can perform rigorous, verifiable reasoning across disciplines such as physics, chemistry, biology, and mathematics. WorkArena (Drouin et al., 2024) and its extension WorkArena++ (Boisvert et al., 2025) propose enterprise task benchmarks where agents must complete ticket-based workflows involving retrieval, summarization, and coordination across documents. OfficeBench (Wang et al., 2024m) simulates a productivity software suite environment with tasks such as creating meeting memos, modifying spreadsheets, and replying to emails, emphasizing goal decomposition and tool selection. PlanBench (Valmeekam et al., 2023a) and FlowBench (Xiao et al., 2024a) test general workflow planning skills with abstracted task graphs and structured dependencies. ACPBench (Kokel et al., 2025) evaluates agents in assistant–collaborator–planner triads, tracking performance in a hybrid role hierarchy. TRAIL (Deshpande et al., 2025) focuses on multi-agent trace debugging and error attribution (Zhang et al., 2024k) in LLM-based systems, providing dense annotations for reasoning chains. PaperMind (Zhao et al., 2026a) evaluates agents' tool-use ability for scientific paper understanding, including evidence retrieval and reasoning over figures, tables, and text. CLIN (Majumder et al., 2023) introduces lifelong few-shot learning benchmarks where agents adapt to distribution shift and task evolution. Agent-as-a-Judge (Zhuge et al., 2024b) studies peer-review style

evaluation with agents grading reasoning chains and correctness of other agents' outputs. AutoMat (Huang et al., 2026) benchmarks coding agents on computational scientific reproducibility in materials science. PRBench (Qiu et al., 2026) evaluates whether agents can reproduce physics research papers end-to-end from scientific materials and computational workflows. InfoDeepSeek (Xi et al., 2025) measures information-seeking abilities in open-domain QA and synthesis tasks. PaperWritingBench (Song et al., 2026) evaluates whether agents can transform raw research materials into submission-ready LaTeX papers. AutoResearchBench (Xiong et al., 2026a) measures whether autonomous research agents can discover relevant scientific literature for research problems and evidence verification. Together, these benchmarks capture the growing demand for agentic reasoning in complex knowledge workflows that involve abstraction, iteration, and evaluation.

7.2.4 Medical and Clinical Agents

These benchmarks test agents' abilities to reason with clinical knowledge, patient data, and multimodal biomedical sources. AgentClinic (Schmidgall et al., 2024) introduces a virtual hospital environment where agents make diagnostic decisions based on patient symptoms and medical imaging. MedAgentBench (Jiang et al., 2025c) combines medical QA, patient simulation, and retrieval tasks in a multi-format benchmark grounded in standardized exams. MedAgentsBench (Tang et al., 2025c) evaluates multi-hop medical reasoning over structured and unstructured data, scoring agents on correctness and evidence alignment. EHRAgent (Shi et al., 2024a) benchmarks agents working over structured electronic health record (EHR) tables and clinical notes to complete tasks like diagnosis code prediction and medication reasoning. MedBrowseComp (Chen et al., 2025e) focuses on browsing-based medical QA, where agents must retrieve and verify information across web pages. ACC (Thakrar et al., 2025) explores trustworthy medical agents with retrieval, hallucination detection, and citation-based support evaluation. MedAgents (Tang et al., 2023b) uses a collaborative multi-agent dialogue setup to simulate patient–doctor–nurse interactions, scoring fluency and factual accuracy. ABRA (Maksudov et al., 2026) evaluates radiology agents that must operate imaging viewers and DICOM tools rather than answer from pre-selected images. PhysicianBench (Liu et al., 2026b) provides execution-grounded evaluation for clinical agents operating in realistic EHR environments. CHI-Bench (Chen et al., 2026b) evaluates agents on long-horizon healthcare operations requiring policy grounding, multi-role handoffs, and tool-based workflow execution. GuardAgent (Xiang et al., 2024b) proposes a clinical privacy safeguard agent with structured risk detection benchmarks on EHR and website forms. These datasets emphasize correctness, trustworthiness, and safety in real-world clinical deployment contexts.

7.2.5 Web Agents

Web agents operate in realistic browsing environments and are benchmarked on their ability to parse layouts, execute actions, and handle dynamic content. WebShop (Yao et al., 2023a) provides an early scalable shopping environment where agents must ground language instructions in product search, comparison, and purchase actions. WebArena (Zhou et al., 2023b) introduces a browser-based benchmark suite containing 90+ realistic websites across domains like shopping and booking, where agents complete tasks with structured goals and click-based APIs. VisualWebArena (Koh et al., 2024a) extends this with visual rendering, requiring agents to parse webpage images and align instructions with rendered components, while VisualWebBench (Liu et al., 2024d) isolates web-page understanding and grounding for multimodal LLMs. WebVoyager (He et al., 2024b) proposes goal-driven navigation with long-horizon tasks involving multi-page traversal and backtracking. Mind2Web (Gou et al., 2025) targets cross-domain web automation with multi-task datasets and rich grounding annotations, whereas Web-Bench (Xu et al., 2025c) evaluates code-level web standards and framework knowledge. WebCanvas (Pan et al., 2024) supports fine-grained layout manipulation, such as drag-drop and resize actions. WebLINX (Lù et al., 2024) simulates information gathering tasks with browsing, summarization, and answer synthesis. AppWorld (Trivedi et al., 2024), OmniACT (Kapoor et al., 2024), and OSWorld (Xie et al., 2024d) broaden the setting from websites to app, desktop, and web tasks for generalist autonomous agents. BrowseComp (Wei et al., 2025a) and BrowseComp-ZH (Zhou et al., 2025b) brings language and infrastructure diversity with Chinese websites, challenging agents on multilingual understanding. LASER (Ma et al., 2023), WebWalker (Wu et al., 2025c), and AutoWebBench (Lai et al., 2024) focus on structured page representation, real-time action execution, and policy learning in web navigation. SaaS-Bench (Shi et al., 2026) evaluates computer-use agents in realistic professional workflows across web

browsers and GUI-based SaaS applications. These benchmarks highlight perception, grounding, and policy generalization challenges in web settings.

7.2.6 General Tool-Use Agents

This group of benchmarks emphasizes LLM agents' ability to invoke, coordinate, and reason over tools and APIs. GTA (Wang et al., 2024d) presents a realistic tool-use benchmark grounded in user queries and deployed software tools, spanning APIs from image generation to analytics dashboards. NESTFUL (Basu et al., 2025) evaluates nested API invocation tasks requiring compositional planning across toolchains. RestGPT (Song et al., 2023b) connects LLMs with RESTful APIs via coarse-to-fine planning pipelines, tested on 60+ tool types. Search-o1 (Li et al., 2025i) frames tool use as sequential retrieval, with benchmarks spanning code search, PDF querying, and scientific tool usage. Agentic RL (Singh et al., 2025b) proposes a reinforcement learning agent with access to tool interfaces and evaluation tasks such as calendar scheduling and translation. ActionReasoningBench (Handa et al., 2024) benchmarks agents' ability to reason about action side effects and downstream consequences using a structured action grammar. R-Judge (Yuan et al., 2024c) introduces safety judgment benchmarks where agents assess risky plans involving tools. Workspace-Bench (Tang et al., 2026b) evaluates agents on realistic workspace tasks requiring cross-file retrieval, reasoning, and adaptive decisions over large file dependencies. CodeAct (Wang et al., 2024h) simulates executable function calling and evaluates agents on parsing, composition, and runtime accuracy. SWE Atlas (Raghavendra et al., 2026) introduces a benchmark suite for coding agents that evaluates real-world software engineering workflows beyond issue fixing, including codebase QA, test writing, and refactoring. DevOps-Gym (Tang et al., 2026a) introduces an end-to-end benchmark for evaluating AI agents across real-world DevOps workflows, including build/configuration, monitoring, issue resolving, and test generation. ProgramBench (Yang et al., 2026) introduces a benchmark for evaluating whether language models can rebuild complete software projects from scratch from high-level specifications, testing multi-file implementation, dependency handling, and cross-language reimplementation ability. CocoaBench (Team et al., 2026) evaluates unified digital agents on long-horizon tasks requiring flexible composition of vision, search, and coding. These datasets jointly reflect the increasing complexity and compositionality of tool-augmented agent environments.

8 Open Problems

In this section, we highlight open problems arising from user-centric personalization, long-horizon interaction and credit assignment, world-model-based reasoning, multi-agent collaboration and training, latent internal reasoning, and the governance of agentic systems operating autonomously in real-world environments.

8.1 User-centric Agentic Reasoning and Personalization

User-centric agentic reasoning (Qian et al., 2025c; Li et al., 2025d) refers to an agent's ability to tailor its reasoning and actions to a specific individual user by modeling user characteristics, preferences, and interaction history over time. Rather than optimizing a fixed, task-defined objective, a user-centric agent treats the user as part of the environment and continuously adapts its strategy through extended, multi-turn interaction. Recent personalization benchmarks such as LaMP and PersonaLens show that user profiles, interaction histories, and task constraints must be evaluated jointly, rather than reducing personalization to generic response quality (Salemi et al., 2024; Zhao et al., 2025i). These studies also expose a gap between retrieving user-specific evidence and deciding when it should override default assistant behavior, especially when preferences are ambiguous or stale. At the system level, this requires the agent to dynamically infer evolving user intent, accommodate changes in goals and behavior styles, and adjust decisions based on explicit or implicit user feedback. Crucially, user-centric agentic reasoning involves balancing short-term task rewards with long-term user experience, satisfaction, and trust, which introduces non-stationary objectives and long-horizon credit assignment challenges beyond conventional agentic reasoning settings.

8.2 Long-horizon Agentic Reasoning from Extended Interaction

A central open challenge in agentic reasoning is robust long-horizon planning and credit assignment across extended interactions. While methods such as ReAct and Tree of Thought improve short-horizon reasoning (Yao et al., 2023c;b), errors still compound rapidly in long tasks, as illustrated by embodied agents like Voyager (Wang et al., 2023a). Benchmarks such as τ-bench reveal the evaluation side of this problem: realistic tool-agent-user interaction requires rule following and behavioral consistency over repeated trials, not merely single-run task completion (Yao et al., 2024). RL-trained agents such as WebRL and Agent-R1 improve performance in realistic environments but rely on heavily engineered, domain-specific rewards and largely treat episodes independently (Qi et al., 2024; Wei et al., 2025e). More recent process-aware approaches attempt to construct finer-grained credit signals (Xiang et al., 2024a; Yan et al., 2025; Ji et al., 2025), while general agent-training frameworks and turn-level reward designs expose the optimization side by decomposing trajectories into trainable transitions or denser feedback (Luo et al., 2025c; Wei et al., 2025b). Yet these methods still require reliable definitions of progress, recovery, and failure attribution in open-ended environments. A core open problem is how to assign credit across tokens, tool calls, skills, and memory updates, and to generalize such learning across a long sequence of episodes and tasks.

8.3 Agentic Reasoning with World Models

World-model-based agents (Carbonneaux et al., 2025; Zhang et al., 2025k) aim to mitigate myopic reasoning by enabling internal simulation and lookahead. From early neural world models to DreamerV3, TD-MPC2, and Genie, model-based systems show how imagined dynamics can support policy learning, long-horizon control, and interactive environment generation (Ha & Schmidhuber, 2018; Hafner et al., 2023; Hansen et al., 2024; Bruce et al., 2024). Recent LLM-based agents adapt world models to web, code, and GUI environments (Tang et al., 2024; Carbonneaux et al., 2025; Chae et al., 2024; Luo et al., 2025a). However, current designs rely on ad hoc representations and are typically trained on short-horizon or environment-specific data, raising concerns about calibration and generalization. Only a few works explore co-evolving world models and agents over long time scales (Fang et al., 2025c; Deng et al., 2026). An open problem is how to jointly train, update, and evaluate world models in non-stationary environments, and how to assess their causal impact on downstream planning reliability.

8.4 Multi-agent Collaborative Reasoning and Training

Multi-agent collaboration has emerged as a powerful paradigm for scaling agentic reasoning through role specialization and division of labor (Li et al., 2023a; Chen et al., 2023b; Wu et al., 2024b). Systems such as MetaGPT show that encoding human organizational routines and role-specific verification can reduce cascading errors in complex collaborations (Hong et al., 2023). While debate- and role-based systems often outperform single agents, most collaboration structures are still manually designed. Recent multi-agent RL approaches begin to treat collaboration itself as a trainable skill (Liu et al., 2026c; Park et al., 2025; Ma et al., 2024b), but credit assignment at the group level remains poorly understood. Beyond LLM-agent implementations, cooperative AI research emphasizes that coordination quality depends on incentives, welfare, and social interaction, not only on aggregate task throughput (Dafoe et al., 2020). Scaling to larger agent populations further introduces challenges in topology adaptation, coordination overhead, and safety (Qian et al., 2024b; Grötschla et al., 2025; Zhang et al., 2024k). A key open problem is how to learn adaptive, interpretable collaboration policies that remain robust under partial observability and adversarial conditions.

8.5 Latent Agentic Reasoning

Latent agentic reasoning (Du et al., 2025; Fu et al., 2025; Zou et al., 2025e) explores performing planning, decision-making, and collaboration in internal latent spaces rather than explicit natural language or symbolic traces. Coconut shows that continuous thoughts can encode alternatives before committing to explicit text, while Quiet-STaR and pause tokens explore unstated rationales or delayed computation as complementary forms of hidden reasoning (Hao et al., 2024b; Zelikman et al., 2024; Goyal et al., 2024). These directions can improve efficiency and scalability, but at the cost of reduced interpretability and controllability. In agentic

settings, this raises additional challenges, including how to align latent reasoning with external objectives, tools, agents, and memory systems. Diagnosing failures becomes particularly difficult when intermediate reasoning steps are not externally observable, especially if latent states directly affect tool invocation, memory editing, or communication with other agents. An open problem is how to design learning objectives, probing methods, and evaluation benchmarks that make latent agentic reasoning both effective and auditable.

8.6 Governance of Agentic Reasoning

Governance is a cross-cutting challenge for agentic reasoning systems that act autonomously over tools, environments, and other agents. Beyond standard LLM safety issues, agentic systems introduce new risks due to long-horizon planning, persistent memory, and real-world action execution (Wang et al., 2025f). ToolEmu and realistic evaluations of autonomous agents show that tool-enabled agents can leak private data, cause financial losses, or pursue risky autonomous behaviors even when individual language outputs appear plausible (Ruan et al., 2024; Kinniment et al., 2023). Failures may arise from interactions across time and components, making attribution and auditing difficult. Existing benchmarks and guardrails mainly focus on short-horizon behaviors (Xiang et al., 2024b; Yuan et al., 2024c), leaving planning-time failures and multi-agent dynamics underexplored. More general safety and governance work highlights the need for dangerous-capability evaluations, deployment-time monitoring, and reporting or compliance mechanisms for high-impact systems (Shevlane et al., 2023; Anderljung et al., 2023). A central open problem is to develop governance frameworks that jointly address model-level alignment, agent-level policies, and ecosystem-level interactions under realistic deployment conditions.

References

Mahyar Abbasian, Iman Azimi, Amir M Rahmani, and Ramesh Jain. Conversational health agents: A personalized llm-powered agent framework. *arXiv preprint arXiv:2310.02374*, 2023.

Tamer Abuelsaad, Deepak Akkil, Prasenjit Dey, Ashish Jagmohan, Aditya Vempaty, and Ravi Kokku. Agent-e: From autonomous web navigation to foundational design principles in agentic systems. *arXiv preprint arXiv:2407.13032*, 2024.

Manish Acharya, Yifan Zhang, Kevin Leach, and Yu Huang. Optimizing code runtime performance through context-aware retrieval-augmented generation. In *2025 IEEE/ACM 33rd International Conference on Program Comprehension (ICPC)*, pp. 1–5. IEEE Computer Society, 2025.

Saaket Agashe, Yue Fan, Anthony Reyna, and Xin Eric Wang. Llm-coordination: evaluating and analyzing multi-agent coordination abilities in large language models. *arXiv preprint arXiv:2310.03903*, 2023.

Saaket Agashe, Jiuzhou Han, Shuyu Gan, Jiachen Yang, Ang Li, and Xin Eric Wang. Agent s: An open agentic framework that uses computers like a human. *arXiv preprint arXiv:2410.08164*, 2024.

Vaibhav Aggarwal, Ojasv Kamal, Abhinav Japesh, Zhijing Jin, and Bernhard Schölkopf. DARS: Dynamic action re-sampling to enhance coding agent performance by adaptive tree traversal, 2025.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

LangChain AI. Langchain library. 2023. URL `https://www.langchain.com/`.

Elif Akata, Lion Schulz, Julian Coda-Forno, Seong Joon Oh, Matthias Bethge, and Eric Schulz. Playing repeated games with large language models. *Nature Human Behaviour*, 9(7):1380–1390, May 2025. ISSN 2397-3374. doi: 10.1038/s41562-025-02172-y. URL `http://dx.doi.org/10.1038/s41562-025-02172-y`.

Renat Aksitov, Sobhan Miryoosefi, Zonglin Li, Daliang Li, Sheila Babayan, Kavya Kopparapu, Zachary Fisher, Ruiqi Guo, Sushant Prakash, Pranesh Srinivasan, et al. Rest meets react: Self-improvement for multi-step reasoning llm agent. *arXiv preprint arXiv:2312.10003*, 2023.

Mohammad Almansoori, Komal Kumar, and Hisham Cholakkal. Self-evolving multi-agent simulations for realistic clinical interactions. *arXiv preprint arXiv:2503.22678*, 2025.

AlphaProof and AlphaGeometry teams. AI achieves silver-medal standard solving International Mathematical Olympiad problems, 2024. URL https://deepmind.google/discover/blog/ai-solves-imo-problems-at-silver-medal-level.

Markus Anderljung, Joslyn Barnhart, Anton Korinek, Jade Leung, Cullen O'Keefe, Jess Whittlestone, Shahar Avin, Miles Brundage, Justin Bullock, Duncan Cass-Beggs, Ben Chang, Tantum Collins, Tim Fist, Gillian Hadfield, Alan Hayes, Lewis Ho, Sara Hooker, Eric Horvitz, Noam Kolt, Jonas Schuett, Yonadav Shavit, Divya Siddarth, Robert Trager, and Kevin Wolf. Frontier AI regulation: Managing emerging risks to public safety, 2023. URL https://arxiv.org/abs/2307.03718.

Mehrad Ansari and Seyed Mohamad Moosavi. Agent-based learning of materials datasets from the scientific literature. *Digital Discovery*, 3(12):2607–2617, 2024.

Antonis Antoniades, Albert Örwall, Kexun Zhang, Yuxi Xie, Anirudh Goyal, and William Wang. Swe-search: Enhancing software agents with monte carlo tree search and iterative refinement. *arXiv preprint arXiv:2410.20285*, 2024.

Abrar Anwar, John Welsh, Joydeep Biswas, Soha Pouya, and Yan Chang. Remembr: Building and reasoning over long-horizon spatio-temporal memory for robot navigation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2838–2845. IEEE, 2025.

Shuang Ao, Flora D Salim, and Simon Khan. Emac+: Embodied multimodal agent for collaborative planning with vlm+ llm. *arXiv preprint arXiv:2505.19905*, 2025.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Self-reflective retrieval augmented generation. In *NeurIPS 2023 workshop on instruction tuning and instruction following*, 2023.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=hSyW5go0v8.

Tajamul Ashraf, Amal Saqib, Hanan Ghani, Muhra AlMahri, Yuhao Li, Noor Ahsan, Umair Nawaz, Jean Lahoud, Hisham Cholakkal, Mubarak Shah, et al. Agent-x: Evaluating deep multimodal reasoning in vision-centric agentic tasks. *arXiv preprint arXiv:2505.24876*, 2025.

Mihir Athale and Vishal Vaddina. Knowledge graph based repository-level code generation. In *2025 IEEE/ACM International Workshop on Large Language Models for Code (LLM4Code)*, pp. 169–176. IEEE, May 2025. doi: 10.1109/llm4code66737.2025.00026. URL http://dx.doi.org/10.1109/LLM4Code66737.2025.00026.

Reza Averly, Frazier N Baker, Ian A Watson, and Xia Ning. Liddia: Language-based intelligent drug discovery agent. *arXiv preprint arXiv:2502.13959*, 2025.

Eser Aygün, Anastasiya Belyaeva, Gheorghe Comanici, Marc Coram, Hao Cui, Jake Garrison, Renee Johnston Anton Kast, Cory Y. McLean, Peter Norgaard, Zahra Shamsi, David Smalling, James Thompson, Subhashini Venugopalan, Brian P. Williams, Chujun He, Sarah Martinson, Martyna Plomecka, Lai Wei, Yuchen Zhou, Qian-Ze Zhu, Matthew Abraham, Erica Brand, Anna Bulanova, Jeffrey A. Cardille, Chris Co, Scott Ellsworth, Grace Joseph, Malcolm Kane, Ryan Krueger, Johan Kartiwa, Dan Liebling, Jan-Matthis Lueckmann, Paul Raccuglia, Xuefei, Wang, Katherine Chou, James Manyika, Yossi Matias, John C. Platt, Lizzie Dorfman, Shibl Mourad, and Michael P. Brenner. An ai system to help scientists write expert-level empirical software, 2026. URL https://arxiv.org/abs/2509.06503.

Ruhana Azam, Tamer Abuelsaad, Aditya Vempaty, and Ashish Jagmohan. Multimodal auto validation for self-refinement in web agents. *arXiv preprint arXiv:2410.00689*, 2024.

Ruhana Azam, Aditya Vempaty, and Ashish Jagmohan. Reflection-based memory for web navigation agents. *arXiv preprint arXiv:2506.02158*, 2025.

Alisson Azzolini, Junjie Bai, Hannah Brandon, Jiaxin Cao, Prithvijit Chattopadhyay, Huayu Chen, Jinju Chu, Yin Cui, Jenna Diamond, Yifan Ding, et al. Cosmos-reason1: From physical common sense to embodied reasoning. *arXiv preprint arXiv:2503.15558*, 2025.

Qianqian Bai, Zhongpu Chen, Ling Luo, Huaming Du, Yuqian Lei, and Ziyun Jiao. Endowing embodied agents with spatial reasoning capabilities for vision-and-language navigation, 2026. URL `https://arxiv.org/abs/2504.08806`.

Ye Bai, Minghan Wang, and Thuy-Trang Vu. Maple: Multi-agent adaptive planning with long-term memory for table reasoning. *arXiv preprint arXiv:2506.05813*, 2025.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Ramakrishna Bairi, Atharv Sonwane, Aditya Kanade, Vageesh D C, Arun Iyer, Suresh Parthasarathy, Sriram Rajamani, Balasubramanyan Ashok, and Shashank Shet. Codeplan: Repository-level coding using llms and planning. *Proceedings of the ACM on Software Engineering*, 1(FSE):675–698, 2024.

Kinjal Basu, Ibrahim Abdelaziz, Kiran Kate, Mayank Agarwal, Maxwell Crouse, Yara Rizk, Kelsey Bradford, Asim Munawar, Sadhana Kumaravel, Saurabh Goyal, et al. Nestful: A benchmark for evaluating llms on nested sequences of api calls. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 33526–33535, 2025.

Yuanchen Bei, Tianxin Wei, Xuying Ning, Yanjun Zhao, Zhining Liu, Xiao Lin, Yada Zhu, Hendrik Hamann, Jingrui He, and Hanghang Tong. Mem-gallery: Benchmarking multimodal long-term conversational memory for mllm agents. *arXiv preprint arXiv:2601.03515*, 2026.

Raja Ben Abdessalem, Shiva Nejati, Lionel C. Briand, and Thomas Stifter. Testing advanced driver assistance systems using multi-objective search and neural networks. In *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, ASE '16, pp. 63–74, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450338455. doi: 10.1145/2970276.2970311. URL `https://doi.org/10.1145/2970276.2970311`.

Shelly Bensal, Umar Jamil, Christopher Bryant, Melisa Russak, Kiran Kamble, Dmytro Mozolevskyi, Muayad Ali, and Waseem AlShikh. Reflect, retry, reward: Self-improving llms via reinforcement learning. *arXiv preprint arXiv:2505.24726*, 2025.

Yarin Benyamin, Argaman Mordoch, Shahaf S. Shperberg, and Roni Stern. Toward pddl planning copilot, 2025. URL `https://arxiv.org/abs/2509.12987`.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, pp. 17682–17690, 2024.

Matteo Bettini, Amanda Prorok, and Vincent Moens. Benchmarl: Benchmarking multi-agent reinforcement learning. *Journal of Machine Learning Research*, 25(217):1–10, 2024.

Xiaohe Bo, Zeyu Zhang, Quanyu Dai, Xueyang Feng, Lei Wang, Rui Li, Xu Chen, and Ji-Rong Wen. Reflective multi-agent collaboration based on large language models. *Advances in Neural Information Processing Systems*, 37:138595–138631, 2024.

Daniil A. Boiko, Robert MacKnight, and Gabe Gomes. Emergent autonomous scientific research capabilities of large language models, 2023a. URL `https://arxiv.org/abs/2304.05332`.

Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023b.

Léo Boisvert, Megh Thakkar, Maxime Gasse, Massimo Caccia, Thibault Le Sellier De Chezelles, Quentin Cappart, Nicolas Chapados, Alexandre Lacoste, and Alexandre Drouin. Workarena++: Towards compositional planning and reasoning-based common knowledge work tasks, 2025. URL `https://arxiv.org/abs/2407.05291`.

Fouad Bousetouane. Physical ai agents: Integrating cognitive intelligence with real-world action. *arXiv preprint arXiv:2501.08944*, 2025.

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

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

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments, 2024. URL `https://arxiv.org/abs/2402.15391`.

The Viet Bui, Tien Mai, and Hong Thanh Nguyen. O-mapl: Offline multi-agent preference learning. *arXiv preprint arXiv:2501.18944*, 2025.

Tianle Cai, Xuezhi Wang, Tengyu Ma, Xinyun Chen, and Denny Zhou. Large language models as tool makers. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=qV83K9d5WB`.

Yue Cao and CS Lee. Robot behavior-tree-based task generation with large language models. *arXiv preprint arXiv:2302.12927*, 2023.

Quentin Carbonneaux, Gal Cohen, Jonas Gehring, Jacob Kahn, Jannik Kossen, Felix Kreuk, Emily McMilin, Michel Meyer, Yuxiang Wei, David Zhang, et al. Cwm: An open-weights llm for research on code generation with world models. *arXiv preprint arXiv:2510.02387*, 2025.

Hyungjoo Chae, Namyoung Kim, Kai Tzu-iunn Ong, Minju Gwak, Gwanwoo Song, Jihoon Kim, Sunghwan Kim, Dongha Lee, and Jinyoung Yeo. Web agents with world models: Learning and leveraging environment dynamics in web navigation. *arXiv preprint arXiv:2410.13232*, 2024.

Jonathan D. Chang, Andrew Drozdov, Shubham Toshniwal, Owen Oertell, Alexander Trott, Jacob Portes, Abhay Gupta, Pallavi Koppol, Ashutosh Baheti, Sean Kulinski, Ivan Zhou, Irene Dea, Krista Opsahl-Ong, Simon Favreau-Lessard, Sean Owen, Jose Javier Gonzalez Ortiz, Arnav Singhvi, Xabi Andrade, Cindy Wang, Kartik Sreenivasan, Sam Havens, Jialu Liu, Peyton DeNiro, Wen Sun, Michael Bendersky, and Jonathan Frankle. Karl: Knowledge agents via reinforcement learning, 2026. URL `https://arxiv.org/abs/2603.05218`.

Tianyou Chang, Shizhan Chen, Guodong Fan, and Zhiyong Feng. A self-iteration code generation method based on large language models. In *International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 275–281, 2023.

Guangyao Chen, Siwei Dong, Yu Shu, Ge Zhang, Jaward Sesay, Börje F Karlsson, Jie Fu, and Yemin Shi. Autoagents: A framework for automatic agent generation. *arXiv preprint arXiv:2309.17288*, 2023a.

Guanzhong Chen, Shaoxiong Yang, Chao Li, Wei Liu, Jian Luan, and Zenglin Xu. Heterogeneous group-based reinforcement learning for llm-based multi-agent systems. *arXiv preprint arXiv:2506.02718*, 2025a.

Guhong Chen, Liyang Fan, Zihan Gong, Nan Xie, Zixuan Li, Ziqiang Liu, Chengming Li, Qiang Qu, Hamid Alinejad-Rokny, Shiwen Ni, et al. Agentcourt: Simulating court with adversarial evolvable lawyer agents. *arXiv preprint arXiv:2408.08089*, 2024a.

Guoxin Chen, Zile Qiao, Wenqing Wang, Donglei Yu, Xuanzhong Chen, Hao Sun, Minpeng Liao, Kai Fan, Yong Jiang, Penguin Xie, Wayne Xin Zhao, Ruihua Song, and Fei Huang. Mars: Co-evolving dual-system deep research via multi-agent reinforcement learning, 2026a. URL https://arxiv.org/abs/2510.04935.

Haolin Chen, Deon Metelski, Leon Qi, Tao Xia, Joonyul Lee, Steve Brown, Kevin Riley, Frank Wang, T. Y. Alvin Liu, Hank Capps MD, Zeyu Tang, Xiangchen Song, Lingjing Kong, Fan Feng, Tianyi Zeng, Zhiwei Liu, Zixian Ma, Hang Jiang, Fangli Geng, Yuan Yuan, Chenyu You, Qingsong Wen, Hua Wei, Yanjie Fu, Yue Zhao, Carl Yang, Biwei Huang, Kun Zhang, Caiming Xiong, Sanmi Koyejo, Eric P. Xing, Philip S. Yu, and Weiran Yao. Chi-bench: Can ai agents automate end-to-end, long-horizon, policy-rich healthcare workflows?, 2026b. URL https://arxiv.org/abs/2605.16679.

Junjie Chen, Haitao Li, Jingli Yang, Yiqun Liu, and Qingyao Ai. Enhancing llm-based agents via global planning and hierarchical execution. *arXiv preprint arXiv:2504.16563*, 2025b.

Kexin Chen, Jiamin Lu, Junyou Li, Xiaoran Yang, Yuyang Du, Kunyi Wang, Qiannuan Shi, Jiahui Yu, Lanqing Li, Jiezhong Qiu, Jianzhang Pan, Yi Huang, Qun Fang, Pheng Ann Heng, and Guangyong Chen. Chemist-x: Large language model-empowered agent for reaction condition recommendation in chemical synthesis, 2025c. URL https://arxiv.org/abs/2311.10776.

Liyi Chen, Panrong Tong, Zhongming Jin, Ying Sun, Jieping Ye, and Hui Xiong. Plan-on-graph: Self-correcting adaptive planning of large language model on knowledge graphs. *Advances in Neural Information Processing Systems*, 37:37665–37691, 2024b.

Mingyang Chen, Linzhuang Sun, Tianpeng Li, Haoze Sun, Chenzheng Zhu, Haofen Wang, Jeff Pan, Wen Zhang, Huajun Chen, Fan Yang, et al. Learning to reason with search for llms via reinforcement learning. *Advances in Neural Information Processing Systems*, 38:85287–85307, 2026c.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*, 2025d.

Shan Chen, Pedro Moreira, Yuxin Xiao, Sam Schmidgall, Jeremy Warner, Hugo Aerts, Thomas Hartvigsen, Jack Gallifant, and Danielle S Bitterman. Medbrowsecomp: Benchmarking medical deep research and computer use. *arXiv preprint arXiv:2505.14963*, 2025e.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *The Twelfth International Conference on Learning Representations*, 2023b.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*, 2023c. URL https://openreview.net/forum?id=YfZ4ZPt8zd.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. Teaching large language models to self-debug. *arXiv preprint arXiv:2304.05128*, 2023d.

Yifei Chen, Guanting Dong, and Zhicheng Dou. Toward effective tool-integrated reasoning via self-evolved preference learning, 2025f. URL https://arxiv.org/abs/2509.23285.

Yixing Chen, Yiding Wang, Siqi Zhu, Haofei Yu, Tao Feng, Muhan Zhang, Mostofa Patwary, and Jiaxuan You. Multi-agent evolve: Llm self-improve through co-evolution, 2025g. URL https://arxiv.org/abs/2510.23595.

Zehui Chen, Weihua Du, Wenwei Zhang, Kuikun Liu, Jiangning Liu, Miao Zheng, Jingming Zhuo, Songyang Zhang, Dahua Lin, Kai Chen, and Feng Zhao. T-eval: Evaluating the tool utilization capability of large language models step by step. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 9510–9529. Association for Computational Linguistics, 2024c. doi: 10.18653/V1/2024.ACL-LONG.515. URL https://doi.org/10.18653/v1/2024.acl-long.515.

Zhenfang Chen, Delin Chen, Rui Sun, Wenjun Liu, and Chuang Gan. Scaling autonomous agents via automatic reward modeling and planning. *arXiv preprint arXiv:2502.12130*, 2025h.

Zhiliang Chen, Xinyuan Niu, Chuan-Sheng Foo, and Bryan Kian Hsiang Low. Broaden your scope! efficient multi-turn conversation planning for llms with semantic space. *arXiv preprint arXiv:2503.11586*, 2025i.

Zhipeng Chen, Kun Zhou, Beichen Zhang, Zheng Gong, Xin Zhao, and Ji-Rong Wen. ChatCoT: Tool-augmented chain-of-thought reasoning on chat-based large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14777–14790, Singapore, December 2023e. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.985. URL https://aclanthology.org/2023.findings-emnlp.985/.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, Vishal Dey, Mingyi Xue, Frazier N. Baker, Benjamin Burns, Daniel Adu-Ampratwum, Xuhui Huang, Xia Ning, Song Gao, Yu Su, and Huan Sun. Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery. *arXiv preprint arXiv:2410.05080*, 2024d. URL https://www.arxiv.org/abs/2410.05080.

Ziru Chen, Michael White, Raymond Mooney, Ali Payani, Yu Su, and Huan Sun. When is tree search useful for llm planning? it depends on the discriminator. *arXiv preprint arXiv:2402.10890*, 2024e.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. Seeclick: Harnessing gui grounding for advanced visual gui agents. *arXiv preprint arXiv:2401.10935*, 2024.

Pengzhou Cheng, Lingzhong Dong, Zeng Wu, Zongru Wu, Xiangru Tang, Chengwei Qin, Zhuosheng Zhang, and Gongshen Liu. Agent-scankit: Unraveling memory and reasoning of multimodal agents via sensitivity perturbations. *arXiv preprint arXiv:2510.00496*, 2025a.

Weihua Cheng, Ersheng Ni, Wenlong Wang, Yifei Sun, Junming Liu, Wangyu Shen, Yirong Chen, Botian Shi, and Ding Wang. Mga: Memory-driven gui agent for observation-centric interaction. *arXiv preprint arXiv:2510.24168*, 2025b.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

Yuan Chiang, Elvis Hsieh, Chia-Hong Chou, and Janosh Riebesell. Llamp: Large language model made powerful for high-fidelity materials knowledge retrieval and distillation. *arXiv preprint arXiv:2401.17244*, 2024.

Chanyeol Choi, Jihoon Kwon, Alejandro Lopez-Lira, Chaewoon Kim, Minjae Kim, Juneha Hwang, Jaeseon Ha, Hojun Choi, Suyeol Yun, Yongjin Kim, et al. Finagentbench: A benchmark dataset for agentic retrieval in financial question answering. In *Proceedings of the 6th ACM International Conference on AI in Finance*, pp. 632–637, 2025.

Jishnu Ray Chowdhury and Cornelia Caragea. Zero-shot verification-guided chain of thoughts. *arXiv preprint arXiv:2501.13122*, 2025.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Jarosław A Chudziak and Adam Kostka. Ai-powered math tutoring: Platform for personalized and adaptive education. In *International Conference on Artificial Intelligence in Education*, pp. 462–469. Springer, 2025.

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

Katherine M Collins, Albert Q Jiang, Simon Frieder, Lionel Wong, Miri Zilka, Umang Bhatt, Thomas Lukasiewicz, Yuhuai Wu, Joshua B Tenenbaum, William Hart, et al. Evaluating language models for mathematics through interactions. *Proceedings of the National Academy of Sciences*, 121(24):e2318124121, 2024.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Cristina Cornelio, Flavio Petruzzellis, and Pietro Lio. Hierarchical planning for complex tasks with knowledge graph-rag and symbolic verification. *arXiv preprint arXiv:2504.04578*, 2025.

Logan Cross, Violet Xiang, Agam Bhatia, Daniel LK Yamins, and Nick Haber. Hypothetical minds: Scaffolding theory of mind for multi-agent tasks with large language models. *arXiv preprint arXiv:2407.07086*, 2024.

Mingxuan Cui, Yilan Jiang, Duo Zhou, Cheng Qian, Yuji Zhang, and Qiong Wang. Shortagesim: Simulating drug shortages under information asymmetry. *arXiv preprint arXiv:2509.01813*, 2025.

Xing Cui, Yueying Zou, Zekun Li, Peipei Li, Xinyuan Xu, Xuannan Liu, and Huaibo Huang. T2agent: A tool-augmented multimodal misinformation detection agent with monte carlo tree search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 175–183, 2026.

Allan Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R. McKee, Joel Z. Leibo, Kate Larson, and Thore Graepel. Open problems in cooperative ai, 2020. URL `https://arxiv.org/abs/2012.08630`.

Nicola Dainese, Matteo Merler, Minttu Alakuijala, and Pekka Marttinen. Generating code world models with large language models guided by monte carlo tree search. In *Conference on Neural Information Processing Systems (NeurIPS)*, pp. 60429–60474, 2024.

Alan Dao and Thinh Le. Rezero: Enhancing llm search ability by trying one-more-time. *arXiv preprint arXiv:2504.11001*, 2025.

Kourosh Darvish, Marta Skreta, Yuchi Zhao, Naruki Yoshikawa, Sagnik Som, Miroslav Bogdanovic, Yang Cao, Han Hao, Haoping Xu, Alán Aspuru-Guzik, et al. Organa: A robotic assistant for automated chemistry experimentation and characterization. *Matter*, 8(2), 2025.

Alex Davies, Petar Veličković, Lars Buesing, Sam Blackwell, Daniel Zheng, Nenad Tomašev, Richard Tanburn, Peter Battaglia, Charles Blundell, András Juhász, Marc Lackenby, Geordie Williamson, Demis Hassabis, and Pushmeet Kohli. Advancing mathematics by guiding human intuition with AI. *Nature*, (7887):70–74, 2021. doi: 10.1038/s41586-021-04086-x.

Francisco Angulo de Lafuente, Teerth Sharma, Vladimir Veselov, Seid Mohammed Abdu, Nirmal Tej Kumar, and Guillermo Perry. Openclaw-p2p v7.0-p2pclaw: Resilient multi-layer persistence, live reference verification, and production-scale evaluation of decentralized ai peer review v7.0 – mathematical corrections & ecosystem developments edition, 2026. URL `https://arxiv.org/abs/2604.19792`.

Mingkai Deng, Jinyu Hou, Zhiting Hu, and Eric Xing. General agentic planning through simulative reasoning with world models, 2026. URL `https://arxiv.org/abs/2507.23773`.

Qixin Deng, Qikai Yang, Ruibin Yuan, Yipeng Huang, Yi Wang, Xubo Liu, Zeyue Tian, Jiahao Pan, Ge Zhang, Hanfeng Lin, et al. Composerx: Multi-agent symbolic music composition with llms. In *The 25th International Society for Music Information Retrieval Conference*, 2024.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114, 2023.

Darshan Deshpande, Varun Gangal, Hersh Mehta, Jitin Krishnan, Anand Kannappan, and Rebecca Qian. Trail: Trace reasoning and agentic issue localization. *arXiv preprint arXiv:2505.08638*, 2025. URL https://www.arxiv.org/abs/2505.08638.

Qianggang Ding, Santiago Miret, and Bang Liu. Matexpert: Decomposing materials discovery by mimicking human experts. *arXiv preprint arXiv:2410.21317*, 2024.

Ruomeng Ding, Chaoyun Zhang, Lu Wang, Yong Xu, Minghua Ma, Wei Zhang, Si Qin, Saravan Rajmohan, Qingwei Lin, and Dongmei Zhang. Everything of thoughts: Defying the law of penrose triangle for thought generation. *arXiv preprint arXiv:2311.04254*, 2023.

Guanting Dong, Licheng Bao, Zhongyuan Wang, Kangzhi Zhao, Xiaoxi Li, Jiajie Jin, Jinghan Yang, Hangyu Mao, Fuzheng Zhang, Kun Gai, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. Agentic entropy-balanced policy optimization, 2025a. URL https://arxiv.org/abs/2510.14545.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. Agentic reinforced policy optimization, 2025b. URL https://arxiv.org/abs/2507.19849.

Qingxiu Dong, Li Dong, Yao Tang, Tianzhu Ye, Yutao Sun, Zhifang Sui, and Furu Wei. Reinforcement pre-training. *arXiv preprint arXiv:2506.08007*, 2025c.

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

Zibin Dong, Jianye Hao, Yifu Yuan, Fei Ni, Yitian Wang, Pengyi Li, and Yan Zheng. Diffuserlite: Towards real-time diffusion planning. *Advances in Neural Information Processing Systems*, 37:122556–122583, 2024b.

Zi-Yi Dou, Cheng-Fu Yang, Xueqing Wu, Kai-Wei Chang, and Nanyun Peng. Re-rest: Reflection-reinforced self-training for language agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 15394–15411, 2024.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. In *International Conference on Machine Learning*, pp. 8469–8488. PMLR, 2023.

Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H Laradji, Manuel Del Verme, Tom Marty, Léo Boisvert, Megh Thakkar, Quentin Cappart, David Vazquez, et al. Workarena: How capable are web agents at solving common knowledge work tasks? *arXiv preprint arXiv:2403.07718*, 2024.

Jianda Du, Youran Sun, and Haizhao Yang. Autonumerics: An autonomous, pde-agnostic multi-agent pipeline for scientific computing, 2026. URL https://arxiv.org/abs/2602.17607.

Yiming Du, Hongru Wang, Zhengyi Zhao, Bin Liang, Baojun Wang, Wanjun Zhong, Zezhong Wang, and Kam-Fai Wong. Perltqa: A personal long-term memory dataset for memory classification, retrieval, and fusion in question answering. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, pp. 152–164, 2024a.

Zhuoyun Du, Lujie Zheng, Renjun Hu, Yuyang Xu, Xiawei Li, Ying Sun, Wei Chen, Jian Wu, Haolei Cai, and Haohao Ying. Llms can simulate standardized patients via agent coevolution. *arXiv preprint arXiv:2412.11716*, 2024b.

Zhuoyun Du, Runze Wang, Huiyu Bai, Zouying Cao, Xiaoyong Zhu, Bo Zheng, Wei Chen, and Haochao Ying. Enabling agents to communicate entirely in latent space. *arXiv preprint arXiv:2511.09149*, 2025.

Zhekai Duan, Yuan Zhang, Shikai Geng, Gaowen Liu, Joschka Boedecker, and Chris Xiaoxuan Lu. Fast ecot: Efficient embodied chain-of-thought via thoughts reuse. *arXiv preprint arXiv:2506.07639*, 2025.

Abhishek Dutta and Yen-Che Hsiao. Adaptive reasoning and acting in medical language agents. *arXiv preprint arXiv:2410.10020*, 2024.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *ArXiv*, abs/2404.16130, 2024. URL `https://api.semanticscholar.org/CorpusID:269363075`.

Jordan S Ellenberg, Cristofero S Fraser-Taliente, Thomas R Harvey, Karan Srivastava, and Andrew V Sutherland. Generative modeling for mathematical discovery. *arXiv preprint arXiv:2503.11061*, 2025.

Assaf Elovic. gpt-researcher, July 2023. URL `https://github.com/assafelovic/gpt-researcher`.

Lutfi Eren Erdogan, Nicholas Lee, Sehoon Kim, Suhong Moon, Hiroki Furuta, Gopala Anumanchipalli, Kurt Keutzer, and Amir Gholami. Plan-and-act: Improving planning of agents for long-horizon tasks. *arXiv preprint arXiv:2503.09572*, 2025.

Adibvafa Fallahpour, Jun Ma, Alif Munim, Hongwei Lyu, and Bo Wang. Medrax: Medical reasoning agent for chest x-ray. *arXiv preprint arXiv:2502.02673*, 2025.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems*, 35:18343–18362, 2022.

Yue Fan, Handong Zhao, Ruiyi Zhang, Yu Shen, Xin Eric Wang, and Gang Wu. Gui-bee: Align gui action grounding to novel environments via autonomous exploration. *arXiv preprint arXiv:2501.13896*, 2025. URL `https://www.arxiv.org/abs/2501.13896`.

Jinyuan Fang, Yanwen Peng, Xi Zhang, Yingxu Wang, Xinhao Yi, Guibin Zhang, Yi Xu, Bin Wu, Siwei Liu, Zihao Li, et al. A comprehensive survey of self-evolving ai agents: A new paradigm bridging foundation models and lifelong agentic systems. *arXiv preprint arXiv:2508.07407*, 2025a.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, et al. Lightmem: Lightweight and efficient memory-augmented generation. *arXiv preprint arXiv:2510.18866*, 2025b.

Tianqing Fang, Hongming Zhang, Zhisong Zhang, Kaixin Ma, Wenhao Yu, Haitao Mi, and Dong Yu. Webevolver: Enhancing web agent self-improvement with coevolving world model. *arXiv preprint arXiv:2504.21024*, 2025c.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in llms. *arXiv preprint arXiv:2504.11536*, 2025a.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *arXiv preprint arXiv:2505.10978*, 2025b.

Xidong Feng, Ziyu Wan, Muning Wen, Stephen Marcus McAleer, Ying Wen, Weinan Zhang, and Jun Wang. Alphazero-like tree-search can guide large language model decoding and training. *arXiv preprint arXiv:2309.17179*, 2023.

Yichun Feng, Jiawei Wang, Lu Zhou, Yikai Zheng, Zhen Lei, and Yixue Li. Real-world doctor agent with proactive consultation through multi-agent reinforcement learning, 2026. URL `https://arxiv.org/abs/2505.19630`.

Dyke Ferber, Omar SM El Nahhas, Georg Wölflein, Isabella C Wiest, Jan Clusmann, Marie-Elisabeth Leßman, Sebastian Foersch, Jacqueline Lammert, Maximilian Tschochohei, Dirk Jäger, et al. Autonomous artificial intelligence agents for clinical decision making in oncology. *arXiv preprint arXiv:2404.04667*, 2024.

Mohamed Amine Ferrag, Norbert Tihanyi, and Merouane Debbah. Reasoning beyond limits: Advances and open problems for llms. *arXiv preprint arXiv:2503.22732*, 2025.

Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, et al. Magentic-one: A generalist multi-agent system for solving complex tasks. *arXiv preprint arXiv:2411.04468*, 2024.

Tianyu Fu, Zihan Min, Hanling Zhang, Jichao Yan, Guohao Dai, Wanli Ouyang, and Yu Wang. Cache-to-cache: Direct semantic communication between large language models. *arXiv preprint arXiv:2510.03215*, 2025.

Kanishk Gandhi, Denise Lee, Gabriel Grand, Muxin Liu, Winson Cheng, Archit Sharma, and Noah D Goodman. Stream of search (sos): Learning to search in language. *arXiv preprint arXiv:2404.03683*, 2024.

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

Chongkai Gao, Haozhuo Zhang, Zhixuan Xu, Zhehao Cai, and Lin Shao. Flip: Flow-centric generative planning as general-purpose manipulation world model. *arXiv preprint arXiv:2412.08261*, 2024.

Hang Gao and Yongfeng Zhang. Inms: Memory sharing for large language model based agents, 2026. URL `https://arxiv.org/abs/2404.09982`.

Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Yiran Wu, et al. A survey of self-evolving agents: What, when, how, and where to evolve on the path to artificial super intelligence. *arXiv preprint arXiv:2507.21046*, 2025b.

Longxi Gao, Li Zhang, Pengzhi Gao, Wei Liu, Jian Luan, and Mengwei Xu. Gui-shift: Enhancing vlm-based gui agents through self-supervised reinforcement learning, 2025c. URL `https://arxiv.org/abs/2505.12493`.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models. In *International Conference on Machine Learning*, pp. 10764–10799. PMLR, 2023.

Shanghua Gao, Richard Zhu, Zhenglun Kong, Ayush Noori, Xiaorui Su, Curtis Ginder, Theodoros Tsiligkaridis, and Marinka Zitnik. Txagent: An ai agent for therapeutic reasoning across a universe of tools. *arXiv preprint arXiv:2503.10970*, 2025d.

Xinzge Gao, Chuanrui Hu, Bin Chen, and Teng Li. Chain-of-memory: Enhancing gui agents for cross-application navigation. *arXiv preprint arXiv:2506.18158*, 2025e.

Aniketh Garikaparthi, Manasi Patwardhan, Lovekesh Vig, and Arman Cohan. Iris: Interactive research ideation system for accelerating scientific discovery. *arXiv preprint arXiv:2504.16728*, 2025.

Longling Geng and Edward Y Chang. Realm-bench: A benchmark for evaluating multi-agent systems on real-world, dynamic planning and scheduling tasks. *arXiv preprint arXiv:2502.18836*, 2025.

Bogdan Georgiev, Javier Gómez-Serrano, Terence Tao, and Adam Zsolt Wagner. Mathematical exploration and discovery at scale. *arXiv preprint arXiv:2511.02864*, 2025.

Alireza Ghafarollahi and Markus J Buehler. Atomagents: Alloy design and discovery through physics-aware multi-modal multi-agent artificial intelligence. *arXiv preprint arXiv:2407.10022*, 2024a.

Alireza Ghafarollahi and Markus J Buehler. Protagents: protein discovery via large language model multi-agent collaborations combining physics and machine learning. *Digital Discovery*, 3(7):1389–1409, 2024b.

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. Robin: A multi-agent system for automating scientific discovery, 2025. URL `https://arxiv.org/abs/2505.13400`.

Fatemeh Ghezloo, Mehmet Saygin Seyfioglu, Rustin Soraki, Wisdom O Ikezogwo, Beibin Li, Tejoram Vivekanandan, Joann G Elmore, Ranjay Krishna, and Linda Shapiro. Pathfinder: A multi-modal multi-agent system for medical diagnostic decision-making applied to histopathology. *arXiv preprint arXiv:2502.08916*, 2025.

Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, et al. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai. *arXiv preprint arXiv:2411.04872*, 2024.

Marc Glocker, Peter Hönig, Matthias Hirschmanner, and Markus Vincze. Llm-empowered embodied agent for memory-augmented task planning in household robotics. *arXiv preprint arXiv:2504.21716*, 2025.

Anna Goldie, Azalia Mirhoseini, Hao Zhou, Irene Cai, and Christopher D. Manning. Synthetic data generation & multi-step rl for reasoning & tool use, 2025. URL `https://arxiv.org/abs/2504.04736`.

Olga Golovneva, Sean O'Brien, Ramakanth Pasunuru, Tianlu Wang, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. Pathfinder: Guided search over multi-step reasoning paths. *arXiv preprint arXiv:2312.05180*, 2023.

Ran Gong, Qiuyuan Huang, Xiaojian Ma, Hoi Vo, Zane Durante, Yusuke Noda, Zilong Zheng, Song-Chun Zhu, Demetri Terzopoulos, Li Fei-Fei, and Jianfeng Gao. Mindagent: Emergent gaming interaction. *arXiv preprint arXiv:2309.09971*, 2023. URL `https://www.arxiv.org/abs/2309.09971`.

Alex J Goodell, Simon N Chu, Dara Rouholiman, and Larry F Chu. Large language model agents can use tools to perform clinical calculations. *npj Digital Medicine*, 8(1):163, 2025.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Petar Sirkovic, Artiom Myaskovsky, Grzegorz Glowaty, Felix Weissenberger, Alessio Orlandi, Dan Popovici, Anil Palepu, Keran Rong, Ryutaro Tanno, Khaled Saab, Fan Zhang, Jacob Blum, Andrew Carroll, Kavita Kulkarni, Nenad Tomašev, Dina Zverinski, Ivor Rendulic, Elahe Vedadi, Florian Hasler, Luka Rimanic, Marina Boia, Ivan Budiselic, Ben Feinstein, Mathias Bellaiche, Tom Sheffer, Jan Freyberg, Jeremy Ratcliff, Ottavia Bertolli, Katherine Chou, Avinatan Hassidim, Burak Gokturk, Amin Vahdat, Yuan Guan, Vikram Dhillon, Eeshit Dhaval Vaishnav, Byron Lee, Tiago R. D. Costa, JoséR. Penadés, Gary Peltz, Yossi Matias, James Manyika, Demis Hassabis, Yunhan Xu, Pushmeet Kohli, Annalisa Pawlosky, Alan Karthikesalingam, and Vivek Natarajan. Accelerating scientific discovery with co-scientist. *Nature*, 2026. doi: 10.1038/s41586-026-10644-y. URL `https://doi.org/10.1038/s41586-026-10644-y`.

Boyu Gou, Zanming Huang, Yuting Ning, Yu Gu, Michael Lin, Weijian Qi, Andrei Kopanev, Botao Yu, Bernal Jiménez Gutiérrez, Yiheng Shu, et al. Mind2web 2: Evaluating agentic search with agent-as-a-judge. *arXiv preprint arXiv:2506.21506*, 2025.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens. In *The Twelfth International Conference on Learning Representations*, 2024. doi: 10.48550/arXiv.2310.02226. URL `https://arxiv.org/abs/2310.02226`.

Florian Grötschla, Luis Müller, Jan Tönshoff, Mikhail Galkin, and Bryan Perozzi. Agentsnet: Coordination and collaborative reasoning in multi-agent llms. *arXiv preprint arXiv:2507.08616v1*, 2025. URL `https://www.arxiv.org/abs/2507.08616v1`.

Lin Guan, Karthik Valmeekam, Sarath Sreedharan, and Subbarao Kambhampati. Leveraging pre-trained large language models to construct and utilize world models for model-based task planning. *Advances in Neural Information Processing Systems*, 36:79081–79094, 2023a.

Xinyan Guan, Jiali Zeng, Fandong Meng, Chunlei Xin, Yaojie Lu, Hongyu Lin, Xianpei Han, Le Sun, and Jie Zhou. Deeprag: Thinking to retrieve step by step for large language models. *arXiv preprint arXiv:2502.01142*, 2025.

Yanchu Guan, Dong Wang, Zhixuan Chu, Shiyu Wang, Feiyue Ni, Ruihua Song, Longfei Li, Jinjie Gu, and Chenyi Zhuang. Intelligent virtual assistants with llm-based process automation. *arXiv preprint arXiv:2312.06677*, 2023b.

Runquan Gui, Zhihai Wang, Jie Wang, Chi Ma, Huiling Zhen, Mingxuan Yuan, Jianye Hao, Defu Lian, Enhong Chen, and Feng Wu. Hypertree planning: Enhancing llm reasoning via hierarchical thinking. *arXiv preprint arXiv:2505.02322*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025a.

JunJia Guo, Yuhang Yao, Jiawei, Zhou, and Jingdi Chen. Vista: An end-to-end benchmark for visual spec-to-web-app coding agents, 2026a. URL `https://arxiv.org/abs/2605.26144`.

Xuehang Guo, Xingyao Wang, Yangyi Chen, Sha Li, Chi Han, Manling Li, and Heng Ji. Syncmind: Measuring agent out-of-sync recovery in collaborative software engineering. In *International Conference on Machine Learning (ICML)*, 2025b.

Zhengkang Guo, Yiyang Li, Lin Qiu, Xiaohua Wang, Jingwen Xv, Dongyu Ru, Xiaoyu Li, Xiaoqing Zheng, Xuezhi Cao, and Xunliang Cai. Agentescapebench: Evaluating out-of-domain tool-grounded reasoning in llm agents, 2026b. URL `https://arxiv.org/abs/2605.07926`.

Tanmay Gupta, Luca Weihs, and Aniruddha Kembhavi. Codenav: Beyond tool-use to using real-world codebases with llm agents. *arXiv preprint arXiv:2406.12276*, 2024.

David Ha and Jürgen Schmidhuber. World models, 2018. URL `https://arxiv.org/abs/1803.10122`.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Dongge Han, Camille Couturier, Daniel Madrigal Diaz, Xuchao Zhang, Victor Rühle, and Saravan Rajmohan. Legomem: Modular procedural memory for multi-agent llm systems for workflow automation. *arXiv preprint arXiv:2510.04851*, 2025a.

Siwei Han, Peng Xia, Ruiyi Zhang, Tong Sun, Yun Li, Hongtu Zhu, and Huaxiu Yao. Mdocagent: A multi-modal multi-agent framework for document understanding, 2025b. URL `https://arxiv.org/abs/2503.13964`.

Yewei Han and Chen Lyu. Multi-stage guided code generation for large language models. *Engineering Applications of Artificial Intelligence*, 139(PA):109491, 2025.

Divij Handa, Pavel Dolin, Shrinidhi Kumbhar, Tran Cao Son, and Chitta Baral. Actionreasoningbench: Reasoning about actions with and without ramification constraints. *arXiv preprint arXiv:2406.04046*, 2024. URL `https://www.arxiv.org/abs/2406.04046`.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *The Twelfth International Conference on Learning Representations*, 2024. doi: 10.48550/arXiv.2310.16828. URL `https://arxiv.org/abs/2310.16828`.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*, 2023a.

Shibo Hao, Tianyang Liu, Zhen Wang, and Zhiting Hu. Toolkengpt: Augmenting frozen language models with massive tools via tool embeddings. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023b. URL `http://papers.nips.cc/paper_files/paper/2023/hash/8fd1a81c882cd45f64958da6284f4a3f-Abstract-Conference.html`.

Shibo Hao, Yi Gu, Haotian Luo, Tianyang Liu, Xiyan Shao, Xinyuan Wang, Shuhua Xie, Haodi Ma, Adithya Samavedhi, Qiyue Gao, et al. LLM reasoners: New evaluation, library, and analysis of step-by-step reasoning with large language models. *arXiv preprint arXiv:2404.05221*, 2024a.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space, 2024b. URL `https://arxiv.org/abs/2412.06769`.

Yilun Hao, Yang Zhang, and Chuchu Fan. Planning anything with rigor: General-purpose zero-shot planning with llm-based formalized programming. *arXiv preprint arXiv:2410.12112*, 2024c.

Chengbo He, Bochao Zou, Xin Li, Jiansheng Chen, Junliang Xing, and Huimin Ma. Enhancing llm reasoning with multi-path collaborative reactive and reflection agents. *arXiv preprint arXiv:2501.00430*, 2024a.

Guanzhong He, Zhen Yang, Jinxin Liu, Bin Xu, Lei Hou, and Juanzi Li. Webseer: Training deeper search agents through reinforcement learning with self-reflection. *arXiv preprint arXiv:2510.18798*, 2025a.

Hang He, Chuhuai Yue, Chengqi Dong, Mingxue Tian, Zhenfeng Liu, Jiajun Chai, Xiaohan Wang, Yufei Zhang, Qun Liao, Guojun Yin, et al. Localsearchbench: Benchmarking agentic search in real-world local life services. *arXiv preprint arXiv:2512.07436*, 2025b.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. Webvoyager: Building an end-to-end web agent with large multimodal models. *arXiv preprint arXiv:2401.13919*, 2024b.

Kaiwen He, Zhiwei Wang, Chenyi Zhuang, and Jinjie Gu. Recon-act: A self-evolving multi-agent browser-use system via web reconnaissance, tool generation, and task execution. *arXiv preprint arXiv:2509.21072*, 2025c.

Yexiao He, Ang Li, Boyi Liu, Zhewei Yao, and Yuxiong He. Medorch: Medical diagnosis with tool-augmented reasoning agents for flexible extensibility. *arXiv preprint arXiv:2506.00235*, 2025d.

Yun He, Di Jin, Chaoqi Wang, Chloe Bi, Karishma Mandyam, Hejia Zhang, Chen Zhu, Ning Li, Tengyu Xu, Hongjiang Lv, et al. Multi-if: Benchmarking llms on multi-turn and multilingual instructions following. *arXiv preprint arXiv:2410.15553*, 2024c.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Chia-Tung Ho, Haoxing Ren, and Brucek Khailany. Verilogcoder: Autonomous Verilog coding agents with graph-based planning and abstract syntax tree (ast)-based waveform tracing tool. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 39, pp. 300–307, 2025.

Samuel Holt, Max Ruiz Luyten, and Mihaela van der Schaar. L2mac: Large language model automatic computer for extensive code generation, 2025. URL `https://arxiv.org/abs/2310.02003`.

Haoyang Hong, Jiajun Yin, Yuan Wang, Jingnan Liu, Zhe Chen, Ailing Yu, Ji Li, Zhiling Ye, Hansong Xiao, Yefei Chen, et al. Multi-agent deep research: Training multi-agent systems with m-grpo. *arXiv preprint arXiv:2511.13288*, 2025.

Sirui Hong, Mingchen Zhuge, Jiaqi Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework, 2023. URL `https://arxiv.org/abs/2308.00352`.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=VtmBAGCN7o`.

Andrew Hoopes, Neel Dey, Victor Ion Butoi, John V. Guttag, and Adrian V. Dalca. Voxelprompt: A vision agent for end-to-end medical image analysis, 2025. URL https://arxiv.org/abs/2410.08397.

Cheng-Yu Hsieh, Si-An Chen, Chun-Liang Li, Yasuhisa Fujii, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, and Tomas Pfister. Tool documentation enables zero-shot tool-usage with large language models, 2023a. URL https://arxiv.org/abs/2308.00675.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8003–8017, 2023b.

Mengkang Hu, Yao Mu, Xinmiao Yu, Mingyu Ding, Shiguang Wu, Wenqi Shao, Qiguang Chen, Bin Wang, Yu Qiao, and Ping Luo. Tree-planner: Efficient close-loop task planning with large language models, 2024a. URL https://arxiv.org/abs/2310.08582.

Mengkang Hu, Tianxing Chen, Yude Zou, Yuheng Lei, Qiguang Chen, Ming Li, Yao Mu, Hongyuan Zhang, Wenqi Shao, and Ping Luo. Text2world: Benchmarking large language models for symbolic world model generation. *arXiv preprint arXiv:2502.13092*, 2025a.

Mengkang Hu, Pu Zhao, Can Xu, Qingfeng Sun, Jian-Guang Lou, Qingwei Lin, Ping Luo, and Saravan Rajmohan. Agentgen: Enhancing planning abilities for large language model based agent via environment and task generation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 496–507, 2025b.

Shengchao Hu, Li Shen, Ya Zhang, and Dacheng Tao. Learning multi-agent communication from graph modeling perspective. *arXiv preprint arXiv:2405.08550*, 2024b.

Yaojie Hu, Qiang Zhou, Qihong Chen, Xiaopeng Li, Linbo Liu, Dejiao Zhang, Amit Kachroo, Talha Oz, and Omer Tripp. Qualityflow: An agentic workflow for program synthesis controlled by llm quality checks, 2025c.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in llm agents via incremental multi-turn interactions. *arXiv preprint arXiv:2507.05257*, 2025d.

Yue Hu, Yuzhu Cai, Yaxin Du, Xinyu Zhu, Xiangrui Liu, Zijie Yu, Yuchen Hou, Shuo Tang, and Siheng Chen. Self-evolving multi-agent collaboration networks for software development. *arXiv preprint arXiv:2410.16946*, 2024c.

Zhaolin Hu, Yixiao Zhou, Zhongan Wang, Xin Li, Weimin Yang, Hehe Fan, and Yi Yang. OSDA agent: Leveraging large language models for de novo design of organic structure directing agents. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025e. URL https://openreview.net/forum?id=9YNyiCJE3k.

Zican Hu, Wei Liu, Xiaoye Qu, Xiangyu Yue, Chunlin Chen, Zhi Wang, and Yu Cheng. Divide and conquer: Grounding llms as efficient decision-making agents via offline hierarchical reinforcement learning. *arXiv preprint arXiv:2505.19761*, 2025f.

Wenyue Hua, Xianjun Yang, Mingyu Jin, Zelong Li, Wei Cheng, Ruixiang Tang, and Yongfeng Zhang. Trustagent: Towards safe and trustworthy llm-based agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 10000–10016, 2024.

Dong Huang, Jie M Zhang, Michael Luck, Qingwen Bu, Yuhao Qing, and Heming Cui. AgentCoder: Multi-agent-based code generation with iterative testing and optimisation, 2023a.

Jerry Huang, Siddarth Madala, Risham Sidhu, Cheng Niu, Hao Peng, Julia Hockenmaier, and Tong Zhang. Rag-rl: Advancing retrieval-augmented generation via rl and curriculum learning. *arXiv preprint arXiv:2503.12759*, 2025a.

Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*, 2023b.

Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*, 2022.

Kexin Huang, Serena Zhang, Hanchen Wang, Yuanhao Qu, Yingzhou Lu, Yusuf Roohani, Ryan Li, Lin Qiu, Gavin Li, Junze Zhang, et al. Biomni: A general-purpose biomedical ai agent. *biorxiv*, 2025b.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. Mlagentbench: Evaluating language agents on machine learning experimentation. *arXiv preprint arXiv:2310.03302*, 2023c. URL `https://www.arxiv.org/abs/2310.03302`.

Shijue Huang, Wanjun Zhong, Jianqiao Lu, Qi Zhu, Jiahui Gao, Weiwen Liu, Yutai Hou, Xingshan Zeng, Yasheng Wang, Lifeng Shang, Xin Jiang, Ruifeng Xu, and Qun Liu. Planning, creation, usage: Benchmarking llms for comprehensive tool utilization in real-world complex scenarios. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pp. 4363–4400. Association for Computational Linguistics, 2024a. doi: 10.18653/V1/2024.FINDINGS-ACL.259. URL `https://doi.org/10.18653/v1/2024.findings-acl.259`.

Tao Huang, Zhihong Sun, Zhi Jin, Ge Li, and Chen Lyu. Knowledge-aware code generation with large language models. In *IEEE/ACM International Conference on Program Comprehension (ICPC)*, pp. 52–63, 2024b.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

Yizheng Huang and Jimmy Huang. A survey on retrieval-augmented text generation for large language models. *arXiv preprint arXiv:2404.10981*, 2024.

Yue Huang, Jiawen Shi, Yuan Li, Chenrui Fan, Siyuan Wu, Qihui Zhang, Yixin Liu, Pan Zhou, Yao Wan, Neil Zhenqiang Gong, and Lichao Sun. Metatool benchmark for large language models: Deciding whether to use tools and which to use. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024c. URL `https://openreview.net/forum?id=R0c2qtalgG`.

Ziyang Huang, Yi Cao, Ali K. Shargh, Jing Luo, Ruidong Mei, Mohd Zaki, Zhan Liu, Wyatt Bunstine, William Jurayj, Somdatta Goswami, Tyrel McQueen, Michael Shields, Jaafar El-Awady, Paulette Clancy, Benjamin Van Durme, Nicholas Andrews, William Walden, and Daniel Khashabi. Can coding agents reproduce findings in computational materials science?, 2026. URL `https://arxiv.org/abs/2605.00803`.

John C. Hull. *Risk Management and Financial Institutions*. Wiley, 5th edition, 2018.

Tatsuro Inaba, Hirokazu Kiyomaru, Fei Cheng, and Sadao Kurohashi. MultiTool-CoT: GPT-3 can use multiple external tools with chain of thought prompting. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1522–1532, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.130. URL `https://aclanthology.org/2023.acl-short.130/`.

Yoshitaka Inoue, Tianci Song, Xinling Wang, Augustin Luna, and Tianfan Fu. Drugagent: Multi-agent large language model-based reasoning for drug-target interaction prediction. *ArXiv*, pp. arXiv–2408, 2025.

Yoichi Ishibashi and Yoshimasa Nishimura. Self-organized agents: A LLM multi-agent framework toward ultra large-scale code generation and optimization, 2024.

Md Ashraful Islam, Mohammed Eunus Ali, and Md Rizwan Parvez. Mapcoder: Multi-agent code generation for competitive problem solving. *arXiv preprint arXiv:2405.11403*, 2024.

Riccardo Andrea Izzo, Gianluca Bardaro, and Matteo Matteucci. Btgenbot: Behavior tree generation for robotic tasks with lightweight llms. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9684–9690. IEEE, 2024.

Lawrence Jang, Yinheng Li, Dan Zhao, Charles Ding, Justin Lin, Paul Pu Liang, Rogerio Bonatti, and Kazuhito Koishida. Videowebarena: Evaluating long context multimodal agents with video understanding web tasks. *arXiv preprint arXiv:2410.19100*, 2024.

Peter Jansen, Marc-Alexandre Côté, Tushar Khot, Erin Bransom, Bhavana Dalvi Mishra, Bodhisattwa Prasad Majumder, Oyvind Tafjord, and Peter Clark. Discoveryworld: A virtual environment for developing and evaluating automated scientific discovery agents. *Advances in Neural Information Processing Systems*, 37: 10088–10116, 2024.

Vindula Jayawardana, Baptiste Freydt, Ao Qu, Cameron Hickert, Zhongxia Yan, and Cathy Wu. Intersectionzoo: Eco-driving for benchmarking multi-agent contextual reinforcement learning. *arXiv preprint arXiv:2410.15221*, 2024.

Jihwan Jeong, Xiaoyu Wang, Jingmin Wang, Scott Sanner, and Pascal Poupart. Reflect-then-plan: Offline model-based planning through a doubly bayesian lens. *arXiv preprint arXiv:2506.06261*, 2025.

Dominik Jeurissen, Diego Perez-Liebana, Jeremy Gow, Duygu Cakmak, and James Kwan. Playing nethack with llms: Potential & limitations as zero-shot agents. *arXiv preprint arXiv:2403.00690*, 2024. URL `https://www.arxiv.org/abs/2403.00690`.

Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. Tree search for llm agent reinforcement learning. *arXiv preprint arXiv:2509.21240*, 2025.

Chengyou Jia, Minnan Luo, Zhuohang Dang, Qiushi Sun, Fangzhi Xu, Junlin Hu, Tianbao Xie, and Zhiyong Wu. Agentstore: Scalable integration of heterogeneous agents as specialized generalist computer assistant. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 8908–8934, 2025a.

Shuyi Jia, Chao Zhang, and Victor Fung. Llmatdesign: Autonomous materials discovery with large language models. *arXiv preprint arXiv:2406.13163*, 2024.

Ziqi Jia, Junjie Li, Xiaoyang Qu, and Jianzong Wang. Enhancing multi-agent systems via reinforcement learning with llm-based planner and graph-based policy. *arXiv preprint arXiv:2503.10049*, 2025b.

Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanmin Wu, Jiayi Lei, Pengshuo Qiu, Pan Lu, Zehui Chen, Chaoyou Fu, Guanglu Song, et al. Mmsearch: Benchmarking the potential of large models as multi-modal search engines. *arXiv preprint arXiv:2409.12959*, 2024a.

Nan Jiang, Xiaopeng Li, Shiqi Wang, Qiang Zhou, Soneya B Hossain, Baishakhi Ray, Varun Kumar, Xiaofei Ma, and Anoop Deoras. Ledex: Training LLMs to better self-debug and explain code. In *Neural Information Processing Systems (NeurIPS)*, pp. 35517–35543, 2024b.

Pengcheng Jiang, Jiacheng Lin, Lang Cao, Runchu Tian, SeongKu Kang, Zifeng Wang, Jimeng Sun, and Jiawei Han. Deepretrieval: Hacking real search engines and retrievers with large language models via reinforcement learning. *arXiv preprint arXiv:2503.00223*, 2025a.

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

Xue Jiang, Yihong Dong, Yongding Tao, Huanyu Liu, Zhi Jin, Wenpin Jiao, and Ge Li. ROCODE: Integrating backtracking mechanism and program analysis in large language models for code generation. In *IEEE/ACM International Conference on Software Engineering (ICSE)*, pp. 670–670, 2025b.

Yixing Jiang, Kameron C. Black, Gloria Geng, Danny Park, James Zou, Andrew Y. Ng, and Jonathan H. Chen. Medagentbench: A realistic virtual ehr environment to benchmark medical llm agents. *arXiv preprint arXiv:2501.14654*, 2025c. URL https://www.arxiv.org/abs/2501.14654.

Yuxuan Jiang and Francis Ferraro. Scribe: Structured mid-level supervision for tool-using language models, 2026. URL https://arxiv.org/abs/2601.03555.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R. Narasimhan. Swe-bench: Can language models resolve real-world github issues? In *International Conference on Learning Representations (ICLR)*, 2024.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025a.

Qiao Jin, Zhizheng Wang, Yifan Yang, Qingqing Zhu, Donald Wright, Thomas Huang, Nikhil Khandekar, Nicholas Wan, Xuguang Ai, W John Wilbur, et al. Agentmd: Empowering language agents for risk prediction with large-scale clinical tool learning. *Nature Communications*, 16(1):9377, 2025b.

Lucas Jing, Xinqi Wang, Liao Zhang, and Simon S. Du. Pbt-bench: Benchmarking ai agents on property-based testing, 2026. URL https://arxiv.org/abs/2605.15229.

Gurusha Juneja, Dylan Lu, Saaket Agashe, Parth Diwane, Edward Gunn, Jayanth Srinivasa, Gaowen Liu, William Yang Wang, Yali Du, and Xin Eric Wang. Enacttom: An evolving benchmark for functional theory of mind in embodied agents, 2026. URL https://arxiv.org/abs/2605.09826.

Tomoyuki Kagaya, Thong Jing Yuan, Yuxuan Lou, Jayashree Karlekar, Sugiri Pranata, Akira Kinose, Koki Oguri, Felix Wick, and Yang You. Rap: Retrieval-augmented planning with contextual memory for multimodal llm agents. *arXiv preprint arXiv:2402.03610*, 2024.

Zhao Kaiya, Michelangelo Naim, Jovana Kondic, Manuel Cortes, Jiaxin Ge, Shuying Luo, Guangyu Robert Yang, and Andrew Ahn. Lyfe agents: Generative agents for low-cost real-time social interactions. *arXiv preprint arXiv:2310.02172*, 2023.

Alok Kamatar, J Gregory Pauloski, Yadu Babuji, Ryan Chard, Mansi Sakarvadia, Daniel Babnigg, Kyle Chard, and Ian Foster. Empowering scientific workflows with federated agents. *arXiv preprint arXiv:2505.05428*, 2025.

Chenglong Kang, Xiaoyi Liu, and Fei Guo. Retrointext: A multimodal large language model enhanced framework for retrosynthetic planning via in-context representation learning. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Li Kang, Xiufeng Song, Heng Zhou, Yiran Qin, Jie Yang, Xiaohong Liu, Philip Torr, Lei Bai, and Zhenfei Yin. Viki-r: Coordinating embodied multi-agent cooperation via reinforcement learning. *arXiv preprint arXiv:2506.09049*, 2025b.

Shyam Sundar Kannan, Vishnunandan L. N. Venkatesh, and Byung-Cheol Min. Smart-llm: Smart multi-agent robot task planning using large language models, 2024. URL https://arxiv.org/abs/2309.10062.

Raghav Kapoor, Yash Parag Butala, Melisa Russak, Jing Yu Koh, Kiran Kamble, Waseem Alshikh, and Ruslan Salakhutdinov. Omniact: A dataset and benchmark for enabling multimodal generalist autonomous agents for desktop and web, 2024. URL https://arxiv.org/abs/2402.17553.

Andrej Karpathy. Vibe coding — wikipedia. https://en.wikipedia.org/wiki/Vibe_coding, 2025.

Michael Katz, Harsha Kokel, Kavitha Srinivas, and Shirin Sohrabi Araghi. Thought of search: Planning with language models through the lens of efficiency. *Advances in Neural Information Processing Systems*, 37: 138491–138568, 2024.

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

Zixuan Ke, Austin Xu, Yifei Ming, Xuan-Phi Nguyen, Ryan Chin, Caiming Xiong, and Shafiq Joty. Mas-zero: Designing multi-agent systems with zero supervision. *arXiv preprint arXiv:2505.14996*, 2025b.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. Dspy: Compiling declarative language model calls into self-improving pipelines, 2023. URL `https://arxiv.org/abs/2310.03714`.

Byeonghwi Kim, Jinyeon Kim, Yuyeong Kim, Cheolhong Min, and Jonghyun Choi. Context-aware planning and environment-aware memory for instruction following embodied agents. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10936–10946, 2023.

Dongyoung Kim, Sumin Park, Huiwon Jang, Jinwoo Shin, Jaehyung Kim, and Younggyo Seo. Robot-r1: Reinforcement learning for enhanced embodied reasoning in robotics. *arXiv preprint arXiv:2506.00070*, 2025.

Minsoo Kim, Victor Bursztyn, Eunyee Koh, Shunan Guo, and Seung-won Hwang. Rada: Retrieval-augmented web agent planning with llms. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 13511–13525, 2024a.

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

Zae Myung Kim, Dongseok Lee, Jaehyung Kim, Vipul Raheja, and Dongyeop Kang. The amazing agent race: Strong tool users, weak navigators, 2026. URL `https://arxiv.org/abs/2604.10261`.

Megan Kinniment, Lucas Jun Koba Sato, Haoxing Du, Brian Goodrich, Max Hasin, Lawrence Chan, Luke Harold Miles, Tao R. Lin, Hjalmar Wijk, Joel Burget, Aaron Ho, Elizabeth Barnes, and Paul Christiano. Evaluating language-model agents on realistic autonomous tasks, 2023. URL `https://arxiv.org/abs/2312.11671`.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of rlhf on llm generalisation and diversity. *arXiv preprint arXiv:2310.06452*, 2023.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649*, 2024a. URL `https://www.arxiv.org/abs/2401.13649`.

Jing Yu Koh, Stephen McAleer, Daniel Fried, and Ruslan Salakhutdinov. Tree search for language model agents. *arXiv preprint arXiv:2407.01476*, 2024b.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023. URL `https://arxiv.org/abs/2205.11916`.

Harsha Kokel, Michael Katz, Kavitha Srinivas, and Shirin Sohrabi. Acpbench: Reasoning about action, change, and planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 26559–26568, 2025.

Patrick Tser Jern Kon, Jiachen Liu, Qiuyi Ding, Yiming Qiu, Zhenning Yang, Yibo Huang, Jayanth Srinivasa, Myungjin Lee, Mosharaf Chowdhury, and Ang Chen. Curie: Toward rigorous and automated scientific experimentation with ai agents. *arXiv preprint arXiv:2502.16069*, 2025.

Deqian Kong, Dehong Xu, Minglu Zhao, Bo Pang, Jianwen Xie, Andrew Lizarraga, Yuhao Huang, Sirui Xie, and Ying Nian Wu. Latent plan transformer for trajectory abstraction: Planning as latent space inference. *Advances in Neural Information Processing Systems*, 37:123379–123401, 2024.

Lingdong Kong, Xian Sun, Wei Chow, Linfeng Li, Kevin Qinghong Lin, Xuan Billy Zhang, Song Wang, Rong Li, Qing Wu, Wei Gao, Yingshuo Wang, Shaoyuan Xie, Jiachen Liu, Leigang Qu, Shijie Li, Lai Xing Ng, Benoit R. Cottereau, Ziwei Liu, Tat-Seng Chua, and Wei Tsang Ooi. AI for Auto-Research: Roadmap & user guide. *arXiv preprint arXiv:2605.18661*, 2026.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*, 2024.

Shrinidhi Kumbhar, Venkatesh Mishra, Kevin Coutinho, Divij Handa, Ashif Iquebal, and Chitta Baral. Hypothesis generation for materials discovery and design using goal-driven and constraint-guided llm agents. *arXiv preprint arXiv:2501.13299*, 2025.

Hanyu Lai, Xiao Liu, Iat Long Iong, Shuntian Yao, Yuxuan Chen, Pengbo Shen, Hao Yu, Hanchen Zhang, Xiaohan Zhang, Yuxiao Dong, et al. Autowebglm: A large language model-based web navigating agent. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5295–5306, 2024.

Hanyu Lai, Xiao Liu, Yanxiao Zhao, Han Xu, Hanchen Zhang, Bohao Jing, Yanyu Ren, Shuntian Yao, Yuxiao Dong, and Jie Tang. Computerrl: Scaling end-to-end online reinforcement learning for computer use agents. *arXiv preprint arXiv:2508.14040*, 2025. URL `https://www.arxiv.org/abs/2508.14040`.

Jakub Lála, Odhran O'Donoghue, Aleksandar Shtedritski, Sam Cox, Samuel G Rodriques, and Andrew D White. Paperqa: Retrieval-augmented generative agent for scientific research. *arXiv preprint arXiv:2312.07559*, 2023.

Man Ho Lam, Chaozheng Wang, Hange Liu, Jingyu Xiao, Haau sing Li, Jen tse Huang, Terry Yue Zhuo, and Michael R. Lyu. Swe-chain: Benchmarking coding agents on chained release-level package upgrades, 2026. URL `https://arxiv.org/abs/2605.14415`.

Tian Lan, Bin Zhu, Qianghuai Jia, Junyang Ren, Haijun Li, Longyue Wang, Zhao Xu, Weihua Luo, and Kaifu Zhang. Deepwidesearch: Benchmarking depth and width in agentic information seeking. *arXiv preprint arXiv:2510.20168*, 2025.

Jon M. Laurent, Joseph D. Janizek, Michael Ruzo, Michaela M. Hinks, Michael J. Hammerling, Siddharth Narayanan, Manvitha Ponnapati, Andrew D. White, and Samuel G. Rodriques. Lab-bench: Measuring capabilities of language models for biology research. *arXiv preprint arXiv:2407.10362*, 2024. URL `https://www.arxiv.org/abs/2407.10362`.

Hung Le, Yue Wang, Akhilesh Deepak Yu, Thanh-Tung Nguyen, Zhiwei Sun, Nan Jiang, Quoc Viet Le, and Steven C. H. Hoi. Coderl: Mastering code generation through pretrained models and deep reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Hung Le, Hailin Chen, Amrita Saha, Akash Gokul, Doyen Sahoo, and Shafiq Joty. Codechain: Towards modular code generation through chain of self-revisions with representative sub-modules, 2024. URL `https://arxiv.org/abs/2310.08992`.

Blake LeBaron. Agent-based computational finance. *Handbook of computational economics*, 2:1187–1233, 2006.

Dong-Ho Lee, Adyasha Maharana, Jay Pujara, Xiang Ren, and Francesco Barbieri. Realtalk: A 21-day real-world dataset for long-term conversation. *arXiv preprint arXiv:2502.13270*, 2025a.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. 2024a.

Jin Hwa Lee, Stefano Sarao Mannelli, and Andrew Saxe. Why do animals need shaping? a theory of task composition and curriculum learning. *arXiv preprint arXiv:2402.18361*, 2024b.

Meng-Chieh Lee, Qi Zhu, Costas Mavromatis, Zhen Han, Soji Adeshina, Vassilis N Ioannidis, Huzefa Rangwala, and Christos Faloutsos. Agent-g: An agentic framework for graph retrieval augmented generation. 2024c.

Namkyeong Lee, Edward De Brouwer, Ehsan Hajiramezanali, Tommaso Biancalani, Chanyoung Park, and Gabriele Scalia. Rag-enhanced collaborative llm agents for drug discovery. *arXiv preprint arXiv:2502.17506*, 2025b.

Sunjae Lee, Junyoung Choi, Jungjae Lee, Munim Hasan Wasi, Hojun Choi, Steven Y Ko, Sangeun Oh, and Insik Shin. Explore, select, derive, and recall: Augmenting llm with human-like memory for mobile task automation. *arXiv preprint arXiv:2312.03003*, 2023.

Sunjae Lee, Junyoung Choi, Jungjae Lee, Munim Hasan Wasi, Hojun Choi, Steve Ko, Sangeun Oh, and Insik Shin. 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*, pp. 1119–1133, 2024d.

Joel Z Leibo, Edgar A Dueñez-Guzman, Alexander Vezhnevets, John P Agapiou, Peter Sunehag, Raphael Koster, Jayd Matyas, Charlie Beattie, Igor Mordatch, and Thore Graepel. Scalable evaluation of multi-agent reinforcement learning with melting pot. In *International conference on machine learning*, pp. 6187–6199. PMLR, 2021.

Hui Yi Leong and Yuqing Wu. Dynaswarm: Dynamically graph structure selection for llm-based multi-agent system. *arXiv preprint arXiv:2507.23261*, 2025.

Pascal Leroy, Pablo G Morato, Jonathan Pisane, Athanasios Kolios, and Damien Ernst. Imp-marl: a suite of environments for large-scale infrastructure management planning via marl. *Advances in neural information processing systems*, 36:53522–53551, 2023.

Elad Levi and Ilan Kadar. Intellagent: A multi-agent framework for evaluating conversational ai systems. *arXiv preprint arXiv:2501.11067*, 2025.

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

Ao Li, Yuexiang Xie, Songze Li, Fugee Tsung, Bolin Ding, and Yaliang Li. Agent-oriented planning in multi-agent systems. *arXiv preprint arXiv:2410.02189*, 2024a.

Binxu Li, Tiankai Yan, Yuanting Pan, Jie Luo, Ruiyang Ji, Jiayuan Ding, Zhe Xu, Shilong Liu, Haoyu Dong, Zihao Lin, et al. Mmedagent: Learning to use medical tools with multi-modal agent. *arXiv preprint arXiv:2407.02483*, 2024b.

Boyi Li, Zhonghan Zhao, Der-Horng Lee, and Gaoang Wang. Adaptive graph pruning for multi-agent communication. *arXiv preprint arXiv:2506.02951*, 2025a.

Chengpeng Li, Guanting Dong, Mingfeng Xue, Ru Peng, Xiang Wang, and Dayiheng Liu. Dotamath: Decomposition of thought with code assistance and self-correction for mathematical reasoning, 2024c. URL `https://arxiv.org/abs/2407.04078`.

Chengpeng Li, Zhengyang Tang, Ziniu Li, Mingfeng Xue, Keqin Bao, Tian Ding, Ruoyu Sun, Benyou Wang, Xiang Wang, Junyang Lin, and Dayiheng Liu. Cort: Code-integrated reasoning within thinking, 2025b. URL `https://arxiv.org/abs/2506.09820`.

Chengpeng Li, Mingfeng Xue, Zhenru Zhang, Jiaxi Yang, Beichen Zhang, Xiang Wang, Bowen Yu, Binyuan Hui, Junyang Lin, and Dayiheng Liu. Start: Self-taught reasoner with tools, 2025c. URL `https://arxiv.org/abs/2503.04625`.

Dawei Li, Zhen Tan, Peijia Qian, Yifan Li, Kumar Satvik Chaudhary, Lijie Hu, and Jiayi Shen. Smoa: Improving multi-agent large language models with sparse mixture-of-agents, 2024d. URL https://arxiv.org/abs/2411.03284.

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

Hao Li, Chenghao Yang, An Zhang, Yang Deng, Xiang Wang, and Tat-Seng Chua. Hello again! llm-powered personalized agent for long-term dialogue. 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)*, pp. 5259–5276, 2025d.

Huao Li, Yu Quan Chong, Simon Stepputtis, Joseph Campbell, Dana Hughes, Michael Lewis, and Katia Sycara. Theory of mind for multi-agent collaboration via large language models. *arXiv preprint arXiv:2310.10701*, 2023b.

Jiaqi Li, Xinyi Dong, Yang Liu, Zhizhuo Yang, Quansen Wang, Xiaobo Wang, Song-Chun Zhu, Zixia Jia, and Zilong Zheng. Reflectevo: Improving meta introspection of small llms by learning self-reflection. In *Findings of the Association for Computational Linguistics (ACL)*, 2025e. URL https://aclanthology.org/2025.findings-acl.871/.

Jierui Li, Hung Le, Yingbo Zhou, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Codetree: Agent-guided tree search for code generation with large language models. *arXiv preprint arXiv:2411.04329*, 2024e.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, et al. Websailor: Navigating super-human reasoning for web agent. *arXiv preprint arXiv:2507.02592*, 2025f.

Long Li, Weiwen Xu, Jiayan Guo, Ruochen Zhao, Xingxuan Li, Yuqian Yuan, Boqiang Zhang, Yuming Jiang, Yifei Xin, Ronghao Dang, et al. Chain of ideas: Revolutionizing research via novel idea development with llm agents. *arXiv preprint arXiv:2410.13185*, 2024f.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. Api-bank: A comprehensive benchmark for tool-augmented llms. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 3102–3116. Association for Computational Linguistics, 2023c. doi: 10.18653/V1/2023.EMNLP-MAIN.187. URL https://doi.org/10.18653/v1/2023.emnlp-main.187.

Ning Li, Xiangmou Qu, Jiamu Zhou, Jun Wang, Muning Wen, Kounianhua Du, Xingyu Lou, Qiuying Peng, and Weinan Zhang. Mobileuse: A gui agent with hierarchical reflection for autonomous mobile operation. *arXiv preprint arXiv:2507.16853*, 2025g.

Ruochen Li, Teerth Patel, Qingyun Wang, and Xinya Du. Mlr-copilot: Autonomous machine learning research based on large language models agents. *arXiv preprint arXiv:2408.14033*, 2024g.

Shilong Li, Xingyuan Bu, Wenjie Wang, Jiaheng Liu, Jun Dong, Haoyang He, Hao Lu, Haozhe Zhang, Chenchen Jing, Zhen Li, et al. Mm-browsecomp: A comprehensive benchmark for multimodal browsing agents. *arXiv preprint arXiv:2508.13186*, 2025h.

Tao Li, Gang Li, Zhiwei Deng, Bryan Wang, and Yang Li. A zero-shot language agent for computer control with structured reflection. *arXiv preprint arXiv:2310.08740*, 2023d.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. *arXiv preprint arXiv:2501.05366*, 2025i.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability. *arXiv preprint arXiv:2504.21776*, 2025j.

Xiaoxi Li, Wenxiang Jiao, Jiarui Jin, Guanting Dong, Jiajie Jin, Yinuo Wang, Hao Wang, Yutao Zhu, Ji-Rong Wen, Yuan Lu, and Zhicheng Dou. Deepagent: A general reasoning agent with scalable toolsets, 2026a. URL https://arxiv.org/abs/2510.21618.

Yangning Li, Yinghui Li, Xinyu Wang, Yong Jiang, Zhen Zhang, Xinran Zheng, Hui Wang, Hai-Tao Zheng, Philip S Yu, Fei Huang, et al. Benchmarking multimodal retrieval augmented generation with dynamic vqa dataset and self-adaptive planning agent. *arXiv preprint arXiv:2411.02937*, 2024h.

Yanlong Li, Jindong Li, Qi Wang, Menglin Yang, He Kong, and Shengsheng Wang. Cogito, ergo sum: A neurobiologically-inspired cognition-memory-growth system for code generation, 2025k.

Yuante Li, Xu Yang, Xiao Yang, Minrui Xu, Xisen Wang, Weiqing Liu, and Jiang Bian. R&d-agent-quant: A multi-agent framework for data-centric factors and model joint optimization. *CoRR*, abs/2505.15155, 2025l. doi: 10.48550/ARXIV.2505.15155. URL https://doi.org/10.48550/arXiv.2505.15155.

Zaijing Li, Yuquan Xie, Rui Shao, Gongwei Chen, Dongmei Jiang, and Liqiang Nie. Optimus-1: Hybrid multimodal memory empowered agents excel in long-horizon tasks. *Advances in neural information processing systems*, 37:49881–49913, 2024i.

Zelong Li, Shuyuan Xu, Kai Mei, Wenyue Hua, Balaji Rama, Om Raheja, Hao Wang, He Zhu, and Yongfeng Zhang. Autoflow: Automated workflow generation for large language model agents. *ArXiv*, abs/2407.12821, 2024j. URL https://api.semanticscholar.org/CorpusID:271270428.

Zhiyu Li, Shichao Song, Hanyu Wang, Simin Niu, Ding Chen, Jiawei Yang, Chenyang Xi, Huayi Lai, Jihao Zhao, Yezhaohui Wang, et al. Memos: An operating system for memory-augmented generation (mag) in large language models. *arXiv preprint arXiv:2505.22101*, 2025m.

Zihao Li, Kaifeng Jin, Yuanchen Bei, Jiaru Zou, Avaneesh Kumar, Xuying Ning, Yanjun Zhao, Mengting Ai, Baoyu Jing, Hanghang Tong, and Jingrui He. Harnessing generalist agents for contextualized time series, 2026b. URL https://arxiv.org/abs/2606.05404.

Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Zhi-Quan Luo, and Ruoyu Sun. Preserving diversity in supervised fine-tuning of large language models. *arXiv preprint arXiv:2408.16673*, 2024k.

Shuquan Lian, Yuhang Wu, Jia Ma, Yifan Ding, Zihan Song, Bingqi Chen, Xiawu Zheng, and Hui Li. Ui-agile: Advancing gui agents with effective reinforcement learning and precise inference-time grounding. *arXiv preprint arXiv:2507.22025*, 2025. URL https://www.arxiv.org/abs/2507.22025.

Xuechen Liang, Yangfan He, Yinghui Xia, Xinyuan Song, Jianhui Wang, Meiling Tao, Li Sun, Xinhang Yuan, Jiayi Su, Keqin Li, Jiaqi Chen, Jinsong Yang, Siyuan Chen, and Tianyu Shi. Self-evolving agents with reflective and memory-augmented abilities, 2025a. URL https://arxiv.org/abs/2409.00872.

Yaobo Liang, Chenfei Wu, Ting Song, Wenshan Wu, Yan Xia, Yu Liu, Yang Ou, Shuai Lu, Lei Ji, Shaoguang Mao, Yun Wang, Linjun Shou, Ming Gong, and Nan Duan. Taskmatrix.ai: Completing tasks by connecting foundation models with millions of apis. *CoRR*, abs/2303.16434, 2023. doi: 10.48550/ARXIV.2303.16434. URL https://doi.org/10.48550/arXiv.2303.16434.

Yichao Liang, Nishanth Kumar, Hao Tang, Adrian Weller, Joshua B Tenenbaum, Tom Silver, João F Henriques, and Kevin Ellis. Visualpredicator: Learning abstract world models with neuro-symbolic predicates for robot planning. In *International Conference on Learning Representations*, volume 2025, pp. 60416–60444, 2025b.

Zhengyang Liang, Yan Shu, Xiangrui Liu, Minghao Qin, Kaixin Liang, Nicu Sebe, Zheng Liu, and Lizi Liao. Video-browser: Towards agentic open-web video browsing, 2026a. URL https://arxiv.org/abs/2512.23044.

Zhengyang Liang, Yan Shu, Xiangrui Liu, Minghao Qin, Kaixin Liang, Nicu Sebe, Zheng Liu, and Lizi Liao. Video-browser: Towards agentic open-web video browsing, 2026b. URL https://arxiv.org/abs/2512.23044.

Junwei Liao, Muning Wen, Jun Wang, and Weinan Zhang. Marft: Multi-agent reinforcement fine-tuning, 2025. URL https://arxiv.org/abs/2504.16129.

Mircea Lică, Ojas Shirekar, Baptiste Colle, and Chirag Raman. Mindforge: Empowering embodied agents with theory of mind for lifelong cultural learning. *arXiv preprint arXiv:2411.12977*, 2024.

Jonathan Light, Min Cai, Sheng Shen, and Ziniu Hu. Avalonbench: Evaluating llms playing the game of avalon. *arXiv preprint arXiv:2310.05036*, 2023. URL https://www.arxiv.org/abs/2310.05036.

Feng Lin, Dong Jae Kim, et al. Soen-101: Code generation by emulating software process models using large language model agents. In *International Conference on Software Engineering (ICSE)*, pp. 1527–1539, 2025a.

Kevin Lin, Charlie Snell, Yu Wang, Charles Packer, Sarah Wooders, Ion Stoica, and Joseph E Gonzalez. Sleep-time compute: Beyond inference scaling at test-time. *arXiv preprint arXiv:2504.13171*, 2025b.

Minhua Lin, Zongyu Wu, Zhichao Xu, Hui Liu, Xianfeng Tang, Qi He, Charu Aggarwal, Xiang Zhang, and Suhang Wang. A comprehensive survey on reinforcement learning-based agentic search: Foundations, roles, optimizations, evaluations, and applications. *arXiv preprint arXiv:2510.16724*, 2025c.

Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, et al. Ra-dit: Retrieval-augmented dual instruction tuning. In *International Conference on Learning Representations*, volume 2024, pp. 19138–19162, 2024.

Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. LLM+P: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477*, 2023a.

Chengwen Liu, Xiaomin Yu, Zhuoyue Chang, Zhe Huang, Shuo Zhang, Heng Lian, Jisheng Dang, Rui Xu, Sen Hu, Jianheng Hou, Chengwei Qin, Xiaobin Hu, Kunyi Wang, Zhi Yang, Hao Peng, Hong Peng, Ronghao Chen, and Huacan Wang. Watching, reasoning, and searching: A video deep research benchmark on open web for agentic video reasoning, 2026a. URL https://arxiv.org/abs/2601.06943.

Gang Liu, Michael Sun, Wojciech Matusik, Meng Jiang, and Jie Chen. Multimodal large language models for inverse molecular design with retrosynthetic planning. *arXiv preprint arXiv:2410.04223*, 2024a.

Guangyi Liu, Pengxiang Zhao, Liang Liu, Zhiming Chen, Yuxiang Chai, Shuai Ren, Hao Wang, Shibo He, and Wenchao Meng. Learnact: Few-shot mobile gui agent with a unified demonstration benchmark. *arXiv preprint arXiv:2504.13805*, 2025a.

Haowei Liu, Xi Zhang, Haiyang Xu, Yuyang Wanyan, Junyang Wang, Ming Yan, Ji Zhang, Chunfeng Yuan, Changsheng Xu, Weiming Hu, and Fei Huang. Pc-agent: A hierarchical multi-agent collaboration framework for complex task automation on pc, 2025b. URL https://arxiv.org/abs/2502.14282.

Haoyang Liu, Yijiang Li, Jinglin Jian, Yuxuan Cheng, Jianrong Lu, Shuyi Guo, Jinglei Zhu, Mianchen Zhang, Miantong Zhang, and Haohan Wang. Toward a team of ai-made scientists for scientific discovery from gene expression data. *arXiv preprint arXiv:2402.12391*, 2024b.

Haoyang Liu, Yijiang Li, and Haohan Wang. Genomas: A multi-agent framework for scientific discovery via code-driven gene expression analysis. *arXiv preprint arXiv:2507.21035*, 2025c.

Jerry Liu. LlamaIndex, 11 2022. URL https://github.com/jerryjliu/llama_index.

Jie Liu, Pan Zhou, Yingjun Du, Ah-Hwee Tan, Cees GM Snoek, Jan-Jakob Sonke, and Efstratios Gavves. Capo: Cooperative plan optimization for efficient embodied multi-agent cooperation. *arXiv preprint arXiv:2411.04679*, 2024c.

Jun Liu, Zhenglun Kong, Changdi Yang, Fan Yang, Tianqi Li, Peiyan Dong, Joannah Nanjekye, Hao Tang, Geng Yuan, Wei Niu, et al. Rcr-router: Efficient role-aware context routing for multi-agent llm systems with structured memory. *arXiv preprint arXiv:2508.04903*, 2025d.

Junpeng Liu, Yifan Song, Bill Yuchen Lin, Wai Lam, Graham Neubig, Yuanzhi Li, and Xiang Yue. Visualwebbench: How far have multimodal llms evolved in web page understanding and grounding?, 2024d. URL https://arxiv.org/abs/2404.05955.

Junteng Liu, Yunji Li, Chi Zhang, Jingyang Li, Aili Chen, Ke Ji, Weiyu Cheng, Zijia Wu, Chengyu Du, Qidi Xu, et al. Webexplorer: Explore and evolve for training long-horizon web agents. *arXiv preprint arXiv:2509.06501*, 2025e.

Kehui Liu, Zixin Tang, Dong Wang, Zhigang Wang, Xuelong Li, and Bin Zhao. Coherent: Collaboration of heterogeneous multi-robot system with large language models. *arXiv preprint arXiv:2409.15146*, 2024e.

Ruoqi Liu, Imran Q. Mohiuddin, Austin J. Schoeffler, Kavita Renduchintala, Ashwin Nayak, Prasantha L. Vemu, Shivam C. Vedak, Kameron C. Black, John L. Havlik, Isaac Ogunmola, Stephen P. Ma, Roopa Dhatt, and Jonathan H. Chen. Physicianbench: Evaluating llm agents in real-world ehr environments, 2026b. URL https://arxiv.org/abs/2605.02240.

Shuo Liu, Zeyu Liang, Xueguang Lyu, and Christopher Amato. Llm collaboration with multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 32150–32158, 2026c.

Siwei Liu, Jinyuan Fang, Han Zhou, Yingxu Wang, and Zaiqiao Meng. SEW: Self-evolving agentic workflows for automated code generation, 2025f.

Sizhe Liu, Yizhou Lu, Siyu Chen, Xiyang Hu, Jieyu Zhao, Yingzhou Lu, and Yue Zhao. Drugagent: Automating ai-aided drug discovery programming through llm multi-agent collaboration. *arXiv preprint arXiv:2411.15692*, 2024f.

Tengxiao Liu, Qipeng Guo, Yuqing Yang, Xiangkun Hu, Yue Zhang, Xipeng Qiu, and Zheng Zhang. Plan, verify and switch: Integrated reasoning with diverse x-of-thoughts. *arXiv preprint arXiv:2310.14628*, 2023b.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*, 2023c. URL https://www.arxiv.org/abs/2308.03688.

Xiao Liu, Bo Qin, Dongzhu Liang, Guang Dong, Hanyu Lai, Hanchen Zhang, Hanlin Zhao, Iat Long Iong, Jiadai Sun, Jiaqi Wang, Junjie Gao, Junjun Shan, Kangning Liu, Shudan Zhang, Shuntian Yao, Siyi Cheng, Wentao Yao, Wenyi Zhao, Xinghan Liu, Xinyi Liu, Xinying Chen, Xinyue Yang, Yang Yang, Yifan Xu, Yu Yang, Yujia Wang, Yulin Xu, Zehan Qi, Yuxiao Dong, and Jie Tang. Autoglm: Autonomous foundation agents for guis. *arXiv preprint arXiv:2411.00820*, 2024g. URL https://www.arxiv.org/abs/2411.00820.

Yanming Liu, Xinyue Peng, Jiannan Cao, Yuwei Zhang, Xuhong Zhang, Sheng Cheng, Xun Wang, Jianwei Yin, and Tianyu Du. Tool-planner: Task planning with clusters across multiple tools. In *The Thirteenth International Conference on Learning Representations*, 2025g. URL https://openreview.net/forum?id=dRz3cizftU.

Yuhang Liu, Pengxiang Li, Zishu Wei, Congkai Xie, Xueyu Hu, Xinchen Xu, Shengyu Zhang, Xiaotian Han, Hongxia Yang, and Fei Wu. Infiguiagent: A multimodal generalist gui agent with native reasoning and reflection. *arXiv preprint arXiv:2501.04575*, 2025h.

Yuhang Liu, Pengxiang Li, Congkai Xie, Xavier Hu, Xiaotian Han, Shengyu Zhang, Hongxia Yang, and Fei Wu. Infigui-r1: Advancing multimodal gui agents from reactive actors to deliberative reasoners. *arXiv preprint arXiv:2504.14239*, 2025i. URL https://www.arxiv.org/abs/2504.14239.

Zichen Liu, Guoji Fu, Chao Du, Wee Sun Lee, and Min Lin. Continual reinforcement learning by planning with online world models. *arXiv preprint arXiv:2507.09177*, 2025j.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. A dynamic llm-powered agent network for task-oriented agent collaboration. In *First Conference on Language Modeling*, 2024h.

Chunlok Lo, Kevin Roice, Parham Mohammad Panahi, Scott M Jordan, Adam White, Gabor Mihucz, Farzane Aminmansour, and Martha White. Goal-space planning with subgoal models. *Journal of Machine Learning Research*, 25(330):1–57, 2024.

Jieyi Long. Large language model guided tree-of-thought. *arXiv preprint arXiv:2305.08291*, 2023.

Lin Long, Yichen He, Wentao Ye, Yiyuan Pan, Yuan Lin, Hang Li, Junbo Zhao, and Wei Li. Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory. *arXiv preprint arXiv:2508.09736*, 2025.

Qian Long, Zhi Li, Ran Gong, Ying Nian Wu, Demetri Terzopoulos, and Xiaofeng Gao. Teamcraft: A benchmark for multi-modal multi-agent systems in minecraft. *arXiv preprint arXiv:2412.05255*, 2024.

Chang Han Low, Ziyue Wang, Tianyi Zhang, Zhitao Zeng, Zhu Zhuo, Evangelos B Mazomenos, and Yueming Jin. Surgraw: Multi-agent workflow with chain-of-thought reasoning for surgical intelligence. *arXiv preprint arXiv:2503.10265*, 2025.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292v3*, 2024a. URL `https://www.arxiv.org/abs/2408.06292v3`.

Fanbin Lu, Zhisheng Zhong, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Arpo:end-to-end policy optimization for gui agents with experience replay. *arXiv preprint arXiv:2505.16282*, 2025a. URL `https://www.arxiv.org/abs/2505.16282`.

Fanbin Lu, Zhisheng Zhong, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Arpo: End-to-end policy optimization for gui agents with experience replay. *arXiv preprint arXiv:2505.16282*, 2025b.

Jianqiao Lu, Wanjun Zhong, Wenyong Huang, Yufei Wang, Qi Zhu, Fei Mi, Baojun Wang, Weichao Wang, Xingshan Zeng, Lifeng Shang, et al. Self: Self-evolution with language feedback. *arXiv preprint arXiv:2310.00533*, 2023a.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models. *Advances in Neural Information Processing Systems*, 36:43447–43478, 2023b.

Pan Lu, Bowen Chen, Sheng Liu, Rahul Thapa, Joseph Boen, and James Zou. Octotools: An agentic framework with extensible tools for complex reasoning, 2025c. URL `https://arxiv.org/abs/2502.11271`.

Quanfeng Lu, Zhantao Ma, Shuai Zhong, Jin Wang, Dahai Yu, Michael K Ng, and Ping Luo. Swirl: A staged workflow for interleaved reinforcement learning in mobile gui control. *arXiv preprint arXiv:2508.20018*, 2025d.

Yifei Lu, Fanghua Ye, Jian Li, Qiang Gao, Cheng Liu, Haibo Luo, Nan Du, Xiaolong Li, and Feiliang Ren. CodeTool: Enhancing programmatic tool invocation of LLMs via process supervision, 2025e.

Yining Lu, Haoping Yu, and Daniel Khashabi. GEAR: Augmenting language models with generalizable and efficient tool resolution. In Yvette Graham and Matthew Purver (eds.), *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 112–138, St. Julian's, Malta, March 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.eacl-long.7. URL `https://aclanthology.org/2024.eacl-long.7/`.

Zhengxi Lu, Yuxiang Chai, Yaxuan Guo, Xi Yin, Liang Liu, Hao Wang, Han Xiao, Shuai Ren, Guanjing Xiong, and Hongsheng Li. Ui-r1: Enhancing efficient action prediction of gui agents by reinforcement learning, 2025f. URL `https://arxiv.org/abs/2503.21620`.

Zhengxi Lu, Jiabo Ye, Fei Tang, Yongliang Shen, Haiyang Xu, Ziwei Zheng, Weiming Lu, Ming Yan, Fei Huang, Jun Xiao, and Yueting Zhuang. Ui-s1: Advancing gui automation via semi-online reinforcement learning. *arXiv preprint arXiv:2509.11543*, 2025g. URL `https://www.arxiv.org/abs/2509.11543`.

Dezhao Luo, Bohan Tang, Kang Li, Georgios Papoudakis, Jifei Song, Shaogang Gong, Jianye Hao, Jun Wang, and Kun Shao. Vimo: A generative visual gui world model for app agents. *arXiv preprint arXiv:2504.13936*, 2025a.

Run Luo, Lu Wang, Wanwei He, Longze Chen, Jiaming Li, and Xiaobo Xia. Gui-r1: A generalist r1-style vision-language action model for gui agents. *arXiv preprint arXiv:2504.10458*, 2025b.

Xufang Luo, Yuge Zhang, Zhiyuan He, Zilong Wang, Siyun Zhao, Dongsheng Li, Luna K. Qiu, and Yuqing Yang. Agent lightning: Train ANY AI agents with reinforcement learning, 2025c. URL https://arxiv.org/abs/2508.03680.

Thang Luong, Dawsen Hwang, Hoang H. Nguyen, Golnaz Ghiasi, Yuri Chervonyi, Insuk Seo, Junsu Kim, Garrett Bingham, Jonathan Lee, Swaroop Mishra, Alex Zhai, Clara Huiyi Hu, Henryk Michalewski, Jimin Kim, Jeonghyun Ahn, Junhwi Bae, Xingyou Song, Trieu H. Trinh, Quoc V. Le, and Junehyuk Jung. Towards robust mathematical reasoning, 2025. URL https://arxiv.org/abs/2511.01846.

Michael Lutz, Arth Bohra, Manvel Saroyan, Artem Harutyunyan, and Giovanni Campagna. Wilbur: Adaptive in-context learning for robust and accurate web agents. *arXiv preprint arXiv:2404.05902*, 2024.

Max Ruiz Luyten, Antonin Berthon, and Mihaela van der Schaar. Strategic planning: A top-down approach to option generation. In *Forty-second International Conference on Machine Learning*, 2025.

Artem Lykov and Dzmitry Tsetserukou. Llm-brain: Ai-driven fast generation of robot behaviour tree based on large language model. In *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, pp. 392–397. IEEE, 2024.

Xing Han Lù, Zdeněk Kasner, and Siva Reddy. Weblinx: Real-world website navigation with multi-turn dialogue, 2024. URL https://arxiv.org/abs/2402.05930.

Chang Ma, Haiteng Zhao, Junlei Zhang, Junxian He, and Lingpeng Kong. Non-myopic generation of language models for reasoning and planning. *arXiv preprint arXiv:2410.17195*, 2024a.

Hao Ma, Tianyi Hu, Zhiqiang Pu, Liu Boyin, Xiaolin Ai, Yanyan Liang, and Min Chen. Coevolving with the other you: Fine-tuning llm with sequential cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 37:15497–15525, 2024b.

Kaixin Ma, Hongming Zhang, Hongwei Wang, Xiaoman Pan, Wenhao Yu, and Dong Yu. Laser: Llm agent with state-space exploration for web navigation. *arXiv preprint arXiv:2309.08172*, 2023. URL https://www.arxiv.org/abs/2309.08172.

Pingchuan Ma, Tsun-Hsuan Wang, Minghao Guo, Zhiqing Sun, Joshua B Tenenbaum, Daniela Rus, Chuang Gan, and Wojciech Matusik. Llm and simulation as bilevel optimizers: A new paradigm to advance physical scientific discovery. *arXiv preprint arXiv:2405.09783*, 2024c.

Yingwei Ma, Rongyu Cao, Yongchang Cao, Yue Zhang, Jue Chen, Yibo Liu, Yuchen Liu, Binhua Li, Fei Huang, and Yongbin Li. Lingma SWE-GPT: An open development-process-centric language model for automated software improvement, 2024d.

Yubo Ma, Zhibin Gou, Junheng Hao, Ruochen Xu, Shuohang Wang, Liangming Pan, Yujiu Yang, Yixin Cao, Aixin Sun, Hany Awadalla, et al. Sciagent: Tool-augmented language models for scientific reasoning. *arXiv preprint arXiv:2402.11451*, 2024e.

Zexiong Ma, Chao Peng, Pengfei Gao, Xiangxin Meng, Yanzhen Zou, and Bing Xie. Sorft: Issue resolving with subtask-oriented reinforced fine-tuning, 2025a. URL https://arxiv.org/abs/2502.20127.

Zexiong Ma, Chao Peng, Qunhong Zeng, Pengfei Gao, Yanzhen Zou, and Bing Xie. Tool-integrated reinforcement learning for repo deep search, 2025b. URL https://arxiv.org/abs/2508.03012.

Zhiyuan Ma, Jiayu Liu, Xianzhen Luo, Zhenya Huang, Qingfu Zhu, and Wanxiang Che. Advancing tool-augmented large language models via meta-verification and reflection learning. *CoRR*, abs/2506.04625, 2025c. doi: 10.48550/ARXIV.2506.04625. URL https://doi.org/10.48550/arXiv.2506.04625.

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

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents. *arXiv preprint arXiv:2402.17753*, 2024.

Sadegh Mahdavi, Raquel Aoki, Keyi Tang, and Yanshuai Cao. Leveraging environment interaction for automated pddl translation and planning with large language models. *Advances in Neural Information Processing Systems*, 37:38960–39008, 2024.

Bodhisattwa Prasad Majumder, Bhavana Dalvi Mishra, Peter Jansen, Oyvind Tafjord, Niket Tandon, Li Zhang, Chris Callison-Burch, and Peter Clark. Clin: A continually learning language agent for rapid task adaptation and generalization. *arXiv preprint arXiv:2310.10134*, 2023. URL https://www.arxiv.org/abs/2310.10134.

Bulat Maksudov, Vladislav Kurenkov, Kathleen M. Curran, and Alessandra Mileo. Abra: Agent benchmark for radiology applications, 2026. URL https://arxiv.org/abs/2605.11224.

Potsawee Manakul, Adian Liusie, and Mark Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 9004–9017, 2023.

Kelong Mao, Zheng Liu, Hongjin Qian, Fengran Mo, Chenlong Deng, and Zhicheng Dou. Rag-studio: Towards in-domain adaptation of retrieval augmented generation through self-alignment. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 725–735, 2024.

Elan Markowitz, Anil Ramakrishna, Jwala Dhamala, Ninareh Mehrabi, Charith Peris, Rahul Gupta, Kai-Wei Chang, and Aram Galstyan. Tree-of-traversals: A zero-shot reasoning algorithm for augmenting black-box language models with knowledge graphs. *arXiv preprint arXiv:2407.21358*, 2024.

Sharath Matada, Luke Bhan, Yuanyuan Shi, and Nikolay Atanasov. Generalizable motion planning via operator learning. *arXiv preprint arXiv:2410.17547*, 2024.

Mathematical Association of America. American invitational mathematics examination. https://www.maa.org/math-competitions/aime.

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

Booker Meghan, Byrd Grayson, Kemp Bethany, Schmidt Aurora, and Rivera Corban. Embodiedrag: Dynamic 3d scene graph retrieval for efficient and scalable robot task planning. *arXiv preprint arXiv:2410.23968*, 2024. URL https://www.arxiv.org/abs/2410.23968.

Jingbiao Mei, Jinghong Chen, Guangyu Yang, Xinyu Hou, Margaret Li, and Bill Byrne. According to me: Long-term personalized referential memory qa, 2026. URL https://arxiv.org/abs/2603.01990.

Silin Meng, Yiwei Wang, Cheng-Fu Yang, Nanyun Peng, and Kai-Wei Chang. Llm-a*: Large language model enhanced incremental heuristic search on path planning. *arXiv preprint arXiv:2407.02511*, 2024a.

Tao Meng, Ninareh Mehrabi, Palash Goyal, Anil Ramakrishna, Aram Galstyan, Richard Zemel, Kai-Wei Chang, Rahul Gupta, and Charith Peris. Attribute controlled fine-tuning for large language models: A case study on detoxification. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 13329–13341, Miami, Florida, USA, November 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.779. URL https://aclanthology.org/2024.findings-emnlp.779/.

Yue Meng and Chuchu Fan. Telograf: Temporal logic planning via graph-encoded flow matching. *arXiv preprint arXiv:2505.00562*, 2025.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*, 2023.

Samuel Miserendino, Michele Wang, Tejal Patwardhan, and Johannes Heidecke. SWE-lancer: Can frontier LLMs earn $1 million from real-world freelance software engineering? In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=xZXhFg43EI.

Ali Modarressi, Ayyoob Imani, Mohsen Fayyaz, and Hinrich Schütze. Ret-llm: Towards a general read-write memory for large language models. *ArXiv*, abs/2305.14322, 2023. URL https://api.semanticscholar.org/CorpusID:258841042.

Jesus Moncada-Ramirez, Jose-Luis Matez-Bandera, Javier Gonzalez-Jimenez, and Jose-Raul Ruiz-Sarmiento. Agentic workflows for improving large language model reasoning in robotic object-centered planning. *Robotics*, 14(3):24, 2025.

Sumeet Ramesh Motwani, Chandler Smith, Rocktim Jyoti Das, Rafael Rafailov, Ivan Laptev, Philip H. S. Torr, Fabio Pizzati, Ronald Clark, and Christian Schroeder de Witt. Malt: Improving reasoning with multi-agent llm training, 2025. URL https://arxiv.org/abs/2412.01928.

Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhai Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. Embodiedgpt: Vision-language pre-training via embodied chain of thought. *Advances in Neural Information Processing Systems*, 36:25081–25094, 2023.

Gabriel Mukobi, Hannah Erlebach, Niklas Lauffer, Lewis Hammond, Alan Chan, and Jesse Clifton. Welfare diplomacy: Benchmarking language model cooperation. *arXiv preprint arXiv:2310.08901*, 2023.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

Daye Nam, Andrew Macvean, Vincent Hellendoorn, Bogdan Vasilescu, and Brad Myers. Using an llm to help with code understanding. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, pp. 1–13, 2024.

Jiayan Nan, Wenquan Ma, Wenlong Wu, and Yize Chen. Nemori: Self-organizing agent memory inspired by cognitive science. *arXiv preprint arXiv:2508.03341*, 2025.

Siddharth Nayak, Adelmo M Orozco, Marina T Have, Vittal Thirumalai, Jackson Zhang, Darren Chen, Aditya Kapoor, Eric Robinson, Karthik Gopalakrishnan, James Harrison, et al. Long-horizon planning for multi-agent robots in partially observable environments. *Advances in Neural Information Processing Systems*, 37:67929–67967, 2024.

Thang Nguyen, Peter Chin, and Yu-Wing Tai. Ma-rag: Multi-agent retrieval-augmented generation via collaborative chain-of-thought reasoning, 2025. URL https://arxiv.org/abs/2505.20096.

Xuan-Phi Nguyen, Shrey Pandit, Senthil Purushwalkam, Austin Xu, Hailin Chen, Yifei Ming, Zixuan Ke, Silvio Savarese, Caiming Xong, and Shafiq Joty. Sfr-rag: Towards contextually faithful llms. *arXiv preprint arXiv:2409.09916*, 2024.

Ansong Ni, Srini Iyer, Dragomir Radev, Veselin Stoyanov, Wen-tau Yih, Sida Wang, and Xi Victoria Lin. Lever: Learning to verify language-to-code generation with execution. In *International Conference on Machine Learning*, pp. 26106–26128. PMLR, 2023.

Fei Ni, Jianye Hao, Shiguang Wu, Longxin Kou, Yifu Yuan, Zibin Dong, Jinyi Liu, MingZhi Li, Yuzheng Zhuang, and Yan Zheng. Peria: Perceive, reason, imagine, act via holistic language and vision planning for manipulation. *Advances in Neural Information Processing Systems*, 37:17541–17571, 2024.

Ruiqi Ni, Zherong Pan, and Ahmed H Qureshi. Physics-informed temporal difference metric learning for robot motion planning. *arXiv preprint arXiv:2505.05691*, 2025a.

Ziyi Ni, Yifan Li, Ning Yang, Dou Shen, Pin Lyu, and Daxiang Dong. Tree-of-code: A self-growing tree framework for end-to-end code generation and execution in complex tasks. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 9804–9819, 2025b.

Xuying Ning, Dongqi Fu, Tianxin Wei, Mengting Ai, Jiaru Zou, Ting-Wei Li, and Jingrui He. Mc-search: Benchmarking multimodal agentic rag with structured reasoning chains. In *NeurIPS 2025 Workshop on Evaluating the Evolving LLM Lifecycle: Benchmarks, Emergent Abilities, and Scaling*, 2025.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, et al. Alphaevolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

Ana Nunez, Nafis Tanveer Islam, Sumit Kumar Jha, and Peyman Najafirad. Autosafecoder: A multi-agent framework for securing llm code generation through static analysis and fuzz testing, 2024.

Theo X. Olausson, Jeevana Priya Inala, Chenglong Wang, Jianfeng Gao, and Armando Solar-Lezama. Is self-repair a silver bullet for code generation? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=y0GJXRungR.

Timothy Ossowski, Danyal Maqbool, Jixuan Chen, Zefan Cai, Tyler Bradshaw, and Junjie Hu. Comma: A communicative multimodal multi-agent benchmark, 2025. URL https://arxiv.org/abs/2410.07553.

Siru Ouyang, Jun Yan, I Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T Le, Samira Daruki, Xiangru Tang, et al. Reasoningbank: Scaling agent self-evolving with reasoning memory. *arXiv preprint arXiv:2509.25140*, 2025a.

Siru Ouyang, Wenhao Yu, Kaixin Ma, Zilin Xiao, Zhihan Zhang, Mengzhao Jia, Jiawei Han, Hongming Zhang, and Dong Yu. Repograph: Enhancing ai software engineering with repository-level code graph. In *International Conference on Learning Representations*, volume 2025, pp. 30098–30121, 2025b.

Laura O'Mahony, Leo Grinsztajn, Hailey Schoelkopf, and Stella Biderman. Attributing mode collapse in the fine-tuning of large language models. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, volume 2, 2024.

Charles Packer, Vivian Fang, Shishir G. Patil, Kevin Lin, Sarah Wooders, and Joseph Gonzalez. Memgpt: Towards llms as operating systems. *ArXiv*, abs/2310.08560, 2023. URL https://api.semanticscholar.org/CorpusID:263909014.

Davide Paglieri, Bartłomiej Cupiał, Samuel Coward, Ulyana Piterbarg, Maciej Wolczyk, Akbir Khan, Eduardo Pignatelli, Łukasz Kuciński, Lerrel Pinto, Rob Fergus, Jakob Nicolaus Foerster, Jack Parker-Holder, and Tim Rocktäschel. Balrog: Benchmarking agentic llm and vlm reasoning on games. *arXiv preprint arXiv:2411.13543*, 2024. URL https://www.arxiv.org/abs/2411.13543.

Ruwei Pan, Hongyu Zhang, and Chao Liu. CodeCoR: An llm-based self-reflective multi-agent framework for code generation, 2025.

Yichen Pan, Dehan Kong, Sida Zhou, Cheng Cui, Yifei Leng, Bing Jiang, Hangyu Liu, Yanyi Shang, Shuyan Zhou, Tongshuang Wu, and Zhengyang Wu. Webcanvas: Benchmarking web agents in online environments. *arXiv preprint arXiv:2406.12373v3*, 2024. URL https://www.arxiv.org/abs/2406.12373v3.

Bhargavi Paranjape, Scott Lundberg, Sameer Singh, Hannaneh Hajishirzi, Luke Zettlemoyer, and Marco Tulio Ribeiro. Art: Automatic multi-step reasoning and tool-use for large language models, 2023. URL https://arxiv.org/abs/2303.09014.

Chanwoo Park, Seungju Han, Xingzhi Guo, Asuman Ozdaglar, Kaiqing Zhang, and Joo-Kyung Kim. Maporl: Multi-agent post-co-training for collaborative large language models with reinforcement learning. *arXiv preprint arXiv:2502.18439*, 2025.

Ajay Patel, Markus Hofmarcher, Claudiu Leoveanu-Condrei, Marius-Constantin Dinu, Chris Callison-Burch, and Sepp Hochreiter. Large language models can self-improve at web agent tasks. *ArXiv*, abs/2405.20309, 2024. URL https://api.semanticscholar.org/CorpusID:270122967.

Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. Gorilla: Large language model connected with massive apis. *Advances in Neural Information Processing Systems*, 37:126544–126565, 2024.

Debjit Paul, Daniel Murphy, Milan Gritta, Ronald Cardenas, Victor Prokhorov, Lena Sophia Bolliger, Aysim Toker, Roy Miles, Andreea-Maria Oncescu, Jasivan Alex Sivakumar, Philipp Borchert, Ismail Elezi, Meiru Zhang, Ka Yiu Lee, Guchun Zhang, Jun Wang, and Gerasimos Lampouras. A benchmark for deep information synthesis, 2026. URL https://arxiv.org/abs/2602.21143.

Huy Nhat Phan, Hoang Nhat Phan, Tien N Nguyen, and Nghi DQ Bui. Repohyper: Search-expand-refine on semantic graphs for repository-level code completion, 2024.

Nearchos Potamitis and Akhil Arora. Are retrials all you need? enhancing large language model reasoning without verbalized feedback. *arXiv preprint arXiv:2504.12951*, 2025.

Viraj Prabhu, Yutong Dai, Matthew Fernandez, Jing Gu, Krithika Ramakrishnan, Yanqi Luo, Silvio Savarese, Caiming Xiong, Junnan Li, Zeyuan Chen, et al. Walt: Web agents that learn tools. *arXiv preprint arXiv:2510.01524*, 2025.

Archiki Prasad, Alexander Koller, Mareike Hartmann, Peter Clark, Ashish Sabharwal, Mohit Bansal, and Tushar Khot. Adapt: As-needed decomposition and planning with language models. *arXiv preprint arXiv:2311.05772*, 2023.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*, 2022.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with" gradient descent" and beam search. *arXiv preprint arXiv:2305.03495*, 2023.

Yingming Pu, Tao Lin, and Hongyu Chen. Piflow: Principle-aware scientific discovery with multi-agent collaboration. *arXiv preprint arXiv:2505.15047*, 2025.

Pranav Putta, Edmund Mills, Naman Garg, Sumeet Motwani, Chelsea Finn, Divyansh Garg, and Rafael Rafailov. Agent q: Advanced reasoning and learning for autonomous ai agents. *arXiv preprint arXiv:2408.07199*, 2024.

Biqing Qi, Kaiyan Zhang, Haoxiang Li, Kai Tian, Sihang Zeng, Zhang-Ren Chen, and Bowen Zhou. Large language models are zero shot hypothesis proposers. *arXiv preprint arXiv:2311.05965*, 2023.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Wenyi Zhao, Yu Yang, Xinyue Yang, Jiadai Sun, Shuntian Yao, et al. Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning. *arXiv preprint arXiv:2411.02337*, 2024.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. Chatdev: Communicative agents for software development. In *ACL 2024*, pp. 15174–15186. Association for Computational Linguistics, 2024a.

Chen Qian, Zihao Xie, Yifei Wang, Wei Liu, Kunlun Zhu, Hanchen Xia, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, et al. Scaling large language model-based multi-agent collaboration. *arXiv preprint arXiv:2406.07155*, 2024b.

Cheng Qian, Chi Han, Yi Fung, Yujia Qin, Zhiyuan Liu, and Heng Ji. CREATOR: Tool creation for disentangling abstract and concrete reasoning of large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 6922–6939, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.462. URL https://aclanthology.org/2023.findings-emnlp.462/.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*, 2025a.

Cheng Qian, Zuxin Liu, Shirley Kokane, Akshara Prabhakar, Jielin Qiu, Haolin Chen, Zhiwei Liu, Heng Ji, Weiran Yao, Shelby Heinecke, et al. xrouter: Training cost-aware llms orchestration system via reinforcement learning. *arXiv preprint arXiv:2510.08439*, 2025b.

Cheng Qian, Zuxin Liu, Akshara Prabhakar, Jielin Qiu, Zhiwei Liu, Haolin Chen, Shirley Kokane, Heng Ji, Weiran Yao, Shelby Heinecke, et al. Userrl: Training interactive user-centric agent via reinforcement learning. *arXiv preprint arXiv:2509.19736*, 2025c.

Haofu Qian, Chenjia Bai, Jiatao Zhang, Fei Wu, Wei Song, and Xuelong Li. Discriminator-guided embodied planning for llm agent. In *The Thirteenth International Conference on Learning Representations*, 2025d.

Shuofei Qiao, Runnan Fang, Ningyu Zhang, Yuqi Zhu, Xiang Chen, Shumin Deng, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. Agent planning with world knowledge model. *Advances in Neural Information Processing Systems*, 37:114843–114871, 2024.

Yiran Qin, Enshen Zhou, Qichang Liu, Zhenfei Yin, Lu Sheng, Ruimao Zhang, Yu Qiao, and Jing Shao. Mp5: A multi-modal open-ended embodied system in minecraft via active perception. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16307–16316. IEEE, 2024a.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024b. URL `https://openreview.net/forum?id=dHng2O0Jjr`.

Yujia Qin, Shengding Hu, Yankai Lin, Weize Chen, Ning Ding, Ganqu Cui, Zheni Zeng, Xuanhe Zhou, Yufei Huang, Chaojun Xiao, Chi Han, Yi R. Fung, Yusheng Su, Huadong Wang, Cheng Qian, Runchu Tian, Kunlun Zhu, Shihao Liang, Xingyu Shen, Bokai Xu, Zhen Zhang, Yining Ye, Bowen Li, Ziwei Tang, Jing Yi, Yuzhang Zhu, Zhenning Dai, Lan Yan, Xin Cong, Yaxi Lu, Weilin Zhao, Yuxiang Huang, Junxi Yan, Xu Han, Xian Sun, Dahai Li, Jason Phang, Cheng Yang, Tongshuang Wu, Heng Ji, Guoliang Li, Zhiyuan Liu, and Maosong Sun. Tool learning with foundation models. *ACM Comput. Surv.*, 57(4):101:1–101:40, 2025. doi: 10.1145/3704435. URL `https://doi.org/10.1145/3704435`.

Jiahao Qiu, Xuan Qi, Tongcheng Zhang, Xinzhe Juan, Jiacheng Guo, Yifu Lu, Yimin Wang, Zixin Yao, Qihan Ren, Xun Jiang, et al. Alita: Generalist agent enabling scalable agentic reasoning with minimal predefinition and maximal self-evolution. *arXiv preprint arXiv:2505.20286*, 2025.

Shi Qiu, Junyi Deng, Yiwei Deng, Haoran Dong, Jieyu Fu, Mao Li, Zeyu Li, Zhaolong Zhang, Huiwen Zheng, Leidong Bao, Anqi Lv, Zihan Mo, Yadi Niu, Yiyang Peng, Yu Tian, Yili Wang, Ziyu Wang, Zi-Yu Wang, Jiashen Wei, Liuheng Wu, Aoran Xue, Leyi Yang, Guanglu Yuan, Xiarui Zhan, Jingjun Zhang, Zifan Zheng, Pengfei Liu, Linrui Zhen, Kaiyang Li, Qichang Li, Ziheng Zhou, Guo-En Nian, Yunwei Xiao, Qing-Hong Cao, Linjie Dai, Xu Feng, Peng Gao, Ying Gu, Chang Liu, Jia Liu, Ming xing Luo, Yan-Qing Ma, Liang-You Peng, Huichao Song, Shufeng Wang, Chenxu Wang, Tao Wang, Yi-Nan Wang, Chengyin Wu, Pengwei Zhao, and Hua Xing Zhu. Prbench: End-to-end paper reproduction in physics research, 2026. URL `https://arxiv.org/abs/2603.27646`.

Ao Qu, Han Zheng, Zijian Zhou, Yihao Yan, Yihong Tang, Shao Yong Ong, Fenglu Hong, Kaichen Zhou, Chonghe Jiang, Minwei Kong, Jiacheng Zhu, Xuan Jiang, Sirui Li, Cathy Wu, Bryan Kian Hsiang Low, Jinhua Zhao, and Paul Pu Liang. Coral: Towards autonomous multi-agent evolution for open-ended discovery, 2026. URL `https://arxiv.org/abs/2604.01658`.

Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-Rong Wen. Tool learning with large language models: A survey. *Frontiers of Computer Science*, 19(8):198343, 2025a.

Yuanhao Qu, Kaixuan Huang, Ming Yin, Kanghong Zhan, Dyllan Liu, Di Yin, Henry C. Cousins, William A. Johnson, Xiaotong Wang, Mihir Shah, Russ B. Altman, Denny Zhou, Mengdi Wang, and Le Cong. Crispr-gpt for agentic automation of gene-editing experiments, 2025b. URL https://arxiv.org/abs/2404.18021.

Marissa Radensky, Simra Shahid, Raymond Fok, Pao Siangliulue, Tom Hope, and Daniel S. Weld. Human-llm compound system for scientific ideation through facet recombination and novelty evaluation, 2026. URL https://arxiv.org/abs/2409.14634.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Mohit Raghavendra, Soham Dan, Miguel Romero Calvo, Yannis Yiming He, Johannes Baptist Mols, Gautam Anand, Cole McCollum, Edgar Arakelyan, Vijay Bharadwaj, Andrew Park, Jeff Da, MohammadHossein Rezaei, Bing Liu, Brad Kenstler, and Yunzhong He. Swe atlas: Benchmarking coding agents beyond issue resolution, 2026. URL https://arxiv.org/abs/2605.08366.

Asif Rahman, Veljko Cvetkovic, Kathleen Reece, Aidan Walters, Yasir Hassan, Aneesh Tummeti, Bryan Torres, Denise Cooney, Margaret Ellis, and Dimitrios S Nikolopoulos. Marco: Multi-agent code optimization with real-time knowledge integration for high-performance computing. *arXiv preprint arXiv:2505.03906*, 2025.

Abhinav Rajvanshi, Karan Sikka, Xiao Lin, Bhoram Lee, Han-Pang Chiu, and Alvaro Velasquez. Saynav: Grounding large language models for dynamic planning to navigation in new environments. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 34, pp. 464–474, 2024.

Joaquin Ramirez-Medina, Mohammadmehdi Ataei, and Alidad Amirfazli. Accelerating scientific research through a multi-llm framework. *arXiv preprint arXiv:2502.07960*, 2025.

Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian D. Reid, and Niko Sünderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, volume 229 of *Proceedings of Machine Learning Research*, pp. 23–72. PMLR, 2023. URL https://proceedings.mlr.press/v229/rana23a.html.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: a temporal knowledge graph architecture for agent memory. *arXiv preprint arXiv:2501.13956*, 2025.

Mrinal Rawat, Ambuje Gupta, Rushil Goomer, Alessandro Di Bari, Neha Gupta, and Roberto Pieraccini. Pre-act: Multi-step planning and reasoning improves acting in llm agents. *arXiv preprint arXiv:2505.09970*, 2025.

Revanth Gangi Reddy, Tanay Dixit, Jiaxin Qin, Cheng Qian, Daniel Lee, Jiawei Han, Kevin Small, Xing Fan, Ruhi Sarikaya, and Heng Ji. Winell: wikipedia never-ending updating with llm agents. *arXiv preprint arXiv:2508.03728*, 2025a.

Revanth Gangi Reddy, Sagnik Mukherjee, Jeonghwan Kim, Zhenhailong Wang, Dilek Hakkani-Tur, and Heng Ji. Infogent: An agent-based framework for web information aggregation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 5745–5758, 2025b.

Allen Z Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, et al. Robots that ask for help: Uncertainty alignment for large language model planners. *arXiv preprint arXiv:2307.01928*, 2023.

Tao Ren, Jinyang Jiang, Hui Yang, Wan Tian, Minhao Zou, Guanghao Li, Zishi Zhang, Qinghao Wang, Shentao Qin, Yanjun Zhao, Rui Tao, Hui Shao, and Yijie Peng. Riskpo: Risk-based policy optimization via verifiable reward for llm post-training, 2025. URL https://arxiv.org/abs/2510.00911.

Cinjon Resnick, Wes Eldridge, David Ha, Denny Britz, Jakob Foerster, Julian Togelius, Kyunghyun Cho, and Joan Bruna. Pommerman: A multi-agent playground. *arXiv preprint arXiv:1809.07124*, 2018.

Alireza Rezazadeh, Zichao Li, Wei Wei, and Yujia Bao. From isolated conversations to hierarchical schemas: Dynamic tree memory representation for llms. *arXiv preprint arXiv:2410.14052*, 2024.

Alireza Rezazadeh, Zichao Li, Ange Lou, Yuying Zhao, Wei Wei, and Yujia Bao. Collaborative memory: Multi-user memory sharing in llm agents with dynamic access control. *arXiv preprint arXiv:2505.18279*, 2025.

Maxime Robeyns, Martin Szummer, and Laurence Aitchison. A self-improving coding agent. *arXiv preprint arXiv:2504.15228*, 2025.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, et al. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024.

Yusuf Roohani, Andrew Lee, Qian Huang, Jian Vora, Zachary Steinhart, Kexin Huang, Alexander Marson, Percy Liang, and Jure Leskovec. Biodiscoveryagent: An ai agent for designing genetic perturbation experiments. *arXiv preprint arXiv:2405.17631*, 2024.

Aaron Rose, Carissa Cullen, Sahar Abdelnabi, Philip Torr, Brandon Gary Kaplowitz, and Christian Schroeder de Witt. Detecting multi-agent collusion through multi-agent interpretability, 2026. URL `https://arxiv.org/abs/2604.01151`.

Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J. Maddison, and Tatsunori Hashimoto. Identifying the risks of LM agents with an LM-emulated sandbox, 2024. URL `https://arxiv.org/abs/2309.15817`.

Anian Ruoss, Grégoire Delétang, Sourabh Medapati, Jordi Grau-Moya, Li K Wenliang, Elliot Catt, John Reid, Cannada A Lewis, Joel Veness, and Tim Genewein. Amortized planning with large-scale transformers: A case study on chess. *Advances in Neural Information Processing Systems*, 37:65765–65790, 2024.

Albert Sadowski and Jaroslaw A Chudziak. On verifiable legal reasoning: A multi-agent framework with formalized knowledge representations. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 2535–2545, 2025.

Rikunari Sagara, Koichiro Terao, and Naoto Iwahashi. Beliefnest: A joint action simulator for embodied agents with theory of mind. *arXiv preprint arXiv:2505.12321*, 2025.

Swarnadeep Saha, Archiki Prasad, Justin Chen, Peter Hase, Elias Stengel-Eskin, and Mohit Bansal. System 1. x: Learning to balance fast and slow planning with language models. In *International Conference on Learning Representations*, volume 2025, pp. 38652–38679, 2025.

Rana Salama, Jason Cai, Michelle Yuan, Anna Currey, Monica Sunkara, Yi Zhang, and Yassine Benajiba. Meminsight: Autonomous memory augmentation for llm agents. *arXiv preprint arXiv:2503.21760*, 2025.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. LaMP: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7370–7392. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.399. URL `https://aclanthology.org/2024.acl-long.399/`.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge, 2019. URL `https://arxiv.org/abs/1902.04043`.

Ranjan Sapkota, Konstantinos I Roumeliotis, and Manoj Karkee. Vibe coding vs. agentic coding: Fundamentals and practical implications of agentic ai. *arXiv preprint arXiv:2505.19443*, 2025.

Ranjan Sapkota, Konstantinos I. Roumeliotis, and Manoj Karkee. Ai agents vs. agentic ai: A conceptual taxonomy, applications and challenges. *Information Fusion*, 126:103599, February 2026. ISSN 1566-2535. doi: 10.1016/j.inffus.2025.103599. URL http://dx.doi.org/10.1016/j.inffus.2025.103599.

Gabriel Sarch, Yue Wu, Michael J Tarr, and Katerina Fragkiadaki. Open-ended instructable embodied agents with memory-augmented large language models. *arXiv preprint arXiv:2310.15127*, 2023.

Gabriel Sarch, Lawrence Jang, Michael J. Tarr, William W. Cohen, Kenneth Marino, and Katerina Fragkiadaki. Vlm agents generate their own memories: Distilling experience into embodied programs of thought, 2025. URL https://arxiv.org/abs/2406.14596.

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

Samuel Schmidgall and Michael Moor. Agentrxiv: Towards collaborative autonomous research. *arXiv preprint arXiv:2503.18102*, 2025.

Samuel Schmidgall, Rojin Ziaei, Carl Harris, Eduardo Reis, Jeffrey Jopling, and Michael Moor. Agentclinic: a multimodal agent benchmark to evaluate ai in simulated clinical environments. *arXiv preprint arXiv:2405.07960*, 2024. URL https://www.arxiv.org/abs/2405.07960.

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

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

John Schultz, Jakub Adamek, Matej Jusup, Marc Lanctot, Michael Kaisers, Sarah Perrin, Daniel Hennes, Jeremy Shar, Cannada Lewis, Anian Ruoss, et al. Mastering board games by external and internal planning with language models. *arXiv preprint arXiv:2412.12119*, 2024.

Bilgehan Sel, Ahmad Al-Tawaha, Vanshaj Khattar, Ruoxi Jia, and Ming Jin. Algorithm of thoughts: Enhancing exploration of ideas in large language models. *arXiv preprint arXiv:2308.10379*, 2023.

Dhruv Shah, Błażej Osiński, Sergey Levine, et al. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Conference on robot learning*, pp. 492–504. pmlr, 2023.

Yixiang Shan, Zhengbang Zhu, Ting Long, Qifan Liang, Yi Chang, Weinan Zhang, and Liang Yin. Contradiff: Planning towards high return states via contrastive learning. In *International Conference on Learning Representations*, volume 2025, pp. 50038–50059, 2025.

Tianqi Shang, Weiqing He, Charles Zheng, Lingyao Li, Li Shen, and Bingxin Zhao. Dynamicare: A dynamic multi-agent framework for interactive and open-ended medical decision-making. *arXiv preprint arXiv:2507.02616*, 2025.

Ye Shang, Quanjun Zhang, Haichuan Hu, Chunrong Fang, Liang Xiao, and Zhenyu Chen. Breaking, stale, or missing? benchmarking coding agents on project-level test evolution, 2026. URL https://arxiv.org/abs/2605.06125.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36:38154–38180, 2023.

Zhili Shen, Chenxin Diao, Pavlos Vougiouklis, Pascual Merita, Shriram Piramanayagam, Enting Chen, Damien Graux, Andre Melo, Ruofei Lai, Zeren Jiang, et al. Gear: Graph-enhanced agent for retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 12049–12072, 2025.

Toby Shevlane, Sebastian Farquhar, Ben Garfinkel, Mary Phuong, Jess Whittlestone, Jade Leung, Daniel Kokotajlo, Nahema Marchal, Markus Anderljung, Noam Kolt, Lewis Ho, Divya Siddarth, Shahar Avin, Will Hawkins, Been Kim, Iason Gabriel, Vijay Bolina, Jack Clark, Yoshua Bengio, Paul Christiano, and Allan Dafoe. Model evaluation for extreme risks, 2023. URL `https://arxiv.org/abs/2305.15324`.

Kean Shi, Zihang Li, Tianyi Ma, Zengji Tu, Jialong Wu, Xinbo Xu, Qingyao Yang, Ruoyu Wu, Weichu Xie, Ming Wu, Jason Zeng, Michael Heinrich, Elvis Zhang, Liang Chen, Kuan Li, and Baobao Chang. Saas-bench: Can computer-use agents leverage real-world saas to solve professional workflows?, 2026. URL `https://arxiv.org/abs/2605.15777`.

Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, et al. Hi robot: Open-ended instruction following with hierarchical vision-language-action models. *arXiv preprint arXiv:2502.19417*, 2025a.

Wenqi Shi, Ran Xu, Yuchen Zhuang, Yue Yu, Jieyu Zhang, Hang Wu, Yuanda Zhu, Joyce Ho, Carl Yang, and May D. Wang. Ehragent: Code empowers large language models for few-shot complex tabular reasoning on electronic health records. *arXiv preprint arXiv:2401.07128*, 2024a. URL `https://www.arxiv.org/abs/2401.07128`.

Wenxuan Shi, Haochen Tan, Chuqiao Kuang, Xiaoguang Li, Xiaozhe Ren, Chen Zhang, Hanting Chen, Yasheng Wang, Lu Hou, and Lifeng Shang. Deepdiver: Adaptive search intensity scaling via open-web reinforcement learning, 2025b. URL `https://arxiv.org/abs/2505.24332`.

Zhengliang Shi, Shen Gao, Xiuyi Chen, Yue Feng, Lingyong Yan, Haibo Shi, Dawei Yin, Pengjie Ren, Suzan Verberne, and Zhaochun Ren. Learning to use tools via cooperative and interactive agents. *arXiv preprint arXiv:2403.03031*, 2024b.

Zhengliang Shi, Shen Gao, Lingyong Yan, Yue Feng, Xiuyi Chen, Zhumin Chen, Dawei Yin, Suzan Verberne, and Zhaochun Ren. Tool learning in the wild: Empowering language models as automatic tool agents. In *Proceedings of the ACM on Web Conference 2025*, pp. 2222–2237, 2025c.

Zhengliang Shi, Yuhan Wang, Lingyong Yan, Pengjie Ren, Shuaiqiang Wang, Dawei Yin, and Zhaochun Ren. Retrieval models aren't tool-savvy: Benchmarking tool retrieval for large language models. *CoRR*, abs/2503.01763, 2025d. doi: 10.48550/ARXIV.2503.01763. URL `https://doi.org/10.48550/arXiv.2503.01763`.

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

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

Shuzheng Si, Haozhe Zhao, Kangyang Luo, Gang Chen, Fanchao Qi, Minjia Zhang, Baobao Chang, and Maosong Sun. A goal without a plan is just a wish: Efficient and effective global planner training for long-horizon agent tasks, 2026. URL `https://arxiv.org/abs/2510.05608`.

Toby Simonds and Akira Yoshiyama. Ladder: Self-improving llms through recursive problem decomposition. *arXiv preprint arXiv:2503.00735*, 2025.

Toby Simonds, Kevin Lopez, Akira Yoshiyama, and Dominique Garmier. Self rewarding self improving. *arXiv e-prints*, pp. arXiv–2505, 2025.

Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. Agentic retrieval-augmented generation: A survey on agentic rag. *arXiv preprint arXiv:2501.09136*, 2025a.

Joykirat Singh, Raghav Magazine, Yash Pandya, and Akshay Nambi. Agentic reasoning and tool integration for llms via reinforcement learning. *arXiv preprint arXiv:2505.01441v1*, 2025b. URL `https://www.arxiv.org/abs/2505.01441v1`.

Arnav Singhvi, Manish Shetty, Shangyin Tan, Christopher Potts, Koushik Sen, Matei Zaharia, and Omar Khattab. Dspy assertions: Computational constraints for self-refining language model pipelines. *arXiv preprint arXiv:2312.13382*, 2023.

Rohan Sinha, Amine Elhafsi, Christopher Agia, Matthew Foutter, Edward Schmerling, and Marco Pavone. Real-time anomaly detection and reactive planning with large language models. *arXiv preprint arXiv:2407.08735*, 2024.

Ved Sirdeshmukh, Kaustubh Deshpande, Johannes Mols, Lifeng Jin, Ed-Yeremai Cardona, Dean Lee, Jeremy Kritz, Willow Primack, Summer Yue, and Chen Xing. Multichallenge: A realistic multi-turn conversation evaluation benchmark challenging to frontier llms. *arXiv preprint arXiv:2501.17399*, 2025.

Michael D Skarlinski, Sam Cox, Jon M Laurent, James D Braza, Michaela Hinks, Michael J Hammerling, Manvitha Ponnapati, Samuel G Rodriques, and Andrew D White. Language agents achieve superhuman synthesis of scientific knowledge. *arXiv preprint arXiv:2409.13740*, 2024.

Alexey Skrynnik, Anton Andreychuk, Anatolii Borzilov, Alexander Chernyavskiy, Konstantin Yakovlev, and Aleksandr Panov. Pogema: A benchmark platform for cooperative multi-agent pathfinding. *arXiv preprint arXiv:2407.14931*, 2024.

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2998–3009, 2023a.

Yifan Song, Weimin Xiong, Dawei Zhu, Wenhao Wu, Han Qian, Mingbo Song, Hailiang Huang, Cheng Li, Ke Wang, Rong Yao, et al. Restgpt: Connecting large language models with real-world restful apis. *arXiv preprint arXiv:2306.06624*, 2023b.

Yiwen Song, Yale Song, Tomas Pfister, and Jinsung Yoon. Paperorchestra: A multi-agent framework for automated ai research paper writing, 2026. URL `https://arxiv.org/abs/2604.05018`.

Yixiao Song, Katherine Thai, Chau Minh Pham, Yapei Chang, Mazin Nadaf, and Mohit Iyyer. Bearcubs: A benchmark for computer-using web agents. *arXiv preprint arXiv:2503.07919*, 2025a.

Yuhang Song, Andrzej Wojcicki, Thomas Lukasiewicz, Jianyi Wang, Abi Aryan, Zhenghua Xu, Mai Xu, Zihan Ding, and Lianlong Wu. Arena: A general evaluation platform and building toolkit for multi-agent intelligence. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7253–7260, 2020.

Zirui Song, Guangxian Ouyang, Mingzhe Li, Yuheng Ji, Chenxi Wang, Zixiang Xu, Zeyu Zhang, Xiaoqing Zhang, Qian Jiang, Zhenhao Chen, et al. Maniplvm-r1: Reinforcement learning for reasoning in embodied manipulation with large vision-language models. *arXiv preprint arXiv:2505.16517*, 2025b.

Henry Sprueill, Carl Edwards, Mariefel Olarte, Udishnu Sanyal, Heng Ji, and Sutanay Choudhury. Monte carlo thought search: Large language model querying for complex scientific reasoning in catalyst design. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 8348–8365, 2023.

Henry W Sprueill, Carl Edwards, Khushbu Agarwal, Mariefel V Olarte, Udishnu Sanyal, Conrad Johnston, Hongbin Liu, Heng Ji, and Sutanay Choudhury. Chemreasoner: Heuristic search over a large language model's knowledge space using quantum-chemical feedback. *arXiv preprint arXiv:2402.10980*, 2024.

Hongjin Su, Shizhe Diao, Ximing Lu, Mingjie Liu, Jiacheng Xu, Xin Dong, Yonggan Fu, Peter Belcak, Hanrong Ye, Hongxu Yin, Yi Dong, Evelina Bakhturina, Tao Yu, Yejin Choi, Jan Kautz, and Pavlo Molchanov. Toolorchestra: Elevating intelligence via efficient model and tool orchestration, 2025a. URL https://arxiv.org/abs/2511.21689.

Yanzhou Su, Tianbin Li, Jiyao Liu, Chenglong Ma, Junzhi Ning, Cheng Tang, Sibo Ju, Jin Ye, Pengcheng Chen, Ming Hu, et al. Gmai-vl-r1: Harnessing reinforcement learning for multimodal medical reasoning. *arXiv preprint arXiv:2504.01886*, 2025b.

Vighnesh Subramaniam, Yilun Du, Joshua B Tenenbaum, Antonio Torralba, Shuang Li, and Igor Mordatch. Multiagent finetuning: Self improvement with diverse reasoning chains. *arXiv preprint arXiv:2501.05707*, 2025.

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. Zerosearch: Incentivize the search capability of llms without searching. *arXiv preprint arXiv:2505.04588*, 2025a.

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. Zerosearch: Incentivize the search capability of llms without searching, 2026. URL https://arxiv.org/abs/2505.04588.

Haotian Sun, Yuchen Zhuang, Lingkai Kong, Bo Dai, and Chao Zhang. Adaplanner: Adaptive planning from feedback with language models. *Advances in neural information processing systems*, 36:58202–58245, 2023.

Jingyun Sun, Chengxiao Dai, Zhongze Luo, Yangbo Chang, and Yang Li. Lawluo: A multi-agent collaborative framework for multi-round chinese legal consultation. *arXiv preprint arXiv:2407.16252*, 2024a.

Linzhuang Sun, Hao Liang, Jingxuan Wei, Bihui Yu, Tianpeng Li, Fan Yang, Zenan Zhou, and Wentao Zhang. Mm-verify: Enhancing multimodal reasoning with chain-of-thought verification. In *ACL*, 2025b. URL https://aclanthology.org/2025.acl-long.689/.

Qi Sun, Pengfei Hong, Tej Deep Pala, Vernon Toh, U Tan, Deepanway Ghosal, Soujanya Poria, et al. Emma-x: An embodied multimodal action model with grounded chain of thought and look-ahead spatial reasoning. *arXiv preprint arXiv:2412.11974*, 2024b.

Xin Sun, Zhongqi Chen, Xing Zheng, Qiang Liu, Shu Wu, Bowen Song, Zilei Wang, Weiqiang Wang, and Liang Wang. Kbqa-r1: Reinforcing large language models for knowledge base question answering. *arXiv preprint arXiv:2512.10999*, 2025c.

Zhongxiang Sun, Qipeng Wang, Weijie Yu, Xiaoxue Zang, Kai Zheng, Jun Xu, Xiao Zhang, Yang Song, and Han Li. Rearter: Retrieval-augmented reasoning with trustworthy process rewarding. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1251–1261, 2025d.

Mirac Suzgun, Mert Yuksekgonul, Federico Bianchi, Dan Jurafsky, and James Zou. Dynamic cheatsheet: Test-time learning with adaptive memory. *arXiv preprint arXiv:2504.07952*, 2025.

Kyle Swanson, Wesley Wu, Nash L Bulaong, John E Pak, and James Zou. The virtual lab: Ai agents design new sars-cov-2 nanobodies with experimental validation. *bioRxiv*, pp. 2024–11, 2024.

Grzegorz Swirszcz, Adam Zsolt Wagner, Geordie Williamson, Sam Blackwell, Bogdan Georgiev, Alex Davies, Ali Eslami, Sebastien Racaniere, Theophane Weber, and Pushmeet Kohli. Advancing geometry with ai: Multi-agent generation of polytopes. *arXiv preprint arXiv:2502.05199*, 2025.

Izumi Takahara, Teruyasu Mizoguchi, and Bang Liu. Accelerated inorganic materials design with generative ai agents. *arXiv preprint arXiv:2504.00741*, 2025.

Haoran Tan, Zeyu Zhang, Chen Ma, Xu Chen, Quanyu Dai, and Zhenhua Dong. Membench: Towards more comprehensive evaluation on the memory of llm-based agents. *arXiv preprint arXiv:2506.21605*, 2025a.

Weihao Tan, Changjiu Jiang, Yu Duan, Mingcong Lei, Jiageng Li, Yitian Hong, Xinrun Wang, and Bo An. Stardojo: Benchmarking open-ended behaviors of agentic multimodal llms in production-living simulations with stardew valley. *arXiv preprint arXiv:2507.07445v2*, 2025b. URL https://www.arxiv.org/abs/2507.07445v2.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long T. Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, Anand Iyer, Tianlong Chen, Huan Liu, Chen-Yu Lee, and Tomas Pfister. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents, 2025c. URL https://arxiv.org/abs/2503.08026.

Hao Tang, Darren Key, and Kevin Ellis. Worldcoder, a model-based llm agent: Building world models by writing code and interacting with the environment. In *Conference on Neural Information Processing Systems (NeurIPS)*, pp. 70148–70212, 2024.

Qiaoyu Tang, Ziliang Deng, Hongyu Lin, Xianpei Han, Qiao Liang, and Le Sun. Toolalpaca: Generalized tool learning for language models with 3000 simulated cases. *CoRR*, abs/2306.05301, 2023a. doi: 10.48550/ARXIV.2306.05301. URL https://doi.org/10.48550/arXiv.2306.05301.

Xiangru Tang, Anni Zou, Zhuosheng Zhang, Ziming Li, Yilun Zhao, Xingyao Zhang, Arman Cohan, and Mark Gerstein. Medagents: Large language models as collaborators for zero-shot medical reasoning. *arXiv preprint arXiv:2311.10537*, 2023b. URL https://www.arxiv.org/abs/2311.10537.

Xiangru Tang, Tianyu Hu, Muyang Ye, Yanjun Shao, Xunjian Yin, Siru Ouyang, Wangchunshu Zhou, Pan Lu, Zhuosheng Zhang, Yilun Zhao, et al. Chemagent: Self-updating library in large language models improves chemical reasoning. *arXiv preprint arXiv:2501.06590*, 2025a.

Xiangru Tang, Tianrui Qin, Tianhao Peng, Ziyang Zhou, Daniel Shao, Tingting Du, Xinming Wei, Peng Xia, Fang Wu, He Zhu, et al. Agent kb: Leveraging cross-domain experience for agentic problem solving. *arXiv preprint arXiv:2507.06229*, 2025b.

Xiangru Tang, Daniel Shao, Jiwoong Sohn, Jiapeng Chen, Jiayi Zhang, Jinyu Xiang, Fang Wu, Yilun Zhao, Chenglin Wu, Wenqi Shi, Arman Cohan, and Mark Gerstein. Medagentsbench: Benchmarking thinking models and agent frameworks for complex medical reasoning. *arXiv preprint arXiv:2503.07459*, 2025c. URL https://www.arxiv.org/abs/2503.07459.

Yuheng Tang, Kaijie Zhu, Bonan Ruan, Chuqi Zhang, Michael Yang, Hongwei Li, Suyue Guo, Tianneng Shi, Zekun Li, Christopher Kruegel, Giovanni Vigna, Dawn Song, William Yang Wang, Lun Wang, Yangruibo Ding, Zhenkai Liang, and Wenbo Guo. Devops-gym: Benchmarking ai agents in software devops cycle, 2026a. URL https://arxiv.org/abs/2601.20882.

Zirui Tang, Xuanhe Zhou, Yumou Liu, Linchun Li, Yukai Wu, Weizheng Wang, Hongzhang Huang, Wei Zhou, Jun Zhou, Jiachen Song, Shaoli Yu, Jinqi Wang, Zihang Zhou, Hongyi Zhou, Yuting Lv, Jinyang Li, Jiashuo Liu, Ruoyu Chen, Chunwei Liu, GuoLiang Li, Jihua Kang, and Fan Wu. Workspace-bench 1.0: Benchmarking ai agents on workspace tasks with large-scale file dependencies, 2026b. URL https://arxiv.org/abs/2605.03596.

Xijia Tao, Yihua Teng, Xinxing Su, Xinyu Fu, Jihao Wu, Chaofan Tao, Ziru Liu, Haoli Bai, Rui Liu, and Lingpeng Kong. Mmsearch-plus: Benchmarking provenance-aware search for multimodal browsing agents. *arXiv preprint arXiv:2508.21475*, 2025a.

Zhengwei Tao, Jialong Wu, Wenbiao Yin, Junkai Zhang, Baixuan Li, Haiyang Shen, Kuan Li, Liwen Zhang, Xinyu Wang, Yong Jiang, et al. Webshaper: Agentically data synthesizing via information-seeking formalization. *arXiv preprint arXiv:2507.15061*, 2025b.

CocoaBench Team, Shibo Hao, Zhining Zhang, Zhiqi Liang, Tianyang Liu, Yuheng Zha, Qiyue Gao, Jixuan Chen, Zilong Wang, Zhoujun Cheng, Haoxiang Zhang, Junli Wang, Hexi Jin, Boyuan Zheng, Kun Zhou, Yu Wang, Feng Yao, Licheng Liu, Yijiang Li, Zhifei Li, Zhengtao Han, Pracha Promthaw, Tommaso Cerruti, Xiaohan Fu, Ziqiao Ma, Jingbo Shang, Lianhui Qin, Julian McAuley, Eric P. Xing, Zhengzhong

Liu, Rupesh Kumar Srivastava, and Zhiting Hu. Cocoabench: Evaluating unified digital agents in the wild, 2026. URL https://arxiv.org/abs/2604.11201.

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025a.

InternAgent Team, Bo Zhang, Shiyang Feng, Xiangchao Yan, Jiakang Yuan, Runmin Ma, Yusong Hu, Zhiyin Yu, Xiaohan He, Songtao Huang, Shaowei Hou, Zheng Nie, Zhilong Wang, Jinyao Liu, Tianshuo Peng, Peng Ye, Dongzhan Zhou, Shufei Zhang, Xiaosong Wang, Yilan Zhang, Meng Li, Zhongying Tu, Xiangyu Yue, Wangli Ouyang, Bowen Zhou, and Lei Bai. Internagent: When agent becomes the scientist – building closed-loop system from hypothesis to verification, 2025b. URL https://arxiv.org/abs/2505.16938.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025c.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025d.

Karishma Thakrar, Shreyas Basavatia, and Akshay Daftardar. Architecting clinical collaboration: Multi-agent reasoning systems for multimodal medical vqa, 2025. URL https://arxiv.org/abs/2507.05520.

Amitayush Thakur, George Tsoukalas, Yeming Wen, Jimmy Xin, and Swarat Chaudhuri. An in-context learning agent for formal theorem-proving, 2024. URL https://arxiv.org/abs/2310.04353.

Lucas-Andrei Thil, Mirela Popa, and Gerasimos Spanakis. Navigating webai: Training agents to complete web tasks with large language models and reinforcement learning. In *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*, pp. 866–874, 2024.

Thibaut Thonet, Jos Rozen, and Laurent Besacier. Elitr-bench: A meeting assistant benchmark for long-context language models. *arXiv preprint arXiv:2403.20262*, 2024.

Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024.

Patara Trirat, Wonyong Jeong, and Sung Ju Hwang. Automl-agent: A multi-agent llm framework for full-pipeline automl. *arXiv preprint arXiv:2410.02958*, 2024.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*, 2022.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. Appworld: A controllable world of apps and people for benchmarking interactive coding agents, 2024. URL https://arxiv.org/abs/2407.18901.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change. *Advances in Neural Information Processing Systems*, 36:38975–38987, 2023a.

Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. On the planning abilities of large language models: A critical investigation. *Advances in Neural Information Processing Systems*, 36:75993–76005, 2023b.

Eugene Vinitsky, Nathan Lichtlé, Xiaomeng Yang, Brandon Amos, and Jakob Foerster. Nocturne: a scalable driving benchmark for bringing multi-agent learning one step closer to the real world. *Advances in Neural Information Processing Systems*, 35:3962–3974, 2022.

Kaustubh Vyas, Damien Graux, Yijun Yang, Sébastien Montella, Chenxin Diao, Wendi Zhou, Pavlos Vougiouklis, Ruofei Lai, Yang Ren, Keshuang Li, et al. From an llm swarm to a pddl-empowered hive: Planning self-executed instructions in a multi-modal jungle. *arXiv preprint arXiv:2412.12839*, 2024.

Luanbo Wan and Weizhi Ma. Storybench: A dynamic benchmark for evaluating long-term memory with multi turns. *arXiv preprint arXiv:2506.13356*, 2025.

Ziyu Wan, Yunxiang Li, Xiaoyu Wen, Yan Song, Hanjing Wang, Linyi Yang, Mark Schmidt, Jun Wang, Weinan Zhang, Shuyue Hu, et al. Rema: Learning to meta-think for llms with multi-agent reinforcement learning. *arXiv preprint arXiv:2503.09501*, 2025.

Bing Wang, Xinnian Liang, Jian Yang, Hui Huang, Zhenhe Wu, ShuangZhi Wu, Zejun Ma, and Zhoujun Li. Scm: Enhancing large language model with self-controlled memory framework. In *International Conference on Database Systems for Advanced Applications*, pp. 188–203. Springer, 2025a.

Borui Wang, Kathleen McKeown, and Rex Ying. Dystil: Dynamic strategy induction with large language models for reinforcement learning. *arXiv preprint arXiv:2505.03209*, 2025b.

Chaojie Wang, Yanchen Deng, Zhiyi Lyu, Liang Zeng, Jujie He, Shuicheng Yan, and Bo An. Q*: Improving multi-step reasoning for llms with deliberative planning. *arXiv preprint arXiv:2406.14283*, 2024a.

Daoyu Wang, Mingyue Cheng, Shuo Yu, Zirui Liu, Ze Guo, Xin Li, and Qi Liu. Paperarena: An evaluation benchmark for tool-augmented agentic reasoning on scientific literature, 2026a. URL `https://arxiv.org/abs/2510.10909`.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023a.

Hang Wang, Xin Ye, Feng Tao, Chenbin Pan, Abhirup Mallik, Burhaneddin Yaman, Liu Ren, and Junshan Zhang. Adawm: Adaptive world model based planning for autonomous driving. *arXiv preprint arXiv:2501.13072*, 2025c.

Haochun Wang, Sendong Zhao, Zewen Qiang, Nuwa Xi, Bing Qin, and Ting Liu. Beyond direct diagnosis: Llm-based multi-specialist agent consultation for automatic diagnosis. *arXiv preprint arXiv:2401.16107*, 2024b.

Hongru Wang, Yujia Qin, Yankai Lin, Jeff Z Pan, and Kam-Fai Wong. Empowering large language models: Tool learning for real-world interaction. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2983–2986, 2024c.

Jingbo Wang, Sendong Zhao, Haochun Wang, Yuzheng Fan, Lizhe Zhang, Yan Liu, and Ting Liu. Optimal-agent-selection: State-aware routing framework for efficient multi-agent collaboration. *arXiv preprint arXiv:2511.02200*, 2025d.

Jize Wang, Zerun Ma, Yining Li, Songyang Zhang, Cailian Chen, Kai Chen, and Xinyi Le. Gta: A benchmark for general tool agents. *arXiv preprint arXiv:2407.08713*, 2024d. URL `https://www.arxiv.org/abs/2407.08713`.

Jize Wang, Xuanxuan Liu, Yining Li, Songyang Zhang, Yijun Wang, Zifei Shan, Xinyi Le, Cailian Chen, Xinping Guan, and Dacheng Tao. Gta-2: Benchmarking general tool agents from atomic tool-use to open-ended workflows, 2026b. URL `https://arxiv.org/abs/2604.15715`.

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

Junyang Wang, Haiyang Xu, Xi Zhang, Ming Yan, Ji Zhang, Fei Huang, and Jitao Sang. Mobile-agent-v: A video-guided approach for effortless and efficient operational knowledge injection in mobile automation, 2025e. URL `https://arxiv.org/abs/2502.17110`.

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

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2609–2634, 2023b.

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

Mianchu Wang, Rui Yang, Xi Chen, Hao Sun, Meng Fang, and Giovanni Montana. Goplan: Goal-conditioned offline reinforcement learning by planning with learned models. *arXiv preprint arXiv:2310.20025*, 2023c.

Miles Wang, Robi Lin, Kat Hu, Joy Jiao, Neil Chowdhury, Ethan Chang, and Tejal Patwardhan. Frontierscience: Evaluating ai's ability to perform expert-level scientific tasks, 2026c. URL `https://arxiv.org/abs/2601.21165`.

Pei Wang, Yanan Wu, Noah Wang, Jiaheng Liu, Xiaoshuai Song, Z. Y. Peng, Ken Deng, Chenchen Zhang, Jiakai Wang, Junran Peng, Ge Zhang, Hangyu Guo, Zhaoxiang Zhang, Wenbo Su, and Bo Zheng. Mtu-bench: A multi-granularity tool-use benchmark for large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025g. URL `https://openreview.net/forum?id=6guG2OlXsr`.

Renxi Wang, Xudong Han, Lei Ji, Shu Wang, Timothy Baldwin, and Haonan Li. Toolgen: Unified tool retrieval and calling via generation. In *International Conference on Learning Representations (ICLR)*, 2025h.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Scienceworld: Is your agent smarter than a 5th grader? *arXiv preprint arXiv:2203.07540*, 2022. URL `https://www.arxiv.org/abs/2203.07540`.

Shilong Wang, Guibin Zhang, Miao Yu, Guancheng Wan, Fanci Meng, Chongye Guo, Kun Wang, and Yang Wang. G-safeguard: A topology-guided security lens and treatment on llm-based multi-agent systems. *arXiv preprint arXiv:2502.11127*, 2025i.

Tevin Wang and Chenyan Xiong. Autorule: Reasoning chain-of-thought extracted rule-based rewards improve preference learning. *arXiv preprint arXiv:2506.15651*, 2025.

Wei Wang, Dan Zhang, Tao Feng, Boyan Wang, and Jie Tang. Battleagentbench: A benchmark for evaluating cooperation and competition capabilities of language models in multi-agent systems. *arXiv preprint arXiv:2408.15971*, 2024g.

Wenhao Wang, Yanyan Li, Long Jiao, and Jiawei Yuan. Gsce: A prompt framework with enhanced reasoning for reliable llm-driven drone control. In *2025 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 441–448. IEEE, 2025j.

Xiaoqiang Wang and Bang Liu. Oscar: Operating system control via state-aware reasoning and re-planning. *arXiv preprint arXiv:2410.18963*, 2024.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. Executable code actions elicit better llm agents. In *Forty-first International Conference on Machine Learning*, 2024h.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, et al. Openhands: An open platform for ai software developers as generalist agents. *arXiv preprint arXiv:2407.16741*, 2024i.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023d. URL https://openreview.net/forum?id=1PL1NIMMrw.

Yiying Wang, Xiaojing Li, Binzhu Wang, Yueyang Zhou, Yingru Lin, Han Ji, Hong Chen, Jinshi Zhang, Fei Yu, Zewei Zhao, et al. Peer: Expertizing domain-specific tasks with a multi-agent framework and tuning methods. *arXiv preprint arXiv:2407.06985*, 2024j.

Yu Wang and Xi Chen. Mirix: Multi-agent memory system for llm-based agents. *arXiv preprint arXiv:2507.07957*, 2025.

Yu Wang, Ryuichi Takanobu, Zhiqi Liang, Yuzhen Mao, Yuanzhe Hu, Julian McAuley, and Xiaojian Wu. Mem-{\alpha}: Learning memory construction via reinforcement learning. *arXiv preprint arXiv:2509.25911*, 2025k.

Yuanfei Wang, Xinju Huang, Fangwei Zhong, Yaodong Yang, Yizhou Wang, Yuanpei Chen, and Hao Dong. Communication-efficient desire alignment for embodied agent-human adaptation, 2025l. URL https://arxiv.org/abs/2505.22503.

Yuyao Wang, Zhongjian Zhang, Mo Chi, Kaichi Yu, Yuhan Li, Miao Peng, Bing Tong, Chen Zhang, Yan Zhou, and Jia Li. Evomembench: Benchmarking agent memory from a self-evolving perspective, 2026d. URL https://arxiv.org/abs/2605.18421.

Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. Unleashing cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. In *Proc. 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL2024)*, 2024k.

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

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, et al. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025n.

Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Ma, and Yitao Liang. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents, 2024l. URL https://arxiv.org/abs/2302.01560.

Ziliang Wang, Xuhui Zheng, Kang An, Cijun Ouyang, Jialu Cai, Yuhang Wang, and Yichao Wu. Stepsearch: Igniting llms search ability via step-wise proximal policy optimization, 2025o. URL https://arxiv.org/abs/2505.15107.

Zilong Wang, Yuedong Cui, Li Zhong, Zimin Zhang, Da Yin, Bill Yuchen Lin, and Jingbo Shang. Officebench: Benchmarking language agents across multiple applications for office automation. *arXiv preprint arXiv:2407.19056*, 2024m. URL https://www.arxiv.org/abs/2407.19056.

Ziqi Wang, Le Hou, Tianjian Lu, Yuexin Wu, Yunxuan Li, Hongkun Yu, and Heng Ji. Enable language models to implicitly learn self-improvement from data. In *Proc. The Twelfth International Conference on Learning Representations (ICLR2024)*, 2024n.

Ziwei Wang, Leyang Yang, Xiaoxuan Tang, Sheng Zhou, Dajun Chen, Wei Jiang, and Yong Li. History-aware reasoning for gui agents. *arXiv preprint arXiv:2511.09127*, 2025p.

Ziyan Wang, Zhicheng Zhang, Fei Fang, and Yali Du. M3hf: Multi-agent reinforcement learning from multi-phase human feedback of mixed quality. *arXiv preprint arXiv:2503.02077*, 2025q.

Ziyue Wang, Junde Wu, Linghan Cai, Chang Han Low, Xihong Yang, Qiaxuan Li, and Yueming Jin. Medagent-pro: Towards evidence-based multi-modal medical diagnosis via reasoning agentic workflow. *arXiv preprint arXiv:2503.18968*, 2025r.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. *ArXiv*, abs/2409.07429, 2024o. URL `https://api.semanticscholar.org/CorpusID:272592995`.

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

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents, 2025a. URL `https://arxiv.org/abs/2504.12516`.

Quan Wei, Siliang Zeng, Chenliang Li, William Brown, Oana Frunza, Wei Deng, Anderson Schneider, Yuriy Nevmyvaka, Yang Katie Zhao, Alfredo Garcia, and Mingyi Hong. Reinforcing multi-turn reasoning in LLM agents via turn-level reward design, 2025b. URL `https://arxiv.org/abs/2505.11821`.

Tianxin Wei, Noveen Sachdeva, Benjamin Coleman, Zhankui He, Yuanchen Bei, Xuying Ning, Mengting Ai, Yunzhe Li, Jingrui He, Ed H Chi, et al. Evo-memory: Benchmarking llm agent test-time learning with self-evolving memory. *arXiv preprint arXiv:2511.20857*, 2025c.

Yuxiang Wei, Olivier Duchenne, Jade Copet, Quentin Carbonneaux, Lingming Zhang, Daniel Fried, Gabriel Synnaeve, Rishabh Singh, and Sida I Wang. Swe-rl: Advancing llm reasoning via reinforcement learning on open software evolution. *arXiv preprint arXiv:2502.18449*, 2025d.

Zhepei Wei, Wenlin Yao, Yao Liu, Weizhi Zhang, Qin Lu, Liang Qiu, Changlong Yu, Puyang Xu, Chao Zhang, Bing Yin, et al. Webagent-r1: Training web agents via end-to-end multi-turn reinforcement learning. *arXiv preprint arXiv:2505.16421*, 2025e.

Hao Wen, Yuanchun Li, Guohong Liu, Shanhui Zhao, Tao Yu, Toby Jia-Jun Li, Shiqi Jiang, Yunhao Liu, Yaqin Zhang, and Yunxin Liu. Autodroid: Llm-powered task automation in android. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pp. 543–557, 2024.

Hao Wen, Shizuo Tian, Borislav Pavlov, Wenjie Du, Yixuan Li, Ge Chang, Shanhui Zhao, Jiacheng Liu, Yunxin Liu, Ya-Qin Zhang, et al. Autodroid-v2: Boosting slm-based gui agents via code generation. In *Proceedings of the 23rd Annual International Conference on Mobile Systems, Applications and Services*, pp. 223–235, 2025a.

Jiaxin Wen, Jian Guan, Hongning Wang, Wei Wu, and Minlie Huang. Codeplan: Unlocking reasoning potential in large language models by scaling code-form planning. In *International Conference on Learning Representations*, volume 2025, pp. 78641–78663, 2025b.

Yixuan Weng, Minjun Zhu, Guangsheng Bao, Hongbo Zhang, Jindong Wang, Yue Zhang, and Linyi Yang. Cycleresearcher: Improving automated research via automated review. *arXiv preprint arXiv:2411.00816*, 2024.

Matt L. Wiemann, Lindsay M. Smith, Peter Melchior, Siddharth Mishra-Sharma, Andrew Gordon Wilson, Pavel Izmailov, and Carolina Cuesta-Lázaro. Discoverphysics: Benchmarking llms for out-of-the-box scientific thinking, 2026. URL `https://arxiv.org/abs/2605.26087`.

Ross Williams, Niyousha Hosseinichimeh, Aritra Majumdar, and Navid Ghaffarzadegan. Epidemic modeling with generative agents. *arXiv preprint arXiv:2307.04986*, 2023.

Simon Willison. Not all ai-assisted programming is vibe coding (but vibe coding rocks). `https://simonwillison.net/2025/Mar/19/vibe-coding/`, 2025. Blog post.

Ryan Wong, Jiawei Wang, Junjie Zhao, Li Chen, Yan Gao, Long Zhang, Xuan Zhou, Zuo Wang, Kai Xiang, Ge Zhang, et al. Widesearch: Benchmarking agentic broad info-seeking. *arXiv preprint arXiv:2508.07999*, 2025.

Bin Wu, Arastun Mammadli, Xiaoyu Zhang, and Emine Yilmaz. Agentsearchbench: A benchmark for ai agent search in the wild, 2026. URL https://arxiv.org/abs/2604.22436.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813*, 2024a.

Feijie Wu, Zitao Li, Fei Wei, Yaliang Li, Bolin Ding, and Jing Gao. Talk to right specialists: Routing and planning in multi-agent system for question answering. *arXiv preprint arXiv:2501.07813*, 2025a.

Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Gang Fu, Yong Jiang, et al. Webdancer: Towards autonomous information seeking agency. *arXiv preprint arXiv:2505.22648*, 2025b.

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, et al. Webwalker: Benchmarking llms in web traversal. *arXiv preprint arXiv:2501.07572*, 2025c.

Juncheng Wu, Wenlong Deng, Xingxuan Li, Sheng Liu, Taomian Mi, Yifan Peng, Ziyang Xu, Yi Liu, Hyunjin Cho, Chang-In Choi, et al. Medreason: Eliciting factual medical reasoning steps in llms via knowledge graphs. *arXiv preprint arXiv:2504.00993*, 2025d.

Junde Wu, Jiayuan Zhu, Yuyuan Liu, Min Xu, and Yueming Jin. Agentic reasoning: A streamlined framework for enhancing llm reasoning with agentic tools. 2025e. URL https://arxiv.org/abs/2502.04644.

Mengsong Wu, YaFei Wang, Yidong Ming, Yuqi An, Yuwei Wan, Wenliang Chen, Binbin Lin, Yuqiang Li, Tong Xie, and Dongzhan Zhou. Chematagent: Enhancing llms for chemistry and materials science through tree-search based tool learning, 2025f. URL https://arxiv.org/abs/2506.07551.

Mengsong Wu, Tong Zhu, Han Han, Xiang Zhang, Wenbiao Shao, and Wenliang Chen. Chain-of-tools: Utilizing massive unseen tools in the cot reasoning of frozen language models. *CoRR*, abs/2503.16779, 2025g. doi: 10.48550/ARXIV.2503.16779. URL https://doi.org/10.48550/arXiv.2503.16779.

Mingrui Wu, Zhaozhi Wang, Fangjinhua Wang, Jiaolong Yang, Marc Pollefeys, and Tong Zhang. From indoor to open world: Revealing the spatial reasoning gap in mllms, 2025h. URL https://arxiv.org/abs/2512.19683.

Penghao Wu, Shengnan Ma, Bo Wang, Jiaheng Yu, Lewei Lu, and Ziwei Liu. Gui-reflection: Empowering multimodal gui models with self-reflection behavior. *arXiv preprint arXiv:2506.08012*, 2025i.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. Autogen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*, 2024b. URL https://openreview.net/forum?id=BAakY1hNKS.

Qinzhuo Wu, Wei Liu, Jian Luan, and Bin Wang. Toolplanner: A tool augmented llm for multi granularity instructions with path planning and feedback, 2024c. URL https://arxiv.org/abs/2409.14826.

Shirley Wu, Shiyu Zhao, Qian Huang, Kexin Huang, Michihiro Yasunaga, Kaidi Cao, Vassilis N. Ioannidis, Karthik Subbian, Jure Leskovec, and James Zou. Avatar: Optimizing llm agents for tool usage via contrastive reasoning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 25981–26010. Curran Associates, Inc., 2024d.

Yue Wu, So Yeon Min, Yonatan Bisk, Ruslan Salakhutdinov, Amos Azaria, Yuanzhi Li, Tom Mitchell, and Shrimai Prabhumoye. Plan, eliminate, and track – language models are good teachers for embodied agents, 2023. URL https://arxiv.org/abs/2305.02412.

Yuheng Wu, Wentao Guo, Zirui Liu, Heng Ji, Zhaozhuo Xu, and Denghui Zhang. How large language models encode theory-of-mind: a study on sparse parameter patterns. *npj Artificial Intelligence*, 1(1):20, 2025j.

Zhiyong Wu, Chengcheng Han, Zichen Ding, Zhenmin Weng, Zhoumianze Liu, Shunyu Yao, Tao Yu, and Lingpeng Kong. Os-copilot: Towards generalist computer agents with self-improvement. *arXiv preprint arXiv:2402.07456*, 2024e.

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, et al. Os-atlas: A foundation action model for generalist gui agents. *arXiv preprint arXiv:2410.23218*, 2024f.

Georg Wölflein, Dyke Ferber, Daniel Truhn, Ognjen Arandjelović, and Jakob Nikolas Kather. Llm agents making agent tools, 2025. URL `https://arxiv.org/abs/2502.11705`.

Yunjia Xi, Jianghao Lin, Menghui Zhu, Yongzhao Xiao, Zhuoying Ou, Jiaqi Liu, Tong Wan, Bo Chen, Weiwen Liu, Yasheng Wang, et al. Infodeepseek: Benchmarking agentic information seeking for retrieval-augmented generation. *arXiv preprint arXiv:2505.15872*, 2025.

Yufei Xiang, Yiqun Shen, Yeqin Zhang, and Nguyen Cam-Tu. Retrospex: Language agent meets offline reinforcement learning critic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 4650–4666. Association for Computational Linguistics, 2024a. doi: 10.18653/v1/2024.emnlp-main.268. URL `http://dx.doi.org/10.18653/v1/2024.emnlp-main.268`.

Zhen Xiang, Linzhi Zheng, Yanjie Li, Junyuan Hong, Qinbin Li, Han Xie, Jiawei Zhang, Zidi Xiong, Chulin Xie, Carl Yang, Dawn Song, and Bo Li. Guardagent: Safeguard llm agents by a guard agent via knowledge-enabled reasoning. *arXiv preprint arXiv:2406.09187*, 2024b. URL `https://www.arxiv.org/abs/2406.09187`.

Ruixuan Xiao, Wentao Ma, Ke Wang, Yuchuan Wu, Junbo Zhao, Haobo Wang, Fei Huang, and Yongbin Li. Flowbench: Revisiting and benchmarking workflow-guided planning for llm-based agents. *arXiv preprint arXiv:2406.14884*, 2024a.

Wei Xiao, Tsun-Hsuan Wang, Chuang Gan, Ramin Hasani, Mathias Lechner, and Daniela Rus. Safediffuser: Safe planning with diffusion probabilistic models. In *The thirteenth international conference on learning representations*, 2023.

Yihang Xiao, Jinyi Liu, Yan Zheng, Xiaohan Xie, Jianye Hao, Mingzhi Li, Ruitao Wang, Fei Ni, Yuxiao Li, Jintian Luo, et al. Cellagent: An llm-driven multi-agent framework for automated single-cell data analysis. *arXiv preprint arXiv:2407.09811*, 2024b.

Amber Xie, Oleh Rybkin, Dorsa Sadigh, and Chelsea Finn. Latent diffusion planning for imitation learning. *arXiv preprint arXiv:2504.16925*, 2025a.

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. Travelplanner: A benchmark for real-world planning with language agents. *arXiv preprint arXiv:2402.01622*, 2024a.

Quanting Xie, So Yeon Min, Pengliang Ji, Yue Yang, Tianyi Zhang, Kedi Xu, Aarav Bajaj, Ruslan Salakhutdinov, Matthew Johnson-Roberson, and Yonatan Bisk. Embodied-rag: General non-parametric embodied memory for retrieval and generation. *arXiv preprint arXiv:2409.18313*, 2024b. URL `https://www.arxiv.org/abs/2409.18313`.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *arXiv preprint arXiv:2404.07972v2*, 2024c. URL `https://www.arxiv.org/abs/2404.07972v2`.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments, 2024d. URL https://arxiv.org/abs/2404.07972.

Yuquan Xie, Zaijing Li, Rui Shao, Gongwei Chen, Kaiwen Zhou, Yinchuan Li, Dongmei Jiang, and Liqiang Nie. Mirage-1: Augmenting and updating gui agent with hierarchical multimodal skills. *arXiv preprint arXiv:2506.10387*, 2025b.

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, James Xu Zhao, Min-Yen Kan, Junxian He, and Michael Xie. Self-evaluation guided beam search for reasoning. *Advances in Neural Information Processing Systems*, 36: 41618–41650, 2023.

Mingzhe Xing, Rongkai Zhang, Hui Xue, Qi Chen, Fan Yang, and Zhen Xiao. Understanding the weakness of large language model agents within a complex android environment. *arXiv preprint arXiv:2402.06596v1*, 2024. URL https://www.arxiv.org/abs/2402.06596v1.

Lei Xiong, Kun Luo, Ziyi Xia, Wenbo Zhang, Jin-Ge Yao, Zheng Liu, Jingying Shao, Jianlyu Chen, Hongjin Qian, Xi Yang, Qian Yu, Hao Li, Chen Yue, Xiaan Du, Yuyang Wang, Yesheng Liu, Haiyu Xu, and Zhicheng Dou. Autoresearchbench: Benchmarking ai agents on complex scientific literature discovery, 2026a. URL https://arxiv.org/abs/2604.25256.

Xubang Xiong, Raymond Chi-Wing Wong, and Yuanfeng Song. Monte carlo tree search with reasoning path refinement for small language models in conversational text-to-nosql. *arXiv preprint arXiv:2602.12574*, 2026b.

Binfeng Xu, Zhiyuan Peng, Bowen Lei, Subhabrata Mukherjee, Yuchen Liu, and Dongkuan Xu. REWOO: Decoupling reasoning from observations for efficient augmented language models. *arXiv preprint arXiv:2305.18323*, 2023a.

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

Haoran Xu, Jiacong Hu, Ke Zhang, Lei Yu, Yuxin Tang, Xinyuan Song, Yiqun Duan, Lynn Ai, and Bill Shi. Sedm: Scalable self-evolving distributed memory for agents. *arXiv preprint arXiv:2509.09498*, 2025b.

Haoyuan Xu, Chang Li, Xinyan Ma, Xianhao Ou, Zihan Zhang, Tao He, Xiangyu Liu, Zixiang Wang, Jiafeng Liang, Zheng Chu, Runxuan Liu, Rongchuan Mu, Dandan Tu, Ming Liu, and Bing Qin. The evolution of tool use in llm agents: From single-tool call to multi-tool orchestration, 2026a. URL https://arxiv.org/abs/2603.22862.

Huan Xu, Jinlin Wu, Guanglin Cao, Zhen Lei, Zhen Chen, and Hongbin Liu. Enhancing surgical robots with embodied intelligence for autonomous ultrasound scanning. *arXiv preprint arXiv:2405.00461*, 2024.

Kai Xu, YiWei Mao, XinYi Guan, and ZiLong Feng. Web-bench: A llm code benchmark based on web standards and frameworks, 2025c. URL https://arxiv.org/abs/2505.07473.

Lin Xu, Zhiyuan Hu, Daquan Zhou, Hongyu Ren, Zhen Dong, Kurt Keutzer, See Kiong Ng, and Jiashi Feng. Magic: Investigation of large language model powered multi-agent in cognition, adaptability, rationality and collaboration. *arXiv preprint arXiv:2311.08562*, 2023b.

Ningning Xu, Yuxuan Jiang, Shubhashis Roy Dipta, and Hengyuan Zhang. Learning how to use tools, not just when: Pattern-aware tool-integrated reasoning. *arXiv preprint arXiv:2509.23292*, 2025d.

Qiantong Xu, Fenglu Hong, Bo Li, Changran Hu, Zhengyu Chen, and Jian Zhang. On the tool manipulation capability of open-source large language models. *CoRR*, abs/2305.16504, 2023c. doi: 10.48550/ARXIV.2305.16504. URL https://doi.org/10.48550/arXiv.2305.16504.

Qinghua Xu, Guancheng Wang, Lionel Briand, and Kui Liu. Hallucination to consensus: Multi-agent llms for end-to-end junit test generation. *ACM Transactions on Software Engineering and Methodology*, March 2026b. ISSN 1557-7392. doi: 10.1145/3803418. URL http://dx.doi.org/10.1145/3803418.

Ran Xu, Kaixin Ma, Wenhao Yu, Hongming Zhang, Joyce C Ho, Carl Yang, and Dong Yu. Retrieval-augmented gui agents with generative guidelines. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 17877–17886, 2025e.

Ran Xu, Wenqi Shi, Yuchen Zhuang, Yue Yu, Joyce C Ho, Haoyu Wang, and Carl Yang. Collab-rag: Boosting retrieval-augmented generation for complex question answering via white-box and black-box llm collaboration. *arXiv preprint arXiv:2504.04915*, 2025f.

Ran Xu, Yuchen Zhuang, Yishan Zhong, Yue Yu, Zifeng Wang, Xiangru Tang, Hang Wu, May D. Wang, Peifeng Ruan, Donghan Yang, Tao Wang, Guanghua Xiao, Xin Liu, Carl Yang, Yang Xie, and Wenqi Shi. Medagentgym: A scalable agentic training environment for code-centric reasoning in biomedical data science, 2025g. URL https://arxiv.org/abs/2506.04405.

Wujiang Xu, Kai Mei, Hang Gao, Juntao Tan, Zujie Liang, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110*, 2025h.

Yilong Xu, Xiang Long, Zhi Zheng, and Jinhua Gao. Ravine: Reality-aligned evaluation for agentic search. *arXiv preprint arXiv:2507.16725*, 2025i.

Haochen Xue, Feilong Tang, Ming Hu, Yexin Liu, Qidong Huang, Yulong Li, Chengzhi Liu, Zhongxing Xu, Chong Zhang, Chun-Mei Feng, et al. Mmrc: A large-scale benchmark for understanding multimodal large language model in real-world conversation. *arXiv preprint arXiv:2502.11903*, 2025a.

Xiangyuan Xue, Yifan Zhou, Guibin Zhang, Zaibin Zhang, Yijiang Li, Chen Zhang, Zhenfei Yin, Philip Torr, Wanli Ouyang, and Lei Bai. Comas: Co-evolving multi-agent systems via interaction rewards, 2025b. URL https://arxiv.org/abs/2510.08529.

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

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Kristian Kersting, Jeff Z. Pan, Hinrich Schütze, Volker Tresp, and Yunpu Ma. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. *arXiv preprint arXiv:2508.19828*, 2025.

Bo Yang, Jiaxian Guo, Yusuke Iwasawa, and Yutaka Matsuo. Large language models as theory of mind aware generative agents with counterfactual reflection. *arXiv preprint arXiv:2501.15355*, 2025a.

Chenyu Yang, Shiqian Su, Shi Liu, Xuan Dong, Yue Yu, Weijie Su, Xuehui Wang, Zhaoyang Liu, Jinguo Zhu, Hao Li, Wenhai Wang, Yu Qiao, Xizhou Zhu, and Jifeng Dai. Zerogui: Automating online gui learning at zero human cost. *arXiv preprint arXiv:2505.23762v1*, 2025b. URL https://www.arxiv.org/abs/2505.23762v1.

Hanqing Yang, Jingdi Chen, Marie Siew, Tania Lorido-Botran, and Carlee Joe-Wong. Llm-powered decentralized generative agents with adaptive hierarchical knowledge graph for cooperative planning. *arXiv preprint arXiv:2502.05453*, 2025c.

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

Jingkang Yang, Yuhao Dong, Shuai Liu, Bo Li, Ziyue Wang, Haoran Tan, Chencheng Jiang, Jiamu Kang, Yuanhan Zhang, Kaiyang Zhou, et al. Octopus: Embodied vision-language programmer from environmental feedback. In *European conference on computer vision*, pp. 20–38. Springer, 2024b.

John Yang, Kilian Lieret, Jeffrey Ma, Parth Thakkar, Dmitrii Pedchenko, Sten Sootla, Emily McMilin, Pengcheng Yin, Rui Hou, Gabriel Synnaeve, Diyi Yang, and Ofir Press. Programbench: Can language models rebuild programs from scratch?, 2026. URL https://arxiv.org/abs/2605.03546.

Kaiyu Yang, Gabriel Poesia, Jingxuan He, Wenda Li, Kristin Lauter, Swarat Chaudhuri, and Dawn Song. Formal mathematical reasoning: A new frontier in AI. *arXiv preprint arXiv:2412.16075*, 2024c.

Ke Yang, Yao Liu, Sapana Chaudhary, Rasool Fakoor, Pratik Chaudhari, George Karypis, and Huzefa Rangwala. Agentoccam: A simple yet strong baseline for llm-based web agents. *arXiv preprint arXiv:2410.13825*, 2024d.

Ling Yang, Zhaochen Yu, Tianjun Zhang, Shiyi Cao, Minkai Xu, Wentao Zhang, Joseph E Gonzalez, and Bin Cui. Buffer of thoughts: Thought-augmented reasoning with large language models. *Advances in Neural Information Processing Systems*, 37:113519–113544, 2024e.

Ruihan Yang, Jiangjie Chen, Yikai Zhang, Siyu Yuan, Aili Chen, Kyle Richardson, Yanghua Xiao, and Deqing Yang. Selfgoal: Your language agents already know how to achieve high-level goals. 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)*, pp. 799–819, 2025d.

Xianliang Yang, Zhihao Liu, Wei Jiang, Chuheng Zhang, Li Zhao, Lei Song, and Jiang Bian. A versatile multi-agent reinforcement learning benchmark for inventory management. *arXiv preprint arXiv:2306.07542*, 2023.

Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze D Gui, Ziran W Jiang, Ziyu Jiang, et al. Crag-comprehensive rag benchmark. *Advances in Neural Information Processing Systems*, 37:10470–10490, 2024f.

Yingxuan Yang, Mulei Ma, Yuxuan Huang, Huacan Chai, Chenyu Gong, Haoran Geng, Yuanjian Zhou, Ying Wen, Meng Fang, Muhao Chen, et al. Agentic web: Weaving the next web with ai agents. *arXiv preprint arXiv:2507.21206*, 2025e.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents, 2023a. URL https://arxiv.org/abs/2207.01206.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023b.

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

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains, 2024. URL https://arxiv.org/abs/2406.12045.

Jiabo Ye, Xi Zhang, Haiyang Xu, Haowei Liu, Junyang Wang, Zhaoqing Zhu, Ziwei Zheng, Feiyu Gao, Junjie Cao, Zhengxi Lu, Jitong Liao, Qi Zheng, Fei Huang, Jingren Zhou, and Ming Yan. Mobile-agent-v3: Fundamental agents for gui automation. *arXiv preprint arXiv:2508.15144*, 2025a. URL https://www.arxiv.org/abs/2508.15144.

Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Beyond autoregression: Discrete diffusion for complex reasoning and planning. *arXiv preprint arXiv:2410.14157*, 2024.

Rui Ye, Shuo Tang, Rui Ge, Yaxin Du, Zhenfei Yin, Siheng Chen, and Jing Shao. Mas-gpt: Training llms to build llm-based multi-agent systems. *arXiv preprint arXiv:2503.03686*, 2025b.

Yining Ye, Xin Cong, Shizuo Tian, Yujia Qin, Chong Liu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Rational decision-making agent with internalized utility judgment. *arXiv preprint arXiv:2308.12519*, 2023.

Da Yin, Faeze Brahman, Abhilasha Ravichander, Khyathi Chandu, Kai-Wei Chang, Yejin Choi, and Bill Yuchen Lin. Agent lumos: Unified and modular training for open-source language agents. *arXiv preprint arXiv:2311.05657*, 2023.

Minjong Yoo, Jinwoo Jang, Wei-Jin Park, and Honguk Woo. Exploratory retrieval-augmented planning for continual embodied instruction following. *Advances in Neural Information Processing Systems*, 37: 67034–67060, 2024.

Jaesik Yoon, Hyeonseo Cho, Doojin Baek, Yoshua Bengio, and Sungjin Ahn. Monte carlo tree diffusion for system 2 planning. *arXiv preprint arXiv:2502.07202*, 2025.

Ori Yoran, Samuel Joseph Amouyal, Chaitanya Malaviya, Ben Bogin, Ofir Press, and Jonathan Berant. Assistantbench: Can web agents solve realistic and time-consuming tasks? *arXiv preprint arXiv:2407.15711*, 2024.

Bangguo Yu, Hamidreza Kasaei, and Ming Cao. L3mvn: Leveraging large language models for visual target navigation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3554–3560. IEEE, 2023a.

Botao Yu, Frazier N Baker, Ziru Chen, Garrett Herb, Boyu Gou, Daniel Adu-Ampratwum, Xia Ning, and Huan Sun. Chemtoolagent: The impact of tools on language agents for chemistry problem solving. *arXiv preprint arXiv:2411.07228*, 2024a.

Hong Qing Yu and Frank McQuade. Rag-kg-il: A multi-agent hybrid framework for reducing hallucinations and enhancing llm reasoning through rag and incremental knowledge graph learning integration, 2025. URL `https://arxiv.org/abs/2503.13514`.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, et al. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. *arXiv preprint arXiv:2507.02259*, 2025a.

Huizi Yu, Jiayan Zhou, Lingyao Li, Shan Chen, Jack Gallifant, Anye Shi, Jie Sun, Xiang Li, Jingxian He, Wenyue Hua, et al. Simulated patient systems powered by large language model-based ai agents offer potential for transforming medical education. *Communications Medicine*, 2025b.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025c.

Tao Yu, Zhengbo Zhang, Zhiheng Lyu, Junhao Gong, Hongzhu Yi, Xinming Wang, Yuxuan Zhou, Jiabing Yang, Ping Nie, Yan Huang, and Wenhu Chen. Browseragent: Building web agents with human-inspired web browsing actions, 2025d. URL `https://arxiv.org/abs/2510.10666`.

Xianhao Yu, Jiaqi Fu, Renjia Deng, and Wenjuan Han. Mineland: Simulating large-scale multi-agent interactions with limited multimodal senses and physical needs. *arXiv preprint arXiv:2403.19267*, 2024b.

Xiao Yu, Maximillian Chen, and Zhou Yu. Prompt-based monte-carlo tree search for goal-oriented dialogue policy planning. *arXiv preprint arXiv:2305.13660*, 2023b.

Yangyang Yu, Haohang Li, Zhi Chen, Yuechen Jiang, Yang Li, Denghui Zhang, Rong Liu, Jordan W. Suchow, and Khaldoun Khashanah. Finmem: A performance-enhanced llm trading agent with layered memory and character design. *arXiv preprint arXiv:2311.13743*, 2023c. URL `https://www.arxiv.org/abs/2311.13743`.

Yangyang Yu, Zhiyuan Yao, Haohang Li, Zhiyang Deng, Yuechen Jiang, Yupeng Cao, Zhi Chen, Jordan W. Suchow, Zhenyu Cui, Rong Liu, Zhaozhuo Xu, Denghui Zhang, Koduvayur Subbalakshmi, Guojun Xiong, Yueru He, Jimin Huang, Dong Li, and Qianqian Xie. Fincon: A synthesized LLM multi-agent system with conceptual verbal reinforcement for enhanced financial decision making. In Amir Globersons, Lester Mackey,

Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024c. URL http://papers.nips.cc/paper_files/paper/2024/hash/f7ae4fe91d96f50abc2211f09b6a7e49-Abstract-Conference.html.

Yi Yu, Liuyi Yao, Yuexiang Xie, Qingquan Tan, Jiaqi Feng, Yaliang Li, and Libing Wu. Agentic memory: Learning unified long-term and short-term memory management for large language model agents. *arXiv preprint arXiv:2601.01885*, 2026.

Zhaochen Yu, Ling Yang, Jiaru Zou, Shuicheng Yan, and Mengdi Wang. Demystifying reinforcement learning in agentic reasoning. *arXiv preprint arXiv:2510.11701*, 2025e.

Jiakang Yuan, Xiangchao Yan, Shiyang Feng, Bo Zhang, Tao Chen, Botian Shi, Wanli Ouyang, Yu Qiao, Lei Bai, and Bowen Zhou. Dolphin: Moving towards closed-loop auto-research through thinking, practice, and feedback. *arXiv preprint arXiv:2501.03916*, 2025a.

Lifan Yuan, Yangyi Chen, Xingyao Wang, Yi Fung, Hao Peng, and Heng Ji. CRAFT: Customizing LLMs by creating and retrieving from specialized toolsets. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=G0vdDSt9XM.

Siyu Yuan, Kaitao Song, Jiangjie Chen, Xu Tan, Yongliang Shen, Ren Kan, Dongsheng Li, and Deqing Yang. Easytool: Enhancing llm-based agents with concise tool instruction. *arXiv preprint arXiv:2401.06201*, 2024b.

Tongxin Yuan, Zhiwei He, Lingzhong Dong, Yiming Wang, Ruijie Zhao, Tian Xia, Lizhen Xu, Binglin Zhou, Fangqi Li, Zhuosheng Zhang, Rui Wang, and Gongshen Liu. R-judge: Benchmarking safety risk awareness for llm agents. *arXiv preprint arXiv:2401.10019*, 2024c. URL https://www.arxiv.org/abs/2401.10019.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. Self-rewarding language models. In *Forty-first International Conference on Machine Learning*, 2024d.

Xinbin Yuan, Jian Zhang, Kaixin Li, Zhuoxuan Cai, Lujian Yao, Jie Chen, Enguang Wang, Qibin Hou, Jinwei Chen, Peng-Tao Jiang, and Bo Li. Enhancing visual grounding for gui agents via self-evolutionary reinforcement learning. *arXiv preprint arXiv:2505.12370*, 2025b. URL https://www.arxiv.org/abs/2505.12370.

Junpeng Yue, Xinrun Xu, Börje F. Karlsson, and Zongqing Lu. Mllm as retriever: Interactively learning multimodal retrieval for embodied agents. *arXiv preprint arXiv:2410.03450*, 2024a. URL https://www.arxiv.org/abs/2410.03450.

Ling Yue, Sixue Xing, Jintai Chen, and Tianfan Fu. 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*, pp. 1–10, 2024b.

Yanwei Yue, Guibin Zhang, Boyang Liu, Guancheng Wan, Kun Wang, Dawei Cheng, and Yiyan Qi. Masrouter: Learning to route llms for multi-agent systems. *arXiv preprint arXiv:2502.11133*, 2025.

Sizhe Yuen, Francisco Gomez Medina, Ting Su, Yali Du, and Adam J. Sobey. Intrinsic memory agents: Heterogeneous multi-agent llm systems through structured contextual memory. *arXiv preprint arXiv:2508.08997*, 2025.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. Textgrad: Automatic" differentiation" via text. *arXiv preprint arXiv:2406.07496*, 2024.

Karina Zainullina, Alexander Golubev, Maria Trofimova, Sergei Polezhaev, Ibragim Badertdinov, Daria Litvintseva, Simon Karasik, Filipp Fisin, Sergei Skvortsov, Maksim Nekrashevich, Anton Shevtsov, and Boris Yangel. Guided search strategies in non-serializable environments with applications to software engineering agents. In *International Conference on Machine Learning (ICML)*, 2025.

Daoguang Zan, Zhirong Huang, Wei Liu, Hanwu Chen, Linhao Zhang, Shulin Xin, Lu Chen, Qi Liu, Xiaojian Zhong, Aoyan Li, Siyao Liu, Yongsheng Xiao, Liangqiang Chen, Yuyu Zhang, Jing Su, Tianyu Liu, Rui Long, Kai Shen, and Liang Xiang. Multi-swe-bench: A multilingual benchmark for issue resolving, 2025. URL `https://arxiv.org/abs/2504.02605`.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*, 2024.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D. Goodman. Quiet-STaR: Language models can teach themselves to think before speaking, 2024. URL `https://arxiv.org/abs/2403.09629`.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. Agenttuning: Enabling generalized agent abilities for llms. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3053–3077, 2024.

Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, et al. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*, 2025a.

Xianghua Zeng, Hang Su, Zhengyi Wang, and Zhiyuan Lin. Graph diffusion for robust multi-agent coordination. In *Forty-second International Conference on Machine Learning*.

Yirong Zeng, Xiao Ding, Yuxian Wang, Weiwen Liu, Yutai Hou, Wu Ning, Xu Huang, Duyu Tang, Dandan Tu, Bing Qin, et al. itool: Reinforced fine-tuning with dynamic deficiency calibration for advanced tool use. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 13901–13916, 2025b.

Zhen Zeng, William Watson, Nicole Cho, Saba Rahimi, Shayleen Reynolds, Tucker Balch, and Manuela Veloso. Flowmind: automatic workflow generation with llms. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, pp. 73–81, 2023.

Zhixiong Zeng, Jing Huang, Liming Zheng, Wenkang Han, Yufeng Zhong, Lei Chen, Longrong Yang, Yingjie Chu, Yuzhi He, and Lin Ma. Uitron: Foundational gui agent with advanced perception and planning. *arXiv preprint arXiv:2508.21767*, 2025c.

Bofei Zhang, Zirui Shang, Zhi Gao, Wang Zhang, Rui Xie, Xiaojian Ma, Tao Yuan, Xinxiao Wu, Song-Chun Zhu, and Qing Li. Tongui: Internet-scale trajectories from multimodal web tutorials for generalized gui agents, 2025a. URL `https://arxiv.org/abs/2504.12679`.

Danqing Zhang, Balaji Rama, Jingyi Ni, Shiying He, Fu Zhao, Kunyu Chen, Arnold Chen, and Junyu Cao. Litewebagent: The open-source suite for vlm-based web-agent applications. *arXiv preprint arXiv:2503.02950*, 2025b.

Ding-Chu Zhang, Yida Zhao, Jialong Wu, Liwen Zhang, Baixuan Li, Wenbiao Yin, Yong Jiang, Yu-Feng Li, Kewei Tu, Pengjun Xie, et al. Evolvesearch: An iterative self-evolving search agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 13134–13147, 2025c.

Dongxu Zhang, Ning Yang, Jihua Zhu, Jinnan Yang, Miao Xin, and Baoliang Tian. Ascot: An adaptive self-correction chain-of-thought method for late-stage fragility in llms. *arXiv preprint arXiv:2508.05282*, 2025d.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. Cut the crap: An economical communication pipeline for llm-based multi-agent systems. *arXiv preprint arXiv:2410.02506*, 2024a.

Guibin Zhang, Yanwei Yue, Xiangguo Sun, Guancheng Wan, Miao Yu, Junfeng Fang, Kun Wang, Tianlong Chen, and Dawei Cheng. G-designer: Architecting multi-agent communication topologies via graph neural networks. *arXiv preprint arXiv:2410.11782*, 2024b.

Guibin Zhang, Muxin Fu, Guancheng Wan, Miao Yu, Kun Wang, and Shuicheng Yan. G-memory: Tracing hierarchical memory for multi-agent systems. *arXiv preprint arXiv:2506.07398*, 2025e.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, et al. The landscape of agentic reinforcement learning for llms: A survey. *arXiv preprint arXiv:2509.02547*, 2025f.

Guibin Zhang, Luyang Niu, Junfeng Fang, Kun Wang, Lei Bai, and Xiang Wang. Multi-agent architecture search via agentic supernet. *arXiv preprint arXiv:2502.04180*, 2025g.

Han Zhang, Langshi Zhou, and Hanfang Yang. Learning to retrieve and reason on knowledge graph through active self-reflection. *arXiv preprint arXiv:2502.14932*, 2025h.

Hongxin Zhang, Zeyuan Wang, Qiushi Lyu, Zheyuan Zhang, Sunli Chen, Tianmin Shu, Behzad Dariush, Kwonjoon Lee, Yilun Du, and Chuang Gan. Combo: compositional world models for embodied multi-agent cooperation. *arXiv preprint arXiv:2404.10775*, 2024c.

Hongxin Zhang, Zheyuan Zhang, Zeyuan Wang, Zunzhe Zhang, Lixing Fang, Qinhong Zhou, and Chuang Gan. Ella: Embodied social agents with lifelong memory. *arXiv preprint arXiv:2506.24019*, 2025i.

Huan Zhang, Wei Cheng, Yuhan Wu, and Wei Hu. A pair programming framework for code generation via multi-plan exploration and feedback-driven refinement. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, pp. 1319–1331, 2024d.

Huan Zhang, Yu Song, Ziyu Hou, Santiago Miret, and Bang Liu. Honeycomb: A flexible llm-based agent system for materials science. *arXiv preprint arXiv:2409.00135*, 2024e.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, et al. Aflow: Automating agentic workflow generation. *arXiv preprint arXiv:2410.10762*, 2024f.

Jiayi Zhang, Chuang Zhao, Yihan Zhao, Zhaoyang Yu, Ming He, and Jianping Fan. Mobileexperts: A dynamic tool-enabled agent team in mobile devices. *arXiv preprint arXiv:2407.03913*, 2024g.

Jingyu Zhang, Haozhu Wang, Eric Michael Smith, Sid Wang, Amr Sharaf, Mahesh Pasupuleti, Benjamin Van Durme, Daniel Khashabi, Jason Weston, and Hongyuan Zhan. The alignment waltz: Jointly training agents to collaborate for safety, 2025j. URL `https://arxiv.org/abs/2510.08240`.

Kai Zhang, Xiangchao Chen, Bo Liu, Tianci Xue, Zeyi Liao, Zhihan Liu, Xiyao Wang, Yuting Ning, Zhaorun Chen, Xiaohan Fu, et al. Agent learning via early experience. *arXiv preprint arXiv:2510.08558*, 2025k.

Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, et al. A survey of reinforcement learning for large reasoning models. *arXiv preprint arXiv:2509.08827*, 2025l.

Kechi Zhang, Zhuo Li, Jia Li, Ge Li, and Zhi Jin. Self-edit: Fault-aware code editor for code generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 769–787, 2023a.

Kechi Zhang, Huangzhao Zhang, Ge Li, Jia Li, Zhuo Li, and Zhi Jin. ToolCoder: Teach code generation models to use API search tools, 2023b.

Kechi Zhang, Jia Li, Ge Li, Xianjie Shi, and Zhi Jin. 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)*, pp. 13643–13658, 2024h.

Lisheng Zhang, Lilong Wang, Xiangyu Sun, Wei Tang, Haoyang Su, Yuehui Qian, Qikui Yang, Qingsong Li, Zhenyu Tang, Haoran Sun, Yingnan Han, Yankai Jiang, Wenjie Lou, Bowen Zhou, Xiaosong Wang, Lei Bai, and Zhengwei Xie. Molclaw: An autonomous agent with hierarchical skills for drug molecule evaluation, screening, and optimization, 2026a. URL https://arxiv.org/abs/2604.21937.

Natalia Zhang, Xinqi Wang, Qiwen Cui, Runlong Zhou, Sham M. Kakade, and Simon S. Du. Preference-based multi-agent reinforcement learning: Data coverage and algorithmic techniques, 2025m. URL https://arxiv.org/abs/2409.00717.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, et al. Agentic context engineering: Evolving contexts for self-improving language models. *arXiv preprint arXiv:2510.04618*, 2025n.

Shengtao Zhang, Jiaqian Wang, Ruiwen Zhou, Junwei Liao, Yuchen Feng, Zhuo Li, Yujie Zheng, Weinan Zhang, Ying Wen, Zhiyu Li, et al. Memrl: Self-evolving agents via runtime reinforcement learning on episodic memory. *arXiv preprint arXiv:2601.03192*, 2026b.

Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. Raft: Adapting language model to domain specific rag. *arXiv preprint arXiv:2403.10131*, 2024i.

Weizhi Zhang, Yangning Li, Yuanchen Bei, Junyu Luo, Guancheng Wan, Liangwei Yang, Chenxuan Xie, Yuyao Yang, Wei-Chieh Huang, Chunyu Miao, Henry Peng Zou, Xiao Luo, Yusheng Zhao, Yankai Chen, Chunkit Chan, Peilin Zhou, Xinyang Zhang, Chenwei Zhang, Jingbo Shang, Ming Zhang, Yangqiu Song, Irwin King, and Philip S. Yu. From web search towards agentic deep research: Incentivizing search with reasoning agents. *arXiv preprint arXiv:2506.18959v3*, 2025o. URL https://www.arxiv.org/abs/2506.18959v3.

Wentao Zhang, Liang Zeng, Yuzhen Xiao, Yongcong Li, Ce Cui, Yilei Zhao, Rui Hu, Yang Liu, Yahui Zhou, and Bo An. Agentorchestra: Orchestrating multi-agent intelligence with the tool-environment-agent(tea) protocol, 2026c. URL https://arxiv.org/abs/2506.12508.

Xuan Zhang, Shengbo Cai, Ziyan Jiang, Rui Meng, Zora Zhiruo Wang, Lu Guangcheng, Zhiyong Wu, Yanyi Shang, and Dehan Kong. WebRAGent: Retrieval-augmented generation for multimodal web agent planning, 2026d. URL https://openreview.net/forum?id=L1VPZFbAcu.

Xueqiao Zhang, Chao Zhang, Jianwen Sun, Jun Xiao, Yi Yang, and Yawei Luo. Eduplanner: Llm-based multi-agent systems for customized and intelligent instructional design. *IEEE Transactions on Learning Technologies*, 2025p.

Yang Zhang, Shixin Yang, Chenjia Bai, Fei Wu, Xiu Li, Zhen Wang, and Xuelong Li. Towards efficient llm grounding for embodied multi-agent collaboration. *arXiv preprint arXiv:2405.14314*, 2024j.

Yao Zhang, Zijian Ma, Yunpu Ma, Zhen Han, Yu Wu, and Volker Tresp. Webpilot: A versatile and autonomous multi-agent system for web task execution with strategic exploration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23378–23386, 2025q.

Yaolun Zhang, Xiaogeng Liu, and Chaowei Xiao. Metaagent: Automatically constructing multi-agent systems based on finite state machines. *arXiv preprint arXiv:2507.22606*, 2025r.

Yilin Zhang, Xinran Zhao, Zora Zhiruo Wang, Chenyang Yang, Jiayi Wei, and Tongshuang Wu. cAST: Enhancing code retrieval-augmented generation with structural chunking via abstract syntax tree. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 8106–8116, Suzhou, China, November 2025s. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.430. URL https://aclanthology.org/2025.findings-emnlp.430/.

Yiran Zhang, Mo Wang, Xiaoyang Li, Kaixuan Ren, Chencheng Zhu, and Usman Naseem. Turnbench-ms: A benchmark for evaluating multi-turn, multi-step reasoning in large language models. *arXiv preprint arXiv:2506.01341*, 2025t.

Yuji Zhang, Sha Li, Jiateng Liu, Pengfei Yu, Yi R Fung, Jing Li, Manling Li, and Heng Ji. Knowledge overshadowing causes amalgamated hallucination in large language models. *arXiv preprint arXiv:2407.08039*, 2024k.

Yuji Zhang, Qingyun Wang, Cheng Qian, Jiateng Liu, Chenkai Sun, Denghui Zhang, Tarek Abdelzaher, Chengxiang Zhai, Preslav Nakov, and Heng Ji. Atomic reasoning for scientific table claim verification. *arXiv preprint arXiv:2506.06972*, 2025u.

Yuqiang Zhang, Jiangming Shu, Ye Ma, Xueyuan Lin, Shangxi Wu, and Jitao Sang. Memory as action: Autonomous context curation for long-horizon agentic tasks. *arXiv preprint arXiv:2510.12635*, 2025v. URL https://arxiv.org/abs/2510.12635.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Arik. Chain of agents: Large language models collaborating on long-context tasks. *Advances in Neural Information Processing Systems*, 37:132208–132237, 2024l.

Zeyu Zhang, Quanyu Dai, Luyu Chen, Zeren Jiang, Rui Li, Jieming Zhu, Xu Chen, Yi Xie, Zhenhua Dong, and Ji-Rong Wen. Memsim: A bayesian simulator for evaluating memory of llm-based personal assistants. *arXiv preprint arXiv:2409.20163*, 2024m.

Zheyuan Zhang, Kaiwen Shi, Zhengqing Yuan, Zehong Wang, Tianyi Ma, Keerthiram Murugesan, Vincent Galassi, Chuxu Zhang, and Yanfang Ye. Agentrouter: A knowledge-graph-guided llm router for collaborative multi-agent question answering. *arXiv preprint arXiv:2510.05445*, 2025w.

Zhong Zhang, Yaxi Lu, Yikun Fu, Yupeng Huo, Shenzhi Yang, Yesai Wu, Han Si, Xin Cong, Haotian Chen, Yankai Lin, Jie Xie, Wei Zhou, Wang Xu, Yuanheng Zhang, Zhou Su, Zhongwu Zhai, Xiaoming Liu, Yudong Mei, Jianming Xu, Hongyan Tian, Chongyi Wang, Chi Chen, Yuan Yao, Zhiyuan Liu, and Maosong Sun. Agentcpm-gui: Building mobile-use agents with reinforcement fine-tuning, 2025x. URL https://arxiv.org/abs/2506.01391.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023c.

Zijing Zhang, Zhanpeng Chen, He Zhu, Ziyang Chen, Nan Du, and Xiaolong Li. Toolexpnet: Optimizing multi-tool selection in llms with similarity and dependency-aware experience networks. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 15706–15722. Association for Computational Linguistics, 2025y. URL https://aclanthology.org/2025.findings-acl.811/.

Zijing Zhang, Ziyang Chen, Mingxiao Li, Zhaopeng Tu, and Xiaolong Li. Rlvmr: Reinforcement learning with verifiable meta-reasoning rewards for robust long-horizon agents. *arXiv preprint arXiv:2507.22844*, 2025z.

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

Baining Zhao, Ziyou Wang, Jianjie Fang, Chen Gao, Fanhang Man, Jinqiang Cui, Xin Wang, Xinlei Chen, Yong Li, and Wenwu Zhu. Embodied-r: Collaborative framework for activating embodied spatial reasoning in foundation models via reinforcement learning. *arXiv preprint arXiv:2504.12680*, 2025a.

Di Zhao, Longhui Ma, Siwei Wang, Miao Wang, and Zhao Lv. Cola: A scalable multi-agent framework for windows ui task automation. *arXiv preprint arXiv:2503.09263*, 2025b.

Kai Zhao, Yanjun Zhao, Jiaming Song, Shien He, Lusheng Zhang, Qiang Zhang, and Tianjiao Li. Saber: Switchable and balanced training for efficient llm reasoning, 2025c. URL https://arxiv.org/abs/2508.10026.

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

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 1702–1713, 2025d.

Shitian Zhao, Haoquan Zhang, Shaoheng Lin, Ming Li, Qilong Wu, Kaipeng Zhang, and Chen Wei. Pyvision: Agentic vision with dynamic tooling. *CoRR*, abs/2507.07998, 2025e. doi: 10.48550/ARXIV.2507.07998. URL https://doi.org/10.48550/arXiv.2507.07998.

Wanjia Zhao, Mert Yuksekgonul, Shirley Wu, and James Zou. Sirius: Self-improving multi-agent systems via bootstrapped reasoning. *arXiv preprint arXiv:2502.04780*, 2025f.

Xufeng Zhao, Mengdi Li, Cornelius Weber, Muhammad Burhan Hafez, and Stefan Wermter. Chat with the environment: Interactive multimodal perception using large language models. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3590–3596. IEEE, 2023a.

Yanjun Zhao, Tianxin Wei, Jiaru Zou, Xuying Ning, Yuanchen Bei, Lingjie Chen, Simmi Rana, Wendy H. Yang, Hanghang Tong, and Jingrui He. Papermind: Benchmarking agentic reasoning and critique over scientific papers in multimodal llms, 2026a. URL https://arxiv.org/abs/2604.21304.

Yu Zhao, Wei-Ning Chen, Huseyin Atahan Inan, Samuel Kessler, Lu Wang, Lukas Wutschitz, Fangkai Yang, Chaoyun Zhang, Pasquale Minervini, Saravan Rajmohan, and Robert Sim. Learning gui grounding with spatial reasoning from visual feedback, 2026b. URL https://arxiv.org/abs/2509.21552.

Yujie Zhao, Lanxiang Hu, Yang Wang, Minmin Hou, Hao Zhang, Ke Ding, and Jishen Zhao. Stronger-mas: Multi-agent reinforcement learning for collaborative llms, 2025g. URL https://arxiv.org/abs/2510.11062.

Zehua Zhao, Zhixian Huang, Junren Li, Siyu Lin, Junting Zhou, Fengqi Cao, Kun Zhou, Rui Ge, Tingting Long, Yuexiang Zhu, Yan Liu, Jie Zheng, Junnian Wei, Rong Zhu, Peng Zou, Wenyu Li, Zekai Cheng, Tian Ding, Yaxuan Wang, Yizhao Yan, Tingru Wei, Haowei Ming, Weijie Mao, Chen Sun, Yiming Liu, Zichen Wang, Zuo Zhang, Tong Yang, Hao Ma, Zhen Gao, and Jian Pei. Superchem: A multimodal reasoning benchmark in chemistry, 2025h. URL https://arxiv.org/abs/2512.01274.

Zheng Zhao, Clara Vania, Subhradeep Kayal, Naila Khan, Shay B. Cohen, and Emine Yilmaz. PersonaLens: A benchmark for personalization evaluation in conversational AI assistants, 2025i. URL https://arxiv.org/abs/2506.09902.

Ziliang Zhao, Zenan Xu, Shuting Wang, Hongjin Qian, Yan Lei, Minda Hu, Zhao Wang, Shihan Dou, Zhicheng Dou, and Pluto Zhou. Planningbench: Generating scalable and verifiable planning data for evaluating and training large language models, 2026c. URL https://arxiv.org/abs/2605.20873.

Zirui Zhao, Wee Sun Lee, and David Hsu. Large language models as commonsense knowledge for large-scale task planning. *Advances in neural information processing systems*, 36:31967–31987, 2023b.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. Gpt-4v (ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*, 2024a.

Boyuan Zheng, Michael Y Fatemi, Xiaolong Jin, Zora Zhiruo Wang, Apurva Gandhi, Yueqi Song, Yu Gu, Jayanth Srinivasa, Gaowen Liu, Graham Neubig, et al. Skillweaver: Web agents can self-improve by discovering and honing skills. *arXiv preprint arXiv:2504.07079*, 2025a.

Lianmin Zheng, Jiacheng Yang, Han Cai, Ming Zhou, Weinan Zhang, Jun Wang, and Yong Yu. Magent: A many-agent reinforcement learning platform for artificial collective intelligence. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Longtao Zheng, Rundong Wang, Xinrun Wang, and Bo An. Synapse: Trajectory-as-exemplar prompting with memory for computer control. *arXiv preprint arXiv:2306.07863*, 2023.

Yu Zheng, Longyi Liu, Yuming Lin, Jie Feng, Guozhen Zhang, Depeng Jin, and Yong Li. Urbanplanbench: A comprehensive urban planning benchmark for evaluating large language models, 2025b. URL https://arxiv.org/abs/2504.21027.

Yuanhang Zheng, Peng Li, Wei Liu, Yang Liu, Jian Luan, and Bin Wang. Toolrerank: Adaptive and hierarchy-aware reranking for tool retrieval. In Nicoletta Calzolari, Min-Yen Kan, Véronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pp. 16263–16273. ELRA and ICCL, 2024b. URL https://aclanthology.org/2024.lrec-main.1413.

Yupeng Zheng, Zebin Xing, Qichao Zhang, Bu Jin, Pengfei Li, Yuhang Zheng, Zhongpu Xia, Kun Zhan, Xianpeng Lang, Yaran Chen, et al. Planagent: A multi-modal large language agent for closed-loop vehicle motion planning. *arXiv preprint arXiv:2406.01587*, 2024c.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *arXiv preprint arXiv:2504.03160*, 2025c.

Zhi Zheng and Wee Sun Lee. Reasoning-cv: Fine-tuning powerful reasoning llms for knowledge-assisted claim verification. *arXiv preprint arXiv:2505.12348*, 2025.

Ruizhe Zhong, Xingbo Du, Shixiong Kai, Zhentao Tang, Siyuan Xu, Jianye Hao, Mingxuan Yuan, and Junchi Yan. Flexplanner: Flexible 3d floorplanning via deep reinforcement learning in hybrid action space with multi-modality representation. *Advances in Neural Information Processing Systems*, 37:49252–49278, 2024.

Wanjun Zhong, Lianghong Guo, Qi-Fei Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. *ArXiv*, abs/2305.10250, 2023. URL https://api.semanticscholar.org/CorpusID:258741194.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024a.

Han Zhou, Xingchen Wan, Ruoxi Sun, Hamid Palangi, Shariq Iqbal, Ivan Vulić, Anna Korhonen, and Sercan Ö Arık. Multi-agent design: Optimizing agents with better prompts and topologies. *arXiv preprint arXiv:2502.02533*, 2025a.

Jianing Zhou, Ziheng Zeng, and Suma Bhat. CLCL: Non-compositional expression detection with contrastive learning and curriculum learning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 730–743, Toronto, Canada, July 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.43. URL https://aclanthology.org/2023.acl-long.43/.

Ming Zhou, Jun Luo, Julian Villella, Yaodong Yang, David Rusu, Jiayu Miao, Weinan Zhang, Montgomery Alban, Iman Fadakar, Zheng Chen, et al. Smarts: Scalable multi-agent reinforcement learning training school for autonomous driving. *arXiv preprint arXiv:2010.09776*, 2020.

Peilin Zhou, Bruce Leon, Xiang Ying, Can Zhang, Yifan Shao, Qichen Ye, Dading Chong, Zhiling Jin, Chenxuan Xie, Meng Cao, Yuxin Gu, Sixin Hong, Jing Ren, Jian Chen, Chao Liu, and Yining Hua. Browsecomp-zh: Benchmarking web browsing ability of large language models in chinese. *arXiv preprint arXiv:2504.19314*, 2025b. URL https://www.arxiv.org/abs/2504.19314.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023b. URL https://www.arxiv.org/abs/2307.13854.

Xiaolin Zhou, Aojie Yuan, Zheng Luo, Zipeng Ling, Xixiao Pan, Yicheng Gao, Haiyue Zhang, Jiate Li, Shuli Jiang, Prince Zizhuang Wang, Zixuan Zhu, Jinbo Liu, Ryan A. Rossi, Hua Wei, and Xiyang Hu. When simulation lies: A sim-to-real benchmark and domain-randomized rl recipe for tool-use agents, 2026. URL https://arxiv.org/abs/2605.11928.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. Archer: Training language model agents via hierarchical multi-turn rl. *arXiv preprint arXiv:2402.19446*, 2024b.

Yifei Zhou, Song Jiang, Yuandong Tian, Jason Weston, Sergey Levine, Sainbayar Sukhbaatar, and Xian Li. Sweet-rl: Training multi-turn llm agents on collaborative reasoning tasks. *arXiv preprint arXiv:2503.15478*, 2025c.

Yifei Zhou, Sergey Levine, Jason Weston, Xian Li, and Sainbayar Sukhbaatar. Self-challenging language model agents. *arXiv preprint arXiv:2506.01716*, 2025d.

Yifei Zhou, Qianlan Yang, Kaixiang Lin, Min Bai, Xiong Zhou, Yu-Xiong Wang, Sergey Levine, and Li Erran Li. Proposer-agent-evaluator (pae): Autonomous skill discovery for foundation model internet agents. In *Forty-second International Conference on Machine Learning*, 2025e.

Yiyun Zhou, Chang Yao, and Jingyuan Chen. Cola: Collaborative low-rank adaptation, 2025f. URL https://arxiv.org/abs/2505.15471.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. MEM1: Learning to synergize memory and reasoning for efficient long-horizon agents. *arXiv preprint arXiv:2506.15841*, 2025g.

Zikang Zhou, Haibo Hu, Xinhong Chen, Jianping Wang, Nan Guan, Kui Wu, Yung-Hui Li, Yu-Kai Huang, and Chun J Xue. Behaviorgpt: Smart agent simulation for autonomous driving with next-patch prediction. *Advances in Neural Information Processing Systems*, 37:79597–79617, 2024c.

Bingwen Zhu, Yuqian Fu, Qiaole Dong, Guolei Sun, Tianwen Qian, Yuzheng Wu, Danda Pani Paudel, Xiangyang Xue, and Yanwei Fu. Egosound: Benchmarking sound understanding in egocentric videos, 2026a. URL https://arxiv.org/abs/2602.14122.

Guobin Zhu, Rui Zhou, Wenkang Ji, and Shiyu Zhao. Lamarl: Llm-aided multi-agent reinforcement learning for cooperative policy generation. *IEEE Robotics and Automation Letters*, 2025a.

Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Zhe Wang, Zhenhailong Wang, Cheng Qian, Xiangru Tang, Heng Ji, et al. Multiagentbench: Evaluating the collaboration and competition of llm agents. *arXiv preprint arXiv:2503.01935*, 2025b.

Yakun Zhu, Shaohang Wei, Xu Wang, Kui Xue, Xiaofan Zhang, and Shaoting Zhang. Menti: Bridging medical calculator and llm agent with nested tool calling. *arXiv preprint arXiv:2410.13610*, 2024a.

Yichen Zhu, Zhicai Ou, Xiaofeng Mou, and Jian Tang. Retrieval-augmented embodied agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17985–17995, 2024b.

Yinghao Zhu, Yifan Qi, Zixiang Wang, Lei Gu, Dehao Sui, Haoran Hu, Xichen Zhang, Ziyi He, Junjun He, Liantao Ma, et al. Healthflow: A self-evolving ai agent with meta planning for autonomous healthcare research. *arXiv preprint arXiv:2508.02621*, 2025c.

Yutao Zhu, Peitian Zhang, Chenghao Zhang, Yifei Chen, Binyu Xie, Zheng Liu, Ji-Rong Wen, and Zhicheng Dou. Inters: Unlocking the power of large language models in search with instruction tuning. *arXiv preprint arXiv:2401.06532*, 2024c.

Zichen Zhu, Hao Tang, Yansi Li, Dingye Liu, Hongshen Xu, Kunyao Lan, Danyang Zhang, Yixuan Jiang, Hao Zhou, Chenrun Wang, et al. Moba: multifaceted memory-enhanced adaptive planning for efficient mobile task automation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (System Demonstrations)*, pp. 535–549, 2025d.

Zilin Zhu, Longteng Guo, Yanghong Mei, Bowen Pang, Zongxun Zhang, Xingjian He, Ruyi Ji, and Jing Liu. When robots do the chores: A benchmark and agent for long-horizon household task execution, 2026b. URL https://arxiv.org/abs/2605.14504.

Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. Toolqa: A dataset for LLM question answering with external tools. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/9cb2a7495900f8b602cb10159246a016-Abstract-Datasets_and_Benchmarks.html.

Yuchen Zhuang, Xiang Chen, Tong Yu, Saayan Mitra, Victor Bursztyn, Ryan Rossi, Somdeb Sarkhel, and Chao Zhang. Toolchain*: Efficient action space navigation in large language models with a* search. In *International Conference on Learning Representations*, volume 2024, pp. 4524–4549, 2024.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. Gptswarm: Language agents as optimizable graphs. In *Forty-first International Conference on Machine Learning*, 2024a.

Mingchen Zhuge, Changsheng Zhao, Dylan Ashley, Wenyi Wang, Dmitrii Khizbullin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, Yangyang Shi, Vikas Chandra, and Jürgen Schmidhuber. Agent-as-a-judge: Evaluate agents with agents. *arXiv preprint arXiv:2410.10934*, 2024b. URL https://www.arxiv.org/abs/2410.10934.

Jiaru Zou, Yikun Ban, Zihao Li, Yunzhe Qi, Ruizhong Qiu, Ling Yang, and Jingrui He. Transformer copilot: Learning from the mistake log in llm fine-tuning, 2025a. URL https://arxiv.org/abs/2505.16270.

Jiaru Zou, Dongqi Fu, Sirui Chen, Xinrui He, Zihao Li, Yada Zhu, Jiawei Han, and Jingrui He. Rag over tables: Hierarchical memory index, multi-stage retrieval, and benchmarking, 2025b. URL https://arxiv.org/abs/2504.01346.

Jiaru Zou, Soumya Roy, Vinay Kumar Verma, Ziyi Wang, David Wipf, Pan Lu, Sumit Negi, James Zou, and Jingrui He. Tattoo: Tool-grounded thinking prm for test-time scaling in tabular reasoning. *arXiv preprint arXiv:2510.06217*, 2025c.

Jiaru Zou, Ling Yang, Yunzhe Qi, Sirui Chen, Mengting Ai, Ke Shen, Jingrui He, and Mengdi Wang. Autotool: Dynamic tool selection and integration for agentic reasoning, 2025d. URL https://arxiv.org/abs/2512.13278.

Jiaru Zou, Xiyuan Yang, Ruizhong Qiu, Gaotang Li, Katherine Tieu, Pan Lu, Ke Shen, Hanghang Tong, Yejin Choi, Jingrui He, James Zou, Mengdi Wang, and Ling Yang. Latent collaboration in multi-agent systems, 2025e. URL https://arxiv.org/abs/2511.20639.

Jiaru Zou, Ling Yang, Jingwen Gu, Jiahao Qiu, Ke Shen, Jingrui He, and Mengdi Wang. Reasonflux-prm: Trajectory-aware prms for long chain-of-thought reasoning in llms. *Advances in Neural Information Processing Systems*, 38:26764–26802, 2026.

Yunheng Zou, Austin H. Cheng, Abdulrahman Aldossary, Jiaru Bai, Shi Xuan Leong, Jorge A. Campos Gonzalez Angulo, Changhyeok Choi, Cher Tian Ser, Gary Tom, Andrew Wang, Zijian Zhang, Ilya Yakavets, Han Hao, Chris Crebolder, Varinia Bernales, and Alán Aspuru-Guzik. El agente: An autonomous agent for quantum chemistry. *CoRR*, abs/2505.02484, 2025f. doi: 10.48550/ARXIV.2505.02484. URL https://doi.org/10.48550/arXiv.2505.02484.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, et al. Ttrl: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*, 2025.

Adam Zweiger, Jyothish Pari, Han Guo, Ekin Akyürek, Yoon Kim, and Pulkit Agrawal. Self-adapting language models. *arXiv preprint arXiv:2506.10943*, 2025.