A Survey on Evaluation of LLM-based Agents

Asaf Yehudai[1,2], Lilach Eden[2], Alan Li[3], Guy Uziel[2],
Yilun Zhao[3], Roy Bar-Haim[2], Arman Cohan[3], Michal Shmueli-Scheuer[2]

[1]The Hebrew University of Jerusalem [2]IBM Research [3]Yale University
{Asaf.Yehudai, Guy.Uziel1}@ibm.com {lilache, roybar, shmueli}@il.ibm.com
{haoxin.li, yilun.zhao, arman.cohan}@yale.edu

Abstract

LLM-based agents represent a paradigm shift in AI, enabling autonomous systems to plan, reason, and use tools while interacting with dynamic environments. This paper provides the first comprehensive survey of evaluation methods for these increasingly capable agents. We analyze the field of agent evaluation across five perspectives: (1) Core LLM capabilities needed for agentic workflows, like planning, and tool use; (2) Application-specific benchmarks such as web and SWE agents; (3) Evaluation of generalist agents; (4) Analysis of agent benchmarks' core dimensions; and (5) Evaluation frameworks and tools for agent developers. Our analysis reveals current trends, including a shift toward more realistic, challenging evaluations with continuously updated benchmarks. We also identify critical gaps that future research must address—particularly in assessing cost-efficiency, safety, and robustness, and in developing fine-grained, scalable evaluation methods [1].

1 Introduction

LLMs have recently made remarkable progress, tackling a wide range of challenging tasks. Yet, LLMs are static, having fixed knowledge, and confined to text-to-text interaction. `LLM-based agents` address those gaps by building on LLMs as a backbone, integrating them into multi-step workflows and equipping them with external tools (Wang et al., 2024a). Hence, LLM agents can perform computations, retrieve up-to-date information, and interact with their environment. Crucially, they can autonomously plan, execute, and adapt complex strategies in real-world settings. This agency enables them to tackle problems once beyond the reach of AI, unlocking innovative applications across diverse domains.

The shift from static models to adaptive, interactive agents calls for a new paradigm for *evaluating* LLM-based agents. Such evaluation must go beyond measuring LLM textual outputs to assess an agent's capacity for sequential decision-making and operation within dynamic environments. It requires benchmarks that can assess the agent's ability to accomplish user tasks via a sequence of actions and interactions. Moreover, benchmarks must co-evolve with agent capabilities, accommodating new classes of tasks and domains.

In this survey, we present the first overview of LLM-based agent evaluation. We aim to benefit developers, benchmark creators, practitioners, and researchers by mapping the current evaluation landscape and identifying key gaps for future research.

We begin by discussing the evaluation of fundamental LLM-based agent capabilities (§2). These include planning, tool use, self-reflection, and memory. We then review benchmarks and evaluation strategies for prominent types of agentic applications: web agents, software engineering agents, scientific agents, and conversational agents (§3). We continue to describe benchmarks and leaderboards for evaluating generalist agents (§4). Consequently, we define and analyze core dimensions of agent benchmarks (§5). §6 reviews current evaluation frameworks for agent developers. These frameworks integrate with the agent's development environment, and support its evaluation throughout the entire development cycle. We conclude with a discussion (§7) of current trends and future research directions in agent evaluation. Figure 1 offers a visual summary of the survey's structure.

2 Agent Capabilities Evaluation

LLM-based agents are composed of a backbone LLM and an agent harness (Yao et al., 2022b). Thus, evaluating the core suite of LLM abilities required for agentic tasks is paramount to understand-

[1]Our GitHub repository tracks works in the field.

Findings of the Association for Computational Linguistics: ACL 2026, pages 26690–26714
July 2-7, 2026 ©2026 Association for Computational Linguistics

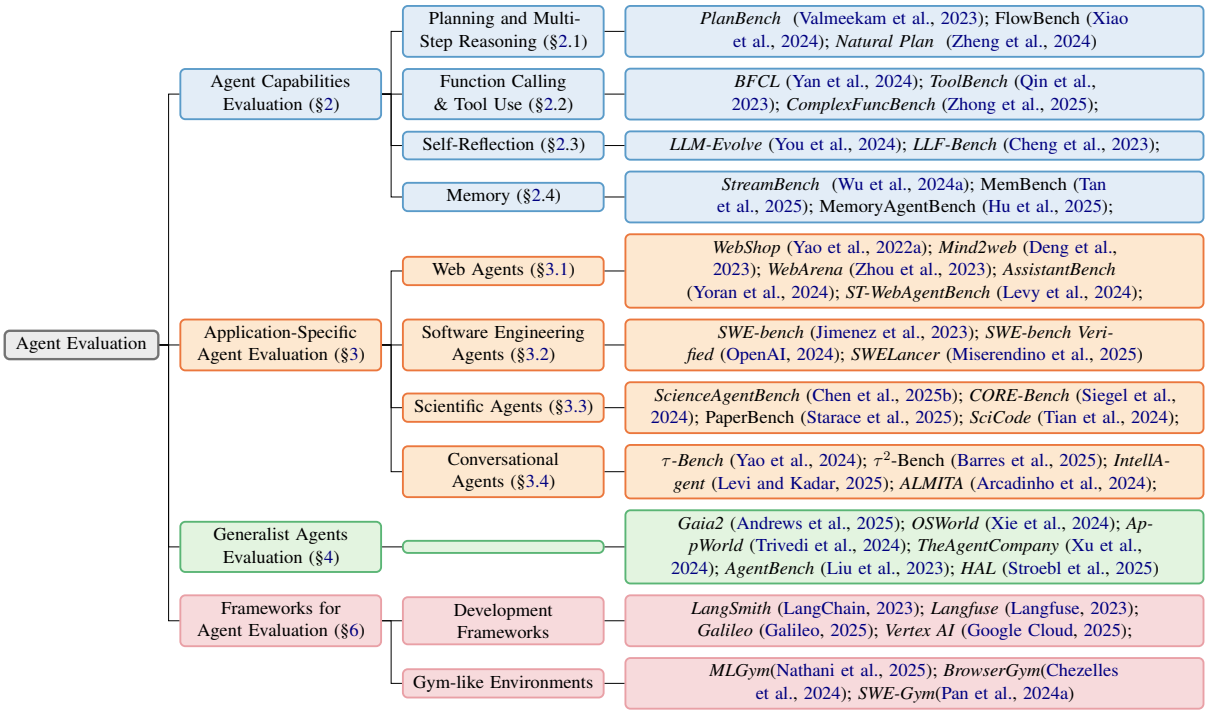

Figure 1: Overview of the paper with core works.

ing the potential and limitations of LLM-based agents. Each ability can be evaluated in isolation or as part of a full agent workflow. Here, we shortly describe four such core agent abilities. In Appendix §B we provide a more detailed review of each one.

Planning and Multi-Step Reasoning enables agents to decompose problems into smaller, more manageable subtasks and create strategic execution paths toward solutions (Gao et al., 2023a).

LLM reasoning benchmarks requiring multiple logical steps, such as HotpotQA (Yang et al., 2018; Cobbe et al., 2021; Suzgun et al., 2022), have been used to evaluate agent-based approaches like ReAct. More specialized planning benchmarks, such as PlanBench (Valmeekam et al., 2023), adapt classical planning tasks to assess LLMs and reveal gaps in long-term planning (Stein et al., 2023). Agent-oriented planning benchmarks further evaluate an agent's ability to follow structured workflows (Xiao et al., 2024) or to manage real-world planning tasks expressed in natural language (Zheng et al., 2024), with focus on long-horizon planning with verifiable constraints (Zhang et al., 2026). Results show that even SOTA models struggle with long-horizon planning.

Function Calling & Tool Use is a fundamental ability for agents to deliver updated, contextually accurate responses (Qin et al., 2023; Tang et al., 2023). Function calling involves several subtasks that work together seamlessly, including intent recognition, function selection, and parameter-value pair-mapping.

Initial benchmarks for tool use focused on these sub-tasks, providing relatively simple, one-step interactions with explicitly predefined parameters. Benchmarks such as ToolAlpaca (Tang et al., 2023), ToolBench (Qin et al., 2023), and the Berkeley Function Calling Leaderboard v1 (BFCL) (Yan et al., 2024) represent this early stage, relying on synthetic data and rule-based matching to measure metrics like pass rates and structural accuracy.

Later versions of BFCL (v2 and v3) introduced multi-turn interactions, organizational tools, and multi-step logic, emphasizing continuous state management. Furthermore, NESTFUL (Basu et al., 2024b) introduces cases where calls are dependent on previous ones, while ComplexFuncBench (Zhong et al., 2025) presents scenarios requiring implicit parameter inference, adherence to user-defined constraints, and efficient long-context processing.

To better reflect real-world scenarios, recent evaluations have become increasingly agentic (Patil et al., 2025a), requiring longer interactions and scaling the number of domains and tools by sourcing them from real MCP servers (Anthropic, Nov 2024). Two current frontier benchmarks,

Scale's MCP Atlas (Bandi et al., 2026) and Tool-Decathlon (Li et al., 2026), further push in this direction. Despite significant model advancements, these benchmarks continue to pose challenges.

Self-Reflection enables agents to self-correct by dynamically adjusting reasoning or actions based on feedback (Renze and Guven, 2024a).

Early evaluation efforts repurposed existing benchmarks into multi-turn feedback loops to gauge LLM's abilities to self-correct (Renze and Guven, 2024b; Huang et al., 2024; Shinn et al., 2023; You et al., 2024). In agentic settings, works like LLM-Evolve (You et al., 2024) reuse past feedback as in-context examples to evaluate self-reflection. Similarly, LLF-Bench (Cheng et al., 2023) utilizes feedback to assess decision-making in diverse environments. Despite these efforts, a standardized benchmark or methodology for assessing self-reflection remains a critical gap.

Memory mechanisms enable LLM agents to manage information and reason across extended interactions (Park et al., 2023), supporting different memory types: episodic (past interactions), semantic (factual knowledge), and procedural (operational information) (Hatalis et al., 2023).

Early studies employed long-context benchmarks (Liu et al., 2024b; Pang et al., 2021) to assess memory mechanisms (Packer et al., 2024; Xu et al., 2025b). More recently, dedicated benchmarks for agentic memory have been introduced. For episodic memory, Benchmarks evaluate how agents leverage prior interactions and feedback to support continual improvement across multi-session agentic tasks (Wu et al., 2024a; He et al., 2026). For semantic memory, benchmarks assess retrieval effectiveness and long-range understanding, revealing that current methods remain limited in maintaining long-range consistency and handling dynamic memory (Tan et al., 2025; Hu et al., 2025; Wu et al., 2025).

3 Application-Specific Agents Evaluation

The landscape of application-specific agents is expanding, with an increasing number of specialized agents emerging across tasks and domains such as web, software, game, embodied, search, and scientific agents (Wang et al., 2024a). Here, we focus on four representative and prominent applications. We review these agents while implicitly addressing several core benchmark dimensions, such as data curation, environment type, and metrics, which we discuss systematically in §5.

3.1 Web Agents

Web agents handle web-related tasks, including e-commerce, information search, and personal assistant tasks. Early work presented simplified simulation environments with limited interaction options (Shi et al., 2017; Liu et al., 2018). For example, WebShop (Yao et al., 2022a) simulates online shopping tasks, from product search to checkout.

More recently, the field shifted toward more realistic evaluation environments. The two most widely used offline and sandboxed online environments are Mind2Web (Deng et al., 2023) and WebArena (Zhou et al., 2023), respectively. These environments serve as the foundation for most current evaluation efforts. Mind2Web provides an offline environment with real websites across diverse domains, supporting rich user interactions (e.g., clicking, selecting, or typing into any element) and enabling intermediate goal evaluation by comparing predicted operations with gold-standard actions. In contrast, WebArena introduces a dynamic environment featuring fully functional websites across multiple domains, enriched with auxiliary tools and knowledge sources. It also defines diverse, long-horizon human tasks, with corresponding functional correctness tests.

Subsequent works build on these environments to evaluate specific dimensions of web interaction, focusing on multi-turn dialogue (Lù et al., 2024), office and enterprise workflows performed by knowledge workers (Drouin et al., 2024; Boisvert et al., 2025), and multi-site, time-intensive tasks (Yoran et al., 2024). Other works refine the evaluation process, offering a more granular analysis of agent performance (Pan et al., 2024b), or employing LLM-as-a-judge methods for more semantic, human-aligned evaluation (Xue et al., 2025). Importantly, Levy et al. (2024) emphasizes safety and trustworthiness by assessing policy compliance and risk mitigation, which are important to real-world deployment.

An important line of research emphasizes the multimodal nature of the modern web, with benchmarks that require agents to integrate visual and textual information, typically interacting via a graphical interface (Koh et al., 2024; Zhang et al., 2024). Notably, WebVoyager (He et al., 2024) has gained significant commercial interest due to its realistic multimodal online evaluation. However, recent

work suggests it exhibits over-optimistic performance estimates, and proposes Online-Mind2Web as a more rigorous alternative that remains challenging for current agents (Xue et al., 2025).

3.2 Software Engineering Agents

The evaluation of software engineering (SWE) agents began with benchmarks that measured LLM fundamental coding capabilities (Chen et al., 2021b; Austin et al., 2021). These early benchmarks focused on short, self-contained, algorithm-specific tasks, thus falling short of addressing the full complexity of real-world SWE tasks.

SWE-bench (Jimenez et al., 2023) was introduced to address the above shortcomings by providing an end-to-end evaluation framework grounded in real-world GitHub issues. Each task includes a detailed issue description, full repository context, executable environments (e.g., Docker), and validation tests, enabling agents to generate and verify code patches automatically. Several follow-up works identified evaluation issues, like overly specific or unrelated unit tests, underspecified issue descriptions, and problematic environmental setups, and thus proposed new variants (SWE-bench Lite, 2024; Xia et al., 2024; OpenAI, 2024; Aleithan et al., 2024). The most widely adopted of these, SWE-bench Verified (OpenAI, 2024), carries extensive human filtration and validation to yield a high-quality subset of 500 samples. It also standardizes execution through containerized environments and provides difficulty annotations, enabling more reproducible and interpretable evaluations. These improvements make SWE-bench Verified the de facto benchmark for assessing SWE agents.

Following the success of SWE-bench, multilingual (Zan et al., 2024; Yang et al., 2025) and multimodal (Yang et al., 2024) versions were proposed. Complementary benchmarks explore additional SWE tasks, including the ability to generate and validate tests from GitHub issues (Ahmed et al., 2024; Mündler et al., 2024), or solving real-world IT automation tasks (Jha et al., 2025). Notably, Terminal-Bench (Merrill et al., 2025) focuses on interactive terminal usage, assessing agents' command-line proficiency.

More recent benchmarks push toward more challenging and realistic evaluation. SWE-Lancer (Miserendino et al., 2025) collected 1,400 freelance tasks from Upwork with total payouts over $\$1M$. Tasks include both technical fixes and managerial decisions, evaluated via verified tests or comparison to human managers' choices. Results expose gaps in long-term reasoning and decision-making. Similarly, SWE-bench Pro (Deng et al., 2025) introduces 1,865 human-verified tasks spanning 41 repositories, often requiring multi-file edits and hours of effort. Model performance remains below 25% Pass@1, highlighting current limitations in handling long-horizon, complex code changes.

3.3 Scientific Agents

Scientific agents automate core research tasks by integrating domain knowledge and scientific tools. Their evaluation has progressed from basic reasoning benchmarks to comprehensive frameworks assessing diverse scientific research capabilities. Early efforts focused on knowledge recall and reasoning (Clark et al., 2018; Lu et al., 2022; Wang et al., 2022a), and literature understanding (Dasigi et al., 2021; Lee et al., 2023; DeYoung et al., 2021). More recent benchmarks like SciRiff (Wadden et al., 2024) broaden the scope to instruction-following across scientific domains.

Recent advancements have shifted the focus toward developing and assessing scientific agents in accelerating scientific research. New benchmarks span the full research pipeline: (1) Scientific Ideation: Evaluates agents' ability to generate novel, expert-level research ideas, emphasizing creativity, relevance, and scientific feasibility (Si et al., 2025). (2) Experiment Design: Benchmarks like AAAR-1.0 (Lou et al., 2025) assess systematic experiment planning, hypothesis formulation, appropriate use of methodologies, and rigor of experimental procedures. (3) Code Generation: Benchmarks such as SciCode (Tian et al., 2024), ScienceAgentBench (Chen et al., 2025b), CORE-Bench (Siegel et al., 2024), and PaperBench (Starace et al., 2025) test the agent's ability to produce accurate, executable scientific code (Chan et al., 2025). (4) Peer Review: Evaluates whether agents can generate substantive, high-quality reviews (Chamoun et al., 2024). Ultimately, the field is moving toward benchmarks that encompass the full research cycle, leading to the evaluation of innovative scientific discovery.

3.4 Conversational Agents

Conversational agents are agents designed to perform goal-directed, multi-turn dialogue with a user to accomplish a specific task, such as booking a flight or resolving a customer support issue. This field builds on Task-Oriented Dialogue Systems

(TODS), extending it from purely textual tasks to real-world tasks requiring environment interaction.

Early TODS benchmarks focused on multi-turn conversations across different domains (Budzianowski et al., 2018; Andreas et al., 2020), and distinct user intents (Chen et al., 2021a). Yet these benchmarks were confined to textual interaction. To address that, Sierra proposed τ-Bench (Yao et al., 2024). It assesses agents' ability to interact with simulated users to accomplish their tasks by effectively utilizing API tools, while adhering to domain-specific policies. τ-Bench spans domains like retail and airline customer service. However, it is limited in scale, user simulation setup, and focuses solely on coarse-grained end-to-end metrics, overlooking policy violations and dialogue flow errors. τ^2-Bench (Barres et al., 2025) addresses these limitations by introducing a Telecom domain, where the user utilizes tools to act in a shared, dynamic environment, and by adding a compositional task generator that programmatically creates diverse, verifiable tasks.

Although τ-Bench is the most dominant benchmark in this category, it builds on prior work that automates benchmark creation and enables dynamic evaluation. IntellAgent (Levi and Kadar, 2025) proposes an automated process for generating synthetic test scenarios from database schemas and policy documents. They showed a high correlation between τ-Bench and their synthetic benchmark results. ALMITA (Arcadinho et al., 2024) used a hybrid approach that starts from user intents and a sequence of intermediate LLM-generated graphs, followed by manual filtering to generate diverse, realistic customer support scenarios.

4 Generalist Agent Evaluation

Similar to the evolution of LLMs from task-specific into general-purpose models, agents are also transitioning from application-specific toward general-purpose agents (Bandel et al., 2026b). Such tasks require integrated capabilities, from planning, reasoning, and tool use to web interaction, file handling, code execution, and more. This shift has led to the development of two complementary approaches to generalist agent evaluation.

One approach aims to address this gap by proposing benchmarks that inherently require a wide range of capabilities. A key benchmark in this category is Gaia (Mialon et al., 2023). Gaia is composed of real-world questions that require abilities such as reasoning, multi-modality handling, web browsing, and tool-use proficiency. Over time, model performance on Gaia improved, saturating the easier portion of the benchmark and emphasizing the need for an updated version. Andrews et al. (2025) proposed Gaia2 as a mobile environment with apps such as email, messaging, and calendar. The set of required capabilities was extended to include handling ambiguity, noise, temporal constraints, and multi-agent collaboration.

This work continues a class of benchmarks that evaluate agents in full computer environments. OSWorld (Xie et al., 2024), AppWorld (Trivedi et al., 2024), and more (Kapoor et al., 2024a; Bonatti et al., 2025) test the agent's ability to execute complex tasks across applications, manage control flows, and ensure stable performance in real settings. They differ in the way the agent interacts with the environment. In OSWorld, the agent interacts via UI, while in AppWorld and Gaia2, it uses code and API calls.

The second approach for generalist agent evaluation relies on unifying several task-specific benchmarks into one. AgentBench (Liu et al., 2023) introduces interactive environments spanning OS operations, databases, games, and household tasks—highlighting core skills like flexibility and tool-based problem solving. HAL (Holistic Agent Leaderboard; Stroebl et al., 2025) provides a unified platform for benchmarks across domains, including coding and web. Yet, this approach does not support the same agent harness evaluation across any benchmark environment. Harbor and Exgentic aim to solve that by providing frameworks with a unified protocol for general agent assessment (Harbor Framework Team, 2026; Bandel et al., 2026a). These represent the first steps toward holistic, standardized, cross-environment agent evaluation (Lacoste et al., 2026).

5 Core Benchmark Dimensions

The preceding sections reviewed benchmarks organized by agent type and application domain. Here, we take a broader perspective and analyze benchmarks along shared orthogonal dimensions: data curation, environment, interface, metric, and safety. This analysis reveals common structural patterns and gaps. In Table 1, we compare representative benchmarks based on these dimensions.

Data Curation High-quality benchmarks rarely rely solely on human-curated data; most employ a

hybrid curation strategy. For example, SWE-bench Verified refines the original harvest of GitHub issues through human validation to improve the robustness and reliability of the benchmark. Similarly, AppWorld constructs tasks using a synthetic "world" of 100 fictitious users but validates them with programmatic checks. Other benchmarks, like Mind2Web, derive data from real-world interaction logs, which are then cleaned and annotated. In contrast, GAIA specifically utilizes humans for crafting and validating the questions, ensuring challenging, unambiguous questions that are conceptually simple for humans but require complex tool usage for agents. This raises the tension between the reliance on human annotation for ensuring data validity and the need to automate the data curation and validation process, allowing scalable and adaptable evaluation. This tension emphasizes the need for methods that automate this process without compromising its quality.

Environment Evaluation environments generally fall into two categories: *static* and *dynamic*. Static environments, such as the original Mind2Web, rely on offline traces or cached web pages where agents predict the next action without influencing the state. While scalable, these fail to capture the cascading effects of errors: incorrect actions have no downstream consequences, missing compounding failures that can ultimately cause task failure. Dynamic environments, conversely, allow agents to interact with a live or simulated world (e.g., Docker containers for SWE, browser sandboxes for web agents), where their actions alter the state the agent observes, enabling diagnosis of failure modes in long-horizon tasks. As agents become more capable, benchmarks must embrace dynamic environments that assess the agent's ability to evaluate long-horizon interactions with the environment.

Interaction Interface The interface defines the communication protocol between the agent and the environment, governing the action and observation spaces. We categorize benchmarks into three primary interfaces: *Code and Terminal* interface requires agents to generate executable scripts, e.g., Python, Bash, or SQL. This interface is predominant in SWE and scientific benchmarks, where success depends on logic and syntactic correctness. *Tools* interface restricts actions to predefined function calls. Benchmarks such as τ-Bench and AppWorld rely on tool calling with schema adherence to achieve their goal. *Graphical User Interfaces (GUI)* simulate human interaction via accessibility trees (e.g., HTML DOM) or visual UI. Web and consumer-facing applications benchmarks utilize this modality to evaluate visual grounding and computer or web navigation.

Benchmark	Data	Env.	Interface	Metric	Safety
SWE-bench Ver.	Hybrid	Dynamic	Code	Unit Tests	No
SWE-Lancer	Hybrid	Dynamic	Code	End-to-end	No
Mind2Web	Hybrid	Static	GUI	Action Match	No
WebArena	Hybrid	Dynamic	GUI	Mix	No
PaperBench	Hybrid	Dynamic	Code	End-to-end	No
TAU-Bench	Hybrid	Dynamic	Tools	State Match	Yes
AppWorld	Hybrid	Dynamic	Tools	State Match	No
GAIA	Human	Dynamic	Mix	Answer Match	No

Table 1: Comparative analysis of representative agent benchmarks. We characterize each benchmark by its data curation strategy, environment dynamicity, interaction interface, evaluation metric, and whether it explicitly incorporates safety constraints.

Metric The most ubiquitous metric is task completion, yet its implementation varies depending on the application and expected output. For SWE tasks, where the output is a piece of functional code, SWE measures are adopted, like execution-based *unit testing*. Tasks that require modifying the environment state, like τ-Bench, deploy *state matching* against the gold state to assess the modification correctness. For multi-step reasoning tasks, such as GAIA, *answer matching* is used to verify unambiguous short-form responses against a gold standard. This metric divergence emphasizes the need for targeted verification steps to ensure the benchmark's validity (Zhu et al., 2025). Notably, such binary outcome metrics are insufficient to understand the intermediate agent's progress, and call for fine-grained evaluation solutions (See §7).

Safety and Robustness Metrics While most benchmarks prioritize capability, safety, and robustness are critical for enterprise adoption. Robustness is often quantified via pass^k, the fraction of tasks where the agent succeeds across all k independent runs. Beyond robustness, enterprise agents must adhere to strict policies, such as data privacy and access control, which are rarely tested in standard benchmarks. For instance, SWE-Lancer does not inherently penalize risky behaviors unless they interfere with replicating the target behavior. Future benchmarks must integrate "guardrail" metrics, penalizing agents that achieve task success via non-compliant actions (e.g., deleting production databases).

6 Frameworks for Agent Evaluation

To meet the growing need for systematic assessment of LLM agents, several general-purpose frameworks have emerged, offering developers tools for continuous monitoring, evaluation, error analysis, and performance optimization. Unlike the benchmarks discussed in the preceding sections, which assess fully developed systems using fixed scenarios and standardized test datasets, these frameworks integrate into the development process, enabling flexible, custom scenario design and supporting a broad range of general use cases across both development and deployment.

There are many frameworks supporting the evaluation of a wide range of agent types, including LangSmith (LangChain, 2023), Langfuse (Langfuse, 2023), Google Vertex AI evaluation service (Google Cloud, 2025), Arize AI's Evaluation Framework (Arize AI, Inc, 2025), Galileo Agentic Evaluation (Galileo, 2025), Patronus AI (Patronus AI, Inc., 2023), W&B Weave (Weights&Biases, 2023), LangChains' AgentEvals (LangChain, 2025); Databricks Mosaic AI Agent Evaluation (Databricks, 2023), which is mostly designed for RAG like tasks; Botpress Multi-Agent Evaluation System (Kargwal, 2025) and AutoGen (Dibia et al., 2024) for multi-agent systems; and more.

All evaluation platforms provide continuous monitoring of agent trajectories, assessing key performance metrics such as task completion rates, latency, execution speed, and, in some cases, throughput and memory usage (LangChain, 2023). Some frameworks utilize the OpenTelemetry (Blanco, 2023) observability framework and their infrastructure, including Langfuse and Google Vertex AI.

Beyond observability, each framework applies quality assessment methods across multiple levels of granularity:

Final Response Evaluation. Frameworks often incorporate LLM-based judges to evaluate agent responses against predefined criteria (such as faithfulness or politeness), with some offering proprietary judge models (e.g., Databricks Mosaic and PatronusAI). Additionally, most platforms allow for customizable assessment metrics, enabling domain-specific evaluation of output quality and relevance. Final-response evaluation is fast, inexpensive, and easy to automate, making it well-suited for large-scale monitoring and regression testing. However, it offers limited insight into agent behavior, as it cannot assess intermediate decisions, execution efficiency, or failure causes within complex workflows.

Stepwise Evaluation. Most frameworks support granular assessments of individual agent steps, such as LLM generations, tool invocations, and routing decisions, enabling error localization and systematic analysis of where and how multi-step executions fail. A common approach evaluates each step independently using predefined or customizable judges, often LLM-based or lightweight classifiers, that assess quality attributes such as correctness, relevance, or instruction adherence. Additionally, many frameworks perform tool-specific evaluation by validating tool choice, parameter schemas, and output usability. To better align evaluation with agent structure, Arize Phoenix provides agent-specific step templates, which tailor evaluation criteria to particular stages such as routing, planning, retrieval, or reflection.

These approaches assume that each action can be meaningfully assessed in isolation, overlooking dependencies between steps. To address this gap, Galileo Agentic Evaluation incorporates a goal-progress-oriented *action advancement metric*, which measures whether each step successfully contributes to or advances toward a user-defined goal.

Trajectory-Based Assessment. Some platforms, such as Google Vertex AI and LangSmith, also support trajectory-based assessments, which move beyond individual steps and analyze how an agent navigates toward task completion. Current approaches fall into two broad categories. ***Reference-Based*** methods compare the trajectory against an expected optimal path, measuring alignment between the observed and gold action sequences. Platforms such as LangSmith, Vertex AI, and AgentEvals support various alignment modes, including exact, partial, unordered, and subset matching. AgentEvals further extends this through graph-based evaluation: for frameworks that model agents as graphs, it assesses whether the agent visits the expected nodes and transitions rather than requiring alignment over a flat sequence of tool calls. However, reference-based methods are inherently limited, as multiple valid paths typically exist and manual reference specification is often infeasible. ***Reference-Free*** methods use LLM-based judges to evaluate trajectory quality without a predefined gold path, assessing properties such as coherence, efficiency, or goal directedness directly from the observed sequence.

Framework	Stepwise Assessment	Monitoring	Trajectory Assessment	Human in the Loop	Synthetic Data Generation	A/B Comparisons
LangSmith (LangChain)	✓	✓	✓	✓	×	✓
Langfuse (Langfuse)	✓	✓	×	✓	×	✓
Google Vertex AI evaluation (Google Cloud)	✓	✓	✓	×	×	✓
Arize AI's Evaluation (Arize AI, Inc)	✓	✓	×	✓	✓	✓
Galileo Agentic Evaluation (Galileo)	✓	✓	×	✓	×	✓
Patronus AI (Patronus AI, Inc.)	✓	✓	×	✓	✓	✓
AgentsEval (LangChain)	×	×	✓	×	×	×
Mosaic AI (Databricks)	✓	✓	×	✓	✓	✓

Table 2: Supported evaluation capabilities of major agent frameworks. Note that some of these capabilities are still in initial phases of development, as discussed further in the text.

A core methodological consideration is the choice between reference-based and reference-free evaluation. Reference-based methods offer precision and reproducibility but depend on predefined expected behavior. Reference-free methods, typically relying on LLM-based judges, provide greater flexibility at the cost of reliability. This tension extends to judge design: general-purpose judges offer broad coverage but lower precision, while task-specific judges excel on their target criteria at the cost of a narrower scope.

Supporting Capabilities. Beyond quality assessment, frameworks provide supporting capabilities for data management and experimentation. Most offer integrated annotation tools and support *human-in-the-loop* evaluation, enabling the extraction of evaluation datasets from production logs. Platforms such as Patronus AI and Databricks Mosaic additionally facilitate *synthetic data generation* using proprietary seed data. Current frameworks also support *A/B comparisons*, enabling side-by-side analysis of inputs, outputs, and metrics across runs, and in some cases (e.g., Patronus AI) of aggregated results across experimental setups.

Table 2 presents key frameworks for agent evaluation along with their support for the evaluation features discussed in this section.

Overall, current evaluation frameworks show great value but still face several open challenges. First, while analyzing individual traces is well supported, deriving insights across large collections of runs remains difficult. Thus, understanding root causes for agent failures at scales is still unsupported. A/B comparisons partially address this by allowing side-by-side analysis of different experimental setups. Yet, causally attributing outcome differences to specific steps or decisions is not yet possible. Second, current frameworks overlook the cost incurred by the evaluation process itself, specifically when scaling over a large number of traces with expensive LLM judges. This emphasizes the need for more efficient evaluation processes, with better resource allocation techniques. Finally, most current frameworks lack built-in support for evaluating safety and policy compliance (See §7).

Gym-like Environments.

Gym-like Environments provide controlled, interactive environment simulations for agent evaluation. Inspired by OpenAI Gym (Brockman et al., 2016), originally designed for training and evaluating Reinforcement Learning algorithms, these frameworks have been adapted to the training and evaluation of LLM agents using realistic task simulations, allowing them to interact with dynamic environments. Moreover, these frameworks enable standardized evaluation across various benchmarks, with environments made for web agents (Chezelles et al., 2024), AI research agents (Nathani et al., 2025), and SWE agents (Pan et al., 2024a).

7 Discussion

7.1 Current Trends

Our review identifies two key trends shaping the current landscape of agent evaluation:

Realistic and Challenging Evaluation. While early evaluations often employed simplified, static environments, there is a clear shift toward more realistic and complex benchmarks. Web agents evolved from basic simulations to dynamic, real-world settings such as *WebArena*. In software engineering, benchmarks like *SWE-bench* utilize real GitHub issues, moving beyond synthetic coding tasks. Furthermore, there is increasing interest in long-horizon tasks typically performed by highly trained human experts, pushing evaluation closer to real-world professional workflows. This change reflects the growing need to assess agents' ability to derive practical value in realistic settings, to stress-test their limitations, and guide their progress.

Live Benchmarks. The rapid pace of LLM and agent development necessitates adaptive and

continuously updated evaluation methods. Static benchmarks quickly become obsolete, saturated, and abandoned. In response, we see a rise in "live" benchmarks. For instance, the few versions of *BFCL*, and the refinement of the *SWE-bench* family (*Verified*, *Pro*, and more). These ongoing adaptations serve several purposes: matching the increased capabilities of agents, addressing shortcomings of previous versions (e.g., by conducting human verification and refining success metrics), and adapting to evolving research ecosystems such as MCP-based tool calling. This dynamic benchmarking approach is essential to maintain relevance in a fast-moving field.

7.2 Future Directions

We also recognize critical gaps in current agent evaluation that future research must address.

Advancing Granular Evaluation. Many current benchmarks rely on coarse-grained, end-to-end success metrics that, while useful for gauging overall performance, fall short in diagnosing specific agent failures. This lack of granularity obscures insights into intermediate decision processes such as tool selection and reasoning quality. Addressing this limitation calls for the development of standardized, fine-grained evaluation metrics targeting the trajectory of an agent's task execution. Future work should explore detailed, step-by-step assessments to provide richer feedback and guide targeted improvements.

Cost and Efficiency Metrics. Current evaluations often prioritize performance while overlooking cost and efficiency measurements. This emphasis can inadvertently drive the development of highly capable but resource-intensive agents, limiting their practical deployment Kapoor et al. (2024b). Future evaluation frameworks should integrate cost efficiency as a core metric, tracking factors such as token usage, API expenses, inference time, and overall resource consumption. Establishing standardized cost metrics will help guide the development of agents that balance performance with operational viability.

Scaling & Automating. The reliance on static, human-annotated evaluation data poses significant scalability challenges. It is resource-intensive, and the resulting benchmarks quickly become outdated in this rapidly evolving field. This shortcoming underscores the need for scalable, automated evaluation, which may be addressed via *synthetic data generation* techniques. Another avenue is automating evaluation by employing an LLM or agent as a Judge Zhuge et al. (2024). This approach not only reduces the reliance on resource-intensive human annotation but also holds the potential to capture more nuanced aspects of agent performance.

Safety and Compliance. Current benchmarks lack sufficient focus on safety, trustworthiness, and policy compliance. While early efforts have begun to address these dimensions, evaluations still lack comprehensive tests for robustness against adversarial inputs, bias mitigation, and organizational and societal policy compliance. Future research should include safety metrics in benchmarks as well as developing safety benchmarks that simulate real-world scenarios, particularly in multi-agent scenarios where emergent risks may arise (Hammond et al., 2025). This can ensure that agents are not only effective but also safe and secure.

Decoupling LLM & Harness Evaluation. Most current agent benchmarks conflate two distinct evaluation targets: (1) the inherent capabilities of the backbone LLM, and (2) the design of the agent Harness (a.k.a. scaffold). Disentangling these is essential for enabling systemic attribution of performance gains. Efforts like Harbor and Exgentic begin to address this by standardizing agent evaluation across models and agent harness settings. Future work should develop controlled evaluation protocols that vary each factor independently, enabling systematic comparison and isolating the contribution of individual components, whether LLM capabilities, harness design, or specific modules such as memory or planning, to overall agent performance.

8 Conclusion

This survey presents an overview of the evolving field of LLM-based agent evaluation, outlining its progression from assessing isolated capabilities in simplified settings to evaluating agents in realistic, dynamic, and challenging environments. While notable progress has been made, several challenges remain. Future research should focus on developing fine-grained, scalable evaluation methods that extend beyond overall success rates, establishing standardized metrics for cost-efficiency, safety, and robustness for responsible deployment. For practical guidance, we provide actionable benchmark recommendations in Appendix §E.

Limitations

While this survey provides a comprehensive overview of the evaluation landscape for LLM-based agents, it is important to acknowledge certain limitations inherent to a review of such a rapidly evolving field.

First, the domain of LLM-based agents and their evaluation is exceptionally dynamic. New benchmarks, evaluation frameworks, agent architectures, and research findings are being published at an unprecedented pace. Although we have striven to incorporate the most current and impactful work up to the time of writing, this survey inevitably represents a snapshot. Some very recent or forthcoming developments might not have been included. Our commitment to maintaining a continuously updated GitHub repository, as mentioned in the abstract, aims to mitigate this challenge over time for the community.

Second, the selection of benchmarks and frameworks, while intended to be representative, is subject to the breadth of the field. To maintain clarity and focus, we prioritized works that illustrate key trends or address significant aspects of agent evaluation. Consequently, some specific or niche evaluation approaches may not have received detailed coverage.

Third, in covering a wide array of topics, the depth of analysis for each individual benchmark or framework is necessarily constrained. Readers requiring an exhaustive understanding of a particular evaluation tool or methodology are encouraged to consult the primary research articles cited throughout this survey.

Finally, the discussion on "Future Directions" (§7.2) and the identification of "critical gaps" inherently involve a degree of foresight and interpretation based on current trends. While these are informed by our analysis of the existing literature, the actual trajectory of future research and the relative importance of these identified areas will continue to evolve.

Despite these limitations, we believe this survey offers a valuable and structured synthesis of the current state of LLM-based agent evaluation, serving as a useful resource for researchers, developers, and practitioners in the field.

References

Toufique Ahmed, Martin Hirzel, Rangeet Pan, Avraham Shinnar, and Saurabh Sinha. 2024. Tdd-bench verified: Can llms generate tests for issues before they get resolved? *arXiv preprint arXiv:2412.02883*.

Reem Aleithan, Haoran Xue, Mohammad Mahdi Mohajer, Elijah Nnorom, Gias Uddin, and Song Wang. 2024. Swe-bench+: Enhanced coding benchmark for llms. *ArXiv*, abs/2410.06992.

Jacob Andreas, John Bufe, David Burkett, Charles Chen, Josh Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, Hao Fang, Alan Guo, David Hall, Kristin Hayes, Kellie Hill, Diana Ho, Wendy Iwaszuk, Smriti Jha, Dan Klein, Jayant Krishnamurthy, Theo Lanman, Percy Liang, Christopher H. Lin, Ilya Lintsbakh, Andy McGovern, Aleksandr Nisnevich, Adam Pauls, Dmitrij Petters, Brent Read, Dan Roth, Subhro Roy, Jesse Rusak, Beth Short, Div Slomin, Ben Snyder, Stephon Striplin, Yu Su, Zachary Tellman, Sam Thomson, Andrei Vorobev, Izabela Witoszko, Jason Wolfe, Abby Wray, Yuchen Zhang, and Alexander Zotov. 2020. Task-oriented dialogue as dataflow synthesis. *Transactions of the Association for Computational Linguistics*, 8:556–571.

Pierre Andrews, Amine Benhalloum, Gerard Moreno-Torres Bertran, Matteo Bettini, Amar Budhiraja, Ricardo Silveira Cabral, Virginie Do, Romain Froger, Emilien Garreau, Jean-Baptiste Gaya, Hugo Laurençon, Maxime Lecanu, Kunal Malkan, Dheeraj Mekala, Pierre Ménard, Grégoire Mialon, Ulyana Piterbarg, Mikhail Plekhanov, Mathieu Rita, Andrey Rusakov, Thomas Scialom, Vladislav Vorotilov, Mengjue Wang, and Ian Yu. 2025. Are: Scaling up agent environments and evaluations. *Preprint*, arXiv:2509.17158.

Anthropic. Nov 2024. Introducing the model context protocol. https://www.anthropic.com/news/model-context-protocol.

Samuel Arcadinho, David Oliveira Aparicio, and Mariana S. C. Almeida. 2024. Automated test generation to evaluate tool-augmented LLMs as conversational AI agents. In *Proceedings of the 2nd GenBench Workshop on Generalisation (Benchmarking) in NLP*, pages 54–68, Miami, Florida, USA. Association for Computational Linguistics.

Arize AI, Inc. 2025. Agent evaluation.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. 2021. Program synthesis with large language models. *ArXiv*, abs/2108.07732.

Elron Bandel, Asaf Yehudai, Lilach Eden, Yehoshua Sagron, Yotam Perlitz, Elad Venezian, Natalia Razinkov, Natan Ergas, Shlomit Shachor Ifergan, Segev Shlomov, Michal Jacovi, Leshem Choshen,

Liat Ein-Dor, Yoav Katz, and Michal Shmueli-Scheuer. 2026a. General agent evaluation. *Preprint*, arXiv:2602.22953.

Elron Bandel, Asaf Yehudai, Alexandre Lacoste, Avijit Ghosh, Graham Neubig, Margaret Mitchell, Michal Shmueli-Scheuer, and Leshem Choshen. 2026b. Agentic systems should be general. *SSRN Electronic Journal*.

Chaithanya Bandi, Ben Hertzberg, Geobio Boo, Tejas Polakam, Jeff Da, Sami Hassaan, Manasi Sharma, Andrew Park, Ernesto Hernandez, Dan Rambado, Ivan Salazar, Rafael Cruz, Chetan Rane, Ben Levin, Brad Kenstler, and Bing Liu. 2026. Mcp-atlas: A large-scale benchmark for tool-use competency with real mcp servers. *Preprint*, arXiv:2602.00933.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. 2025. τ^2-bench: Evaluating conversational agents in a dual-control environment. *Preprint*, arXiv:2506.07982.

Kinjal Basu, Ibrahim Abdelaziz, Subhajit Chaudhury, Soham Dan, Maxwell Crouse, Asim Munawar, Sadhana Kumaravel, Vinod Muthusamy, Pavan Kapanipathi, and Luis A Lastras. 2024a. Api-blend: A comprehensive corpora for training and benchmarking api llms. *arXiv preprint arXiv:2402.15491*.

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

Pratik Bhavsar. 2025. Agent leaderboard. `https://huggingface.co/spaces/galileo-ai/agent-leaderboard`.

Daniel Gomez Blanco. 2023. *Practical OpenTelemetry*. Springer.

Ben Bogin, Kejuan Yang, Shashank Gupta, Kyle Richardson, Erin Bransom, Peter Clark, Ashish Sabharwal, and Tushar Khot. 2024. Super: Evaluating agents on setting up and executing tasks from research repositories. *Preprint*, arXiv:2409.07440.

Léo Boisvert, Megh Thakkar, Maxime Gasse, Massimo Caccia, Thibault de Chezelles, Quentin Cappart, Nicolas Chapados, Alexandre Lacoste, and Alexandre Drouin. 2025. Workarena++: Towards compositional planning and reasoning-based common knowledge work tasks. *Advances in Neural Information Processing Systems*, 37:5996–6051.

Rogerio Bonatti, Dan Zhao, Francesco Bonacci, Dillon Dupont, Sara Abdali, Yinheng Li, Yadong Lu, Justin Wagle, Kazuhito Koishida, Arthur Bucker, Lawrence Keunho Jang, and Zheng Hui. 2025. Windows agent arena: Evaluating multi-modal OS agents at scale. In *Forty-second International Conference on Machine Learning*.

Nikos I Bosse, Jon Evans, Robert G Gambee, Daniel Hnyk, Peter Mühlbacher, Lawrence Phillips, Dan Schwarz, Jack Wildman, et al. 2025. Deep research bench: Evaluating ai web research agents. *arXiv preprint arXiv:2506.06287*.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. Openai gym. *arXiv preprint arXiv:1606.01540*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

David Castillo-Bolado, Joseph Davidson, Finlay Gray, and Marek Rosa. 2024. Beyond prompts: Dynamic conversational benchmarking of large language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Eric Chamoun, Michael Schlichtkrull, and Andreas Vlachos. 2024. Automated focused feedback generation for scientific writing assistance. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9742–9763, Bangkok, Thailand. Association for Computational Linguistics.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Lilian Weng, and Aleksander Mądry. 2025. Mle-bench: Evaluating machine learning agents on machine learning engineering. *Preprint*, arXiv:2410.07095.

Derek Chen, Howard Chen, Yi Yang, Alexander Lin, and Zhou Yu. 2021a. Action-based conversations dataset: A corpus for building more in-depth task-oriented dialogue systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3002–3017, Online. Association for Computational Linguistics.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé, Jared Kaplan, Harrison Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mo Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, David W. Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William H. Guss, Alex Nichol, Igor Babuschkin, Suchir Balaji, Shantanu Jain, Andrew Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew M. Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob

McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021b. Evaluating large language models trained on code. *ArXiv*, abs/2107.03374.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, et al. 2025a. Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent. *arXiv preprint arXiv:2508.06600*.

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. 2025b. Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *The Thirteenth International Conference on Learning Representations*.

Ching-An Cheng, Andrey Kolobov, Dipendra Misra, Allen Nie, and Adith Swaminathan. 2023. Llf-bench: Benchmark for interactive learning from language feedback. *Preprint*, arXiv:2312.06853.

De Chezelles, Thibault Le Sellier, Maxime Gasse, Alexandre Lacoste, Alexandre Drouin, Massimo Caccia, Léo Boisvert, Megh Thakkar, Tom Marty, Rim Assouel, et al. 2024. The browsergym ecosystem for web agent research. *arXiv preprint arXiv:2412.05467*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

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

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. *Preprint*, arXiv:2105.03011.

Databricks. 2023. Mosaic ai agent evaluation: Assessing ai application performance.

Xiang Deng, Jeff Da, Edwin Pan, Yannis Yiming He, Charles Ide, Kanak Garg, Niklas Lauffer, Andrew Park, Nitin Pasari, Chetan Rane, Karmini Sampath, Maya Krishnan, Srivatsa Kundurthy, Sean Hendryx, Zifan Wang, Chen Bo Calvin Zhang, Noah Jacobson, Bing Liu, and Brad Kenstler. 2025. Swe-bench pro: Can ai agents solve long-horizon software engineering tasks? *Preprint*, arXiv:2509.16941.

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

Jay DeYoung, Iz Beltagy, Madeleine van Zuylen, Bailey Kuehl, and Lucy Lu Wang. 2021. Ms2: Multi-document summarization of medical studies. *Preprint*, arXiv:2104.06486.

Victor Dibia, Jingya Chen, Gagan Bansal, Suff Syed, Adam Fourney, Erkang Zhu, Chi Wang, and Saleema Amershi. 2024. Autogen studio: A no-code developer tool for building and debugging multi-agent systems. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 72–79.

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

Mingxuan Du, Benfeng Xu, Chiwei Zhu, Xiaorui Wang, and Zhendong Mao. 2025. Deepresearch bench: A comprehensive benchmark for deep research agents. *arXiv preprint arXiv:2506.11763*.

Galileo. 2025. Introducing agentic evaluations.

Chen Gao, Xiaochong Lan, Nian Li, Yuan Yuan, Jingtao Ding, Zhilun Zhou, Fengli Xu, and Yong Li. 2023a. Large language models empowered agent-based modeling and simulation: A survey and perspectives. *Preprint*, arXiv:2312.11970.

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

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Google Cloud. 2025. Evaluate your ai agents with vertex gen ai evaluation service.

Boyu Gou, Zanming Huang, Yuting Ning, Yu Gu, Michael Lin, Weijian Qi, Andrei Kopanev, Botao Yu, Bernal Jiménez Gutiérrez, Yiheng Shu, Chan Hee Song, Jiaman Wu, Shijie Chen, Hanane Nour Moussa, Tianshu Zhang, Jian Xie, Yifei Li, Tianci Xue, Zeyi Liao, Kai Zhang, Boyuan Zheng, Zhaowei Cai, Viktor Rozgic, Morteza Ziyadi, Huan Sun, and Yu Su. 2025. Mind2web 2: Evaluating agentic search with agent-as-a-judge. *Preprint*, arXiv:2506.21506.

Zhicheng Guo, Sijie Cheng, Hao Wang, Shihao Liang, Yujia Qin, Peng Li, Zhiyuan Liu, Maosong Sun, and Yang Liu. 2024. Stabletoolbench: Towards stable

large-scale benchmarking on tool learning of large language models. *arXiv preprint arXiv:2403.07714*.

Lewis Hammond, Alan Chan, Jesse Clifton, Jason Hoelscher-Obermaier, Akbir Khan, Euan McLean, Chandler Smith, Wolfram Barfuss, Jakob Foerster, Tomáš Gavenčiak, et al. 2025. Multi-agent risks from advanced ai. *arXiv preprint arXiv:2502.14143*.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Luke Benson, Lucy Sun, E. Zubova, Yujie Qiao, Matthew Burtell, David Peng, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Shafiq R. Joty, Alexander R. Fabbri, Wojciech Kryscinski, Xi Victoria Lin, Caiming Xiong, and Dragomir R. Radev. 2022. Folio: Natural language reasoning with first-order logic. *EMNLP 2024*.

Simeng Han, Aaron Yu, Rui Shen, Zhenting Qi, Martin Riddell, Wenfei Zhou, Yujie Qiao, Yilun Zhao, Semih Yavuz, Ye Liu, Shafiq Joty, Yingbo Zhou, Caiming Xiong, Dragomir R. Radev, Rex Ying, and Arman Cohan. 2024. P-folio: Evaluating and improving logical reasoning with abundant human-written reasoning chains. *EMNLP 2024 Findings*.

Harbor Framework Team. 2026. Harbor: A framework for evaluating and optimizing agents and models in container environments.

Kostas Hatalis, Despina Christou, Joshua Myers, Steven Jones, Keith Lambert, Adam Amos-Binks, Zohreh Dannenhauer, and Dustin Dannenhauer. 2023. Memory matters: The need to improve long-term memory in llm-agents. In *Proceedings of the AAAI Symposium Series*, volume 2, pages 277–280.

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

Zexue He, Yu Wang, Churan Zhi, Yuanzhe Hu, Tzu-Ping Chen, Lang Yin, Ze Chen, Tong Arthur Wu, Siru Ouyang, Zihan Wang, Jiaxin Pei, Julian McAuley, Yejin Choi, and Alex Pentland. 2026. Memoryarena: Benchmarking agent memory in interdependent multi-session agentic tasks. *Preprint*, arXiv:2602.16313.

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, and Jacob Steinhardt. 2021a. Measuring coding challenge competence with apps. *Preprint*, arXiv:2105.09938.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

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

Yuanzhe Hu, Yu Wang, and Julian McAuley. 2025. Evaluating memory in LLM agents via incremental multi-turn interactions. In *ICML 2025 Workshop on Long-Context Foundation Models*.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. *Preprint*, arXiv:2310.01798.

Kung-Hsiang Huang, Akshara Prabhakar, Sidharth Dhawan, Yixin Mao, Huan Wang, Silvio Savarese, Caiming Xiong, Philippe Laban, and Chien-Sheng Wu. 2025. Crmarena: Understanding the capacity of llm agents to perform professional crm tasks in realistic environments. *Preprint*, arXiv:2411.02305.

Alexis Huet, Zied Ben Houidi, and Dario Rossi. 2025. Episodic memories generation and evaluation benchmark for large language models. *Preprint*, arXiv:2501.13121.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. *Preprint*, arXiv:2403.07974.

Peter Jansen, Marc-Alexandre Côté, Tushar Khot, Erin Bransom, Bhavana Dalvi Mishra, Bodhisattwa Prasad Majumder, Oyvind Tafjord, and Peter Clark. 2024. Discoveryworld: A virtual environment for developing and evaluating automated scientific discovery agents. *Preprint*, arXiv:2406.06769.

Saurabh Jha, Rohan Arora, Yuji Watanabe, Takumi Yanagawa, Yinfang Chen, Jackson Clark, Bhavya Bhavya, Mudit Verma, Harshit Kumar, Hirokuni Kitahara, et al. 2025. Itbench: Evaluating ai agents across diverse real-world it automation tasks. *arXiv preprint arXiv:2502.05352*.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2023. Swe-bench: Can language models resolve real-world github issues? *ArXiv*, abs/2310.06770.

Raghav Kapoor, Yash Parag Butala, Melisa Russak, Jing Yu Koh, Kiran Kamble, Waseem AlShikh, and Ruslan Salakhutdinov. 2024a. Omniact: A dataset and benchmark for enabling multimodal generalist autonomous agents for desktop and web. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXVIII*, page 161–178, Berlin, Heidelberg. Springer-Verlag.

Sayash Kapoor, Benedikt Stroebl, Zachary S Siegel, Nitya Nadgir, and Arvind Narayanan. 2024b. Ai agents that matter. *arXiv preprint arXiv:2407.01502*.

Aryan Kargwal. 2025. Mastering multi-agent evaluation systems in 2025. *Botpress Blog*.

Ehud Karpas, Omri Abend, Yonatan Belinkov, Barak Lenz, Opher Lieber, Nir Ratner, Yoav Shoham, Hofit Bata, Yoav Levine, Kevin Leyton-Brown, et al. 2022. Mrkl systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning. *arXiv preprint arXiv:2205.00445*.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

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

Harsha Kokel, Michael Katz, Kavitha Srinivas, and Shirin Sohrabi. 2024. Acpbench: Reasoning about action, change, and planning. *arXiv preprint arXiv:2410.05669*.

Alexandre Lacoste, Nicolas Gontier, Oleh Shliazhko, Aman Jaiswal, Kusha Sareen, Shailesh Nanisetty, Joan Cabezas, Manuel Del Verme, Omar G. Younis, Simone Baratta, Matteo Avalle, Imene Kerboua, Xing Han Lù, Elron Bandel, Michal Shmueli-Scheuer, Asaf Yehudai, Leshem Choshen, Jonathan Lebensold, Sean Hughes, Massimo Caccia, Alexandre Drouin, Siva Reddy, Tao Yu, Yu Su, Graham Neubig, and Dawn Song. 2026. Cube: A standard for unifying agent benchmarks. *Preprint*, arXiv:2603.15798.

LangChain. 2025. Agentevals: Evaluating agent trajectories.

Inc LangChain. 2023. Langsmith: Evaluation framework for ai applications.

Langfuse. 2023. Langfuse: Observability for ai applications.

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. 2024. Lab-bench: Measuring capabilities of language models for biology research. *Preprint*, arXiv:2407.10362.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. 2024. A human-inspired reading agent with gist memory of very long contexts. *Preprint*, arXiv:2402.09727.

Yoonjoo Lee, Kyungjae Lee, Sunghyun Park, Dasol Hwang, Jaehyeon Kim, Hong-in Lee, and Moontae Lee. 2023. Qasa: advanced question answering on scientific articles. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

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

Ido Levy, Ben Wiesel, Sami Marreed, Alon Oved, Avi Yaeli, and Segev Shlomov. 2024. St-webagentbench: A benchmark for evaluating safety and trustworthiness in web agents. *arXiv preprint arXiv:2410.06703*.

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

Junlong Li, Wenshuo Zhao, Jian Zhao, Weihao Zeng, Haoze Wu, Xiaochen Wang, Rui Ge, Yuxuan Cao, Yuzhen Huang, Wei Liu, Junteng Liu, Zhaochen Su, Yiyang Guo, Fan Zhou, Lueyang Zhang, Juan Michelini, Xingyao Wang, Xiang Yue, Shuyan Zhou, Graham Neubig, and Junxian He. 2026. The tool decathlon: Benchmarking language agents for diverse, realistic, and long-horizon task execution. In *The Fourteenth International Conference on Learning Representations*.

Lingyu Li, Yixu Wang, Haiquan Zhao, Shuqi Kong, Yan Teng, Chunbo Li, and Yingchun Wang. 2024. Reflection-bench: probing ai intelligence with reflection. *Preprint*, arXiv:2410.16270.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. 2023. Api-bank: A comprehensive benchmark for tool-augmented llms. *Preprint*, arXiv:2304.08244.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.

Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, Tianlin Shi, and Percy Liang. 2018. Reinforcement learning on web interfaces using workflow-guided exploration. *Preprint*, arXiv:1802.08802.

Fengyuan Liu, Nouar AlDahoul, Gregory Eady, Yasir Zaki, and Talal Rahwan. 2025. Self-reflection makes

large language models safer, less biased, and ideologically neutral. *Preprint*, arXiv:2406.10400.

Na Liu, Liangyu Chen, Xiaoyu Tian, Wei Zou, Kaijiang Chen, and Ming Cui. 2024a. From llm to conversational agent: A memory enhanced architecture with fine-tuning of large language models. *ArXiv*, abs/2401.02777.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024b. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. 2023. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*.

Zuxin Liu, Thai Hoang, Jianguo Zhang, Ming Zhu, Tian Lan, Juntao Tan, Weiran Yao, Zhiwei Liu, Yihao Feng, Rithesh RN, et al. 2024c. Apigen: Automated pipeline for generating verifiable and diverse function-calling datasets. *Advances in Neural Information Processing Systems*, 37:54463–54482.

Renze Lou, Hanzi Xu, Sijia Wang, Jiangshu Du, Ryo Kamoi, Xiaoxin Lu, Jian Xie, Yuxuan Sun, Yusen Zhang, Jihyun Janice Ahn, Hongchao Fang, Zhuoyang Zou, Wenchao Ma, Xi Li, Kai Zhang, Congying Xia, Lifu Huang, and Wenpeng Yin. 2025. Aaar-1.0: Assessing ai's potential to assist research. *Preprint*, arXiv:2410.22394.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Felix Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, et al. 2024. Toolsandbox: A stateful, conversational, interactive evaluation benchmark for llm tool use capabilities. *arXiv preprint arXiv:2408.04682*.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Preprint*, arXiv:2209.09513.

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

Xing Han Lù, Zdeněk Kasner, and Siva Reddy. 2024. Weblinx: Real-world website navigation with multi-turn dialogue. *arXiv preprint arXiv:2402.05930*.

Ziyang Luo, Zhiqi Shen, Wenzhuo Yang, Zirui Zhao, Prathyusha Jwalapuram, Amrita Saha, Doyen Sahoo, Silvio Savarese, Caiming Xiong, and Junnan Li. 2025. Mcp-universe: Benchmarking large language models with real-world model context protocol servers. *Preprint*, arXiv:2508.14704.

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

Mike Merrill, Chris Rytting Alex Shaw, Ludwig Schmidt, and Andy Konwinski. 2025. Introducing terminal-bench.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: a benchmark for general ai assistants. *Preprint*, arXiv:2311.12983.

Samuel Miserendino, Michele Wang, Tejal Patwardhan, and Johannes Heidecke. 2025. Swe-lancer: Can frontier llms earn $1 million from real-world freelance software engineering? *arXiv preprint arXiv:2502.12115*.

Niels Mündler, Mark Müller, Jingxuan He, and Martin Vechev. 2024. Swt-bench: Testing and validating real-world bug-fixes with code agents. *Advances in Neural Information Processing Systems*, 37:81857–81887.

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

Deepak Nathani, Lovish Madaan, Nicholas Roberts, Nikolay Bashlykov, Ajay Menon, Vincent Moens, Amar Budhiraja, Despoina Magka, Vladislav Vorotilov, Gaurav Chaurasia, et al. 2025. Mlgym: A new framework and benchmark for advancing ai research agents. *arXiv preprint arXiv:2502.14499*.

OpenAI. 2024. Introducing swe-bench verified. urlhttps://openai.com/index/introducing-swe-bench-verified/.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2024. Memgpt: Towards llms as operating systems. *Preprint*, arXiv:2310.08560.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa : A large-scale multi-subject multi-choice dataset for medical domain question answering. *Preprint*, arXiv:2203.14371.

Jane Pan, Ryan Shar, Jacob Pfau, Ameet Talwalkar, He He, and Valerie Chen. 2025. When benchmarks talk: Re-evaluating code llms with interactive feedback. *Preprint*, arXiv:2502.18413.

Jiayi Pan, Xingyao Wang, Graham Neubig, Navdeep Jaitly, Heng Ji, Alane Suhr, and Yizhe Zhang. 2024a. Training software engineering agents and verifiers with swe-gym. *arXiv preprint arXiv:2412.21139*.

Yichen Pan, Dehan Kong, Sida Zhou, Cheng Cui, Yifei Leng, Bing Jiang, Hangyu Liu, Yanyi Shang, Shuyan Zhou, Tongshuang Wu, et al. 2024b. Webcanvas: Benchmarking web agents in online environments. *arXiv preprint arXiv:2406.12373*.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, et al. 2021. Quality: Question answering with long input texts, yes! *arXiv preprint arXiv:2112.08608*.

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

Liana Patel, Negar Arabzadeh, Harshit Gupta, Ankita Sundar, Ion Stoica, Matei Zaharia, and Carlos Guestrin. 2025. Deepscholar-bench: A live benchmark and automated evaluation for generative research synthesis. *Preprint*, arXiv:2508.20033.

Shishir G. Patil, Huanzhi Mao, Charlie Cheng-Jie Ji, Fanjia Yan, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. 2025a. The berkeley function calling leaderboard (bfcl): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*.

Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. 2025b. Gorilla: Large language model connected with massive apis. *Advances in Neural Information Processing Systems*, 37:126544–126565.

Patronus AI, Inc. 2023. Patronus ai: Automated testing and evaluation platform for generative ai applications.

Yun Peng, Shuqing Li, Wenwei Gu, Yichen Li, Wenxuan Wang, Cuiyun Gao, and Michael Lyu. 2021. Revisiting, benchmarking and exploring api recommendation: How far are we? *Preprint*, arXiv:2112.12653.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*.

Matthew Renze and Erhan Guven. 2024a. The benefits of a concise chain of thought on problem-solving in large language models. In *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, page 476–483. IEEE.

Matthew Renze and Erhan Guven. 2024b. Self-reflection in llm agents: Effects on problem-solving performance. *Preprint*, arXiv:2405.06682.

Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J. Maddison, and Tatsunori Hashimoto. 2024. Identifying the risks of lm agents with an lm-emulated sandbox. *Preprint*, arXiv:2309.15817.

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

Tianlin Shi, Andrej Karpathy, Linxi Fan, Jonathan Hernandez, and Percy Liang. 2017. World of bits: An open-domain platform for web-based agents. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3135–3144. PMLR.

Noah Shinn, Federico Cassano, Beck Labash, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Neural Information Processing Systems*.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. Alfworld: Aligning text and embodied environments for interactive learning. *Preprint*, arXiv:2010.03768.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. 2025. Can LLMs generate novel research ideas? a large-scale human study with 100+ NLP researchers. In *The Thirteenth International Conference on Learning Representations*.

Zachary S. Siegel, Sayash Kapoor, Nitya Nagdir, Benedikt Stroebl, and Arvind Narayanan. 2024. Core-bench: Fostering the credibility of published research through a computational reproducibility agent benchmark. *Preprint*, arXiv:2409.11363.

Shruti Singh, Nandan Sarkar, and Arman Cohan. 2024. SciDQA: A deep reading comprehension dataset over scientific papers. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

Yifan Song, Weimin Xiong, Dawei Zhu, Wenhao Wu, Han Qian, Mingbo Song, Hailiang Huang, Cheng Li, Ke Wang, Rong Yao, et al. 2023. Restgpt: Connecting large language models with real-world restful apis. *arXiv preprint arXiv:2306.06624*.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. 2023. Musr: Testing the limits of chain-of-thought with multistep soft reasoning. *arXiv preprint arXiv:2310.16049*.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. 2025. Paperbench: Evaluating ai's ability to replicate ai research. *Preprint*, arXiv:2504.01848.

Katharina Stein, Daniel Fišer, Jörg Hoffmann, and Alexander Koller. 2023. Autoplanbench: Automatically generating benchmarks for llm planners from pddl. *arXiv preprint arXiv:2311.09830*.

Benedikt Stroebl, Sayash Kapoor, and Arvind Narayanan. 2025. Hal: A holistic agent leaderboard for centralized and reproducible agent evaluation. https://github.com/princeton-pli/hal-harness.

Haotian Sun, Yuchen Zhuang, Lingkai Kong, Bo Dai, and Chao Zhang. 2023. Adaplanner: Adaptive planning from feedback with language models. *Preprint*, arXiv:2305.16653.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

SWE-bench Lite. 2024. Swe-bench lite. urlhttps://www.swebench.com/lite.html.

Haoran Tan, Zeyu Zhang, Chen Ma, Xu Chen, Quanyu Dai, and Zhenhua Dong. 2025. MemBench: Towards more comprehensive evaluation on the memory of LLM-based agents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19336–19352, Vienna, Austria. Association for Computational Linguistics.

Qiaoyu Tang, Ziliang Deng, Hongyu Lin, Xianpei Han, Qiao Liang, Boxi Cao, and Le Sun. 2023. Toolalpaca: Generalized tool learning for language models with 3000 simulated cases. *arXiv preprint arXiv:2306.05301*.

Nexusflow.ai team. 2023. Nexusraven-v2: Surpassing gpt-4 for zero-shot function calling.

Minyang Tian, Luyu Gao, Dylan Zhang, Xinan Chen, Cunwei Fan, Xuefei Guo, Roland Haas, Pan Ji, Kittithat Krongchon, Yao Li, Shengyan Liu, Di Luo, Yutao Ma, HAO TONG, Kha Trinh, Chenyu Tian, Zihan Wang, Bohao Wu, Shengzhu Yin, Minhui Zhu, Kilian Lieret, Yanxin Lu, Genglin Liu, Yufeng Du, Tianhua Tao, Ofir Press, Jamie Callan, Eliu A Huerta, and Hao Peng. 2024. Scicode: A research coding benchmark curated by scientists. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. 2024. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16022–16076, Bangkok, Thailand. Association for Computational Linguistics.

Ada Defne Tur, Nicholas Meade, Xing Han Lù, Alejandra Zambrano, Arkil Patel, Esin DURMUS, Spandana Gella, Karolina Stanczak, and Siva Reddy. 2025. Safearena: Evaluating the safety of autonomous web agents. In *Forty-second International Conference on Machine Learning*.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2023. Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change. *Advances in Neural Information Processing Systems*, 36:38975–38987.

David Wadden, Kejian Shi, Jacob Morrison, Aakanksha Naik, Shruti Singh, Nitzan Barzilay, Kyle Lo, Tom Hope, Luca Soldaini, Shannon Zejiang Shen, Doug Downey, Hannaneh Hajishirzi, and Arman Cohan. 2024. Sciriff: A resource to enhance language model instruction-following over scientific literature. *Preprint*, arXiv:2406.07835.

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

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022a. ScienceWorld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11279–11298, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. 2023. Mint: Evaluating llms in multi-turn interaction with tools and language feedback. *arXiv preprint arXiv:2309.10691*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022b. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Zixuan Wang, Bo Yu, Junzhe Zhao, Wenhao Sun, Sai Hou, Shuai Liang, Xing Hu, Yinhe Han, and Yiming Gan. 2024b. Karma: Augmenting embodied ai agents with long-and-short term memory systems. *Preprint*, arXiv:2409.14908.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. 2025. Browsecomp: A simple yet challenging benchmark for browsing agents. *Preprint*, arXiv:2504.12516.

Weights&Biases. 2023. W&b evaluations: Tools for evaluating llm applications and agents. https://wandb.ai/site/evaluations/. Accessed: 2025-05-11.

Cheng-Kuang Wu, Zhi Rui Tam, Chieh-Yen Lin, Yun-Nung Chen, and Hung-yi Lee. 2024a. Streambench: Towards benchmarking continuous improvement of language agents. *arXiv preprint arXiv:2406.08747*.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2025. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *Preprint*, arXiv:2410.10813.

Mengsong Wu, Tong Zhu, Han Han, Chuanyuan Tan, Xiang Zhang, and Wenliang Chen. 2024b. Seal-tools: Self-instruct tool learning dataset for agent tuning and detailed benchmark. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 372–384. Springer.

Chun Xia, Yinlin Deng, Soren Dunn, and Lingming Zhang. 2024. Agentless: Demystifying llm-based software engineering agents. *ArXiv*, abs/2407.01489.

Ruixuan Xiao, Wentao Ma, Ke Wang, Yuchuan Wu, Junbo Zhao, Haobo Wang, Fei Huang, and Yongbin Li. 2024. Flowbench: Revisiting and benchmarking workflow-guided planning for llm-based agents. *arXiv preprint arXiv:2406.14884*.

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. 2024. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Frank F. Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Z. Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, Mingyang Yang, Hao Yang Lu, Amaad Martin, Zhe Su, Leander Maben, Raj Mehta, Wayne Chi, Lawrence Jang, Yiqing Xie, Shuyan Zhou, and Graham Neubig. 2024. Theagentcompany: Benchmarking llm agents on consequential real world tasks. *Preprint*, arXiv:2412.14161.

Jing Xu, Arthur Szlam, and Jason Weston. 2021. Beyond goldfish memory: Long-term open-domain conversation. *arXiv preprint arXiv:2107.07567*.

Qiantong Xu, Fenglu Hong, Bo Li, Changran Hu, Zhengyu Chen, and Jian Zhang. 2023. On the tool manipulation capability of open-source large language models. *arXiv preprint arXiv:2305.16504*.

Tianze Xu, Pengrui Lu, Lyumanshan Ye, Xiangkun Hu, and Pengfei Liu. 2025a. Researcherbench: Evaluating deep ai research systems on the frontiers of scientific inquiry. *arXiv preprint arXiv:2507.16280*.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025b. A-mem: Agentic memory for llm agents. *Preprint*, arXiv:2502.12110.

Tianci Xue, Weijian Qi, Tianneng Shi, Chan Hee Song, Boyu Gou, Dawn Song, Huan Sun, and Yu Su. 2025. An illusion of progress? assessing the current state of web agents. *arXiv preprint arXiv:2504.01382*.

Fanjia Yan, Huanzhi Mao, Charlie Cheng-Jie Ji, Tianjun Zhang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2024. Berkeley function calling leaderboard. https://gorilla.cs.berkeley.edu/blogs/8_berkeley_function_calling_leaderboard.html.

John Yang, Carlos E. Jimenez, Alex L. Zhang, Kilian Lieret, Joyce Yang, Xindi Wu, Ori Press, Niklas Muennighoff, Gabriele Synnaeve, Karthik R. Narasimhan, Diyi Yang, Sida I. Wang, and Ofir Press. 2024. Swe-bench multimodal: Do ai systems generalize to visual software domains? *ArXiv*, abs/2410.03859.

John Yang, Kilian Lieret, Carlos E. Jimenez, Alexander Wettig, Kabir Khandpur, Yanzhe Zhang, Binyuan Hui, Ofir Press, Ludwig Schmidt, and Diyi Yang. 2025. Swe-smith: Scaling data for software engineering agents. *Preprint*, arXiv:2504.21798.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *Preprint*, arXiv:1809.09600.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022a. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *Preprint*, arXiv:2406.12045.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022b. React: Synergizing reasoning and acting in language models. *ArXiv*, abs/2210.03629.

Ori Yoran, Samuel Joseph Amouyal, Chaitanya Malaviya, Ben Bogin, Ofir Press, and Jonathan Berant. 2024. Assistantbench: Can web agents solve realistic and time-consuming tasks? *arXiv preprint arXiv:2407.15711*.

Jiaxuan You, Mingjie Liu, Shrimai Prabhumoye, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2024. LLM-evolve: Evaluation for

LLM's evolving capability on benchmarks. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16937–16942, Miami, Florida, USA. Association for Computational Linguistics.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. *arXiv preprint arXiv:1809.08887*.

Daoguang Zan, Zhirong Huang, Ailun Yu, Shaoxin Lin, Yifan Shi, Wei Liu, Dong Chen, Zongshuai Qi, Hao Yu, Lei Yu, et al. 2024. Swe-bench-java: A github issue resolving benchmark for java. *arXiv preprint arXiv:2408.14354*.

Yinger Zhang, Shutong Jiang, Renhao Li, Jianhong Tu, Yang Su, Lianghao Deng, Xudong Guo, Chenxu Lv, and Junyang Lin. 2026. Deepplanning: Benchmarking long-horizon agentic planning with verifiable constraints. *Preprint*, arXiv:2601.18137.

Ziniu Zhang, Shulin Tian, Liangyu Chen, and Ziwei Liu. 2024. Mmina: Benchmarking multihop multimodal internet agents. *arXiv preprint arXiv:2404.09992*.

Huaixiu Steven Zheng, Swaroop Mishra, Hugh Zhang, Xinyun Chen, Minmin Chen, Azade Nova, Le Hou, Heng-Tze Cheng, Quoc V Le, Ed H Chi, et al. 2024. Natural plan: Benchmarking llms on natural language planning. *arXiv preprint arXiv:2406.04520*.

Lucen Zhong, Zhengxiao Du, Xiaohan Zhang, Haiyi Hu, and Jie Tang. 2025. Complexfuncbench: Exploring multi-step and constrained function calling under long-context scenario. *arXiv preprint arXiv:2501.10132*.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, et al. 2021. Qmsum: A new benchmark for query-based multi-domain meeting summarization. *arXiv preprint arXiv:2104.05938*.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. Agieval: A human-centric benchmark for evaluating foundation models. *Preprint*, arXiv:2304.06364.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. 2023. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*.

Yuxuan Zhu, Tengjun Jin, Yada Pruksachatkun, Andy K Zhang, Shu Liu, Sasha Cui, Sayash Kapoor, Shayne Longpre, Kevin Meng, Rebecca Weiss, Fazl Barez, Rahul Gupta, Jwala Dhamala, Jacob Merizian, Mario Giulianelli, Harry Coppock, Cozmin Ududec, Antony Kellermann, Jasjeet S Sekhon, Jacob Steinhardt, Sarah Schwettmann, Arvind Narayanan, Matei Zaharia, Ion Stoica, Percy Liang, and Daniel Kang. 2025. Establishing best practices in building rigorous agentic benchmarks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Mingchen Zhuge, Changsheng Zhao, Dylan Ashley, Wenyi Wang, Dmitrii Khizbullin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, et al. 2024. Agent-as-a-judge: Evaluate agents with agents. *arXiv preprint arXiv:2410.10934*.

A Literature Review Methodology

To ensure the comprehensiveness and validity of this survey, we employed a rigorous, multi-stage literature review process designed to capture the rapidly evolving landscape of LLM agent evaluation. Our methodology consisted of three primary phases: systematic search, structured selection, and expert validation.

Search Strategy and Citation Chasing Our initial data collection involved a systematic search across major academic repositories and archives, including Google Scholar, the ACL Anthology, HuggingFace Papers, and arXiv. We utilized a targeted set of keywords and their combinations, such as "LLM agent evaluation," "agent benchmark," "tool use evaluation," and "web agent benchmark." Following the identification of seminal and highly cited works, we applied an iterative "snowballing" technique. We performed both forward and backward citation chasing on these seed papers to uncover foundational prior art as well as emerging methodologies that might not yet be indexed by standard keywords.

Inclusion and Exclusion Criteria We defined strict criteria to maintain the survey's focus and quality. Papers were *included* if they introduced a novel benchmark, an evaluation framework, or a significant methodological contribution to the assessment of LLM-based agents. Conversely, we *excluded* works that: 1. Proposed new agent architectures without a distinct contribution to evaluation methodology. 2. Focused solely on traditional LLM evaluation (e.g., reasoning on static datasets), lacking dynamic, agentic, or interactive components. While acceptance to top-tier conferences was utilized as a primary indicator of quality, we also included high-impact preprints to ensure the survey reflects the most recent advancements in this fast-paced field.

Expert Consultation To mitigate coverage gaps and ensure an accurate representation of specific subdomains, we consulted with domain experts corresponding to the key categories in our taxonomy (e.g., SWE and web agent researchers). We specifically engaged with creators of leading benchmarks and dominant agent architectures to discuss the selected manuscripts. This validation step ensured that our survey captures the nuances of state-of-the-art evaluation protocols.

B Agent Capabilities Evaluation

In this appendix, we expand on the evaluation of LLM-based agent capabilities. While the main paper presents the core information, here we offer a more detailed account of each benchmark, its interrelations, and how the evaluation of each capability evolves.

B.1 Planning and Multi-Step Reasoning

Planning and multi-step reasoning form the foundation of an LLM agent's ability to tackle complex tasks effectively which enables agents to decompose problems into smaller, more manageable subtasks and create strategic execution paths toward solutions (Gao et al., 2023a).

Multi-step reasoning in LLMs typically involves executing sequential logical operations—typically requiring 3-10 intermediate steps—to arrive at solutions that cannot be derived through single-step inference (Cobbe et al., 2021; Yang et al., 2018; Suzgun et al., 2022). This foundational need for multi-step planning has led to the development of specialized benchmarks and evaluation frameworks that systematically assess these capabilities across diverse domains, including: mathematical reasoning (GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021b), AQUA-RAT (Ling et al., 2017)), multi-hop question answering (HotpotQA (Yang et al., 2018), StrategyQA (Geva et al., 2021), MultiRC (Khashabi et al., 2018)), scientific reasoning (ARC (Clark et al., 2018)), logical reasoning (FOLIO, P-FOLIO (Han et al., 2024, 2022)) constraint satisfaction puzzles (Game of 24 (Yao et al., 2023)), everyday common sense (MUSR (Sprague et al., 2023)), and challenging reasoning tasks (BBH (Suzgun et al., 2022)). Several of these benchmarks, particularly HotpotQA, ALFWorlds, and Game of 24, have been specifically adapted for evaluating agent-based approaches like ReAct, where planning and calling the tools proposed by the agent are interleaved in interactive problem-solving settings.

Recent work has developed more specialized frameworks targeting LLM planning capabilities. ToolEmu (Ruan et al., 2024) introduces a simulator-based approach for evaluating tool-using agents, revealing that successful planning requires explicit state tracking and the ability to recover from errors. The MINT benchmark (Wang et al., 2023) evaluates planning in interactive environments, demonstrating that even advanced LLMs

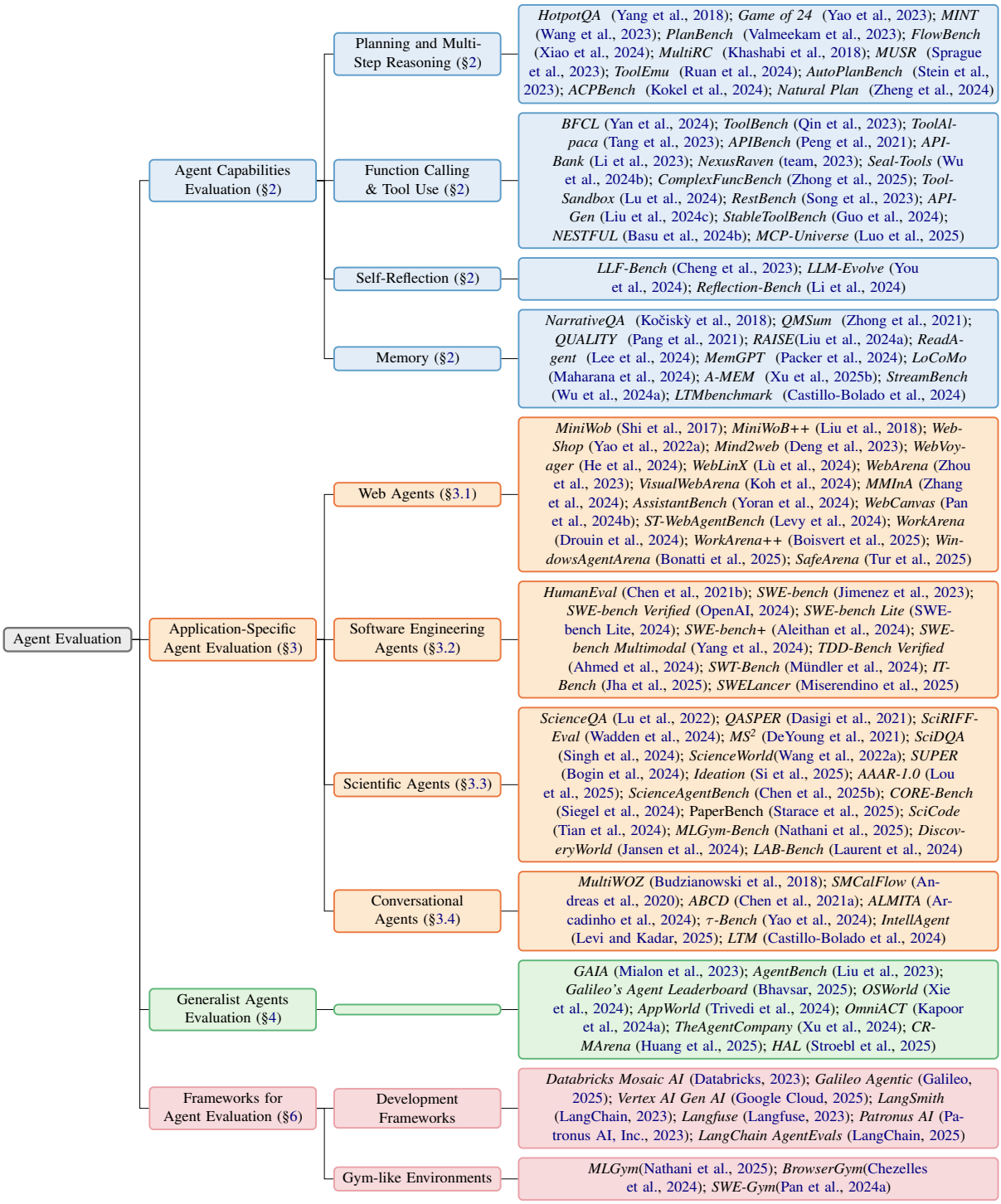

Figure 2: Overview of the paper.

struggle with long-horizon tasks that require multiple steps.

PlanBench (Valmeekam et al., 2023) provides a comprehensive evaluation framework specifically designed to assess planning capabilities in LLM agents across diverse domains, revealing that current models excel at short-term tactical planning but struggle with strategic long-horizon planning. Complementing this, AutoPlanBench (Stein et al., 2023) focuses on evaluating planning in everyday scenarios, demonstrating that even SoTA LLM agents lag behind classical symbolic planners.

FlowBench (Xiao et al., 2024) evaluates workflow planning abilities, with a focus on expertise-intensive tasks. ACPBench (Kokel et al., 2024) focuses on evaluating LLMs on core reasoning skills.

The Natural Plan benchmark (Zheng et al., 2024) is designed to evaluate how LLMs handle real-world planning tasks presented in natural language. SoTA LLM agents perform poorly on this benchmark, particularly as complexity increases.

These benchmarks highlight key abilities essential for effective agent planning: (1) task decomposition for breaking down complex problems, (2) state tracking and belief maintenance for accurate multi-step reasoning, (3) self-correction to detect and recover from errors, (4) causal understanding to predict action outcomes, and (5) meta-planning to refine planning strategies.

B.2 Function Calling & Tool Use

The ability of LLMs to interact with external tools through function calling is fundamental for building intelligent agents capable of delivering real-time, contextually accurate responses (Qin et al., 2023; Tang et al., 2023). Early works utilized targeted tools, such as retrieval in approaches by augmented language models with retrieval capabilities (Lewis et al., 2020; Gao et al., 2023b; Nakano et al., 2021). Later developments included more general-purpose tools, exemplified by ToolFormer (Schick et al., 2023), Chameleon (Lu et al., 2023), and MRKL (Karpas et al., 2022).

Function calling involves several sub-tasks that work together seamlessly. Intent recognition identifies when a function is needed based on user requests. Function selection determines the most appropriate tool for the task. Parameter-value-pair mapping extracts relevant arguments from the conversation and assigns them to function parameters. Function execution invokes the selected function with those parameters to interact with external systems. Finally, response generation processes the function output and incorporates it into the LLM's reply to the user. This integrated process ensures accurate and efficient function calling within the LLM's workflow.

Early evaluation efforts offered approaches to evaluate the above sub-tasks while focusing on relatively simple, one-step interactions with explicitly provided parameters. Benchmarks such as ToolAlpaca (Tang et al., 2023), APIBench (Patil et al., 2025b), ToolBench (Qin et al., 2023), and the Berkeley Function Calling Leaderboard v1 (BFCL) (Yan et al., 2024) exemplify this phase, employing synthetic datasets and rule-based matching (e.g., via Abstract Syntax Trees) to establish baseline metrics like pass rates and structural accuracy. However, these methods were limited in capturing the complexities of real-world scenarios, which might include multistep conversations, parameters that are not explicitly mentioned in the conversation, and tools with complex input structures and long, intricate outputs. BFCL v2 and v3 address these gaps by adding organizational tools and integrated multi-turn, multi-step evaluation logic, offering a more realistic simulation and highlighting the need for continuous state management.

Complementing this evolution, several benchmarks have broadened the evaluation landscape. For example, ToolSandbox (Lu et al., 2024) differs from previous benchmarks by incorporating stateful tool execution, implicit state dependencies, on-policy conversational evaluation with a built-in user simulator, and dynamic evaluation strategies for intermediate and final milestones across arbitrary trajectories. Seal-Tools (Wu et al., 2024b) adopts a self-instruct (Wang et al., 2022b) methodology to generate nested tool calls, effectively modeling layered and interdependent interactions. In parallel, API-Bank (Li et al., 2023) emphasizes realistic API engagements by utilizing dialogue-based evaluations and extensive training datasets. Frameworks like NexusRaven (team, 2023) further enrich this landscape by focusing on generalized tool-use scenarios that mirror the diverse challenges encountered in practice. API-Blend (Basu et al., 2024a) suggested a comprehensive approach focusing on identifying, curating, and transforming existing datasets into a large corpus for training and systematic testing of tool-augmented LLMs. API-Blend mimics real-world scenarios involving API tasks such as API/tool detection, slot filling, and sequencing of detected APIs, providing utility for both training and benchmarking purposes. RestBench (Song et al., 2023) facilitates exploration of utilizing multiple APIs to address complex real-world user instructions. APIGen (Liu et al., 2024c) provides a comprehensive automated data generation pipeline that synthesizes high-quality function-calling datasets verified through hierarchical stages. StableToolBench (Guo et al., 2024) addresses the challenges of function-calling evaluation by introducing a virtual API server with caching and simulators to alleviate API status changes.

Addressing the inherent complexity of multi-step interactions, ComplexFuncBench (Zhong et al., 2025) was specifically designed to assess scenarios requiring implicit parameter inference, adherence to user-defined constraints, and efficient

long-context processing. NESTFUL (Basu et al., 2024b) focuses on adding complexity by evaluating LLMs on nested sequences of API calls where outputs from one call serve as inputs to subsequent calls.

B.3 Self-Reflection

An emerging line of research focuses on whether agents can self-reflect and improve their reasoning through interactive feedback, thereby reducing errors in multi-step interactions. This requires the model to understand the feedback and dynamically update its beliefs to carry out adjusted actions or reasoning steps over extensive trajectories.

Early efforts to gauge LLM agent self-reflection were often indirect, repurposing existing reasoning or planning tasks, such as AGIEval (Zhong et al., 2023), MedMCQA (Pal et al., 2022), ALFWorld (Shridhar et al., 2021), MiniWoB++ (Liu et al., 2018), etc., into multi-turn feedback loops, to see if models could recognize or correct their own errors given external feedback in confined settings (Renze and Guven, 2024b; Huang et al., 2024; Shinn et al., 2023; You et al., 2024; Sun et al., 2023; Liu et al., 2025). Improvement was typically measured by determining if the final answer was corrected, providing only a coarse evaluation and potentially ill-defined measurement, as observed improvements may depend on specific prompting techniques lacking proper standardization (Huang et al., 2024; Liu et al., 2025).

LLF-Bench (Cheng et al., 2023) was introduced to standardize benchmarks for interactive self-reflection. It includes diverse decision-making tasks and treats task instructions as part of the environment. To reduce overfitting, it allows randomization of task instruction descriptions and agent feedback.

Similarly, LLM-Evolve (You et al., 2024) was introduced to evaluate LLM agents' self-reflection capabilities on standard benchmarks such as MMLU (Hendrycks et al., 2020). This approach evaluates agents based on past experiences by collecting previous queries with feedback and extracting them as in-context demonstrations. To provide more granular insights into different feedback types, (Pan et al., 2025) focused specifically on coding agents, extending existing coding benchmarks like APPS (Hendrycks et al., 2021a) and LiveCodeBench (Jain et al., 2024) to interactive settings.

From a cognitive science perspective, ReflectionBench (Li et al., 2024) was designed to assess LLMs' cognitive reflection capabilities, breaking down reflection into components like perception of new information, memory usage, belief updating following surprise, decision-making adjustments, counterfactual reasoning, and meta-reflection.

B.4 Memory

Memory mechanisms in LLM-based agents improve their handling of long contexts and information retrieval, overcoming static knowledge limits and supporting reasoning and planning in dynamic scenarios (Park et al., 2023). Unlike tool use, which connects agents to external resources, memory ensures context retention for extended interactions like processing documents or maintaining conversations. Agents rely on short-term memory for real-time responses and long-term memory for deeper understanding and applying knowledge over time. Together, these memory systems allow LLM-based agents to adapt, learn, and make well-informed decisions in tasks requiring persistent information access.

One prominent line of research focuses on addressing the challenge of limited context lengths in LLMs by incorporating memory mechanisms to enhance reasoning and retrieval across extended contexts and conversations. Recent works, such as ReadAgent (Lee et al., 2024), MemGPT (Packer et al., 2024), and A-MEM (Xu et al., 2025b), investigate these methods and evaluate their efficacy through reasoning and retrieval metrics.

Specifically, ReadAgent structures reading by grouping content, condensing episodes into memories, and retrieving passages, with effectiveness shown on datasets like QUALITY (Pang et al., 2021), NarrativeQA (Kočiskỳ et al., 2018), and QMSum (Zhong et al., 2021).

Similarly, A-MEM introduces an advanced memory architecture evaluated using the LoCoMo benchmark (Maharana et al., 2024), while MemGPT manages a tiered memory system tested on NaturalQuestions-Open (Liu et al., 2024b) and multi-session chat datasets (Xu et al., 2021).

For episodic memory evaluation, (Huet et al., 2025) proposes a specialized benchmark to assess how LLMs generate and manage memories that capture specific events with contextual details. This benchmark utilizes synthetically created book chapters and events with LLMs-based judge evaluation metrics to measure accuracy and relevance. StreamBench (Wu et al., 2024a) represents a more challenging setting, evaluating how agents lever-

age external memory components—including the memory of previous interactions and external feedback—to continuously improve performance over time, with quality and efficiency assessed across diverse datasets including text-to-SQL tasks (e.g., Spider (Yu et al., 2018)), ToolBench (Xu et al., 2023), and HotpotQA (Yang et al., 2018).

Beyond context length optimization, memory mechanisms also enhance real-time decision-making and learning in agent settings, focusing on action optimization (Liu et al., 2024a; Shinn et al., 2023; Wang et al., 2024b). For example, Reflexion (Shinn et al., 2023) tracks success rate on tasks like HotPotQA (Yang et al., 2018) and ALFWorld (Shridhar et al., 2021), while RAISE (Liu et al., 2024a) enhances the ReAct framework with a two-part memory system evaluated through human judgment on quality metrics and efficiency. Similarly, KARMA (Wang et al., 2024b) tests memory in household tasks using metrics such as success rate, retrieval accuracy, and memory hit rate, demonstrating how memory mechanisms significantly improve agent performance across diverse domains requiring complex reasoning and persistent information retention. LTM-benchmark (Castillo-Bolado et al., 2024) evaluates conversational agents through extended, multi-task interactions with frequent context switching to test long-term memory and information integration capabilities. The results demonstrate that while LLMs generally perform well in single-task scenarios, they struggle with interleaved tasks, and interestingly, short-context LLMs equipped with long-term memory systems can match or exceed the performance of models with larger context windows.

C Application-Specific Agents Evaluation

C.1 Scientific Agents

Scientific frameworks Complementing task-level benchmarks, unified evaluation frameworks are introduced to assess agents across sequential, end-to-end research workflows. These efforts range from gym-style workflow simulators (Nathani et al., 2025) and virtual discovery environments (Jansen et al., 2024) to domain-focused suites like LAB-Bench (Laurent et al., 2024) for biologically orientated evaluation. Such frameworks enable holistic measurement of experimental design, iterative hypothesis refinement, and capabilities central to autonomous, innovative scientific discovery.

Deep Research Recently, research agents at the intersection of search, web, and scientific agents have become common, due to a wide commercial offering. Such agents retrieve and synthesize information before returning comprehensive citation-backed answers. This process poses a difficult evaluation task with multi-step, multi-faceted, time-insensitive consideration. Patel et al. (2025) proposed evaluating those agents based on three core functions: retrieval quality, knowledge synthesis, and verifiability. Gou et al. (2025) utilized agent-as-a-judge for evaluation, and shows agents achieve $50-70\%$ of human performance. This is an active research field with many works (Wei et al., 2025; Chen et al., 2025a; Xu et al., 2025a; Bosse et al., 2025; Du et al., 2025).

D Generalist Agent Evaluation

Realistic Workplaces Benchmarks Another venue of generalist agent evaluation focuses on realistic workplaces with different roles and personas. TheAgentCompany (Xu et al., 2024) simulates a software company where agents must browse internal sites, write code, and collaborate. CRMArena (Huang et al., 2025) replicates a Customer Relationship Management environment, requiring agents to use UI and APIs, follow domain policies, and integrate diverse data to complete enterprise-level tasks.

E Benchmark Recommendations

Navigating the extensive landscape of agent evaluation is a prerequisite for effective benchmarking. In this section, we distill our survey findings into actionable recommendations for practitioners and researchers. Our selection criteria prioritize community adoption, active maintenance status, and the reporting standards observed in recent literature from leading LLM and agent developers.

E.1 Web Agents

The choice of benchmark in this domain depends heavily on the environment's dynamicity and the agent's modality.

- **Dynamic Interaction: WebArena** remains the leading option for agents operating in dynamic environments. We note that as of April 19, 2026, top performance has reached 74.3%, based on a submission from February 2026,

suggesting there is still room for improvement.

- **Static Evaluation:** For offline evaluation using cached traces, **Mind2Web** remains the standard.
- **Online & Multimodal:** For agents heavily reliant on visual signals, **WebVoyager** is the recommended choice. Researchers seeking updated, reactive environments should also consider the new but promising **Mind2Web-Live** and **Online-Mind2Web**.

E.2 Software Engineering (SWE) Agents

SWE-bench Verified continues to serve as the dominant gold standard for evaluating coding agents. However, with top performances now reaching approximately 80%, suggesting it is close to being saturated.

- **Higher Difficulty:** We recommend **SWE-bench-pro** (Scale) as a more rigorous alternative, where current state-of-the-art (SOTA) performance hovers around 46%.
- **Specialized Contexts: SWE-Lancer** is valuable for evaluating freelance-style task completion, while **Terminal Bench** is essential for agents specializing in command-line interface (CLI) interactions.
- **Multi options:** Multilingual and multi-modal focused benchmarks are mentioned in the paper.

E.3 Scientific Agents

Evaluation in scientific domains is highly task-dependent. In the main paper, we try to mention the best options for each scientific task.

E.4 Conversational Agents

For agents requiring robust user simulation and tool usage in dialogue, τ**-bench** is the community standard. While current models achieve high success rates on this benchmark, it remains the most common option, and no widely adopted alternative has yet emerged to displace it.

E.5 Generalist Agent Evaluation

For general-purpose agents, the recommendation varies by the task focus and the agent's interaction interface:

- **Reasoning Focus: GAIA** remains the primary recommendation for testing general logical reasoning and tool selection.
- **Tools Interface: AppWorld** is the preferred environment for agents that interact via coding or structured tool calling.
- **GUI Interface:** For agents interacting via User Interface, **OS-World** is the standard.
- **Holistic Evaluation:** For cross-benchmark assessment, we recommend starting with the **HAL** (Holistic Agent Leaderboard) framework to standardize comparisons across these disparate domains.