

# Evaluation and Benchmarking of LLM Agents: A Survey

Mahmoud Mohammadi  
 mohmoud.mohammadi@sap.com  
 SAP Labs  
 Bellevue, WA, USA

Jane Lo  
 jane.lo@sap.com  
 SAP Labs  
 Palo Alto, CA, USA

Yipeng Li  
 yipeng.li@sap.com  
 SAP Labs  
 Bellevue, WA, USA

Wendy Yip  
 wendy.yip@sap.com  
 SAP Labs  
 Palo Alto, CA, USA

## Abstract

The rise of LLM-based agents has opened new frontiers in AI applications, yet evaluating these agents remains a complex and underdeveloped area. This survey provides an in-depth overview of the emerging field of LLM agent evaluation, introducing a two-dimensional taxonomy that organizes existing work along (1) evaluation objectives—what to evaluate, such as agent behavior, capabilities, reliability, and safety—and (2) evaluation process—how to evaluate, including interaction modes, datasets and benchmarks, metric computation methods, and tooling. In addition to taxonomy, we highlight enterprise-specific challenges, such as role-based access to data, the need for reliability guarantees, dynamic and long-horizon interactions, and compliance, which are often overlooked in current research. We also identify the future research directions, including holistic, more realistic, and scalable evaluation. This work aims to bring clarity to the fragmented landscape of agent evaluation and provide a framework for systematic assessment, enabling researchers and practitioners to evaluate LLM agents for real-world deployment.

## CCS Concepts

- Computing methodologies → Natural language processing;
- Software and its engineering → Software verification and validation;
- Human-centered computing → Human computer interaction (HCI).

## Keywords

LLM Agents; Agent Evaluation; Evaluation Taxonomy; Agent Behavior, Benchmarks, Safety; Enterprise AI

## ACM Reference Format:

Mahmoud Mohammadi, Yipeng Li, Jane Lo, and Wendy Yip. 2025. Evaluation and Benchmarking of LLM Agents: A Survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada*. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3711896.3736570>



This work is licensed under a Creative Commons Attribution 4.0 International License.  
*KDD '25, Toronto, ON, Canada*

© 2025 Copyright held by the owner/author(s).  
 ACM ISBN 979-8-4007-1454-2/2025/08  
<https://doi.org/10.1145/3711896.3736570>

## 1 Introduction

Agents based on LLMs are autonomous or semi-autonomous systems that use LLMs to reason, plan, and act, and represent a rapidly growing frontier in artificial intelligence [69, 105]. From customer service bots and coding copilots to digital assistants, LLM agents are redefining how we build intelligent systems.

As these agents move from research prototypes to real-world applications [23, 62], the question of how to rigorously evaluate them becomes both pressing and complex. However, evaluating LLM agents is more complex than evaluating LLMs in isolation. Unlike LLMs, which are primarily assessed for text generation or question answering, LLM agents operate in dynamic, interactive environments. They reason and make plans, execute tools, leverage memory, and even collaborate with humans or other agents [20]. This complex behavior and dependence on real-world effects make standard LLM evaluation approaches insufficient. To make an analogy, LLM evaluation is like examining the performance of an engine. In contrast, agent evaluation assesses a car's performance comprehensively, as well as under various driving conditions.

LLM agent evaluation also differs from traditional software evaluation. While software testing focuses on deterministic and static behavior, LLM agents are inherently probabilistic and behave dynamically; therefore, they require new approaches to assessing their performance. The evaluation of LLM agents is at the intersection of natural language processing (NLP), human-computer interaction (HCI), and software engineering, which demands additional perspectives.

Despite increasing interest in this space, existing surveys focus narrowly on LLM evaluation or cover specific agent capabilities without a holistic perspective [121]. In addition, enterprise applications bring additional requirements to agents, including secure access to data and systems, a high degree of reliability for audit and compliance purposes, and more complex interaction patterns, which are rarely addressed in the existing literature [107]. This survey aims to serve as a helpful reference for practitioners and researchers in the field of agent evaluation. Our contributions in this survey are twofold.

- We propose a taxonomy of LLM agent evaluation that organizes prior work by evaluation objectives (what to evaluate, such as behavior, capabilities, reliability, and safety) and evaluation process (how to evaluate, including interaction modes, datasets and benchmarks, metrics computation methods, evaluation tooling, and evaluation environments).

- We highlight enterprise-specific challenges, including role-based access control, reliability guarantees, long-term interaction, and compliance requirements.

The remainder of this paper is structured as follows. Section 2 describes the taxonomy used in this survey paper to analyze the agent evaluation landscape. Section 3 dives into the first dimension of the taxonomy, evaluation objectives, and focuses on the aspects of the agent to be evaluated. Section 4 describes the second dimension, the evaluation process, and focuses on the evaluation method. Section 5 discusses the challenges of assessing LLM agents in enterprise environments. Section 6 outlines open questions and future research directions to guide the next phase of work in evaluating LLM agents.

## 2 Taxonomy for LLM-based Agent Evaluation

We propose a two-dimensional taxonomy to organize different aspects of the evaluation of LLM-based agents, structured along the axes of **Evaluation Objectives** (what to evaluate) and **Evaluation Process** (how to evaluate). This taxonomy is visualized as a hierarchical tree in 1.

The *Evaluation Objectives* dimension is concerned with the targets of evaluation. The first category, *Agent Behavior*, in this dimension focuses on outcome-oriented aspects such as task completion and output quality, capturing how well an agent meets end-users' expectations. Next, *Agent Capabilities* emphasize process-oriented competencies, including tool use, planning and reasoning, memory and context retention, and multi-agent collaboration. These capabilities provide insights into how agents achieve their goals and how well they meet their design specification. *Reliability* assesses whether an agent behaves consistently for the same input and robustly when input varies or the system encounters errors. Finally, *Safety and Alignment* evaluates the agent's trustworthiness and security, including fairness, compliance, and the prevention of harmful or unethical behaviors.

The *Evaluation Process* dimension describes how agents are assessed. *Interaction Mode* distinguishes between static evaluation, where agents respond to fixed inputs, and interactive assessment, where agents engage with users. *Evaluation Data* discusses both synthetic and real-world datasets, as well as benchmarks tailored to specific domains such as software engineering, healthcare, and finance [23, 62]. *Metrics Computation Methods* encompasses quantitative measures, such as task success and factual accuracy, as well as qualitative evaluations based on human or LLM judgments. *Evaluation Tooling* refers to the supporting infrastructure, such as instrumentation frameworks (e.g., LangSmith, Arize AI) and public leaderboards (e.g., Holistic Evaluation of Agents), that enable scalable and reproducible assessment. Lastly, *Evaluation Contexts* define the environment in which evaluations are conducted, from controlled simulations to open-world settings such as web browsers or APIs.

This taxonomy serves both as a conceptual framework and a practical guide, enabling systematic comparison and analysis of LLM agents across a wide range of goals, methodologies, and deployment conditions. In the following sections, we examine each dimension in detail, highlighting key evaluation practices and representative studies.

As LLM agents are deployed in increasingly diverse and complex settings, factors such as single-turn versus multi-turn interactions, multilingualism, and multimodality all become more important. While the taxonomy remains applicable across these variations, tailored metrics and evaluation strategies are often required. We will discuss these specific adaptations in the relevant sections that follow.

## 3 Evaluation Objectives

### 3.1 Agent Behavior

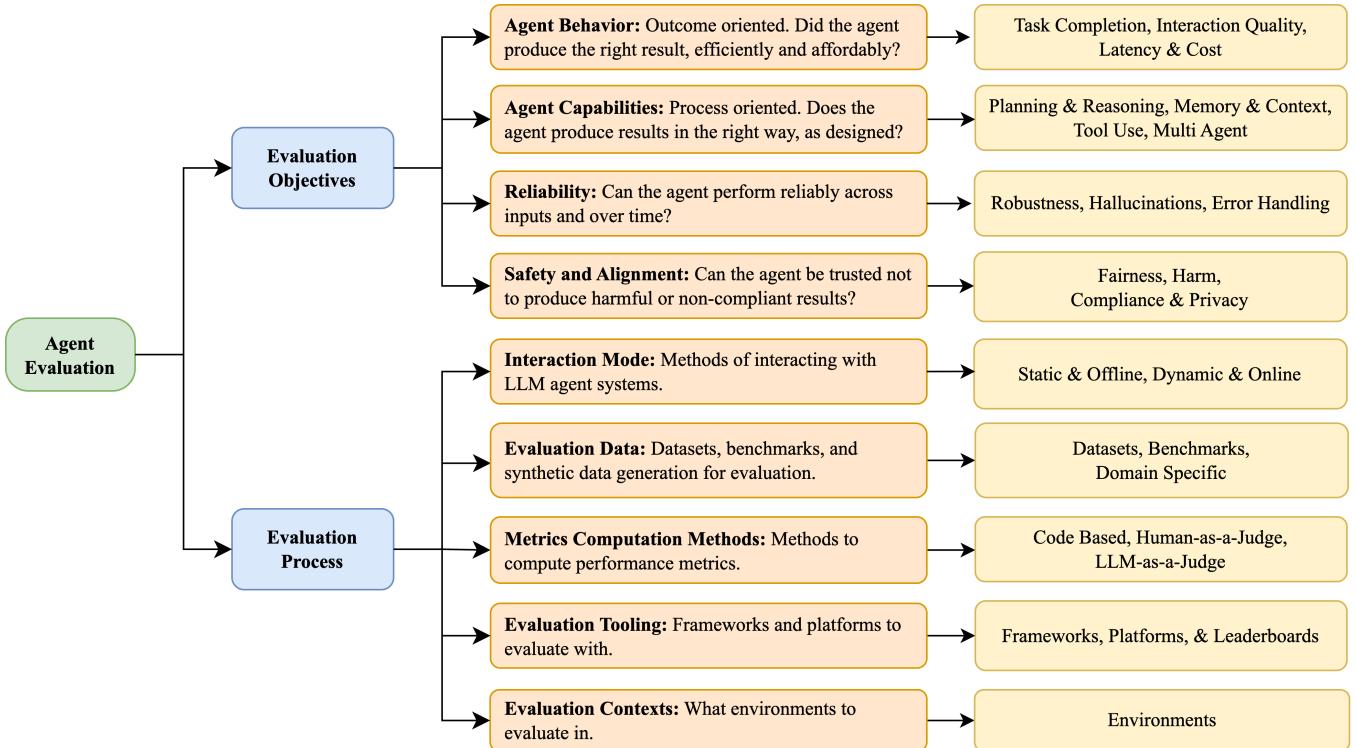
Agent behavior refers to the overall performance of the agent as perceived by a user, treating the agent as a black box. It represents the highest-level view in evaluation and offers the most direct insight into the user experience. This category encompasses aspects such as task completion, output quality, latency, and cost.

**3.1.1 Task Completion:** Task completion is a fundamental objective of agent evaluation, assessing whether an agent successfully achieves the predefined goals of a given task [11, 80, 96, 115]. It involves determining whether a desired state is reached or if specific criteria defined for task success are met [57, 93]. Although sometimes noted for providing limited fine-grained insight into failures, especially when most models achieve low success rates [64], task completion remains a predominant and essential measure of overall agent performance [64].

Task completion is commonly quantified using metrics such as *Success Rate* (SR) [11, 57, 64], which can also be referred to as *Task Success Rate* [115] or *Overall Success Rate* [57]. Other related metrics include *Task Goal Completion* (TGC) [91] and *Pass Rate* [76]. Some evaluations employ binary indicators, such as a reward function that returns 0 or 1 for goal achievement [42]. Metrics such as *pass@k* and *pass^k* extend this by considering success over multiple trials [104].

This crucial objective is applied across a wide range of LLM agent evaluation domains and benchmarks [93]. This includes tasks related to coding and software engineering, such as resolving GitHub issues (SWE-bench [40]), scientific data analysis programming (ScienceAgentBench [11]), reproducing research (CORE-Bench [86], PaperBench [87]), and interactive coding in apps (AppWorld [91]). It is also extensively used for agents interacting with web environments, including general web navigation (BrowserGym [14], WebArena [126], WebCanvas [73]), multimodal web tasks (Visual-WebArena [42], MMInA [124]), and realistic time-consuming web tasks (ASSISTANTBENCH [109]).

**3.1.2 Output Quality:** Output quality refers to the characteristics of responses by an LLM agent. It is an umbrella term encompassing aspects such as accuracy, relevance, clarity, coherence, and adherence to agent specifications or task requirements [80]. An agent may complete a task yet still deliver a subpar user experience if the interaction lacks the qualities mentioned above. Output quality is particularly relevant in evaluating conversational agents, where user goals are often achieved over multiple turns. Many metrics in this category overlap with those used in large language model (LLM) evaluation. For example, the fluency metric is used to measure the degree to which the output of an LLM satisfies the conventions of natural language [120]. The logical coherence metric focuses on



**Figure 1: Taxonomy of LLM Agent Evaluation**

the rigor in arguments [120]. As LLM agents may utilize tools for retrieving grounding information and providing context-aware text answers, standard metrics used in retrieval-augmented generation (RAG) systems also apply. Such metrics include Response Relevance or Factual Correctness [21].

**3.1.3 Latency & Cost:** Latency is a critical aspect of agent behavior, especially in scenarios where users interact with agents synchronously. Long wait times can significantly degrade user experience and erode trust in the system. A commonly used metric in this context is *Time To First Token* (TTFT), which measures the delay before a user sees the first token of an LLM’s response in streaming mode. For use cases where the agent operates asynchronously, *End-to-End Request Latency*—the time to receive the complete response—is often more relevant [70].

While cost is not directly observable by end-users, it plays a crucial role in determining the practicality of deploying agents at scale. We include cost as a measure of an agent’s monetary efficiency. It is typically estimated based on the number of input and output tokens, which directly correlate with usage-based pricing in most LLM deployments.

## 3.2 Agent Capabilities

Beyond external behavior, evaluations often target specific capabilities of LLM-based agents that enable their performance. Key aspects of this category include tool use, planning and reasoning,

memory and context retention, and multi-agent collaboration. Evaluating these capabilities helps determine an agent’s strengths and weaknesses on a more granular level.

**3.2.1 Tool Use:** Tool use is a core capability for LLM-based agents, enabling them to retrieve grounding information, perform actions, and interact with external environments. In this survey, tool use involves invocation of a single tool and is interchangeable with function calling; more complex cases of determining tool sequences for complex tasks will be discussed in 3.2.2. Recent advances have allowed LLMs, such as ChatGPT-3.5 and beyond, to support function calling natively. These models can autonomously decide whether to invoke a function, select the appropriate one from a candidate set, and generate the required parameters. As a result, LLM agents can directly build on the functions of the underlying model, allowing many of the evaluation techniques originally developed for LLMs to use tools [47].

The evaluation of the tool’s use involves answering several key questions. First, can the agent correctly determine whether tool invocation is necessary for a given task? If so, can it select the appropriate tool from a defined set of candidates? Once the tool is selected, the agent must be able to identify the correct parameters required by the tool and then generate appropriate values for each parameter to ensure proper execution. In cases where the candidate toolset is extensive, the agent may also need to retrieve the correct tool from a repository based on a natural language description of the task [83].

Several metrics have been proposed to assess these abilities. *Invocation Accuracy* [54] evaluates whether the agent makes the correct decision about whether to call a tool at all. *Tool Selection Accuracy* measures whether the proper tool is chosen from a list of options. *Retrieval Accuracy* focuses on whether the system can retrieve the correct tool from a larger toolset, often measured using rank accuracy  $k$ . For ranking-based evaluation, *Mean Reciprocal Rank (MRR)* quantifies the position of the correct tool in the ranked list. In contrast, *Normalized Discounted Cumulative Gain (NDCG)* reflects how well the system ranks all relevant tools[54].

Parameter-related evaluation involves two aspects. The *parameter name F1 score* [83] measures the agent’s ability to identify the parameter names required for a given function correctly and then correctly assign values to them. While some evaluations rely on the correctness of abstract syntax trees (ASTs) to check if the tool call is syntactically valid, this approach may miss semantic errors, such as incorrect or hallucinated parameter values, especially for parameters constrained to enumerated types [75]. To address this limitation, recent work, such as the Gorilla paper, has proposed execution-based evaluation, in which the system runs the tool calls and assesses their outcomes, offering a more comprehensive and grounded assessment of tool use capability [75].

**3.2.2 Planning and Reasoning:** Planning and reasoning are essential capabilities for LLM-based agents, especially in complex tasks that require multiple steps or making decisions under uncertainty. Planning involves selecting the correct set of tools in an appropriate order. At the same time, reasoning enables agents to make context-aware decisions, either ahead of time or dynamically during task execution [37]. T-eval [12] formulated planning evaluation as comparing the set of predicted tools against a reference. Since tool order and dependency also matter, some benchmarks adopt graph-based representations and introduce metrics such as *Node F1* for tool selection and *Edge F1* or *Normalized Edit Distance* for assessing tool invocation sequences and structural accuracy [83].

In dynamic environments, agents often need to interleave planning and execution, adapting their actions in response to evolving context [37]. This pattern is illustrated by the ReAct paradigm, where agents alternate between reasoning steps and tool usage [106]. Evaluating such adaptive reasoning requires more than comparing static plans—it demands metrics that reflect decision-making in real time. The T-Eval framework [12] addresses this by introducing a reasoning metric that assesses how closely an agent’s predicted next tool call aligns with the expected one at each step. This captures the agent’s ability to make informed decisions when tool outputs are not known in advance. Similarly, AgentBoard [64] proposes the metric *Progress Rate*, which compares the agent’s actual trajectory against the expected one, offering a fine-grained measure of how effectively the agent advances toward its goal.

When agents are instructed to plan in the form of generating complete multi-step programs, evaluation methods from code generation become relevant. Benchmarks like ScienceAgentBench compare the generated plans against annotated references using program similarity metrics [11]. Additionally, the *Step Success Rate* has been proposed to measure the percentage of steps in the generated plan that are successfully executed, providing a holistic view of planning quality during execution [28].

**3.2.3 Memory and Context Retention:** A critical capability for long-running agents is the ability to retain information throughout many interactions and apply previous context to current requests. Guan et al. [31] categorize memory evaluation in multi-turn conversations by *Memory Span* (how long information is stored) and *Memory Forms* (how information is represented). For example, LongEval [43] and SocialBench [9] are benchmarks that test an agent’s context retention in long dialogues (40+ turns). An agent might be given a conversation that spans dozens of exchanges and later asked questions that require recalling details from early in the conversation. Maharana et al. [65] demonstrate evaluation with dialogues spanning hundreds of turns (600+ turns), and Li et al. [50] introduce memory-enhanced evaluation techniques, tracking how well agents maintain consistency in long-horizon tasks. These evaluations often use synthetic or logged conversations as datasets, and metrics include *Factual Recall Accuracy* or *Consistency Score* (no contradictions between turns). Memory evaluation may also consider working memory for tool-using agents (i.e., whether the agent keeps track of intermediate results) and forgetting strategies (i.e., whether it appropriately forgets irrelevant details to avoid confusion).

**3.2.4 Multi-Agent Collaboration:** Evaluating multi-agent collaboration in LLM-based systems requires different methodologies compared to traditional reinforcement learning-driven coordination [7, 48, 89]. Unlike conventional agents that rely on predefined reward structures, LLM agents coordinate through natural language, strategic reasoning, and decentralized problem-solving [32, 33]. These capabilities are crucial in real-world applications such as financial decision-making and structured data analysis, where autonomous agents must exchange information, negotiate, and synchronize decision-making processes efficiently [50, 55]. Autonomous Agents for Collaborative Tasks [55] evaluates *Collaborative Efficiency*, assessing how well multiple agents share responsibilities and distribute tasks dynamically.

### 3.3 Reliability

Reliability is a crucial objective, especially as LLM agents are considered for use in enterprise and safety-critical applications. It encompasses consistency, robustness to variations, and trustworthiness of the agents’ outputs. Unlike task performance (which might measure best-case capabilities), reliability evaluation probes worst-case and average-case scenarios.

**3.3.1 Consistency:** Consistency refers to the stability of the output when the same task is repeated multiple times [54]. Since LLMs are inherently non-deterministic, LLM-based agents also exhibit variability in their behavior. For agents to be trusted in enterprise or other high-stakes applications, they must demonstrate consistent performance across repeated runs of the same task. A commonly used metric in this context is *pass@k*, which measures the probability that an agent succeeds at least once over  $k$  attempts. However, a stricter measure of consistency is whether the agent succeeds in all  $k$  attempts. This is formalized in the  $\tau$ -benchmark as the *pass<sup>τ</sup>k* metric, which better captures the consistency requirements of mission-critical deployments.

**3.3.2 Robustness:** Robustness refers to the stability of an agent’s output when faced with input variations or changes in the environment. To remain effective and trustworthy, LLM-based agents must consistently deliver high performance under a range of challenging conditions. Evaluating robustness often involves stress-testing the agent with perturbed inputs—such as paraphrased instructions, irrelevant or misleading context, or linguistic variations like typos and dialects—to assess whether it can still complete the task successfully. For example, robustness evaluations may involve applying systematic transformations to standard prompts and measuring the resulting drop in task success rate or output quality. The HELM benchmark [51] explicitly incorporates such tests, tracking how model performance degrades under input variation.

Robustness also encompasses adaptive resilience—the agent’s ability to recover from dynamic changes in the environment. For instance, WebLinX [63] examines how agents behave when the structure of a web page changes during execution. In such settings, an effective agent must adjust its strategy rather than stall or fail.

In tool-using agents, robustness is further reflected in error-handling capabilities. As demonstrated in ToolEmu’s evaluation [82], agents must be able to respond to tool failures or unexpected outputs gracefully. Robustness tests may include intentionally injecting failures—such as API errors or null responses—to observe whether the agent recovers (e.g., retries, switches tools, or explains the issue to the user) or breaks down. A key metric could be the proportion of induced failures that are handled appropriately, reflecting the agent’s reliability in uncertain or imperfect conditions.

### 3.4 Safety and Alignment

Safety covers an agent’s adherence to ethical guidelines, avoidance of harmful behavior, and compliance with legal or policy constraints. As LLM agents become more powerful and autonomous, the risk of unintended adverse outcomes (e.g., generating disinformation, hate speech, or unsafe instructions) grows, making safety evaluation indispensable. These evaluations are especially critical in fields such as financial services, cybersecurity, and autonomous decision-making, where agent vulnerabilities can lead to severe consequences [22, 26, 34, 49].

**3.4.1 Fairness:** The lack of fairness and transparency in AI agents can result in biased outcomes, decreased users’ trust, and unintended societal consequences [16]. In financial applications, for example, biased decision-making in loan approvals or investment strategies can reinforce systemic inequalities (FinCon [111], Auto-Guide [24]). Ethical concerns also arise in multi-agent interactions, where decision-making frameworks must ensure compliance with standards and social norms [67].

Explainability is crucial in enhancing user trust, especially in interactive systems where AI agents provide recommendations or automated assistance. Methods such as guideline-driven decision-making (AutoGuide [24]) and structured transparency mechanisms (MATSA [66], FinCon [111]) provide users with clear reasoning paths. Meanwhile, Rjudge [112] analyzes how agents perceive risk when making autonomous decisions, emphasizing transparency and trustworthiness in AI interactions. Evaluating these dimensions ensures that AI agents align with ethical standards while maintaining fairness in their operational contexts.

**3.4.2 Harm, Toxicity, and Bias:** One aspect of safety is ensuring that an agent’s outputs do not contain harmful content such as hate speech, harassment, or extremely biased statements. Evaluation for toxicity often uses specialized test sets and metrics, such as the RealToxicityPrompts dataset [27] – a collection of prompts likely to elicit toxic content—where its responses are checked with automated toxicity detectors or human raters. Metrics include the percentage of responses containing toxic language or the average toxicity score (as given by a classifier). HELM [51] includes toxicity and bias metrics as part of holistic evaluation, indicating how frequently a model produces offensive content or exhibits undesired biases. For an interactive agent, one might evaluate it by giving provocative or ethically challenging inputs (red-teaming) and then measuring its failure rate (how often it responds in an unsafe manner). Safety-focused datasets, such as CoSafe, target exactly this: Yu et al. introduce CoSafe [110] to evaluate conversation agents on adversarial prompts designed to trick them into breaking safety rules (e.g., a user subtly asks for self-harm advice or illicit instructions). CoSafe revealed that even advanced agents had vulnerabilities, such as falling for coreference-based attacks (where a user refers to something ambiguously to bypass filters). The evaluation process involved monitoring the agent’s responses for policy violations when faced with these adversarial queries. Having a numeric score (like “agent produced a disallowed response in X% of adversarial cases”) quantifies safety.

**3.4.3 Compliance and Privacy:** Beyond avoiding overt toxicity, many deployments require agents to comply with specific regulatory or policy constraints [6, 123]. For instance, a finance chatbot must not disclose confidential information or provide particular types of financial advice, and a medical assistant must not deviate from established medical guidelines. Evaluating compliance may be highly domain-specific, as it involves scenarios crafted to test whether the agent respects boundaries (e.g., a user asks the medical bot for a prescription-only drug recommendation—the correct, safe behavior is to refuse and advise consulting a doctor).

In enterprise contexts, compliance evaluation may require proprietary test cases that reflect actual policies. One approach is to integrate those concerns into evaluation frameworks. For example, the HELM benchmark [51] for enterprises was proposed to include domain-specific prompts and metrics (such as accuracy on financial jargon or compliance in responses) for fields like finance and law. The process involves gathering representative enterprise scenarios (which may contain confidential or custom data) and designing evaluation metrics that reflect real-world success criteria (e.g., Did the agent follow all legal disclaimer requirements in its response?). For example, TheAgentCompany [97] evaluates enterprise AI agents under structured correctness constraints, requiring them to follow predefined organizational policies when completing tasks.

## 4 Evaluation Process

### 4.1 Interaction Mode

Evaluating LLM agents can occur in various interaction modes and with different tooling. A fundamental distinction is between offline evaluation (using pre-generated, static datasets) and online assessment (involving reactive simulations, humans in the loop, or live system monitoring).

**Table 1: Evaluation Objectives and Their Metrics.**

Objectives	Category	Metrics	Relevant Papers
Agent Behavior	Task Completion	Success Rate (SR), F1-score, Pass@k, Progress Rate, Execution Accuracy, Transfer Learning Success, Zero-Shot Generalization Accuracy	AgentBoard [8], WebShop [103], AgentBench [58], Tool Use Evaluation [53], InformativeBench, SQuAD [79][78] [56], ResearchArena [41], AgentBoard [8], AppWorld [91], TheAgentCompany [97], MAGIC [99], Mobile-Env [117], Re-ReST [19], XMC-AGENT [59], SWE-bench [40]
	Output Quality	Coherence, User Satisfaction, Usability, Likability, Overall Quality	PredictingIQ [80], EnDex [98], PsychoGAT [101]
	Latency & Cost	Latency, Token Usage, Cost	Cluster diagnosis [84], MobileBench [18], MobileAgent-Bench [92], LangSuitE [39], WebArena [126], Mobile-env [116], GUI Agents [94], GPTDroid [60], Spa-bench [10]
Agent Capability	Tool Use	Task Completion Rate, Tool Selection Accuracy	ToolEmu [81], MetaTool [38], AutoCodeRover [119]
	Planning & Reasoning	Reasoning Quality, Accuracy, Fine-Grained Progress Rate, Self Consistency, Plan Quality	AgentBoard [8] Massive Multitask Language Understanding [36], LLM-Augmented Autonomous Agents [55], Cluster diagnosis questions [84], SimuCourt [35], Magis [90]
	Memory & Context Retention	Factual Accuracy Recall, Consistency Scores	LongEval [43], SocialBench [9], LoCoMo [65], Optimus-1 [50]
	Multi-Agent Collaboration	Information Sharing Effectiveness, Adaptive Role Switching, Reasoning Rating	AgentSims [52], WebArena [126], MATSA [66], GAMEBENCH [15], BALROG [72], TheAgentCompany [97]
Reliability	Consistency	pass^k	$\tau$ -Bench [104]
	Robustness	Accuracy, Task Success Rate Under Perturbation	HELM [51], WebLinX [63]
Safety	Fairness	Awareness Coverage, Violation Rate, Transparency, Ethics, Morality	CASA [77], R-Judge [112], SimuCourt [35], MATSA [66], FinCon [111], AutoGuide [24]
	Harm	Adversarial Robustness, Prompt Injection Resistance, Harmfulness, Bias Detection	Agent Security Bench(ASB) [118], AgentPoison [13], AgentDojo [17], Backdoor Attacks [102], SafeAgentBench [108], Agent-Safety Bench [122], AgentHarm [5], Adaptive Attacks [113], RealToxicityPrompts [27]
	Compliance & Privacy	Risk Awareness, Task Completion Under Constraints	R-Judge [112], Cybench [114], TheAgentCompany [97]

**4.1.1 Static & Offline Evaluation** Often performed as a baseline, offline evaluations typically rely on datasets and static test cases: collections of tasks, prompts, or conversations that represent challenges the agent might face. Simulated conversations may be used to help develop these data, but are ultimately inert between different runs. Though comparatively cheaper and simpler to run and maintain, offline evaluations typically lack the nuance to fully address the wide range of responses an LLM agent may be able or expected to provide. As such, they are more prone to error propagation and are generally less accurate representations of system performance.

**4.1.2 Dynamic & Online Evaluation** As with other ML systems, online evaluation often occurs after an LLM agent has been deployed. Instead of relying on synthetic, historical, or manually crafted data, online evaluations leverage simulations or fundamental user interactions. This adaptive data is crucial for identifying pain points and issues that are not discovered during static testing and is often rich in domain context that is more difficult to capture with synthetic or generalized benchmarks. Dynamic evaluations may use proxies to **simulate users or environments** in reactive, real-time response

to agent behavior. For example, in the assessment of web agents, researchers built web simulators (MiniWoB [85], WebShop [103], WebArena [126], etc.) where the behavior of an agent (clicking links, filling forms) can be programmed to verify the correct sequence.

The concept of **Evaluation-driven Development (EDD)** has also been proposed [95], proposing making evaluation an integral part of the agent development cycle. It advocates for continuous evaluation of the agent, both offline (during development) and online (after deployment), to detect regressions and adapt to new use cases. They further outline a reference architecture in which an AgentOps component monitors agent performance in production and provides insights back to developers. While still an emerging idea, it underscores that evaluation is not a one-time task but an ongoing process, especially for agents that learn or evolve.

## 4.2 Evaluation Data

The growing interest in evaluating LLM-based agents has led to the development of diverse datasets, benchmarks, and leaderboards specifically targeting the agent capabilities discussed in Section 3.

Many of these resources are designed to reflect real-world complexity and are built using a mix of human-annotated, synthetic, and interaction-generated data. For instance, datasets such as AAAR-1.0 [61], ScienceAgentBench [11], and TaskBench [83] provide structured, expert-labeled benchmarks for assessing research reasoning, scientific workflows, and multi-tool planning. Others, such as FlowBench [96], ToolBench [38], and API-Bank [47], focus on tool use and function-calling across large API repositories. These benchmarks typically include not only the gold tool sequences but also expected parameter structures, enabling fine-grained evaluation.

In parallel, datasets like AssistantBench [109], AppWorld [91], and WebArena [126] simulate more open-ended and interactive agent behaviors in web and application environments. They emphasize dynamic decision-making, long-horizon planning, and user-agent interactions. Several benchmarks also support safety and robustness testing—for example, AgentHarm [5] assesses potentially harmful behaviors, while AgentDojo [17] evaluates resilience against prompt injection attacks. Leaderboards such as the Berkeley Function-Calling Leaderboard (BFCL) [100] and Holistic Agent Leaderboard [88] consolidate these evaluations by providing standardized test cases, automated metrics (e.g., AST correctness, Win Rate), and ranking mechanisms to compare systems.

### 4.3 Metrics Computation Methods

The code-based method is the most deterministic and objective approach [8, 53, 57]. [8] It relies on explicit rules, test cases, or assertions to verify whether an agent’s response meets predefined criteria. This method is particularly effective for tasks with well-defined outputs, such as numerical calculations, structured query generation, or syntactic correctness in programming tasks. Its primary advantage is its consistency and reproducibility, making it highly reliable for benchmarking. However, code-based methods are typically inflexible. They struggle with evaluating open-ended or qualitative responses, such as natural language generation or creative problem-solving, where correctness is subjective. Despite this, it remains a fundamental technique for evaluating structured tasks where correctness is well-defined.

The LLM-as-a-Judge approach [125] leverages the reasoning capabilities of LLMs to evaluate agent responses based on qualitative criteria, assessing responses according to the criteria provided through instructions. This method has gained traction due to its ability to handle tasks that are subjective and nuanced, such as summarization, reasoning, and conversational interactions. One recent extension of this method is Agent-as-a-Judge [127], where the evaluation process involves multiple AI agents interacting to refine the assessment, potentially improving evaluation reliability. This method is highly scalable and can adapt to complex tasks. As a result, it has received increasing attention [30, 46].

Human-in-the-loop evaluation remains the gold standard for subjective aspects (like naturalness and user satisfaction) and safety-critical judgment calls. Human evaluations can take the form of user studies, expert audits (where domain experts review agent outputs), or Crowdworker annotations (where outputs are rated along dimensions such as relevance, correctness, and tone). This method offers the highest reliability in open-ended tasks, such as content generation, strategic decision-making, or dialogue coherence. However, it is expensive, time-consuming, and challenging to

scale, making it impractical for large-scale automated systems that require frequent evaluations.

### 4.4 Evaluation Tooling

A notable aspect in the process dimension is the emergence of software frameworks and platforms that support automated, scalable, and continuous agent evaluation workflows. These tools enable integration of evaluation directly into the development lifecycle, reflecting a growing movement toward Evaluation-driven Development (EDD) [95] in agent building. OpenAI Eval is an open-source framework that allows developers to specify evaluation tasks and metrics for models, automating the execution and reporting of results (though not formally described in academic literature, it reflects practical needs) [71]. Other open-source or commercial tools such as DeepEval [2], InspectAI [3], Phoenix [1], and GALILEO [25] provide rich analytics, evaluation orchestration, and debugging capabilities. Moreover, agent development platforms like Azure AI Foundry [68], Google Vortex AI [29], LangGraph [44], and Amazon Bedrock [4] increasingly incorporate evaluation features, helping developers monitor performance, detect regressions, and adapt agents to evolving user needs. Xia et al. [95] further propose an AgentOps architecture to continuously monitor deployed agents, closing the loop between development and deployment through real-time feedback and quality control.

### 4.5 Evaluation Contexts

The evaluation context pertains to the environment in which an evaluation is performed. Similar to software engineering, a tradeoff exists between more realistic (but often more costly and potentially less secure) and simpler, more controlled (but usually less representative of final performance) environments. The context in which a system is assessed is typically guided by the system’s intended use; a simpler LLM agent without edit access might be tested in its working environment directly, whereas an LLM agent designed to work with and make changes to many intertwined systems will likely be evaluated in a mocked API or sandbox environment. For less contained systems, the evaluation context may take the form of a web simulator, such as MiniWoB [85], WebShop [103], or WebArena [126]. As the development of an agent continues, the evaluation context often evolves with it, from smaller, mocked API environments to live deployment as agent performance and trustworthiness are determined.

## 5 Enterprise-Specific Challenges

As LLM-based agents transition from research demos to deployment in enterprise settings, new challenges are emerging. Enterprises often demand high performance in conjunction with predictable reliability, compliance with regulations, data security, and maintainability, which are usually overlooked during evaluation. To address these gaps, we discuss the concerns outlined in the following sections and outline future directions.

### 5.1 Complexity from Role-based Access

A key challenge in evaluating LLM-based agents in enterprise settings is the need to account for Role-Based Access Control (RBAC), which governs users’ permissions to access data and services. In these environments, users operate with varying levels of access

depending on their roles, and agents acting on their behalf must adhere to the same constraints. This introduces complexity into agent evaluation, as an agent's ability to retrieve or act on information is not uniform but contextually bound to the user's permissions.

To address this, some evaluation frameworks have begun incorporating access control constraints into their design. For example, IntellAgent [45] includes evaluation tasks that require authentication of user identity and enforce policies that deny access to other users' information. By embedding role-specific restrictions into task generation, these approaches more accurately model how agents behave in permission-sensitive enterprise contexts.

## 5.2 Reliability Guarantees

Reliability guarantees are especially important in enterprise settings, where agents are expected to operate within compliance and auditing frameworks that require deterministic or repeatable behavior that is explainable. In such contexts, occasional success is insufficient; agents must perform reliably across time and usage scenarios to be considered production-ready.

Evaluating reliability is nontrivial. Because LLM-based agents are inherently stochastic, measuring consistency requires executing the same task multiple times and observing the variation in outcomes. This introduces significant evaluation overhead: running multiple trials per input can be computationally expensive, especially when testing complex tasks involving tools, memory, or multi-agent coordination. Moreover, to draw meaningful conclusions, benchmarks must include a representative dataset that reflects the types of tasks and conditions the agent may encounter.

Some efforts have begun to tackle this challenge. For example, the  $\tau$ -benchmark [104] explicitly incorporates the pass<sup>k</sup> metric to evaluate the consistency of an agent. By applying this to domains such as retail and airline booking, the authors show that current agents struggle with consistency.

## 5.3 Dynamic and Long-Horizon Interactions

One major challenge in evaluating LLM-based agents is assessing their performance on long-horizon tasks in dynamic, evolving environments. Unlike most current benchmarks that focus on short episodes or single interactions, real-world enterprise agents often operate continuously over extended periods while interacting with users, systems, and data.

Addressing this challenge is essential for understanding how agents behave over time, particularly in enterprise settings where reliability, adaptability, and goal alignment are crucial throughout the agent's lifecycle. Standard, short-term evaluations cannot capture phenomena such as performance drift, context retention, or the cumulative effect of decisions on business outcomes.

To begin tackling this issue, some research efforts have introduced long-running simulations and extended dialogues as evaluation tools. For instance, Park et al. [74] observed generative agents in a continuously running simulated town environment to study emergent behaviors across multi-day interactions. Similarly, Mahara et al. [65] evaluated long-term conversational memory through 600-turn dialogues, focusing on how well agents maintain coherence and context over extended conversations.

## 5.4 Adherence to Domain-Specific Policies and Compliance Requirements

Another significant challenge in evaluating enterprise agents is ensuring that they can operate within domain-specific policies and compliance constraints. Enterprises often enforce strict operational rules—such as approval workflows, data retention policies, usage quotas, and legal regulations like GDPR or HIPAA—that must be respected by agents throughout task execution. Evaluating agents in such contexts requires more than measuring task success; it demands verification that agent behaviors align with formal policy constraints and legal compliance standards. For instance, an agent generating financial reports must avoid unauthorized access to confidential forecasts and ensure that generated content adheres to regulatory reporting standards. Without explicit modeling of these constraints during evaluation, agents deemed "correct" in traditional benchmarks may still fail in production due to policy violations or compliance risks.

## 6 Future Research Directions

As LLM-based agents continue to grow in complexity and application scope, future research must push toward more robust, practical, and scalable evaluation methodologies. We highlight four key directions that can significantly advance the field:

**Holistic Evaluation Frameworks:** Current evaluation efforts often focus on isolated dimensions such as task success, planning quality, or tool use. However, agents in real-world applications must simultaneously balance multiple competencies. Future work should develop holistic evaluation frameworks that assess agent performance across multiple, interdependent dimensions.

**More Realistic Evaluation Settings:** To bridge the gap between lab settings and production environments, agent evaluation must move toward more realistic conditions. This includes creating evaluation environments that incorporate enterprise-specific elements such as dynamic multi-user interactions, role-based access controls, and domain-specific knowledge. These settings can be achieved through real-world deployment trials or through simulated environments that mimic enterprise workflows.

**Automated and Scalable Evaluation Techniques:** Manual evaluation of agent behavior, especially in multi-turn or long-horizon scenarios, is costly and complicated to scale. Future research should explore automated evaluation approaches to reduce human effort and improve reproducibility. This includes using synthetic data generation for controllable test cases, leveraging simulated environments to emulate task contexts, and advancing LLM-based evaluation techniques such as LLM-as-a-judge or Agent-as-a-judge.

**Time- and Cost-Bounded Evaluation Protocols:** Evaluation must be efficient and able to support iterative agent development. Today's methods—especially those requiring repeated trials or human-in-the-loop assessments—can be both time- and resource-intensive. Future research should aim to develop time- and cost-bounded evaluation protocols that strike a balance between depth and efficiency.

In summary, future research should focus on developing evaluation methods that are holistic, realistic, scalable, and efficient. These directions are essential for building reliable and trustworthy LLM-based agents at scale.

## References

- [1] Arize AI. 2025. Phoenix. <https://github.com/Arize-ai/phoenix>
- [2] Confident AI. 2025. DeepEval. <https://github.com/confident-ai/deepeval>
- [3] UK AI Security Institute. 2024. *Inspect AI: Framework for Large Language Model Evaluations*. [https://github.com/UKGovernmentBEIS/inspect\\_ai](https://github.com/UKGovernmentBEIS/inspect_ai)
- [4] Amazon. 2024. Amazon Bedrock Agents. <https://aws.amazon.com/bedrock/agents/>
- [5] Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Dueñas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, Zico Kolter, Matt Fredrikson, Eric Winsor, Jerome Wynne, Yarin Gal, and Xander Davies. 2025. AgentHarm: a benchmark for measuring harmfulness of LLM agents. doi:10.48550/arXiv.2410.09024
- [6] Nik Bear Brown. 2024. Enhancing trust in llms: Algorithms for comparing and interpreting llms. *arXiv preprint arXiv:2406.01943* (2024).
- [7] Chi-Min Chan, Weizhi Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201* (2023).
- [8] Ma Chang, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. 2024. AgentBoard: An Analytical Evaluation Board of Multi-turn LLM Agents. *Advances in Neural Information Processing Systems* 37 (2024), 74325–74362.
- [9] Hongzhan Chen, Hehong Chen, Ming Yan, Wenshen Xu, Xing Gao, Weizhou Shen, Xiaojuan Quan, Chenliang Li, Ji Zhang, Fei Huang, and Jingren Zhou. 2024. SocialBench: Sociality Evaluation of Role-Playing Conversational Agents. arXiv:2403.13679 [cs.CL] <https://arxiv.org/abs/2403.13679>
- [10] Jingxuan Chen, Derekl Yuen, Bin Xie, Yuhao Yang, Gongwei Chen, Zhihao Wu, Li Yixing, Xurui Zhou, Weiwen Liu, Shuai Wang, et al. 2024. Spa-bench: A comprehensive benchmark for smartphone agent evaluation. In *NeurIPS 2024 Workshop on Open-World Agents*.
- [11] 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. 2025. ScienceAgentBench: toward rigorous assessment of language agents for data-driven scientific discovery. doi:10.48550/arXiv.2410.05080
- [12] Zehui Chen, Weihua Du, Wenwei Zhang, Kuikun Liu, Jiangning Liu, Miao Zheng, Jingming Zhuo, Songyang Zhang, Dahua Lin, Kai Chen, and Feng Zhao. 2024. T-eval: evaluating the tool utilization capability of large language models step by step. doi:10.48550/arXiv.2312.14033
- [13] Zhaorun Chen, Zhen Xiang, Chaowei Xiao, Dawn Song, and Bo Li. 2024. Agent-Poison: Red-teaming LLM Agents via Poisoning Memory or Knowledge Bases. arXiv:2407.12784 [cs.LG] <https://arxiv.org/abs/2407.12784>
- [14] Thibault Le Sellier De Chezelles, Maxime Gasse, Alexandre Drouin, Massimo Caccia, Léo Boisvert, Megh Thakkar, Tom Marty, Rim Assouel, Sahar Omidi Shayegan, Lawrence Keunho Jang, Xing Han Lü, Ori Yoran, Dehan Kong, Frank F. Xu, Siva Reddy, Quentin Cappart, Graham Neubig, Ruslan Salakhutdinov, Nicolas Chapados, and Alexandre Lacoste. 2025. The BrowserGym ecosystem for web agent research. doi:10.48550/arXiv.2412.05467
- [15] Anthony Costarelli, Mat Allen, Roman Hauksson, Grace Sodunke, Suhas Harisharan, Carlson Cheng, Wenjie Li, Joshua Clymer, and Arjun Yadav. 2024. Gamebench: Evaluating strategic reasoning abilities of llm agents. *arXiv preprint arXiv:2406.06613* (2024).
- [16] José Antonio Siqueira de Cerqueira, Mamia Agbese, Rebekah Rousi, Nannan Xi, Juho Hamari, and Pekka Abrahamsson. 2024. Can we trust AI agents? An experimental study towards trustworthy LLM-based multi-agent systems for AI ethics. *arXiv preprint arXiv:2411.08881* (2024).
- [17] Edoardo Debenedetti, Jie Zhang, Mislav Balunović, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. 2024. AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents. doi:10.48550/arXiv.2406.13352
- [18] Shihan Deng, Weikai Xu, Hongda Sun, Wei Liu, Tao Tan, Jianfeng Liu, Ang Li, Jian Luan, Bin Wang, Rui Yan, et al. 2024. Mobile-bench: An evaluation benchmark for llm-based mobile agents. *arXiv preprint arXiv:2407.00993* (2024).
- [19] Zi-Yi Dou, Cheng-Fu Yang, Xueqing Wu, Kai-Wei Chang, and Nanyun Peng. 2024. Re-rest: Reflection-reinforced self-training for language agents. *arXiv preprint arXiv:2406.01495* (2024).
- [20] Zane Durante, Qiuyuan Huang, Naoki Wake, Ran Gong, Jae Sung Park, Bidipta Sarkar, Rohan Taori, Yusuke Noda, Demetri Terzopoulos, Yejin Choi, et al. 2024. Agent ai: Surveying the horizons of multimodal interaction. *arXiv preprint arXiv:2401.03568* (2024).
- [21] Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2025. Ragas: Automated Evaluation of Retrieval Augmented Generation. arXiv:2309.15217 [cs.CL] <https://arxiv.org/abs/2309.15217>
- [22] Richard Fang, Rohan Bindu, Akul Gupta, and Daniel Kang. 2024. Llm agents can autonomously exploit one-day vulnerabilities. *arXiv preprint arXiv:2404.08144* 13 (2024), 14.
- [23] Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Erkang Zhu, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, Peter Chang, Ricky Loynd, Robert West, Victor Dibia, Ahmed Awadallah, Ece Kamar, Rafah Hosn, and Saleema Amershi. 2024. Magentic-One: A Generalist Multi-Agent System for Solving Complex Tasks. doi:10.48550/arXiv.2411.04468
- [24] Yao Fu, Dong-Ki Kim, Jaekyeom Kim, Sungryull Sohn, Lajanugen Logeswaran, Kyunghoon Bae, and Honglak Lee. 2024. AutoGuide: Automated Generation and Selection of Context-Aware Guidelines for Large Language Model Agents. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 119919–119948.
- [25] Galileo. 2025. Introducing agentic evaluations. <https://www.galileo.ai/blog/introducing-agentic-evaluations> Accessed: 2025-05-20.
- [26] Yuyou Gan, Yong Yang, Zhe Ma, Ping He, Rui Zeng, Yiming Wang, Qingming Li, Chunyi Zhou, Songze Li, Ting Wang, et al. 2024. Navigating the risks: A survey of security, privacy, and ethics threats in llm-based agents. *arXiv preprint arXiv:2411.09523* (2024).
- [27] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models. arXiv:2009.11462 [cs.CL] <https://arxiv.org/abs/2009.11462>
- [28] Luca Gioacchini, Giuseppe Siracusano, Davide Sanvito, Kiril Gashteovski, David Friede, Roberto Bifulco, and Carolin Lawrence. 2024. AgentQuest: A Modular Benchmark Framework to Measure Progress and Improve LLM Agents. doi:10.48550/arXiv.2404.06411
- [29] Google. 2024. Google Vortex AI. <https://cloud.google.com/vertex-ai>
- [30] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594* (2024).
- [31] Zhenyu Guan, Xiangyu Kong, Fangwei Zhong, and YIzhou Wang. 2024. Richelle: Self-Evolving LLM-Based Agents for AI Diplomacy. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 38. Curran Associates, Inc., 0.
- [32] Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680* (2024).
- [33] Shanshan Han, Qifan Zhang, Yuhang Yao, Weizhao Jin, Zhaozhuo Xu, and Chaoyang He. 2024. LLM multi-agent systems: Challenges and open problems. *arXiv preprint arXiv:2402.03578* (2024).
- [34] Feng He, Tianqiang Zhu, Dayong Ye, Bo Liu, Wanlei Zhou, and Philip S Yu. 2024. The emerged security and privacy of llm agent: A survey with case studies. *arXiv preprint arXiv:2407.19354* (2024).
- [35] Zhitao He, Pengfei Cao, Chenhao Wang, Zhuoran Jin, Yubo Chen, Jiexin Xu, Huajun Li, Xiaojian Jiang, Kang Liu, and Jun Zhao. 2024. AgentsCourt: Building Judicial Decision-Making Agents with Court Debate Simulation and Legal Knowledge Augmentation. arXiv:2403.02959v3 (2024). <https://arxiv.org/pdf/2403.02959.pdf>
- [36] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring Massive Multitask Language Understanding. arXiv:2009.03300 [cs.CY] <https://arxiv.org/abs/2009.03300>
- [37] Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. 2024. Understanding the planning of LLM agents: A survey. doi:10.48550/arXiv.2402.02716
- [38] Yue Huang, Jiawen Shi, Yuan Li, Chenrui Fan, Siyu Wu, Qihui Zhang, Yixin Liu, Pan Zhou, Yao Wan, Neil Zhenqiang Gong, and Lichao Sun. 2024. MetaTool Benchmark for Large Language Models: Deciding Whether to Use Tools and Which to Use. arXiv:2310.03128 [cs.SE] <https://arxiv.org/abs/2310.03128>
- [39] Zixia Jia, Mengmeng Wang, Baichen Tong, Song-Chun Zhu, and Zilong Zheng. 2024. LangSuite: Planning, Controlling and Interacting with Large Language Models in Embodied Text Environments. *arXiv preprint arXiv:2406.16294* (2024).
- [40] Carlos E. Jimenez, John Yang, Alexander Wetzig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2024. SWE-bench: Can Language Models Resolve Real-World GitHub Issues? doi:10.48550/arXiv.2310.06770
- [41] H Kang and C Xiong. 2024. ResearchArena: Benchmarking LLMs' Ability to Collect and Organize Information as Research Agents. (2024).
- [42] Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvuri, 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. doi:10.48550/arXiv.2401.13649
- [43] Kalpesh Krishna, Erin Bransom, Bailey Kuehl, Mohit Iyyer, Pradeep Dasigi, Arman Cohan, and Kyle Lo. 2023. LongEval: Guidelines for Human Evaluation of Faithfulness in Long-form Summarization. arXiv:2301.13298 [cs.CL] <https://arxiv.org/abs/2301.13298>
- [44] LangChain. 2024. LangGraph Platform. <https://www.langchain.com/langgraph-platform>
- [45] Elad Levi and Ilan Kadar. 2025. IntellAgent: a multi-agent framework for evaluating conversational AI systems. doi:10.48550/arXiv.2501.11067

- [46] Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024. Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579* (2024).
- [47] 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. doi:10.48550/arXiv.2304.08244
- [48] Xinyi Li, Sai Wang, Siqi Zeng, Yu Wu, and Yi Yang. 2024. A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth* 1, 1 (2024), 9.
- [49] Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu, Wenxing Xu, Xiang Wang, Yi Sun, et al. 2024. Personal llm agents: Insights and survey about the capability, efficiency and security. *arXiv preprint arXiv:2401.05459* (2024).
- [50] Zaijing Li, Yuquan Xie, Rui Shao, Gongwei Chen, Dongmei Jiang, and Liqiang Nie. 2024. Optimus-1: Hybrid Multimodal Memory Empowered Agents Excel in Long-Horizon Tasks. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 49881–49913.
- [51] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladha, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2023. Holistic Evaluation of Language Models. doi:10.48550/arXiv.2211.09110
- [52] Jiaju Lin, Haoran Zhao, Aochi Zhang, Yiting Wu, Huqiyue Ping, and Qin Chen. 2023. Agentsims: An open-source sandbox for large language model evaluation. *arXiv preprint arXiv:2308.04026* (2023).
- [53] Bing Liu, Zhou Jianxiang, Dan Meng, and Haonan Lu. 2024. An Evaluation Mechanism of LLM-based Agents on Manipulating APIs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 4649–4662. doi:10.18653/v1/2024.findings-emnlp.267
- [54] Bang Liu, Xinfeng Li, Jiayi Zhang, Jinlin Wang, Tanjin He, Sirui Hong, Hongzhang Liu, Shaokun Zhang, Kaitao Song, Kunlun Zhu, Yuheng Cheng, Suyuchen Wang, Xiaoqiang Wang, Yuyu Luo, Haibo Jin, Peiyang Zhang, Ollie Liu, Jiaqi Chen, Huan Zhang, Zhaoyang Yu, Haochen Shi, Boyan Lei, Dekun Wu, Fengwei Teng, Xiaojun Jia, Jiawei Xu, Jinyu Xiang, Yizhang Lin, Tianming Liu, Tongliang Liu, Yu Su, Huan Sun, Glen Berseth, Jianyun Nie, Ian Foster, Logan Ward, Qingyun Wu, Yu Gu, Mingchen Zhuge, Xiangru Tang, Haohan Wang, Jiaxuan You, Chi Wang, Jian Pei, Qiang Yang, Xiaoliang Qi, and Chenglin Wu. 2025. Advances and Challenges in Foundation Agents: From Brain-Inspired Intelligence to Evolutionary, Collaborative, and Safe Systems. doi:10.48550/arXiv.2504.01990
- [55] Wei Liu, Chenxi Wang, Yifei Wang, Zihao Xie, Rennai Qiu, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, and Chen Qian. 2024. Autonomous agents for collaborative task under information asymmetry. *arXiv preprint arXiv:2406.14928* (2024).
- [56] Wei Liu, Chenxi Wang, YiFei Wang, Zihao Xie, Rennai Qiu, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, and Chen Qian. 2024. Autonomous Agents for Collaborative Task under Information Asymmetry. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- [57] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2023. AgentBench: Evaluating LLMs as Agents. doi:10.48550/arXiv.2308.03688
- [58] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2023. AgentBench: Evaluating LLMs as Agents. arXiv:2308.03688 [cs.AI]. <https://arxiv.org/abs/2308.03688>
- [59] Yanjiang Liu, Tianyun Zhong, Yacie Lu, Hongyu Lin, Ben He, Shuheng Zhou, Huijia Zhu, Weiqiang Wang, Zhongyi Liu, Xianpei Han, et al. 2024. XMC-Agent: Dynamic Navigation over Scalable Hierarchical Index for Incremental Extreme Multi-label Classification. In *Findings of the Association for Computational Linguistics ACL 2024*. 5659–5672.
- [60] Zhe Liu, Chunyang Chen, Junjie Wang, Mengzhuo Chen, Boyu Wu, Xing Che, Dandan Wang, and Qing Wang. 2023. Chatting with gpt-3 for zero-shot human-like mobile automated gui testing. *arXiv preprint arXiv:2305.09434* (2023).
- [61] 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. doi:10.48550/arXiv.2410.22394
- [62] Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292* (2024).
- [63] Xing Han Lu, Zdeněk Kasner, and Siva Reddy. 2024. WebLINX: Real-World Website Navigation with Multi-Turn Dialogue. (Feb. 2024). arXiv:2402.05930 [cs.CL]
- [64] Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujin Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. 2024. AgentBoard: An Analytical Evaluation Board of Multi-turn LLM Agents. doi:10.48550/arXiv.2401.13178
- [65] 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:2402.17753 [cs.CL]. <https://arxiv.org/abs/2402.17753>
- [66] Puneet Mathur, Alexa Siu, Nedim Lipka, and Tong Sun. 2024. MATSA: Multi-Agent Table Structure Attribution. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 250–258.
- [67] Siddharth Mehrotra, Carolina Centeio Jorge, Catholijn M Jonker, and Myrthe L Tielman. 2024. Integrity-based explanations for fostering appropriate trust in AI agents. *ACM Transactions on Interactive Intelligent Systems* 14, 1 (2024), 1–36.
- [68] Microsoft. 2024. Azure Foundry. <https://azure.microsoft.com/en-us/products/ai-foundry>
- [69] Yohei Nakajima. 2023. Babyagi. *GitHub repository* (2023).
- [70] Nvidia. 2024. LLM Benchmark Metrics. <https://docs.nvidia.com/nim/benchmarking/lm/latest/metrics.htm>
- [71] OpenAI. 2023. OpenAI Eval. <https://github.com/openai/evals>
- [72] Davide Paglieri, Bartłomiej Cupial, Samuel Coward, Ulyana Piterbarg, Maciej Wolczyk, Akbir Khan, Eduard Pignatelli, Łukasz Kuciński, Lerrel Pinto, Rob Fergus, Jakob Nicolaus Foerster, Jack Parker-Holder, and Tim Rocktäschel. 2024. BALROG: Benchmarking Agentic LLM and VLM Reasoning On Games. arXiv:2411.13543 [cs.AI]. <https://arxiv.org/abs/2411.13543>
- [73] Yichen Pan, Dehong Kong, Sida Zhou, Cheng Cui, Yifei Leng, Bing Jiang, Hangyu Liu, Yanyi Shang, Shuyan Zhou, Tongshuang Wu, and Zhengyang Wu. 2024. WebCanvas: benchmarking web agents in online environments. doi:10.48550/arXiv.2406.12373
- [74] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*. 1–22.
- [75] Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. 2023. Gorilla: Large Language Model Connected with Massive APIs. doi:10.48550/arXiv.2305.15334 [cs]. arXiv:2305.15334 [cs]
- [76] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2023. ToolLLM: facilitating large language models to master 16000+ real-world APIs. doi:10.48550/arXiv.2307.16789
- [77] Haoyi Qiu, Alexander Fabbri, Divyansh Agarwal, Kung-Hsiang Huang, Sarah Tan, Nanyun Peng, and Chien-Sheng Wu. 2025. Evaluating Cultural and Social Awareness of LLM Web Agents. arXiv:2410.23252 (2025).
- [78] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. *arXiv preprint arXiv:1806.03822* (2018).
- [79] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. arXiv:1606.05250 [cs.CL]
- [80] Merle M. Reimann, Catharine Oertel, Florian A. Kunneman, and Koen V. Hindriks. 2023. Predicting interaction quality aspects using level-based scores for conversational agents. In *Proceedings of the 23rd ACM International Conference on Intelligent Virtual Agents*. ACM, Würzburg Germany, 1–8. doi:10.1145/3570945.3607332
- [81] Wenjie Ruan, Wei Huang, Xiaowei Huang, Gaojie Jin, Yi Dong, Changshun Wu, Saddek Bensemalem, Ronghui Mu, Yi Qi, Xingyu Zhao, et al. 2024. ToolEmu. In *Proceedings of t*.
- [82] Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J Maddison, and Tatsunori Hashimoto. 2023. Identifying the risks of lm agents with an lm-emulated sandbox. *arXiv preprint arXiv:2309.15817* (2023).
- [83] Yongliang Shen, Kaitao Song, Xu Tan, Wenqi Zhang, Kan Ren, Siyu Yuan, Weiming Lu, Dongsheng Li, and Yueteng Zhuang. 2024. TaskBench: Benchmarking Large Language Models for Task Automation. doi:10.48550/arXiv.2311.18760
- [84] Honghao Shi, Longkai Cheng, Wenli Wu, Yuhang Wang, Xuan Liu, Shaokai Nie, Weixv Wang, Xuebin Min, Chunlei Men, and Yonghua Lin. 2024. Enhancing Cluster Resilience: LLM-agent Based Autonomous Intelligent Cluster Diagnosis System and Evaluation Framework. arXiv:2411.05349 (2024). <https://arxiv.org/pdf/2411.05349.pdf>
- [85] Tianlin Shi, Andrej Karpathy, Linxi Fan, Jonathan Hernandez, and Percy Liang. 2017. World of bits: An open-domain platform for web-based agents. In *International Conference on Machine Learning*. PMLR, 3135–3144.

- [86] Zachary S. Siegel, Sayash Kapoor, Nitya Nagdir, Benedikt Stroebel, and Arvind Narayanan. 2024. CORE-bench: fostering the credibility of published research through a computational reproducibility agent benchmark. doi:10.48550/arXiv.2409.11363
- [87] Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaeser, and Tejal Patwardhan. 2025. PaperBench: Evaluating AI's Ability to Replicate AI Research. doi:10.48550/arXiv.2504.01848
- [88] Benedikt Stroebel, Sayash Kapoor, and Arvind Narayanan. 2025. HAL: A Holistic Agent Leaderboard for Centralized and Reproducible Agent Evaluation. <https://github.com/princeton-pli/hal-harness>.
- [89] Yashar Talebirad and Amirhossein Nadiri. 2023. Multi-agent collaboration: Harnessing the power of intelligent llm agents. *arXiv preprint arXiv:2306.03314* (2023).
- [90] Wei Tao, Yucheng Zhou, Yanlin Wang, Wenqiang Zhang, Hongyu Zhang, and Yu Cheng. 2024. Magis: Llm-based multi-agent framework for github issue resolution. *Advances in Neural Information Processing Systems 37* (2024), 51963–51993.
- [91] 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. doi:10.48550/arXiv.2407.18901
- [92] Luyuan Wang, Yongyu Deng, Yiwei Zha, Guodong Mao, Qinmin Wang, Tianchen Min, Wei Chen, and Shoufa Chen. 2024. MobileAgentBench: An Efficient and User-Friendly Benchmark for Mobile LLM Agents. *arXiv preprint arXiv:2406.08184* (2024).
- [93] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Ji-Rong Wen. 2024. A Survey on Large Language Model based Autonomous Agents. *Frontiers of Computer Science* 18, 6 (Dec. 2024), 186345. doi:10.1007/s11704-024-40231-1
- [94] Shuai Wang, Weiven Liu, Jingxuan Chen, Yuqi Zhou, Weinan Gan, Xingshan Zeng, Yuhuan Che, Shuai Yu, Xinlong Hao, Kun Shao, et al. 2024. Gui agents with foundation models: A comprehensive survey. *arXiv preprint arXiv:2411.04890* (2024).
- [95] Boming Xia, Qinghua Lu, Liming Zhu, Zhenchang Xing, Dehai Zhao, and Hao Zhang. 2024. An Evaluation-Driven Approach to Designing LLM Agents: Process and Architecture. *arXiv preprint arXiv:2411.13768* (2024).
- [96] 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. doi:10.48550/arXiv.2406.14884
- [97] Frank F Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Z Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, et al. 2024. Theagent-company: benchmarking llm agents on consequential real world tasks. *arXiv preprint arXiv:2412.14161* (2024).
- [98] Guangxuan Xu, Ruibo Liu, Fabrice Harel-Canada, Nischal Chandra, and Nanyun Peng. 2022. EnDex: Evaluation of Dialogue Engagingness at Scale. 4884–4893. doi:10.18653/v1/2022.findings-emnlp.359
- [99] Lin Xu, Zhiyuan Hu, Daquan Zhou, Hongyu Ren, Zhen Dong, Kurt Keutzer, See Kiong Ng, and Jiashi Feng. 2023. Magic: Investigation of large language model powered multi-agent in cognition, adaptability, rationality and collaboration. *arXiv preprint arXiv:2311.08562* (2023).
- [100] 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](https://gorilla.cs.berkeley.edu/blogs/8_berkeley_function_calling_leaderboard.html).
- [101] Qisen Yang, Zekun Wang, Honghui Chen, Shenzhi Wang, Yifan Pu, Xin Gao, Wenhao Huang, Shiji Song, and Gao Huang. 2024. PsychoGAT: A Novel Psychological Measurement Paradigm through Interactive Fiction Games with LLM Agents. *arXiv:2402.12326v2* (2024). <https://arxiv.org/pdf/2402.12326.pdf>
- [102] Wenkai Yang, Xiaohan Bi, Yankai Lin, Sishuo Chen, Ji Zhou, and Xu Sun. 2024. Watch Out for Your Agents! Investigating Backdoor Threats to LLM-Based Agents. *arXiv:2402.11208* [cs.CR]. <https://arxiv.org/abs/2402.11208>
- [103] Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems 35* (2022), 20744–20757.
- [104] Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024.  $\tau$ -bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. doi:10.48550/arXiv.2406.12045
- [105] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*. <https://arxiv.org/abs/2210.03629v3>
- [106] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. doi:10.48550/arXiv.2210.03629
- [107] Asaf Yehudai, Lilach Eden, Alan Li, Guy Uziel, Yilun Zhao, Roy Bar-Haim, Arman Cohan, and Michal Shmueli-Scheuer. 2025. Survey on Evaluation of LLM-based Agents. doi:10.48550/arXiv.2503.16416
- [108] Sheng Yin, Xianghe Pang, Yuanzhuo Ding, Menglan Chen, Yutong Bi, Yichen Xiong, Wenhao Huang, Zhen Xiang, Jing Shao, and Siheng Chen. 2024. SafeAgentBench: A Benchmark for Safe Task Planning of Embodied LLM Agents. *arXiv preprint arXiv:2412.13178* (2024).
- [109] 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? doi:10.48550/arXiv.2407.15711
- [110] Erxin Yu, Jing Li, Ming Liao, Siqi Wang, Zuchen Gao, Fei Mi, and Lanqing Hong. 2024. Cosafe: Evaluating large language model safety in multi-turn dialogue coreference. *arXiv preprint arXiv:2406.17626* (2024).
- [111] Yangyang Yu, Zhiyuan Yao, Haohang Li, Zhiyang Deng, Yuechen Jiang, Yupeng Cao, Zhi Chen, Jordan Suchow, Zhenyu Cui, Rong Liu, Zhaozhuo Xu, Denghui Zhang, Koduvayur (Suba) Subbalakshmi, GUOJUN XIONG, Yueru He, Jimin Huang, Dong Li, and Qianqian Xie. 2024. FinCon: A Synthesized LLM Multi-Agent System with Conceptual Verbal Reinforcement for Enhanced Financial Decision Making. In *Advances in Neural Information Processing Systems 37* (Eds., A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang), Vol. 37. Curran Associates, Inc., 137010–137045.
- [112] Tongxin Yuan, Zhiwei He, Lingzhong Dong, Yiming Wang, Ruijie Zhao, Tian Xia, Lizhen Xu, Binglin Zhou, Fangqi Li, Zhusheng Zhang, et al. 2024. R-judge: Benchmarking safety risk awareness for llm agents. *arXiv preprint arXiv:2401.10019* (2024).
- [113] Qiusi Zhan, Richard Fang, Henil Shalin Panchal, and Daniel Kang. 2025. Adaptive Attacks Break Defenses Against Indirect Prompt Injection Attacks on LLM Agents. *arXiv:2503.00061* [cs.CR]. <https://arxiv.org/abs/2503.00061>
- [114] Andy K. Zhang, Neil Perry, and Riya Dulepet et al. 2024. Cybench: A Framework for Evaluating Cybersecurity Capabilities and Risks of Language Models. *arXiv:2408.08926v3* (2024). <https://arxiv.org/pdf/2408.08926.pdf>
- [115] Chaoyun Zhang, Shilin He, Jiaxu Qian, Bowen Li, Lijun Li, Si Qin, Yu Kang, Minghua Ma, Guyue Liu, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. 2025. Large Language Model-Brained GUI Agents: A Survey. doi:10.48550/arXiv.2411.18279
- [116] Danyang Zhang, Lu Chen, and Kai Yu. 2023. Mobile-env: A universal platform for training and evaluation of mobile interaction. *CoRR* (2023).
- [117] Danyang Zhang, Zhennan Shen, Rui Xie, Situo Zhang, Tianbao Xie, Zihan Zhao, Siyuan Chen, Lu Chen, Hongshen Xu, Ruisheng Cao, and Kai Yu. 2024. Mobile-Env: Building Qualified Evaluation Benchmarks for LLM-GUI Interaction. *arXiv:2305.08144* [cs.AI]. <https://arxiv.org/abs/2305.08144>
- [118] Hanrong Zhang, Jingyuan Huang, Kai Mei, Yifei Yao, Zhenting Wang, Chenlu Zhan, Hongwei Wang, and Yongfeng Zhang. 2024. Agent security bench (asb): Formalizing and benchmarking attacks and defenses in llm-based agents. *arXiv preprint arXiv:2410.02644* (2024).
- [119] Yuntong Zhang, Haifeng Ruan, Zhiyu Fan, and Abhik Roychoudhury. 2024. AutoCodeRover: Autonomous Program Improvement (*ISSTA 2024*). Association for Computing Machinery, New York, NY, USA, 1592–1604. doi:10.1145/3650212.3680384
- [120] Yue Zhang, Ming Zhang, Haipeng Yuan, Shichun Liu, Yongyao Shi, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. LLMEval: A Preliminary Study on How to Evaluate Large Language Models. *arXiv:2312.07398* [cs.AI]. <https://arxiv.org/abs/2312.07398>
- [121] Zeyu Zhang, Xiaobei Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2024. A Survey on the Memory Mechanism of Large Language Model based Agents. doi:10.48550/arXiv.2404.13501
- [122] Zhenxin Zhang, Shiyao Cui, Yida Lu, Jingzhuo Zhou, Junxiao Yang, Hongning Wang, and Minlie Huang. 2024. Agent-SafetyBench: Evaluating the Safety of LLM Agents. *arXiv preprint arXiv:2412.14470* (2024).
- [123] Zhiping Zhang, Michelle Jia, B Yao, S Das, A Lerner, D Wang, and T Li. 2023. It's a fair game, or is it? examining how users navigate disclosure risks and benefits when using llm-based conversational agents. *arXiv preprint arXiv:2309.11653* (2023).
- [124] Ziniu Zhang, Shulin Tian, Liangyu Chen, and Ziwei Liu. 2024. MMINa: benchmarking multihop multimodal internet agents. doi:10.48550/arXiv.2404.09992
- [125] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems 36* (2023), 46595–46623.
- [126] 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* (2023).
- [127] Mingchen Zhuge, Changsheng Zhao, Dylan Ashley, Wenyi Wang, Dmitrii Khizbulin, 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* (2024).