https://doi.org/10.1007/s11704-026-51590-2

REVIEW ARTICLE

Evolutionary perspectives on the evaluation of LLM-based AI agents: a comprehensive survey

Jiachen ZHU[1*], Menghui ZHU[2*], Renting RUI[1], Rong SHAN[1], Congmin ZHENG[1], Bo CHEN[2], Yunjia XI[1], Jianghao LIN[3], Weiwen LIU[1], Ruiming TANG[2], Yong YU[1], Weinan ZHANG[1]✉

1. Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
2. Huawei Noah's Ark Lab, Shanghai 201700, China
3. Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai 200240, China

Received September 30, 2025; accepted January 11, 2026

E-mail: wnzhang@sjtu.edu.cn. * These authors contributed equally to this work.

© The Author(s) 2026. This article is published with open access at link.springer.com and journal.hep.com.cn

Abstract

The advent of large language models (LLMs), such as GPT, Gemini, and DeepSeek, has significantly advanced natural language processing, giving rise to sophisticated chatbots capable of diverse language-related tasks. The transition from these traditional LLM chatbots to more advanced AI agents represents a pivotal evolutionary step. However, existing evaluation frameworks often blur the distinctions between LLM chatbots and AI agents, leading to confusion among researchers selecting appropriate benchmarks. To bridge this gap, this paper introduces a systematic analysis of current evaluation approaches, grounded in an evolutionary perspective. We provide a detailed analytical framework that clearly differentiates AI agents from LLM chatbots along five key aspects: complex environment, multi-source instructor, dynamic feedback, multi-modal perception, and advanced capability. Further, we categorize existing evaluation benchmarks based on external environments driving forces, and resulting advanced internal capabilities. For each category, we delineate relevant evaluation attributes, presented comprehensively in practical reference tables. Finally, we synthesize current trends and outline future evaluation methodologies through four critical lenses: environment, agent, evaluator, and metrics. Our findings offer actionable guidance for researchers, facilitating the informed selection and application of benchmarks in AI agent evaluation, thus fostering continued advancement in this rapidly evolving research domain.

Keywords

AI agent evaluation; large language models; evolutionary perspective; evaluation taxonomy; benchmark selection

1 Introduction

The advent of Transformer [1] has revolutionized natural language processing (NLP) and enabled large language models (LLMs) for chatbots, such as GPT [2], LLaMA [3], Gemini [4], Qwen [5], and DeepSeek [6], achieving unprecedented performance across diverse text-based tasks. These models, trained on massive corpora, exhibit emergent capabilities in text generation, comprehension, and reasoning. Their ability to generalize across domains has positioned LLM chatbots as the foundation for modern AI systems, ranging from conversational interfaces to knowledge-intensive problem-solving.

The emerging AI agents mark a further significant evolution beyond traditional LLM chatbots by enabling rich environmental interaction and broader functionality [7–9]. Unlike chatbots, which only respond to human prompts in isolation, AI agents can interact with the web [10], invoke APIs [11], and adapt based on real-world feedback, allowing them to handle more complex tasks [12]. Essentially, the transition from LLM chatbots to fully functional AI agents is an evolutionary process. Therefore, it is imperative to first clearly delineate the advancements that have occurred during this evolution. As shown in Fig. 1, LLM chatbots operate as reactive conversational engines, isolated from their surroundings and dependent solely on human input. In contrast, AI agents can be systematically delineated across five primary dimensions: complex environments, multi-source instructors, dynamic feedback, multi-modal perception and advanced capabilities. Details for the evolution from LLM chatbots to AI agents are illustrated in Section 2.

Such a huge evolution of AI agents necessitates new philosophical and methodological approaches to AI agent evaluation, which raises our core question in this paper:

Given the rapid advancement of AI agents, how can we systematically evaluate LLM-based AI agents from an evolutionary perspective?

In response to this question, numerous evaluation benchmarks for

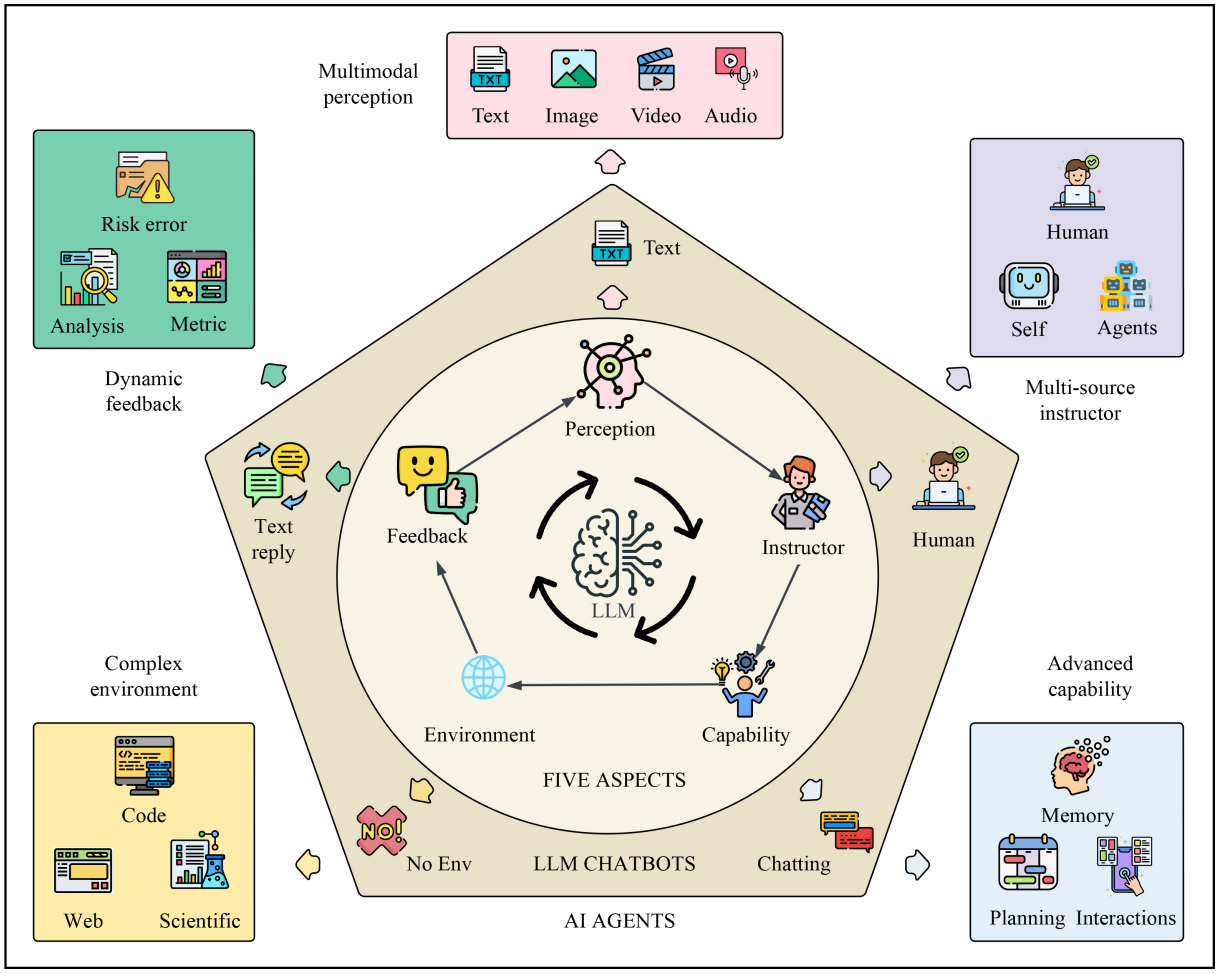


Fig. 1 Evolution between AI agents and LLM chatbots in five key aspects

AI agents have emerged, alongside a growing number of surveys dedicated to LLM evaluation [13−15]. However, these works tend to focus either solely on LLM chatbots or provide only broad overviews of AI agent benchmarks without clearly defining the key distinctions between LLM chatbots and AI agents. Furthermore, they often overlook the evolutionary process from chatbot to agent, as well as the nuanced differences among evaluation benchmarks. As a result, researchers still face substantial ambiguity when selecting appropriate benchmarks to evaluate specific types of agents amidst a rapidly expanding landscape of evaluation tools.

To address these gaps, we adopt an evolutionary perspective, illustrating that the development of AI agents is fundamentally driven by the interplay between **external environmental driving forces** and the resulting advanced **internal capabilities**. Accordingly, as shown in Fig. 2, we systematically categorize and synthesize the landscape of existing AI agent evaluation work using the dual lenses of Environment and Capability—the two most critical dimensions in the agent evolution process.

For the external environments where agents operate, we identify several key categories in Section 3: coding environments, web environments, operating system (OS) environments, mobile environments, scientific environments, and game environments. In terms of internal capabilities in Section 4, we identify planning, memory, self-reflection, and interaction as core competencies, and further discuss the current landscape of general capability evaluation. Moreover, for each environment and capability, we summarize a set of valuable attributes for evaluation. These attributes form the basis for detailed tables provided in Appendix A, which serve as practical references for future researchers selecting benchmarks for agent evaluation.

Furthermore, Section 5 offers a synthesis and future outlook on the evolving trends in evaluation methodology from four distinct perspectives to address a central question facing the research community: *When a new agent is developed, how should the appropriate evaluation benchmark be chosen*? We explore this question from both a present-oriented and a forward-looking perspective, aiming to provide actionable guidance for the continued advancement of agent evaluation research.

The main contributions are summarized as follows:

- We propose an analytical framework to distinguish AI agents from LLM chatbots from five key aspects (i.e., environment, instructor, feedback, perception, and capability), systematically characterizing the evolutionary progression from simple chatbots to advanced AI agent systems.
- We categorize existing evaluation benchmarks for AI agents along two critical axes: external environments and internal

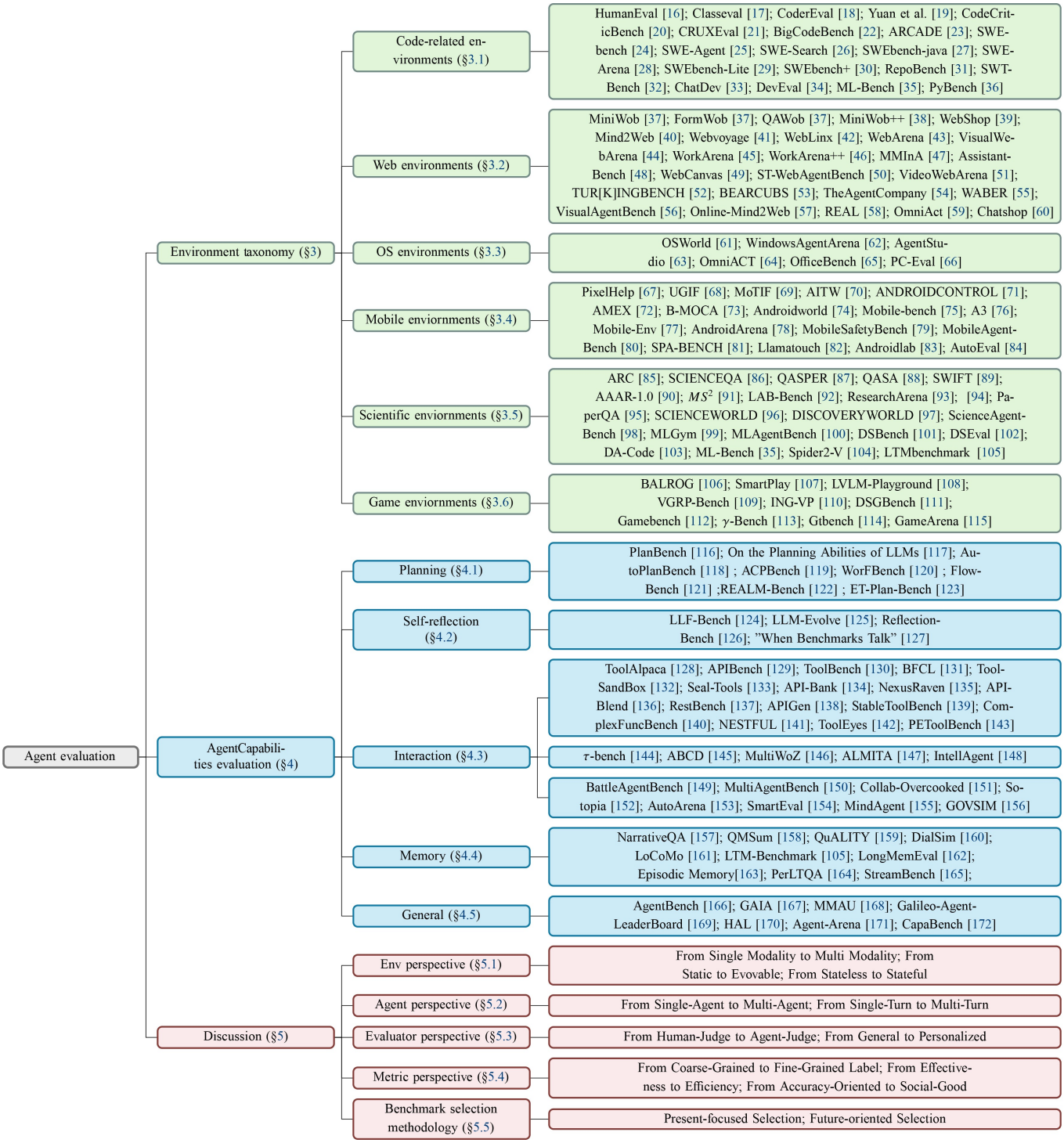


Fig. 2 The overall structure of this paper

capabilities. For each category, detailed attribute tables and discussions are provided to further benefit the research community.

- We analyze the evolutionary trends in evaluation benchmarks across four aspects: environment, agent, evaluator, and metric. Based on these trends, we discuss future directions and offer practical guidelines on selecting appropriate benchmarks, supporting continued progress in agent evaluation research.

2 From LLM chatbots to AI agents: preliminary

In this section, we discuss the evolutionary progression from LLM chatbots to AI agents from five key aspects, which leads to the taxonomy on agent evaluation benchmarks in later sections.

2.1 Background: large language model (LLM) chatbots

The advent of Transformer architectures has laid the foundation for the current era of large language models (LLMs). These models, trained on vast and diverse textual corpora, have demonstrated

remarkable performance in language understanding [2], generation [172], and reasoning [6]. Within this landscape, the LLM chatbot paradigm has emerged as a dominant application. LLM chatbots are designed as reactive conversational systems that receive textual prompts from users, process these inputs using pretrained knowledge, and generate coherent textual responses. Typical use cases include open-domain dialogue, customer support, question answering [6]. Despite their versatility, traditional LLM chatbots are limited in that they operate in closed environments, devoid of active perception or real-world context. Their interactions are fundamentally constrained to static, turn-based exchanges, lacking the means to sense or affect their operational environment.

2.2 The emergence and definition of AI agents

Recent advances have shifted the focus from reactive LLM chatbots to proactive AI agents. AI agents can actively interact with the environment and make decisions by processing language, perceiving, reasoning about multi-modal inputs, and acting within dynamic contexts. Representative examples include web search agents [10], API-calling agents for online tasks [11], and code agents that iteratively debug based on execution feedback [12]. Figure 1 summarizes five key distinctions—perception, instructor, capability, environment, and feedback—showcasing how AI agents advance far beyond traditional chatbots in scope and intelligence.

Complex environment. The most fundamental distinction lies in the environment dimension, which serves as the primary external driving force behind agent evolution. Traditional LLM chatbots are confined to closed environments, interacting solely with humans through static dialogue and lacking awareness or control over their surroundings. In contrast, AI agents operate within diverse and complex environments, e.g., software platforms, scientific computing, Internet ecosystems and operating systems. This enables them to interpret dynamic contexts and take actions that affect the external world, transforming them from passive responders into proactive collaborators and task executors.

Multi-source instructor. AI agents also advance the instructor dimension. Unlike LLM chatbots that depend heavily on human prompts, agents integrate instructions from multiple sources, including self-reflection, collaboration with other agents, and hierarchical commands in multi-agent systems. This multi-source guidance empowers agents to make complex decisions, self-correct, resulting in higher autonomy and robustness.

Dynamic feedback. While LLM chatbots primarily receive feedback through conversation or user correction, AI agents operate in environments with continuous, multifaceted feedback—including metric-based analyses, risk assessments, explicit error signals, and environment-derived rewards or penalties. This rich feedback ecosystem enables ongoing adaptation, self-improvement, and long-term optimization.

Multimodal perception. To function in real-world settings and respond to complex instructions, AI agents are equipped with multimodal sensing —processing not just text, but also visual, auditory, and even tactile or environmental sensor data. The development of Multimodal Large Language Models (MLLMs) exemplifies this leap, allowing agents to understand and reason across diverse modalities and vastly expanding their intelligence and applicability.

Advanced capability. Advancements across environment, instructor, feedback, and perception dimensions collectively drive the evolution of agents' internal capabilities. Dynamic environments, richer instructions and feedback, and enhanced perception propel AI agents far beyond basic conversations. Agents now exhibit complex planning, persistent memory, adaptive reasoning, and autonomous task execution. This marks a transformative step in intelligent systems, demonstrating how external demands and internal advances together drive the shift from reactive LLM chatbots to autonomous AI agents.

The progression from LLM chatbot to AI agent is primarily catalyzed by two forces: **external environment** complexity and the corresponding development of **internal capabilities**. External driving forces stimulate internal growth. As agents are deployed in richer, more demanding environments, they must evolve to incorporate sophisticated perception, planning, and adaptation mechanisms. This external-internal interplay underlies the rapid advancement of modern AI agents. In the next sections, as shown in Fig. 2, we categorize and discuss agent evaluation from both external environment and internal capability perspectives.

As previously discussed, the evolution of AI agents compared to LLM chatbots is manifested in five key aspects. To construct a systematic and clear evaluation benchmark taxonomy, we select "Complex Environment" and "Advanced Capability" as the two core dimensions for our main classification axes. This is because the "Environment" constitutes the primary external driving force behind agent evolution, while "Capability" represents the direct manifestation of internal evolution. The other three aspects ("Multimodal Perception," "Dynamic Feedback," and "Multi-source Instructor") are closely embedded within these two main axes. For instance, "Multimodal Perception" and "Dynamic Feedback" are typically inherent attributes and evaluation dimensions of a "Complex Environment" (such as the Web Environment in Subsection 3.2 or the Mobile Environment in Subsection 3.4). Meanwhile, "Multi-source Instructor" serves as a prerequisite for realizing and evaluating "Advanced Capability" (such as the Self-reflection in Subsection 4.2 or the Interaction in Subsection 4.3). Therefore, classifying benchmarks based on the twin pillars of environment and capability allows for the most systematic synthesis of existing evaluation work.

3 Evaluation for different environments

The environment, as an abstraction and simplification of the world, plays a pivotal role in the evolution from LLM chatbots to AI agents. It is an external system that the agent interacts directly with. Agents are equipped with different types of environments tailored to task categories and complexity levels, enabling capabilities such as self-reflection and multi-step planning. In this section, a systematic categorization and analysis of the environments employed in various agent evaluations is presented. The objective is to delineate their

developmental trajectory and discern the similarities and distinctions among different environments.

3.1 Coding environments

In the context of general-purpose agents, codes represent a pivotal means of interactive engagement with the external environment and exerting influence on it. Consequently, the coding abilities are of paramount importance for agents. The common environments for code agents include referenced code files, code repositories, code executors and so on, which are determined by specific tasks.

It is worthy of mentioning that a number of the earlier works are not related to code environments, in which case the benchmarks are only evaluated on simple tasks and the agents involved are more akin to code LLMs. We will only briefly list these benchmarks and describe the simple tasks on which they were evaluated [16–17−22,173]. As a pioneer of this area, HumanEval [16] tests the LLMs by asking them to synthesise programs from docstrings. Likewise, CoderEval [18] enhances HumanEval mainly on the complexity of the tasks, especially on the cyclomatic complexity and lines of codes. Similarly, CodeCriticBench [20] assesses LLMs' aptitude for code critique, with a particular emphasis on code generation and quality assurance (QA) tasks. BigCodeBench [22] enhances HumanEval [16] and offers more challenging and practical tasks, which has higher requirements for LLMs like utilizing diverse function calls as tools. ARCADE [23] is also an evaluation on code generation, but it pays attention to interactive notebooks in data science.

As code-related tasks become more complex and comprehensive, it is no longer enough for agents to take in only input from the instructors. They need to operate on more objects and performs skills such as self-reflection, where the environment provides the basis for the agents.

To better understand the environments in code agents, here we take the representative benchmark on code agents SWE-bench [24] as the example. In SWE-bench, agents are prompted to understand the real issues of GitHub and generate the code patches to resolve the issues. During inference, related codes and demonstrations are inserted into the prompts beforehand and LLMs directly generate the patches in text, where the models behave more like semi-agents. The environment now only includes the codebases and the automated unit tests. In original SWE-bench paper, the best-performing model, Claude 2, though equipped with a simple retrieval mechanism, could only solve 1.96% of the issues. After that, in order to achieve better performances on these tasks, a series of agent frameworks are proposed to challenge the tasks on SWE-bench such as SWE-Agent [25] and SWE-Search [26]. To support the agents' abilities such as memory and self-reflection, an external terminal is added into the environment, through which the agents could perform some higher-level operations like file editing and searching. Except SWE-bench, a series of similar benchmarks are proposed and their environments differs from that in SWE-bench mainly on the modal [25], programming language [27], agents' comparison [28], execution-free evaluation [29], and data quality [30].

In addition, in the field of code agents, there are various benchmarks with different environments towards different specific code-related tasks. RepoBench [31] evaluates on the tasks of code retrieval and code completion and provides the agents with the code repositories as the environment. Similarly, for the task of code completion at the repository level, the environment in RepoEval [174] provides the capability of retrieval. This is the basis of the iterative retrieval-generation mechanism in the paper. SWT-Bench [32] is based on SWE-bench and adopts similar environments to benchmark agents on the generation of test cases according to the issues. Based on ChatDev [33], DevEval's [34] environments allow to evaluate agent on the full stages of software development, including software design, environment setup, implementation and testing. ML-Bench [35] offers a ML-related repository and a Linux sandbox and ask the agents to perform machine learning tasks. Pybench [36] introduces a Python code interpreter to the environment. The agents in the benchmark operate on the different type of files in the environment to complete some real-world tasks like chart analysis, text analysis, image & audio editing and so on.

In summary, evaluation benchmarks for coding environments have evolved from static code snippet generation (e.g., HumanEval [16]) to dynamic, interactive repository-level problem solving (e.g., SWE-bench [25]), as shown in Table 1. The environment itself has developed from simple code files into complex sandboxes containing codebases, terminals, interpreters, and test frameworks. The core evaluation focus has also expanded from mere code correctness (Pass@k) to measuring the "Resolved Rate" of real-world tasks and task completion across the full software development lifecycle.

3.2 Web environments

Web agents, autonomous systems that navigate, interact with, and extract information from web pages, have emerged as a powerful paradigm for tasks ranging from form filling and information retrieval to complex multi-step workflows on enterprise platforms. Evaluating their performance is crucial for monitoring progress and identifying remaining challenges. Over the past few years numerous benchmarks have been proposed; we summarize them in Table 2 and organize our discussion by the *realism* of the environment.

Early benchmarks use fully synthetic web pages, designed to stress reinforcement-learning agents in a controlled setting. Two pioneering works are MiniWoB [37] and MiniWoB++ [38]. MiniWoB provides a suite of simple tasks (e.g., clicking buttons, filling out toy mail and calendar forms) rendered in canvas elements. Its synthetic nature allows reproducible offline RL training and evaluation. MiniWoB++ extends MiniWoB with longer sequences, random layouts, and soft-text reasoning tasks (e.g., checkbox grids, multi-layout navigation, simple social-media interactions), further challenging sequence modeling and exploration. These early synthetic suites established the importance of standardized evaluation for web agents and paved the way for more realistic benchmarks.

As language-model-based agents matured, purely synthetic data proved insufficient [175–176]. Semi-real benchmarks load snapshots of real websites and host simplified or anonymous modifications on data replicas to afford reproducible evaluation on genuine layouts and data. FormWoB [37] offers form-filling tasks to assess agents'

Table 1 Code benchmarks and their taxonomy

Code environment/ Dataset	Static/ Interactive	Tasks	Evaluation metric	Evaluation dimension
HumanEval [16]	Static	Code generation from docstring	Pass@k	Code correctness
Classeval [17]	Static	Class-level code generation	Pass@k, Method/Field Dependencies	Code correctness, Independency to the contexts
[19]	Static	Defect detection, Clone detection, Assert generation, Code summarization	Accuracy, F1 score, EM(Exact Match)	Correctness of detection, assert and summarization
CodeCriticBench [20]	Static	Code generation, Code QA	Accuracy, MSE, Pass@1	Code correctness, Code evaluation accuracy
CRUXEval [21]	Static	Code input and output prediction	Pass@k	Code prediction correctness
BigCodeBench [22]	Static	Code generation with tools	Pass@k	Code correctness
ARCADE [23]	Static	Code generation in interactive data science notebooks	Pass@k	Code correctness
SWE-bench [24]	Interactive	Code generation for resolving the issues in the repository	Resolved Rate	Correctness of patch files
Swe-bench M [25]	Interactive	Code generation for resolving the issues in the repository with images	Resolved Rate, Average cost	Correctness of patch files
SWE-bench-java-verified [27]	Interactive	Code generation for resolving the issues in the repository	Resolved Rate	Correctness of patch files
Swe arena [28]	Interactive	Code generation for resolving the issues in the repository	Resolved Rate	Correctness of patch files
Swe-bench+ [30]	Interactive	Code generation for resolving the issues in the repository	Resolved Rate, Average Cost	Correctness of patch files
RepoBench [31]	Interactive	Code retrieval and code completion on code repositories	Accuracy, Exact Match, Edit Similarity	Accuracy of code retrieval, Accuracy of code completion
SWT-Bench [32]	Interactive	Code generation for resolving the issues in the repository	Success Rate, Line Coverage,	Success rate, Change coverage, Patch well-Formedness
DevEval [34]	Interactive	Software design, environment setup, implementation and testing	LLM-as-a-judge, Success/Pass Rate	General principles and faithfulness for software design, Correctness for environment setup, Correctness of implementation, Correctness and code coverage of testing
ML-Bench [35]	Interactive	Machine learning tasks like package installing, model code generation	Pass@k	Completion of machine learning tasks
Pybench [36]	Interactive	Python code generation for Chart Analysis, Text Analysis, Image & Audio Editing, Complex Math and Software & Website Development	Pass Rate, Average Turns, LLM-as-a-judge	Completion of python related tasks

abilities in handling structured inputs. QAWoB [37] designs question-answering tasks to test agents' information retrieval and comprehension skills. WebShop [39] simulates online shopping scenarios, challenging agents in multi-step decision-making and understanding user preferences. ChatShop [60] combines conversational systems with shopping tasks, evaluating agents' consistency and task completion in multi-turn dialogues. WebArena [43] constructs simulated environments based on real websites, encompassing various web interaction tasks to test agents' generalization capabilities. Its variants VisualWebArena [44] and VideoWebArena [51] incorporates visual information and video tutorials, respectively, challenging agents' performance in processing mixed text, image content and long-context videos. WABER [55] provides a multilingual, multi-task evaluation environment to test agents' cross-lingual capabilities and task adaptability. REAL [58] recreates eleven high-fidelity website simulations with deterministic snapshots and combines script-based state validation with an LLM-based rubric for open-ended retrieval tasks. ST-WebAgentBench [50] offers a framework for evaluating agents in multi-turn dialogues and complex tasks, testing long-term planning and context understanding. TUR[K]INGBENCH [52] uses original Mechanical Turk HTML pages, evaluating agents on naturally crowdsourced web tasks with metrics like ROUGE-L and IoU, underscoring diversity of real-world page designs. TheAgentCompany [54] simulates a small software company environment, focusing on agents' performance in real-world professional scenarios.

In contrast to the aforementioned semi-realistic benchmarks, a growing number of fully dynamic and real-world benchmarks have recently been proposed. These benchmarks either encapsulate real environments into snapshot-based simulations or directly connect to the live Internet for task execution. Such evaluation paradigms better reflect real-world complexities and are more effective at assessing agents' actual performance and robustness in dynamic conditions. Mind2Web [40] requires agents to understand rich page structures

Table 2 Web benchmarks and their taxonomy

Web environment	Realism	Offline/ Online	Metrics	Evaluator	Input modality	Main challenge
MiniWoB [37]	Synthetic	Offline	Task Success Rate	Human labels + Automated evaluation	Text + Images	A reproducible offline RL benchmark for open-domain web tasks
FormWoB [37]	Semi-real	Offline	Average Reward	Human labels + Automated evaluation	Text + Images	Same as above
QAWoB [37]	Semi-real	Offline	Test Trajectory Likelihood	Human labels + Automated evaluation	Text + Images	Same as above
MiniWoB++ [38]	Synthetic	Offline	Task Success Rate	Human labels + Automated evaluation	Text + Images	Added longer sequences, random layouts, and soft-text reasoning tasks
WebShop [39]	Semi-real	Offline	Independent scoring on attributes/options/price/ type	Human labels + Automated evaluation	Text + Images	Requires agents to complete full purchase flow
mind2web [40]	Real (snapshot replay)	Offline	Element Accuracy, Operation F1	Human labels + Automated evaluation	Text + Images	Very large number of web pages
webvoyage [41]	Real (live connection)	Online	Human vs GPT-4V Agreement	Human +LLM joint evaluation	Text + Images	Online real environment judged directly by humans
WEBLINX [42]	Real (snapshot replay)	Offline	IoU, URL-F1	Human labels + Automated evaluation	Text + Images	Introduces “conversational web navigation”
webarena [43]	Semi-real	Offline	Intermediate page states	Human labels + Automated evaluation	Text + Images	Self-hosted multi-site environment that closely mirrors real websites
visualwebarena [44]	Semi-real	Offline	Task Success Rate	Human labels + LLM-based similarity	Text + Images	Visual-dependent tasks: “see then do” scenarios
WorkArena [45]	Real (live connection)	Online	Task Success Rate	Fully automated by real web	Text + Images	Runs on official ServiceNow developer instance; fully realistic
WorkArena++ [46]	Real (live connection)	Online	Task Success Rate	Fully automated by real web	Text + Images	Evaluates “compositional planning and reasoning.”
MMInA [47]	Real (live connection)	Online	Hop Success Rate	Human labels + LLM-based similarity	Text + Images	Multi-hop (cross-site chain operations)
AssistantBench [48]	Real (live connection)	Online	Task Success Rate	Human labels + LLM-based similarity	Text + Images	Time-consuming, multi-step workflows
WebCanvas [49]	Real	Online	Task Success Rate	Human labels + Automated evaluation	Text + Images	Continuously updated with robust intermediate checks
ST-WebAgentBench [50]	Semi-real	Online	Completion-under-Policies, Risk Ratio	Human labels + Automated evaluation	Text + Images	Policy compliance focus under risk constraints
VideoWebArena [51]	Semi-real	Online	Video QA Accuracy	Human labels + Automated evaluation	Video + Text + Images	Long-video + web interaction (Skill vs Factual tasks)
TUR[K]INGBENCH [52]	Semi-real	Offline	ROUGE-L, IoU	Human labels + Automated evaluation	Text + Images	Natural HTML pages originally designed for crowdsourcing
BEARCUBS [53]	Real	Online	error rate	Human labels + Automated evaluation	Video + Text + Images	Fine-grained answer labeling with live multimedia items
THEAGENTCOMPANY [54]	Semi-real	Offline	Full-Completion Score	Human + Automated + LLM similarity	Text + Images	Performing real company tasks
WABER [55]	Semi-real	Online	Reliability, Efficiency	Human labels + Automated evaluation	Text + Images	Injected failures + efficiency benchmarking
VisualAgentBench [56]	Semi-real	Offline	Task Success Rate	Human labels + Automated evaluation	Text + Images	Multi-domain, multi-scenario LMM visual agent benchmark
Online-Mind2Web [57]	Real (live connection)	Online	Task Success Rate	Human labels + LLM-based similarity	Text + Images	Online adaptation of Mind2Web
REAL [58]	Semi-real	Offline	Rubric-Guided Retrieval Score	Human labels + LLM/Rubric similarity	Text + Images	Tests both robustness and reliability simultaneously
OmniAct [59]	Real	Offline	Sequence Score, Action Score	Human labels + Automated evaluation	Text + Images	Desktop + web automation with DetACT module
ChatShop [60]	Semi-real	Offline	RL Learning Curves (ChatShopBin only)	Human labels + Automated evaluation	Text + Images	Ambiguous initial info requiring strategic questioning

and plan multi-step goals. Online-Mind2Web [57] further extends this by introducing challenges related to dynamic content changes and real-time DOM variations. WebVoyager [41] emphasizes effective intermediate state modeling and reasoning over long-term dependencies between distant pages. WebLINX [42] focuses on reasoning through numerous branching paths and integrating historical click sequences to reach target information. WorkArena [45] and WorkArena++ [46] assess agents' long-context understanding, reflecting more realistic workflows encountered by office or workers in real enterprises. MMInA [47] introduces challenges in multimodal and multi-hop web tasks. AssistantBench [48] mainly tests agents' ability to follow time-consuming and multi-turn workflows. WebCanvas [49] poses challenges in visual parsing and UI element grounding under dynamic layouts by examining intermediate nodes. BEARCUBS [53] centers on knowledge-intensive tasks that require agents retrieve and utilize webpage content and external knowledge sources.

The evolution of web environment benchmarks is clearly reflected in the continuous enhancement of "Realism". Evaluation has progressed from early "fully synthetic" environments (e.g., MiniWoB [37]) to "semi-real" website snapshots (e.g., WebShop [39], WebArena [43]), and finally to "fully dynamic" benchmarks connected to the live Internet (e.g., WebVoyager [41], WorkArena [45]). The primary evaluation challenges lie in handling dynamic content, multimodal information (e.g., VisualWebArena [44]), and complex multi-step workflows requiring long-term planning.

3.3 OS environments

Despite the web environments, there are some benchmarks that are based on the desktop, which is more complex and challenging. OSWorld [61] serves as a representative desktop benchmark, demonstrating the optimal characteristics of a desktop environment. It employs virtual machines (VMs) to facilitate a desktop environment that is executable and controllable. The agents could interact with the environment just like operating the computer in reality. The environments are initialized by config files, which describes some operations like open a file after starting the virtual machine. During the interaction process with the environment, the observation of agents encompasses the capture of screenshots of the desktop and the accessibility (a11y) tree. The action space defined in OSWorld is diverse and it supports all mouse and keyboard actions, including movement, clicks and so on. There are totally 369 evaluated tasks in OSWorld, including Office tasks, OS tasks, Daily tasks, Workflow tasks and Professional tasks.

Furthermore, there are several distinct benchmarks that emphasize disparate fine-grained scenarios. Different from OSWorld, whose tasks are mainly on Ubuntu, WindowsAgentArena [62] offers a general environment focusing on the Windows operating system. AgentStudio [63] constitutes a triad of virtual agent research environments, tools, and benchmarks. AgentStudio's environment is conducive to the observation of text, image, and video formats. Its action space supports both GUI operations and API calls. OmniACT [59] provides a static dataset for the evaluation of generalist autonomous agents. This dataset contains tasks in natural language, screenshots, and PyAutoGUI scripts that serve as ground truth. The environment at OfficeBench [64] is characterized by its concentration on office tasks, which distinguishes it from the general desktop agent evaluation environments, such as OSWorld. The action space in OfficeBench is restricted and depends on the applications that agents are currently utilizing. For instance, the action of *run_command* could be operated in Shell but invalid in other applications like Word. In OfficeBench, agents could observe both the current and the previous states and actions, which is a departure from the settings in previous benchmarks. Although analogous to OSWorld, PC-Eval [65] augments it with more practical and challenging tasks, which are checked by human annotators.

The desktop (OS) environment is a crucial scenario for evaluating agent generalization, characterized by high environmental complexity, as shown in Table 3. Such benchmarks (e.g., OSWorld [61], WindowsAgentArena [62]) utilize virtual machines (VMs) to provide complex environments encompassing multiple applications and modalities. Agents are required to synthesize observational information such as screenshots and accessibility trees (a11y trees), and execute diverse operations including keyboard and mouse actions as well as API calls. The core evaluation point lies in the agent's task completion capability within open-ended tasks and cross-application office automation (e.g., OfficeBench [64]).

3.4 Mobile environments

The advent of mobile Internet has led to the gradual integration of mobile devices, such as smartphones and tablet computers, into our daily life. These devices have become increasingly indispensable, playing a crucial role in various aspects of our lives. However, the increasing functionality of mobile devices poses challenges to their usability. Many mobile agents such as Siri, Bixby and XiaoAi have been proposed to relieve the operation issues. Nevertheless, in general, there is still significant room for improvement in task completion capabilities, personalization, and other aspects for these agents. Therefore, many benchmarks are proposed to provide the testbeds and promote the development of mobile agents, as shown in Table 4.

The environments of earlier benchmarks are usually static, or in other way to say, are just fixed datasets to evaluate LLM chatbots. Early works such as PixelHelp [66], UGIF [67], MoTIF [68], and AITW [69] offer the LLMs with the screenshots and tree-based representations such as View Hierarchy (VH) and regard the demonstrations from the humans as ground truths to evaluate the agents. Among these benchmarks, the process of evaluation is static and does not need the involvement of executable environments, where the LLMs could not interact with the environments in multiple rounds. Besides the ones mentioned above, ANDROIDCONTROL [70] mainly improves on the granularity of the instructions and the freshness of the APPs. AMEX [71] further offers the screen descriptions and screen elements for the agents.

However, when the environments are asked to evaluate AI agents rather than LLM chatbots, things get a little different. The complexity of the tasks inherently determines the requirements of multi-step execution. In addition, the agents usually need multi-round

Table 3 OS Benchmarks and their taxonomy

Benchmark	Task	Observation	Action space	Platform	Evaluation method
OSWorld [61]	Office tasks, OS tasks, Daily tasks, Workflow tasks and Professional tasks	Screenshots, a11y-tree	Keyboard, Mouse	Ubuntu	Predefined post-processing scripts to calculate success rate
WindowsAgentArena [62]	Office, Web browsing, Windows system, Coding, Media & Video, Windows utilities	Task instruction, Clipboard content, Metadata of the current session, Representation of the current screen	Keyboard, Mouse	Windows	Predefined post-processing scripts to calculate success rate
AgentStudio [63]	Office tasks, OS tasks, Daily tasks and so on	Screen recording, Screenshots, Code execution outputs	Keyboard, Mouse, APIs/Tools	Linux, Windows, macOS	User-defined functional auto-evaluators based on reward and language feedback
OmniACT [59]	Shopping tasks, Entertainment tasks, Service tasks, Government tasks, Travel tasks, Health tasks	Screenshots	Actions in PyAutoGUI library	Linux, Windows, macOS	Action sequences matching, Inaccurate behavior penalty
OfficeBench [64]	Office tasks, like Word, Excel, PDF, Email and so on	Text description of current state and previous outputs of actions	APIs	Simulated desktop	Exact matching and fuzzy matching on the final outputs, Scripts to check the final state
PC-Eval [65]	Commom tasks on Chrome, Word, Excel and so on	Screenshots, Operation histories	Keyboard, Mouse, APIs	Windows	Human evaluation on success rate and subtask success rate

interactions to handle their inner mechanisms such as planning and self-reflection. Also, there are usually multiple trajectories to complete the tasks, which makes static datasets even less suitable for evaluating mobile agents. In the era of agents, the environments typically include an emulator and an adapter. The emulator, which is the core of the environment, behaves like a real mobile device and responds to the actions of the agents. For example, if the agents give the action of tapping and dragging up, the emulator will scroll down the screen. The adapter, on the other hand, acts as a bridge between the emulator and the agents. When the agents perform actions, the adapter translates the agents' instructions into shells and passes them to the emulator through tools such as ADB. Similarly, it gets the states of the environment such as the screenshot and XML from the emulator and converts them into the format that the agents can understand.

With the above background knowledge, we now proceed to a detailed characterization of existing evaluations. B-MOCA [72] is an earlier benchmark with dynamic environments and includes a simple range of daily tasks like web tasks, shopping tasks, system tasks and so on. Androidworld [73] expands the scope of APPs. It is equipped with the ability to dynamically construct the initial states of tasks and vary the task parameters in unlimited ways. Mobile-bench [74] extends the agent's action space with API operations, allowing the agent to perform tasks such as opening a specific APP without screen operations. Additionally, to evaluate the agents more accurately, it proposes checkpoints to evaluate the agents on whether reach essential points during the planning and reasoning steps. A3 [75] focuses on the daily used APPs and offers a more flexible action space for the agents, which supports additional actions like Long Press. The environments in Mobile-Env [76] supports intermediate rewards and intermediate instructions for the agents. AndroidArena [77] pays more attention to the cross-APP tasks and constrained tasks. The environment of cross-APP tasks involves multiple APPs and the agents require cooperation between multiple APPs. Besides for the regular instructions, the environment should also offer constraints such as *"do not click the payment button"*. MobileSafetyBench [78] develops a set of safety-related tasks to evaluate the mobile agents, such as messaging and banking applications.

In addition to the aforementioned benchmarks, there are works that mainly improve the evaluation methods. This is also a key point of the environments. MobileAgentBench [79] utilizes Android Accessibility Service to capture app events and uses them to check whether the task is completed. SPA-BENCH [80] proposes a coarse-to-fine success detection pipeline to judge if an agent succeeds. It first applies coarse detection with Optical Character Recognition (OCR) and matches the trajectories with the key components. After that, it deploys a trained multimodal LLM for final success determination. Llamatouch [81] automatically evaluates agents by whether the agent traverses all manually annotated application / system states. Similarly, in Androidlab [82], each task is divided into multiple page states as sub-goals. A task is considered complete only if all sub-goals are correctly addressed. AutoEval [83] adopts an automatic judge system to evaluate the agents, which composed of three components: Capturer, Reasoner, and Checker. The Capturer generates descriptions about the screenshots and the Reasoner judge the agent's performance according to the screen and task descriptions. Finally, the Checker is added to guarantee the Reasoner's output is acceptable and consistent with the reasoning process.

Evaluation benchmarks for mobile environments demonstrate a clear shift from "static datasets" to "executable simulators." Early works (e.g., PixelHelp [66]) primarily evaluated model understanding of static screenshots, whereas modern benchmarks (e.g., Androidworld [73], Mobile-Env [76]) provide dynamic

Table 4 Mobile benchmarks and their taxonomy

Benchmark	Static/ Executable	Task	Observation type	Action space type	Platform	Evaluation method
PixelHelp [66]	Static	Instructions following	Text view hierarchy	Screen operations	Android	Action sequences match
UGIF [67]	Static	Instructions following	Text view hierarchy	Screen operations + Macro Functions	Android	Action sequences match
MoTIF [68]	Static	Instruction following and instruction feasible classification	Screenshots + Text view hierarchy	Screen operations	Android	Calculate F1-score on the binary classification problem
AITW [69]	Static	Instructions following	Screenshots + Screen representation	Screen operations	Android	Action sequences match + Human validation
ANDROIDCONTROL [70]	Static	Instructions following	Screenshots + Text view hierarchy	Screen operations + Macro functions	Android	Action sequences match
AMEX [71]	Static	Instructions following	Screenshots + Screen representation (description, element grounding)	Screen operations	Android	Action sequences match
B-MOCA [72]	Executable	Instructions following	Screenshots + Text view hierarchy	Screen operations	Android	Self-developed success detector with ADB and Appium
Androidworld [73]	Executable	Instructions following	Screenshots + Text view hierarchy	Screen operations + APIs	Android	Inferring task success with ADB
Mobile-bench [74]	Executable	Instructions following	HTML converted from XML	Screen operations + APIs	Android	Checkpoints + GPT4 evaluate task completion status
A3 [75]	Executable	Instructions following	Screenshots + Text view hierarchy	Screen operations	Android	Task-specific evaluation function (element matching, action matching) + LLM Evaluation System
Mobile-Env [76]	Executable	Instructions following	Screenshots + Text view hierarchy + Set-of-Marks	Screen operations	Android	Evaluation from screen text, VH, system log and RHU
AndroidArena [77]	Executable	Cross-APP instructions following	Screenshots + Text view hierarchy	Screen operations + APP-level actions + System-level actions	Android	Action sequences match + GPT4 judgement on success
MobileSafetyBench [78]	Executable	Safety-related instructions following (e.g. Finance, Device/Data Management)	Screenshots + Text view hierarchy	Screen operations	Android	Rule-based evaluators on goal achievement and harm prevention
MobileAgentBench [79]	Executable	Instructions following	Screenshots + Text view hierarchy	Screen operations	Android	Final UI state matching with self-developed app
SPA-BENCH [80]	Executable	Single-APP and Cross-APP instructions following	Screenshots + Text view hierarchy	Screen operations	Android	Key components matching + MLLM evaluation
Llamatouch [81]	Executable	Instructions following	Screenshots + Text view hierarchy	Screen operations	Android	Evaluating on whether the agent traverses all manually annotated application/system state
Androidlab [82]	Executable	Instructions following	Screenshots + Text view hierarchy	Screen operations	Android	Evaluate on the completion of sub-goals
AutoEval [83]	Executable	Instructions following	Screenshots + Text view hierarchy	Screen operations	Android	LLM-based automatic judge system, including Capturer, Reasoner and Checker

simulators that support multi-turn interactions. Core challenges in this field include handling cross-APP tasks and ensuring operational safety. Concurrently, evaluation methods have become increasingly refined, evolving from action sequence matching to success detection based on UI states, checkpoints, or automatic evaluation systems (e.g., AutoEval [83]).

3.5 Scientific environments

The advancements of AI agents have sparked interest in building the entire automated process of scientific discovery or assist researchers in different researching stages, as shown in Table 5. Early benchmarks mainly assess the scientific reasoning and knowledge retrieval capabilities of LLM chatbots. For example, ARC [84] and

Table 5 Scientific benchmarks and their taxonomy

Scientific benchmark	Static/ Interactive	Environment	Evaluation metric	Task
ARC [84]	Static	None	Accuracy	Scientific question answering
SCIENCEQA [85]	Static	None	Accuracy	Multimodal scientific question answering
QASPER [86]	Static	None	Span-level F1 score	Research paper information seeking and answer anchoring
QASA [87]	Static	None	Precision, recall, F1 score, ROUGE scores	Question answering on scientific articles
SWIFT [88]	Static	None	Specificity, actionability, reading comprehension, overall helpfulness	Feedback generation for scientific writing
AAAR-1.0 [89]	Static	None	Equation inference: Accuracy; Other tasks: human expert ratings	Research assistance: equation inference, paper weakness, experiment design, review critique
MS^2 [90]	Static	None	ΔEI	Multi-document summarization of medical studies
LAB-Bench [91]	Static	None	Accuracy, precision, coverage	Multi-choice questions on biology research
ResearchArena [92]	Static	Academic and survey papers	Information retrieval: recall, precision; Information selection: nDCG, MRR; Information Organization: Heading Entity Recall, Heading Soft Recall	Academic surveying
[93]	Static	Research paper pool	Novelty, excitement, feasibility, expected effectiveness	Research idea generation
PaperQA [94]	Static	Scientific article pool	Retrieval AUC, retrieval probability	Scientific article question answering
SCIENCEWORLD [95]	Interactive	Simulated text environment	Task completion rate, step success rate, average completion steps, error rate, learning curve	Elementary-level scientific reasoning
DISCOVERYWORLD [96]	Interactive	Simulated text environment	Task completion rate, task-relevant actions taken, discovered explanatory knowledge	Perform complete cycles of novel scientific discovery
ScienceAgentBench [97]	Interactive	Workspace	Valid execution rate, success rate, codeBERTScore, API Cost	Data-driven scientific discovery
MLGym [98]	Interactive	Workspace	Area Under the Performance Profile (AUP) score	Realistic research workflows
MLAgentBench [99]	Interactive	Workspace	Success rate, consumed tokens and time	Machine learning experimentation tasks
DSBench [100]	Interactive	Workspace	Success rate, relative performance gap	Complex data science tasks
DSEval [101]	Interactive	Workspace	Accuracy, relative performance gap	Data analysis tasks and data modeling tasks throughout the entire data science lifecycle
DA-Code [102]	Interactive	Workspace	Accuracy	Data science code generation
ML-Bench [35]	Interactive	Workspace	Pass@k, success rate	Repository-level code understanding and execution tasks
Spider2-V [103]	Interactive	Workspace	Success rate	Automate data science and engineering workflows

SCIENCEQA [85] evaluate LLM chatbots on scientific multi-choice question answering. QASPER [86] and QASA [87] focus on information seeking and answer anchoring in research papers. SWIFT [88] measures the research paper comment generation and weakness identification abilities. AAAR-1.0 [89] evaluates LLM chatbots on research tasks of equation inference, experiment design, paper weakness, and review critique. MS^2 [90] emphasizes multi-document summarization of medical studies, while LAB-Bench [91] measures a broad range of tasks for scientific research, e.g. analysis of tables and figures. These benchmarks are typically conducted in an offline static manner, and there is no clear involvement of an external environment.

Instead of a straightforward evaluation, there are also benchmarks involving a scientific article pool as an external environment, which is utilized for retrieval and assists in different research stages, including academic surveying, idea generation, experiment design, writing assistance and so on. For example, ResearchArena [92] benchmarks agents' ability to collect surveys and organize information. [93] compares the ideas generated by research agents with expert NLP researchers, which are based on paper retrieval for RAG. PaperQA [94] performs information retrieval across full-text scientific articles for question answering. These agents harness extracted knowledge from the environment, but lack explicit interaction.

Since developing and evaluating an agent's capacity for real-world scientific discovery is often expensive and challenging, some benchmarks construct a simulated, interactive and text-based environment to evaluate the agents' capabilities. Representatives of them are SCIENCEWORLD [95] and DISCOVERYWORLD [96]. SCIENCEWORLD tests agents' scientific reasoning capabilities in a text environment with a simulation for thermodynamics, electrical circuits, chemical reactions, and biological processes. DISCOVERYWORLD benchmarks agents' ability to perform complete cycles of scientific discovery.

As the applications of scientific agents become more complex, the demands on their capabilities continue to grow, and the environments in which they interact are also becoming more complex and realistic. Specifically, in order to automate the complete cycle of scientific discovery, agents may need to interact with the environment by reading in-memory data, handling file permission, editing files, executing scripts, and collecting results. In this case, the environment is complex, which can be composed of file systems, code interpreter, shell, database and so on. We refer to this compound environment as *workspace*, which is typically a jupyter notebook or a virtual machine in many cases. ScienceAgentBench [97] evaluates agents in an environment equipped with bash shell, web browser, code interpreter on coding for workflow of scientific discovery. MLGym [98] designs a gym-like environment where agents can manipulate with shell, tools, python dependencies and permission for various files. MLAgentBench [99], DSBench [100], and MLE-bench [177] mainly focus on Kaggle challenges and completions, assessing agents' end-to-end machine learning engineering capabilities. The workspace allows file editing and code running by these research agents. DSEval [101] also assesses the performance of agents throughout the entire data science lifecycle, with a runtime session (i.e., jupyter notebook) composed of in-memory data, external files, execution history and code executor. DA-Code [102] and ML-Bench [35] evaluate agents in a Docker on data science code generation and repository-level code manipulation respectively. Spider2-V [103] focuses on multimodal agents for data science automation, which also relies on a Docker environment for file transfer, application launch, remote API calls, script execution and Playwright automation.

The evolution of scientific environment benchmarks reflects a progressive deepening of task complexity and interactivity. Evaluation has developed from early static scientific question answering (e.g., SCIENCEQA [85]) and paper-repository-based retrieval (e.g., ResearchArena [92]) to simulated interactive text environments (e.g., SCIENCEWORLD [95]). A recent trend involves constructing complex "Workspaces", such as MLAgentBench [99] and DSEval [101], which integrate file systems, code interpreters, and Shells to evaluate the agent's ability to automate end-to-end scientific discovery and data science workflows.

3.6 Game environments

Prior to the advent of large language models, game environments were extensively utilized as testbeds for AI agents. In contrast to agents such as code agents and web agents, the value in application of agents in game environments is relatively low. In general, benchmarks with game environments do not evaluate agents on specific tasks; rather, they aim to evaluate general capabilities such as planning and reasoning. In this section, we primarily introduce several game environments that have been utilized in existed benchmarks for agents, as in detail shown in Table 6.

One branch of game environments for agent evaluations involve only one agent and focus on the absolute capabilities of the agents. BALROG [105] evaluates agents on common games like BabyAI [178], Crafter [179], TextWorld [180] and so on. For VLMs (Vision Language Model), the environments provide directly the screenshots of the games. While for LLMs, all of the observations in games are transformed into texts for the agents to understand. SmartPlay [106] provides a comprehensive evaluation of the agent's capabilities in various domains, including long text understanding, reasoning, instruction/rule following, and so on. The evaluation process encompasses a diverse array of both visual and text-based games, including Bandits, Rock-Paper-Scissors, Tower of Hanoi, Minecraft, Messenger, and Crafter. In certain environments, internal rewards are utilized as the evaluation metric, while in others, completion rates are employed to assess the performance of the agents. In a similar manner, LVLM-Playground [107] adopts the games including TicTacToe, Reversi, Sudoku to evaluate the agents' capabilities on perception, reasoning, decision and adversary. The evaluation metrics are multifaceted and differ depending on the specific task. For instance, for perceiving task, which asks the agents to convert a visual game state into a structured matrix representation, and the accuracy is utilized as the metric. In contrast to the aforementioned work, VGRP-Bench [108] focuses on puzzle games and employs them as a means to assess the agents. ING-VP [109] focuses evaluation of multi-step planning based on spatial relationships in images. The environments in ING-VP include six representative games: Sokoban, Maze, Sudoku, 8-queens, Tower of Hanoi, and 15-puzzle. In the evaluation of agents, three metrics are considered: accuracy, completion degree, and action efficiency. DSGBench [110] provides a rigorous evaluation platform and includes six complex strategic games like StarCraft II, Civilization and so on to evaluate the agents on decision making. In all of the environments, the original observations —such as screenshots —are transformed into text prompts for the agents. For each game, there exist numerous human-designed metrics that are employed to assess the performance of agents. For instance, in StarCraft II, the following metrics are adopted: RPM (Resource Per Minute), EER (Efficiency of Resource Utilization), SUR (Supply Usage Rate), TCR (Technology Completion Rate), APM (Actions Per Minute), EPM (Ecffective Actions Per Minute), WR (Win Rate), and GA (Grounding Accuracy).

Another branches of benchmarks try to compare the agents and their environments usually support multiple agents. Gamebench [111] evaluates strategic reasoning abilities of AI agents, where nine multi-player board games are served as the testbeds, including Air, Land, Sea (ALS), Arctic Scavengers (ARC) and so on. The Bradley-

Table 6 Game benchmarks and their taxonomy

Benchmark	Game	Single-agent/ Multi-agent	Needed agent capability
BALROG [105]	BabyAI, TextWorld, Crafter, Baba Is AI, MiniHack, NLE	Single-agent	Navigation, Exploration, Resource Management, Complex Credit Assignment, Deducing Env. Dynamics, Long-term Planning
SmartPlay [106]	Bandits, Rock-Paper-Scissors, Hanoi, Messenger, Crafter, Minecraft	Single-agent	Long Text Understanding, Multi-hop Reasoning, Instruction/Rule Following, Planning, Generalization, Understanding the Odds, Learning from Interactions, Error/Mistake Handling, Spatial Reasoning
LVLM-Playground [107]	TicTacToe, Reversi, Sudoku, Minesweeper, Gomoku, Chess	Single-agent	Perception, Reasoning, Decision, Adversary
VGRP-Bench [108]	Binairo, Star-Battle, Colored-Sudoku, Killer-Sudoku (20 visual reasoning puzzles)	Single-agent	Reasoning, Rule-following, Overall Perception, Cell-level Perception
ING-VP [109]	Sokoban, Maze, Sudoku, 8-queens, Tower of Hanoi, 15-puzzle	Single-agent	Perception, Reasoning, Planning, Text Understanding, Spatial Imagination
DSGBench [110]	Starcraft II, Civilization, Diplomacy, Stratego, Street Fighter III, Werewolf	Single-agent	Strategic Planning, Real-Time Decision Making, Adaptive Learning, Social Reasoning, Team Collaboration
Gamebench [111]	Air, Land, Sea, Arctic Scavenger, Are You the Traitor?, Codenames, Hive, Pit, Santorini, Two Rooms and a Boom, Sea Battle	Multi-agent	Reasoning capabilities on Abstract Strategy, Non-Deterministic, Hidden Information, Language Communication, Social Deduction and Cooperation
γ-Bench [112]	Guess 2/3 of the Average, El Farol Bar, Divide the Dollar, Public Goods Game, Diner's Dilemma, Sealed-Bid Auction, Battle Royale, Pirate Game	Multi-agent	Reasoning capabilities on game theoretic environments
Gtbench [113]	Tic-Tac-Toe, Connect-4, Kuhn Poker, Breakthrough, Liar's Dice, Blind Auction, Negotiation, Nim, Pig, Iterated Prisoner's Dilemma	Multi-agent	Board Strategy, Bids, Collaboration, Bluff, Math
GameArena [114]	Akinator, Taboo, Bluffing	Multi-agent	Deductive reasoning, Abductive reasoning, Inductive reasoning, Multi-hop reasoning

Terry system [181] is employed to calculate the scores of the agents in order to make a comparison of their relative abilities. In a similar vein, the γ-Bench [112] and Gtbench [113] assess the performance of agents in a manner analogous to that of competing models, albeit within the context of game-theoretic environments. GameArena [114] is distinguished from prior multi-agent game environments in that it incorporates human participation and provides human feedback during the evaluation process. For instance, in the AI Akinator game of GameArena, the agents determine what object the user is thinking of. During the game, the user responds to the agents' questions with a "yes" or "no" response.

Game environments are primarily utilized as testbeds for evaluating agents' general core capabilities (such as planning and reasoning). Evaluation benchmarks are mainly divided into two branches: one is "Single-Agent" environments (e.g., SmartPlay [106], DSGBench [110]), used to assess agent decision-making and multi-step planning capabilities under complex rules; the other is "Multi-Agent" environments (e.g., Gamebench [111], τ-Bench [143]), which focus on evaluating agents' strategic reasoning and social interaction capabilities in competitive or collaborative settings.

4 Evaluation on agent capabilities

Evaluating AI agents necessitates a nuanced understanding of their advanced internal capabilities, which are fundamental to their performance across various tasks. This section categorizes existing evaluation methods based on these intrinsic abilities, as shown in

Table 7 Planning benchmarks and their taxonomy

Benchmark	Domain category	Task type	Format	Key focus
AQUA-RAT [182]	Mathematical Reasoning	Algebraic word problem solving with natural-language rationales	Static dataset	Evaluating generated rationales in algebra problems
GSM8K [183]	Mathematical Reasoning	Elementary arithmetic word problem solving (CoT)	Static dataset	Evaluating CoT performance on basic arithmetic problems
MATH [184]	Mathematical Reasoning	Competition-level math contest problem solving	Static dataset	Evaluating multi-step reasoning on challenging competition problems
Game of 24 [185]	Mathematical Reasoning	Cumulative arithmetic game problem solving	Static dataset	Evaluating cumulative multi-step arithmetic reasoning to reach a target value
SVAMP [186]	Mathematical Reasoning	Math word problem solving with simple variations	Static dataset	Evaluating brittleness of SOTA models under simple textual variations

Table 7 (Continued)

Benchmark	Domain category	Task type	Format	Key focus
HotpotQA [187]	Document Navigation	Multi-hop question answering	Static dataset	Evaluating cross-document multi-hop reasoning
StrategyQA [188]	Document Navigation	Implicit multi-hop inference	Static dataset	Evaluating implicit reasoning for strategy questions
ScienceQA [85]	Science Knowledge	Multimodal multiple-choice	Static dataset	Incorporating charts and images
ARC [84]	Science Knowledge	Text-based multiple-choice	Static dataset	Purely textual science questions including both Easy and Challenge subsets
FOLIO [189]	Logic Inference	Natural-language reasoning with first-order logic annotations	Static dataset	Evaluating NL reasoning and NL→FOL translation with formal proof verification
P-FOLIO [190]	Logic Inference	Step-by-step annotated natural-language proofs for FOL reasoning	Static dataset	Assessing and improving multi-step logical reasoning in LLMs through human-written proofs
PlanBench [115]	Classical Planning	Planning in canonical IPC domains (Blocksworld & Logistics)	PDDL (Static)	Evaluating eight core planning capabilities (generation, optimality, verification, execution reasoning, robustness, reuse, replanning, generalization)
On Planning Abilities of LLMs [116]	Classical Planning	Autonomous plan generation and heuristic-guided planning	PDDL + LLM prompts	Evaluating LLM's autonomous planning success and its use as heuristic guidance for external planners
AutoPlanBench [117]	Classical Planning	Automated NL prompt generation from PDDL domains	NL prompts (Static)	Comparing auto-generated vs. manual and template-based NL prompts; enabling large-scale PDDL planning evaluation
Exploring Planning Capabilities [191]	Planning Evaluation	PDDL planning (BlocksWorld, Logistics, Mini-Grid); NL planning (Trip Planning, Calendar Scheduling)	PDDL & NL prompts (Static)	Automated instance generation; many-shot in-context learning; chain-of-thought reasoning; fine-tuning; out-of-distribution generalization; failure mode analysis
ACPBench [118]	Planning Evaluation	Seven atomic planning reasoning tasks: Applicability, Progression, Reachability, Validation, Action Reachability, Justification, Landmark Detection	PDDL-based NL prompts (Static)	Benchmarking core planning skills via automatically synthesized Boolean and MCQ questions across 13 domains
NATURALPLAN [192]	Natural‐Language Planning	Real‐world NL planning tasks: Trip Planning, Meeting Planning, Calendar Scheduling	Static dataset with in‐context tool outputs	Evaluating LLMs on realistic NL planning: complexity variation; self-correction; few-shot generalization; long-context planning
FlowBench [120]	Workflow Planning	Workflow-guided planning tasks across 51 scenarios in six domains	Static prompts with workflow knowledge in text, code, and flowchart formats	Formalizing diverse workflow knowledge formats; multi-tiered evaluation (static turn-level and dynamic session-level); comparative analysis of LLM performance across formats
WorFBench [119]	Workflow Planning	Graph-structured workflow generation across four scenarios: problem-solving, function calling, embodied planning, open-grounded planning	Static dataset (task descriptions + action lists)	Multi-scenario DAG workflow generation; evaluation via WORFEVAL's subsequence and subgraph matching; analyzing sequence vs. graph planning gap; assessing generalization and downstream task enhancement
MINT [193]	Interactive Planning	Multi-turn interactive tool-augmented task solving with natural language feedback	Simulated online environment (tool execution + LLM-based user feedback)	Quantifying per-turn gains from tool use and language feedback; analyzing effects of instruction fine-tuning on multi-turn reasoning
REALM-Bench [121]	Interactive Planning	Single- and multi-agent real-world scenario planning with dynamic environment feedback	Simulator-based interactive environment	Evaluating adaptive planning under parallelism, complex dependencies, and unexpected disruptions

Table 7 in detail.

4.1 Planning

As the complexity and diversity of environments confronted by AI agents increase, single-step agents designed for isolated question answering fail to tackle tasks requiring multi-step reasoning, such as solving advanced mathematical problems or navigating across multiple documents. To address these limitations, researchers have introduced methods to stimulate multi-step reasoning in AI agents, including chain-of-thought prompting (CoT) [194] and tree-of-

thoughts (ToT) [195] strategies.

The simplest approach to assess multi-step reasoning is through static datasets spanning diverse domains. In mathematical reasoning, AQUA-RAT [182] evaluates algebraic planning with generated rationales, GSM8K [183], and MATH [184] cover elementary and secondary school problem solving, Game of 24 [185] measures arithmetic CoT planning, and SVAMP [186] introduces perturbations such as operand swaps and paraphrases to increase difficulty on Math Word Problems. Document-navigation benchmarks like HotpotQA [187] and StrategyQA [188] require multi-hop navigations across multiple passages. ScienceQA [85] extends the standard multiple-choice science questions with multimodal information (e.g., charts and images), whereas ARC [84] consists of text-based multiple-choice science questions. Logic-inference workflows are evaluated by FOLIO [189] and P-FOLIO [190], which treat logical inference chains as planning workflows.

Beyond implicit reasoning, explicit planning evaluation employs the Planning Domain Definition Language (PDDL) from classical AI planning. PlanBench leverages canonical domains such as Blocksworld and Logistics to test logical, causal, and spatial planning across eight subtasks [115]. [116] provides a critical investigation showing agent planning success is limited but improves when used as heuristic guidance for external planners. AutoPlanBench [117] automates the conversion of PDDL domains into natural language prompts, enabling large-scale evaluation of agent planning methods and demonstrating that automatically generated prompts match or exceed manually crafted ones. [191] introduces a benchmark suite covering both classical and natural-language planning tasks, investigating many-shot in-context learning and fine-tuning strategies to enhance planning performance. ACPBench [118] further decomposes planning into seven atomic reasoning tasks, such as action applicability, progression, reachability, and landmark detection, across thirteen domains to probe constraint-based planning abilities.

While traditional benchmarks focus on abstract or virtual tasks, workflow-based benchmarks evaluate an agent's ability to decompose real-world tasks into executable subtasks. NATURALPLAN [192] introduces realistic planning tasks like trip planning, meeting scheduling, and calendar management by providing tool outputs as context, eliminating the need for external execution environments. FlowBench [120] revisits workflow-guided planning by formalizing multiple formats of workflow knowledge and covering 51 scenarios across six domains, revealing that current agents struggle with workflow hallucinations and require significant improvement. WorFBench [119] offers a unified benchmark for workflow generation with complex graph structures and a systematic evaluation protocol utilizing subsequence and subgraph matching, highlighting distinct gaps between sequence and graph planning capabilities.

Finally, to incorporate environment feedback and iterative instruction, benchmarks such as MINT [193] and REALM-Bench [121] assess agents in interactive, online settings. MINT evaluates multi-turn interaction capabilities by combining tool use and natural language feedback, showing that feedback can substantially improve performance but also revealing that instruction fine-tuning may hurt multi-turn reasoning. REALM-Bench provides an online simulator with eleven real-world planning scenarios, such as supply chain and disaster response, that can be scaled in parallelism, dependency complexity, and disturbance frequency, enabling rigorous testing of both single-agent and multi-agent planning under dynamic conditions.

In conclusion, the evaluation of planning capability has progressed from implicit reasoning on static datasets (e.g., GSM8K [183], ScienceQA [85]) to explicit, executable planning in dynamic environments. While early benchmarks focused on Chain-of-Thought reasoning for isolated problems, modern benchmarks (e.g., PlanBench [115], FlowBench [120]) emphasize the rigorous translation of natural language into executable actions (PDDL) and the management of complex real-world workflows. The current frontier, represented by benchmarks like REALM-Bench [121], challenges agents to perform iterative planning with environmental feedback, highlighting the shift from abstract reasoning to embodied, adaptive task execution.

4.2 Self-reflection

As LLM chatbots evolve into more autonomous AI agents, the ability to self-reflect and improve through interaction—rather than relying solely on human supervision—has become increasingly critical. A growing body of research explores whether agents can engage in self-reflection through interactive feedback and subsequently refine their reasoning to reduce errors in multi-step tasks. This process demands not only an understanding of feedback but also the dynamic updating of internal beliefs, allowing agents to adaptively revise actions or reasoning paths over extended interaction trajectories.

Early evaluations of agent self-reflection are often simplistic and limited. These studies typically employ static environments with repetitive access and measure improvement only by checking whether the final answer is corrected after multiple attempts [196–197]. However, such evaluations tend to be coarse and lack the capacity to generalize to new data or real-world feedback scenarios.

Furthermore, nascent benchmarks primarily focused on internal self-reflection, providing minimal environmental feedback and relying heavily on prompts such as "Review your previous answer and find problems with your answer" [196,198]. This internal-only mode of reflection limits the performance ceiling of agents, as relying solely on intrinsic reasoning cannot fully replace the adaptive learning signal provided by interactive rewards from the environment. Therefore, there is a pressing need for evaluation protocols that assess an agent's ability to perform self-reflection grounded in external feedback signals.

To address this gap, LLF-Bench [123] was proposed as a standardized benchmark for interactive self-reflection. LLF-Bench constructs domain-specific language feedback across multiple tasks, enabling a systematic evaluation of AI agents' ability to learn from external linguistic corrections. The benchmark simulates diverse feedback scenarios and tests whether AI agents can adapt and improve based on natural language guidance. Similarly, LLM-Evolve [124] targets the self-reflection ability of AI agents on standard

benchmarks such as MMLU [199]. In this framework, agents are exposed to dynamic feedback and allowed to retrieve few-shot examples for reflection and correction. The benchmark measures whether the integration of feedback leads to meaningful performance improvements across tasks.

Reflection-Bench [125] introduces novel evaluation metrics from a cognitive perspective to assess the quality, coherence, and logical consistency of reflective outputs. It explicitly examines whether a model can articulate the cause of its errors, propose plausible fixes, and ultimately achieve better decision-making as a result of reflection. Finally, [126] focuses on the domain of code generation, where self-reflection is evaluated in code environments based on human-provided feedback. Instead of only assessing whether a model generates correct code on the first try, the benchmark emphasizes the ability to revise and improve through iterative reflection, simulating real-world debugging and correction scenarios on three existing benchmarks APPS [200], LiveCodeBench [201], and ClassEval [17].

To summarize, the evaluation of self-reflection has evolved from internal self-correction to interaction-driven refinement. Initial assessments primarily relied on agents reviewing their own outputs with static prompts, which often limited performance gains. Recent benchmarks (e.g., LLF-Bench [123], Reflection-Bench [12]) have introduced external feedback mechanisms —ranging from natural language corrections to code execution errors—to test whether agents can dynamically update their beliefs and debug their reasoning. This shift underscores that true reflective capability is defined not just by identifying errors, but by the ability to leverage external signals for tangible improvement.

4.3 Interaction

To systematically evaluate the interaction capabilities of AI agents, we propose a three-level taxonomy of agent-environment interaction, each representing a progressively more complex and autonomous behavior.

4.3.1 Interaction with static systems

First, we consider interaction with static systems, where the agent interacts with predefined tools or user interfaces (e.g., GUI APIs) via *tool use* or *function calling*. These interactions typically rely on a fixed set of functionalities. The focus of evaluation here is whether the agent can correctly interpret user intent, identify the appropriate tool, and execute the call correctly. This ability constitutes the foundation of agent competence.

Numerous benchmarks have been developed to assess the performance of tool agents across various tasks. These benchmarks evaluate aspects such as tool selection, parameter filling, output formatting, and the ability to handle complex, multi-step processes. A comprehensive overview of these benchmarks is presented in Tables 8 and 9.

General-purpose tool agent benchmarks Several benchmarks have emerged to evaluate agents' capabilities in handling a wide range of tool interactions. These tools involves a wide domain as weather reports, news retrieval and translation. Benchmarks like ToolAlpaca [127], ToolBench [129], BFCL [130], Seal-Tools [132] exemplify this phase. To be specific, early ones ToolAlpaca [127] focuses on generalized tool learning by providing a diverse set of API interactions, and ToolBench [129] is designed to evaluate agents' ability to manipulate real-world tools. BFCL [130] maintains the quality and consistency of live documentation by updating BFCLv2 and BFCLv3 to assess function calling capabilities. Seal-Tools [132] employs self-instructed learning to generate tool usage scenarios. API-Bank [133] focuses on evaluating agents' ability to search and retrieve specific APIs. NexusRaven [134] assesses agents' performance in nested and parallel function calling scenarios. API-Blend [135] emphasizes accurate slot filling and sequencing in complex scenarios. APIGen [137] focuses on multi-turn function calling scenarios. StableToolBench [138] aims to provide stable and realistic tool evaluation by simulating real-world API interactions.

Specialized tool agent benchmarks In addition to general-purpose benchmarks, several specialized benchmarks have been developed to assess agents' capabilities of tool calling in specific domains. APIBench [128] challenges agents on evaluating the performance of agents in utilizing Python APIs for machine learning tasks. ToolSandBox [131] evaluates agents' ability to interact with mobile app tools. Its primary difficulty is managing stateful interactions and implicit dependencies between tools. RestBench [136] assesses agents' performance in interacting with movie and music databases. ComplexFuncBench [139] focuses on handling complex function calls, such as booking and reservation APIs while managing user constraints and preferences. NESTFUL [140] evaluates agents' ability to handle nested sequences of API calls, where outputs from one call serve as inputs to another. A core challenge is managing dependencies and ensuring correct sequencing.

4.3.2 Interaction with human

Second, we extend to interaction with humans, where the agent must engage in multi-turn interactions to accomplish a single goal, and humans exhibit changing states or behaviors. Agents have necessity to ask clarifying questions or request further instructions from a human user, due to the inherently dynamic and underspecified nature of human intent.

Early benchmarks focused on constructing multi-turn human-agent dialogues manually to evaluate human-agent interactions like ABCD [144] and MultiWoZ [145]. The Action-Based Conversations Dataset (ABCD) [144] constructs dialogues requiring unique sequences of actions, constrained by company policies, to achieve task success. This dataset emphasizes the importance of aligning agent actions with organizational guidelines in realistic customer service scenarios. Similarly, the MultiWOZ dataset [145] offers a large-scale collection of human-human written conversations spanning multiple domains. It serves as a valuable resource for developing and evaluating task-oriented dialogue systems across diverse contexts.

With the advent of AI agents, researchers began exploring the simulation of human users by agents instead of manual construction to enhance the realism of evaluation scenarios. The τ-bench [143] benchmark emulates dynamic conversations between a user agent and a language agent equipped with domain-specific API tools and policy guidelines. This setup allows for the assessment of an agent's

Table 8 Tool Benchmark and their taxonomy 1

Benchmark	Task domain	Task type	API data source and authenticity	Instruction data source and authenticity
ToolAlpaca [127]	Multiple daily life domains	Single-turn API call, Multi-turn API call	Public APIs repository on GitHub	LLMs (e.g., GPT-3.5) auto-generated tool usage dialogues
APIBench [128]	Broad machine learning tasks	Single-turn API call	PyTorch Hub, HuggingFace Official	GPT-4 self-instructed data synthesis
ToolBench [129]	Multiple daily life domains	Single-turn API call, Multi-turn chained API calls	RapidAPI Hub	DFSDT algorithm on ChatGPT for searching and annotating
BFCL [130]	Multiple daily life domains	Simple Function, Multiple Functions, Parallel Functions	Real usable network APIs and hand-written functions with executable feasibility	Automatically generated synthetic data
ToolSandbox [131]	Mobile environment tasks	Single/Multiple Tool Call, Single/Multiple User Turn	Real usable RapidAPI interfaces and local functions	GPT-4o simulated dialogue and execution paths
Seal-Tools [132]	Multiple daily life domains	Single-tool, multiple-tool, parallel calls, and nested calls	Real usable RapidAPI interfaces	Tool definition and example calls are synthetic
API-Bank [133]	Multiple daily life domains	Single call, Retrieval+Call, Plan+Retrieval+Call	Hand-implemented and deployed by engineering teams in the same framework	Annotated by multiple computer science students through a discussion-based method
NexusRaven [134]	Multiple daily life domains	Single call, parallel call, nested/composite call	Handwritten but verified for multi-turn executable feasibility	Synthetic generated, but from real open-source projects
API-Blend [135]	Multiple daily life domains	Single API recognition, parameter extraction, multi-step API calls	LLM-assisted generated synthetic examples	5 real or high-quality open-source benchmarks
RestBench [136]	TMDB movie database and Spotify music player	Multi-step API calls	From TMDB and Spotify official OpenAPI documentation, real and callable	6 NLP experts designed from bottom-up using real-world usage scenarios
APIGen [137]	Multiple daily life domains	Simple Function, Multiple Functions, Parallel Functions, Parallel Multiple Functions	RapidAPI Hub real APIs	Self-guided synthesis based on models like DeepSeek-V2-Chat, Mixtral
StableTool Bench [138]	Multiple daily life domains	Single-turn API calls, Multi-turn API chain calls	From ToolBench published data, responses will consider GPT-generated API responses	Human-designed real-world instruction set
ComplexFunc Bench [139]	Booking.com-related APIs	Constraint-based multi-step calls	Synthetic generated and human-annotated through multi-turn validation	Human-designed real-world instruction set
NESTFUL [140]	Mathematical computation tools, general programming functions	Single-turn Multi-step, parameter filling	Real executable production-level functions	Manually constructed and filtered from real datasets
ToolEyes [141]	Seven high-demand real-world scenarios	Multi-tool calls and multi-step processes	Follows OpenAPI/JSON specification HTTP API	LangChain official repository (GitHub), SerpAPI platform, OpenAI documentation
PEToolBench [142]	Personalized tool calling tasks	Instruction calls with user historical data	Real RESTful APIs from RapidAPI	LLM self-synthesis and multi-turn filtering, non-real user data
ACEBench [202]	Multiple daily life domains	multi-step call, in-complete instruction following, multi-agent collaboration	Synthetic APIs referenced by Real APIs	Synthetic instruction data

ability to interact with simulated users.

In recent years, benchmarks have begun to combine automated and manual processes to generate diver see and realistic dialogues. Specifically, ALMITA [146] proposes a new dialogue dataset and framework specifically aiming at evaluating tool-augmented dialogue AI agents in customer support scenarios. IntellAgent [147] employs a graph-based representation to model the relationships, likelihoods, and complexities of various policies. This approach enables the simulation of multi-policy scenarios, capturing the nuanced interplay between agent capabilities and policy constraints.

We have systematically compiled these works into a comparative Table 10. From this table, it becomes evident that the primary task in human-agent interaction benchmarks is often tool calling. Consequently, the ability to effectively utilize tools serves as the cornerstone of agent interaction capabilities. Subsequent interactions with humans are typically aimed at obtaining feedback or further instructions to enhance the completion of tool-based tasks.

4.3.3 Interactions with other agents

Once agents acquire the capability to independently solve problems, a third mode of interaction emerges, which is agent-agent interaction. The purpose of this interaction is two-fold: to collaboratively accomplish tasks that a single agent cannot achieve alone, or to

Table 9 Tool benchmark and their taxonomy 2

Benchmark	Metric	Evaluation label and method	Input modality	Main challenges
ToolAlpaca [127]	Procedure accuracy, Response accuracy	Label by API name + automated evaluation	Text	Generalizing to unseen real APIs using a compact language model trained with only 4K diverse simulated examples.
APIBench [128]	Accuracy, Hallucination Rate	Label by API name + automated evaluation	Text	Generalizing a compact LLM to accurately call unseen APIs in a large, overlapping, and ever-changing API set.
ToolBench [129]	Pass Rate, Win Rate	Label by ChatGPT-generated + LLM evaluation similarity	Text	Accurate retrieval of relevant APIs and generalization to unseen APIs and complex instructions through multi-step, depth-first decision tree strategy.
BFCL'[130]	AST Evaluation, Executable Function Evaluation	Label by experts + automated evaluation similarity	Text	Generalizing to unseen functions in multi-domain, multi-language, and complex multi-step and parallel call scenarios.
ToolSandbox [131]	Milestone Similarity, Turn Count	Milestones and Minefields provided by humans, automated evaluation based on DAG similarity	Text	Generalizing tool call and interaction strategies in complex dialogue scenarios with state dependency and incomplete information.
Seal-Tools [132]	Format Accuracy, Tool Precision/Recall/F1, Parameter Precision/Recall/F1	Label generated through LLM self-instructed multi-turn validation, automated evaluation	Text	Evaluating multi-layer nested calls and implicit dependencies, with evaluation criteria including correct parameter passing.
API-Bank [133]	ROUGE-L, Accuracy	Label by manual annotators + fully automated evaluation	Text	Generalizing LLMs to retrieve and call unknown tools accurately in multi-domain, multi-API pool, and hybrid information retrieval and planning scenarios, with real execution environment verification.
NexusRaven [134]	Nested/Composite Success Rate	Label manually written + automated evaluation script	Text	Handling high complexity in unseen function signatures, multi-level nesting, and parallel calls under zero-shot conditions.
API-Blend [135]	API-F1, Parameter-F1, LCS-F1	Label: Dataset original annotators or paper authors provide + automated evaluation scripts	Text	Cross-domain generalization in heterogeneous data and sparse sequencing requirements.
RestBench [136]	Correct Path Rate, Δ Solution Length	Label manually written + automated evaluation script	Text	Breaking down complex tasks into coarse-to-fine online planning while maintaining high robustness and low redundant calls during execution.
APIGen [137]	AST Evaluation, Executable Function Evaluation	Label by experts + automated evaluation similarity	Text	Building a pipeline for automatic data generation: "Format verification → Execution verification → Semantic verification."
StableTool Bench [138]	Solvable Pass Rate, Solvable Win Rate	Label by three LLM votes + GPT-4 evaluation	Text	Reducing fluctuations caused by real API downtime or changes through caching and LLM simulation, and evaluating solvable tasks with GPT-4 assessment.
ComplexFunc Bench [139]	Rule-Based + Response-Based Precision/Recall/F1	Label manually written + automated evaluation script	Text	Inferring optimal call sequences in multi-user constraint and implicit parameter scenarios while handling long parameters.
NESTFUL [140]	F1 Func/Param, Partial Sequence Accuracy	Label manually written + automated evaluation script	Text	Correctly generating function names, parameters, and managing output variables in multi-layer data dependency and variable reference nested calls.
ToolEyes [141]	Format alignment, Intent understanding, Behavior planning, Tool selection, Answer organization with average score across five dimensions	Label manually written + GPT-4 evaluation score	Text	Capturing LLM capabilities in format compliance, intent reasoning, behavior planning, tool matching, and answer integration at a fine-grained level.
PEToolBench [142]	Preference Alignment Improvement, Tool Call Accuracy	Label: LLM synthesized + manual writing, automated evaluation matching	Text	Identifying and selecting the best tool with user implicit preferences from tools with similar functionalities and generating precise parameters.
ACEBench [202]	End-to-End Accuracy, Process Accuracy	Label: Automated synthesized + expert checking, Eval: automated	Text	Achieving efficient, low-cost, and fine-grained automated evaluation of tool invocation in real-world scenarios involving multi-turn, multi-step interactions with incomplete instructions and multiple agents.

Table 10 Human-agent interaction benchmarks and their taxonomy

Benchmark	Dialogue task type	Dialogue data source	Metrics	Labeling and similarity evaluation	Modality	Explicit interaction evaluation beyond task accuracy
τ-bench [143]	Airline and retail	Simulated users via agent LLM	pass^k	Labels annotated by humans +manual similarity assessment	Text	No
ABCD [144]	55 distinct user intents	Agent interacting with simulated users	Action State Tracking, Cascading Dialogue Success	Human annotation and manual similarity assessment	Text	No
MultiWOZ [145]	Dialogues across multiple domains	Crowd-sourced	Belief tracking, dialogue act classification	Labels are human-provided; evaluations comparing agent responses to human benchmarks	Text	No
ALMITA [146]	192 conversations for 14 intents	Combination of automated generation and manual curation	API correctness	Human annotation and automated similarity assessment	Text	No
IntellAgent [147]	Dialogues across multiple domains	Automated process generation of diverse synthetic test cases	Task accuracy	Human annotation and automated similarity assessment	Text	No

engage in competitive scenarios to determine a winner. To systematically evaluate these interactions, we categorize existing benchmarks into two paradigms: cooperative and competitive. These paradigms serve as the foundation for assessing agent-agent interactions, as shown in Table 11.

According to cooperative agent evaluations, Collab-Overcooked [150] extends the Overcooked-AI game to evaluate AI agents' collaborative capabilities by incorporating process-oriented evaluation metrics such as Trajectory Efficiency Score (TES) and Incremental Trajectory Efficiency Score (ITES). SmartEvals [153] encompasses six distinct games, including Rock-Paper-Scissors, Tower of Hanoi, and Minecraft, each augmented with language descriptors for visual observation. The benchmark uniquely challenges nine critical capabilities of intelligent AI agents, such as reasoning with object dependencies, planning ahead, spatial reasoning, learning from history, and understanding randomness.

Table 11 Multi-agent benchmarks and their taxonomy

Benchmark	Interaction paradigm	Modality	Metrics	Application domain	Agent architecture	Key contributions
BattleAgent Bench [148]	Cooperation and Competition	Text	Rule Understanding, Spatial Perception, Cooperation and Competition Score	Game	Homogeneous	Multiple difficulty levels, fine-grained evaluation
MultiAgent Bench [149]	Cooperation and Competition	Text	Task Completion, Collaboration Quality, Communication Effectiveness, Cognitive Planning	Collaborative Programming, Game Simulation	Homogeneous	Diverse communication topologies, KPI-based evaluation of individual agent contributions
Collab-Overcooked [150]	Cooperation	Text	Progress Completeness, Initiating Capability, Responding Capability	Game	Homogeneous	Process-oriented collaboration metrics, assessing collaborative abilities
Sotopia [151]	Cooperation and Competition	Text	Task Completion, Social Commonsense Reasoning, Strategic Communication	Social Tasks	Homogeneous	Simulates complex social interactions among AI agents, evaluating social intelligence
AutoArena [152]	Competition	Text	Elo Ranking, Correlation with Human Preferences	Logical Reasoning, Coding Ability	Homogeneous	Swiss-system tournament and Elo ranking for evaluation
SmartEval [153]	Cooperation	Text	Completion Rate, Reward, Score	Game	Homogeneous	Each game challenges different abilities, including reasoning, planning, spatial perception, etc.
MindAgent [154]	Cooperation	Text	Task Completion, Coordination Efficiency, Instruction Understanding	Game	Homogeneous	Requires coordination and cooperation with human players, demonstrating planning and execution abilities
GOVSIM [155]	Cooperation	Text	Resource Duration, Total Yield, Efficiency	Fisheries, Pastures, Pollution Management	Homogeneous	Simulates shared resource environments, requiring agents to cooperate to prevent resource depletion

MindAgent [154] leverages existing gaming frameworks to require not only understanding of a multi-agent system but also collaboration with human players via un-finetuned instructions. GOVSIM [155] presents scenarios such as fishery, pasture, and pollution, focusing on agents engaged in cooperative behavior and effective governance to achieve sustainable outcomes

For competitive agents evaluation, AutoArena [152] is an automated evaluation framework that assesses AI agents through peer battles and committee discussions. It utilizes an agent examiner to generate questions, followed by multi-round peer battles between agent candidates, and concludes with a committee of agent judges collaboratively deciding the winner. This approach reduces bias and enhances evaluation fairness

Recently, researchers have established multi-agent benchmarks that simultaneously consider cooperation and competition. BattleAgentBench [148] evaluates AI agents across seven sub-stages with three varying difficulty levels. It assesses single-agent navigation, paired-agent task execution, and multi-agent collaboration and competition capabilities. MultiAgentBench [149] captures coordination dynamics and competitive interactions, providing tailored metrics such as Key Performance Indicators (KPIs), structured planning and communication scores, and competition scores to systematically assess agent performance. Sotopia [151] simulates complex social interactions between artificial agents and humans. It evaluates agents' social intelligence through role-play interactions under various scenarios, assessing dimensions like believability, relationship, knowledge, secrecy, social rules, financial/material benefits, and goal completion.

In summary, the evaluation of interaction capabilities follows a hierarchy of increasing complexity: from static tools to humans, and finally to other agents. Benchmarks have expanded from assessing functional correctness in API calling (e.g., ToolBench [129]) to measuring social intelligence in multi-turn dialogues with humans (e.g., τ-bench [143]). The most advanced evaluations now focus on multi-agent systems (e.g., Sotopia [151], MultiAgentBench [149]), testing coordination, negotiation, and competition. This trajectory reflects a fundamental shift in agent evaluation from solitary task execution to social integration and collaborative intelligence.

4.4 Memory

Memory encompasses an agent's ability to store, retrieve, and apply information over time. A robust memory system allows agents to maintain coherence in long interactions, retrieves relevant knowledge when needed, and adapts behavior based on past experiences. As memory becomes an increasingly vital component for complex reasoning and decision-making tasks, several benchmarks have been

Table 12 Memory benchmarks and their taxonomy

Benchmark	Data source	Task domain	Task type	Memory type	Metric
NarrativeQA [156]	Books and Movie Scripts+ Human-Generated QA Pairs	Narrative Understanding of Books and Movies	QA	Long-range, Causal, Temporal-Spatial	BLEU-1/4, METEOR, ROUGE-L, MRR
QMSum [157]	Real Meeting Transcripts + Human-Annotated Queries and Summaries	Product Design, Academic, Committee discussions	Query-Based Text Summarization	Long-range	ROUGE-1/2/L
QuALITY [158]	English source articles+Human-Generated Question	Fiction, Nonfiction, Magazine	QA	Long-range, Causal	Accuracy
DialSim [159]	TV Show Scripts + Fan QA + LLM-Generation QA	Everyday Interaction Scenes from TV Shows	QA	Long-range, Temporal-Spatial	Accuracy
LoCoMo [160]	LLM Generation Dialogue with Human Verification	Open-Domain Social Dialogue	QA, Event summarization, Multimodal dialogue generation	Long-range, Causal, Multimodal	F1 Score, Recall@K, ROGUE-1/2/L, FactScore, MMRelevance
LTM-Benchmark [104]	Synthetic conversation + Real Data	Everyday Conversations, Cognitive and Reasoning Scenarios	QA, Information Integration, Conflict Resolution	Long-range, Causal, Spatial, Foresight	Score
LongMemEval [161]	Manually created questions	Personalized Long-term Dialogue	Information Extraction, Multi-session Reasoning, Temporal Reasoning, Knowledge Updates, Abstention	Long-range, Dynamic Update, Temporal-Spatial	Accuracy, Recall@k, NDCG@k
Episodic Memory[162]	Synthetic episodic memory data	Fictional Narrative Scenes	QA, Temporal Reasoning, Hallucination Detection	Temporal-Spatial, Long-term State Tracking	F1-Score, Kendall's τ Coefficient, Hallucination Rate
PerLTQA [163]	LLM Generation with Human Verification	Personal Long-term Memory Simulation	QA, Memory Synthesis, Classification, Retrieval	Semantic, Episodic	F1 Score, Accuracy, Recall@K, MAP
StreamBench [164]	Real, Human-Annotated Public Datasets	Text-to-SQL, Programming, Tool Use, Medical, Encyclopedic	Online Streaming Learning and Continuous Model Optimization	Long-range, Incremental Improvement	Accuracy, Pass@1, Exact Match

developed to systematically assess different aspects of agent memory. We summarize these benchmarks in Table 12.

Some benchmarks focus on evaluating specific memory capabilities, such as question answering (QA) or summarization. NarrativeQA [156] tests an agent's ability to answer questions based on long books or screenplays, emphasizing deep narrative understanding. QMSum [157] evaluates memory through the task of generating multi-domain meeting summaries from lengthy transcripts. QuALITY [158] targets reading comprehension, challenging agents to answer multiple-choice questions that require understanding and recalling extended documents. DialSim [159] focuses on dialogue-based QA for TV series, requiring agents to maintain and reason over dialogue history and spatiotemporal memory.

Other benchmarks adopt a more holistic approach, examining multiple memory skills or diverse content types. LoCoMo [160] assesses agents through question answering, event summarization, and multimodal dialogue generation tasks. LTM-Benchmark [104] evaluates conversational agents' long-term memory and information integration within a single dynamic multitask dialogue. LongMemEval [161] tests agents across information extraction, multi-session reasoning, temporal reasoning, knowledge updates, and abstention. Episodic Memory Benchmark [162] assesses episodic memory capabilities, focusing on spatiotemporal-contextual event recall, cross-event relational reasoning, and entity state tracking. PerLTQA [163] evaluates agents' ability to integrate and apply semantic and episodic memories in personalized long-term memory question answering. Finally, StreamBench [164] focuses on online, streaming learning, assessing an agent's capacity to incrementally update its memory and refine performance based on feedback streams.

The evaluation of memory capability is transitioning from static context comprehension to dynamic, lifecycle management. While traditional benchmarks (e.g., NarrativeQA [156]) assesses the retrieval of information from fixed long texts, contemporary approaches (e.g., LongMemEval [161], StreamBench [164]) focus on the agent's ability to maintain episodic memory over extended interactions and update knowledge in response to streaming inputs. The core challenge has shifted to evaluating how well agents can synthesize past experiences to maintain coherence and adaptability in long-term, evolving scenarios.

4.5 General

Beyond the aforementioned agent benchmarks that assess performance in specific capabilities, there exist several benchmarks that explicitly evaluate multiple abilities, including planning and memory. These benchmarks enable a holistic assessment of an agent's generalization capabilities, transcending the limitations of particular environments or individual skills.

AgentBench [165] pioneered the realm of general capabilities benchmarks. It is the first multi-dimensional evaluation framework for AI agents, encompassing eight distinct environments, including operating systems, databases, and games. The benchmark evaluates 27 commercial LLMs, revealing performance variations across complex scenarios. Similarly, GAIA [166] and MMAU [167] also offer offline evaluation benchmarks. GAIA focuses on real-world assistant tasks. It consists of 466 human-designed and annotated questions that are text-based and may include files like images or spreadsheets. These questions are intended to reflect real-world challenges. GAIA evaluates AI systems against real-world tasks through these questions, testing fundamental abilities like reasoning, multi-modality handling, web browsing, and tool-use proficiency. MMAU constructs five major domains encompassing tool usage, question answering, mathematics, and machine learning coding. It assesses AI agents from multiple perspectives, including reasoning, planning, problem-solving, and self-correction.

As research progresses, the limitations of offline evaluations, such as timeliness and lack of realism, have prompted the development of online evaluation benchmarks. Notably, Galileo's Agent Leaderboard [168] introduces a novel evaluation framework that assesses language models' proficiency in real-world agentic tasks by measuring Tool Selection Quality (TSQ), offering a comprehensive ranking across multiple domains such as mathematics, entertainment, education, and retail. HAL (Holistic Agent Leaderboard) [169] provides a standardized, cost-aware platform for evaluating AI agents, integrating a unified evaluation harness that supports various benchmarks and facilitates reproducible assessments, thereby enabling transparent comparisons across different agent implementations. Additionally, AgentArena [170], akin to ChatbotArena [203], allows human evaluators to anonymously score two agents, facilitating authentic preference assessments. This approach not only provides evaluations but also generates human preference data, fostering a positive feedback loop for further agent training.

Furthermore, several innovative general agent benchmarks have emerged. Agent-as-a-Judge [204] departs from traditional human evaluations by employing agents to assess other agents, thereby streamlining the evaluation process and mitigating the cognitive limitations inherent in human assessments. We consider this a promising trend, enhancing scalability in agent evaluations and addressing the challenges associated with human evaluators, which will be discussed in Section 5. Lastly, Capabench [171] utilizes Shapley values to analyze the contributions of different modules within an agent across various tasks and environments, offering insights into the modular performance and optimization of AI agents.

Finally, the landscape of general capability evaluation is moving from offline, multi-task datasets to online, holistic leaderboards. Early frameworks (e.g., AgentBench [165] establishes the standard by aggregating diverse offline environments. However, to address the issues of contamination and static evaluation, the field is adopting online platforms (e.g., HAL [169], Galileo [168]) and "Agent-as-a-Judge" mechanisms (e.g., AgentArena [170]). These evolving benchmarks aim to provide more scalable, reproducible, and human-preference-aligned assessments, offering a comprehensive view of an agent's overall utility in real-world applications.

5 Discussion: the evolution of agent evaluation

After presenting a variety of agent evaluation methods and

benchmarks, a natural question arises: *Given the rapid advancement of AI agents, how can we systematically evaluate LLM-based AI agents from an evolutionary perspective*? This question will be answered in the final section of this discussion. In this section, we will discuss the evolution of agent evaluation from four perspectives, providing both a summary of past work and guidance for future research.

The evolutionary framework is shown in Fig. 3, where we can observe the evaluation process and its four key modules. First, a task is extracted from the environment for the agent to solve. Then, the agent produces a response, which is evaluated by an evaluator through processes such as calculation and judgment. Finally, various evaluation metrics are generated. We will explore each of these four modules in detail to discuss the future directions of Agent Evaluation.

5.1 Environment perspective

From single modality to multi modality. Early AI agents were primarily text-based, and consequently, most early benchmarks focused on evaluating agents using text-based inputs. This was reasonable at the time. As shown in Tables 8 and 10, most benchmarks involving tools, APIs, and human-agent interactions relied on text. However, as the complexity of environments has increased, we are now witnessing the introduction of additional modalities, such as images in web environments or chat audio in mobile environments. Studies have demonstrated that agents perform better when they have access to visual information [44]. Thus, incorporating multimodal evaluation is crucial and beneficial. We are already seeing multimodal benchmarks related to images and videos, such as VisualAgentBench [56] and VideoWebArena [44], and we believe this will be the future trend. Interestingly, in gaming environment benchmarks (e.g., SmartPlay [106], τ-bench [143]), while game images are present, benchmarks have chosen to represent the positions of all characters using coordinates, converting images into text-based inputs for agents. We believe that future evaluations will involve more direct image input rather than converting it into text.

From static to evolving. Earlier benchmarks are relatively simple, where researchers collect data and publish it in offline systems for evaluation. Such benchmarks are static. However, in the real world, the environment evolves rapidly, particularly on the Internet, where new information emerges daily. If benchmarks are not regularly updated, they risk becoming obsolete. Under this philosophy, creating evolving, scalable benchmarks is essential. One approach is to manually update benchmarks, as seen with BFCL [130], BFCLv2, BFCLv3, and the swe-bench [27] series. By continually injecting new knowledge into benchmarks, they can stay relevant. Another approach involves connecting to real-time online evaluations, such as webvoyage [41] and InfoDeepSeek [205]. These benchmarks are highly scalable because they are directly connected to the ever-evolving Internet, maintaining their relevance and expansion capabilities.

From stateless to stateful. As agent evaluation evolves, there is a shift from stateless evaluations to stateful ones. Early benchmarks typically provide isolated, single-instance evaluations. However, agents in the real world interact with complex, evolving states. Thus, incorporating stateful evaluation, where the agent's past interactions and context influence its performance, is increasingly important. This approach allows for more accurate evaluations of an agent's ability to handle dynamic and persistent tasks over time, rather than just isolated scenarios.

Earlier benchmarks, especially in web or mobile environments, are generally stateless, focusing only on the agent's final state. Most benchmarks initialize the agent to a starting state, required the agent to complete a task, and then assessed whether it reached the final state (e.g., a booking confirmation page or a retrieved piece of information). However, the environments in these benchmarks were limited in their ability to evaluate the intermediate states of the agent' s process. As the field progresses, there is a shift toward considering the entire sequence of an agent's actions, not just the final outcome. Some benchmarks, such as ToolSandBox [131] and OmniAct [59], now assess intermediate states by introducing milestones or calculating a sequence score instead of just the final result. This evolution towards stateful evaluation is increasingly relevant in real-world tasks where agents must consider multiple intermediate states, such as handling interruptions in complex workflows. For example, in a multi-task scenario like "book a flight in 20 minutes, organize a

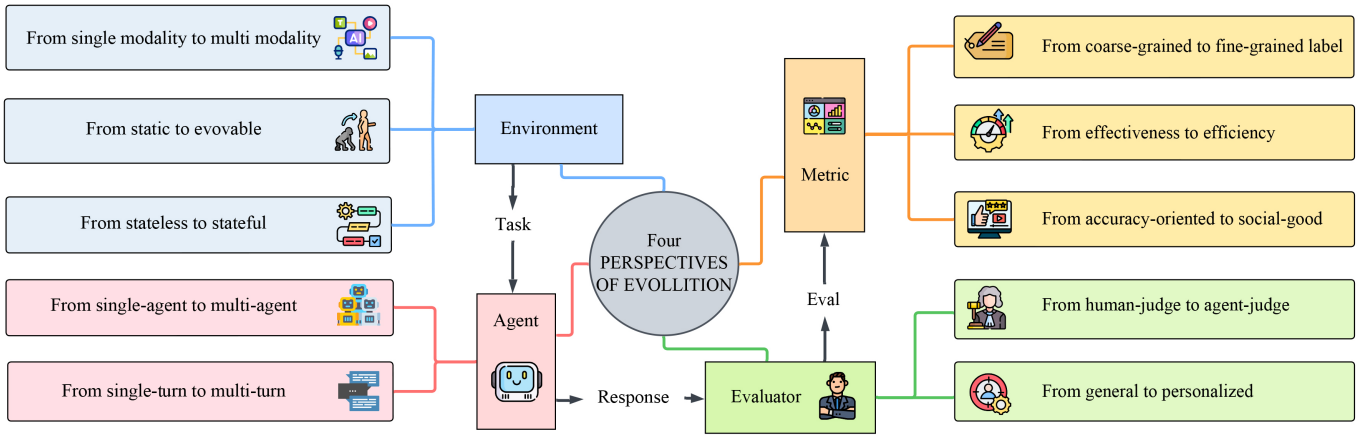


Fig. 3 Four perspectives of evolution prospect about AI agent evaluation

document, and search for information simultaneously," agents need to manage both task-specific states and more complex intermediate states. Recent benchmarks, like Worfbench [119], have already begun addressing such multi-task inputs, signaling a broader trend from stateless to stateful evaluation.

5.2 Agent perspective

From single-agent to multi-agent. Most current benchmarks focus on single-agent tasks, where the agent operates independently to solve a problem. However, as the field matures, the focus is shifting toward multi-agent systems, where agents collaborate or compete to achieve their respective goals. Multi-agent environments introduce new challenges in coordination, communication, and negotiation, all of which require novel benchmarking approaches. We believe multi-agent benchmarks hold significant potential. Currently, they face two key challenges.

Primarily, current benchmarks predominantly feature game-based or virtual tasks, lacking scenarios reflective of real-world collaborative activities, such as internet-based teamwork. Benchmarks like MultiAgentBench [149] and BattleAgentBench [148], while valuable for evaluating coordination and competition, do not adequately represent practical collaborative contexts.

Furthermore, most existing benchmarks focus on homogeneous multi-agent systems, neglecting the essential role of heterogeneous agents with specialized skills—such as reasoning (Deepseek [6]), retrieval (GPT [2]), and coding (Claude [172])—and hierarchical structures commonly seen in real-world settings (e.g., manager-executer dynamics). Although frameworks like DeepClaude [206] and AgentVerse [207] have begun exploring these specialized and hierarchical collaborations, standardized benchmarks for evaluating heterogeneous agent teams remain underdeveloped.

From single-turn to multi-turn. In the early stages of AI agent evaluation, benchmarks are often designed for single-turn interactions, where the agent receives a prompt and responds immediately. Tasks like single-turn retrieval or end-to-end task execution require minimal interaction. Benchmarks like WebArena [43] are designed for such scenarios. As agent capabilities improve, agents now engage in multi-turn dialogues to refine user needs or receive further instructions. Benchmarks like τ-Bench [143] or Weblinx [42] now accommodate these extended interactions, and we believe the future of agent evaluation will increasingly involve multi-turn conversations. This trend will naturally lead to the emergence of benchmarks that evaluate the ability to manage long contextual dialogue sequences.

5.3 Evaluator perspective

From human-judge to agent-judge. Initially, humans are seen as the best evaluators for agents, given that agents were designed to assist humans with tasks. Early evaluations, such as WebArena [43] or AgentArena [170], rely on human judges to assess agent performance. However, over time, it becomes apparent that using agent-based evaluators (agent-judges) could offer significant advantages. First, agent-judges can scale effectively without requiring additional human resources. Second, as AI agents' capabilities evolve and surpass human performance in some domains, human evaluation may no longer align with the agents' full potential. As such, using agents to evaluate other agents is increasingly seen as a necessary evolution in the field.

From general to personalized. Currently, most agents are designed to be general-purpose, aimed at completing tasks based on fixed instructions. Benchmarks reflect this approach, with tasks such as retrieving information or booking tickets. However, once agents are deployed in real-world commercial applications, they must consider factors like user preferences, profiles, and personal history [208–210]. Personalized agent benchmarks, such as PetoolBench [142], have emerged to evaluate how well agents adapt to individual users. These benchmarks will play a crucial role as personalized agents become more common in commercial settings.

5.4 Metric perspective

From coarse-grained to fine-grained label. Most benchmarks provide coarse-grained evaluations, offering broad feedback based on the final task outcome. However, a more granular evaluation system is necessary to capture the nuances of an agent's performance. Fine-grained labels, such as textual comments on an agent's decision-making process (e.g., "this agent's planning incurs high time costs" or "this agent's decisions are fast but risky"), could provide more valuable insights to developers. Although such fine-grained feedback may not be suitable for directly comparing agents by score or ranking which one is better, it can provide developers with more valuable insights, enabling the creation of improved agents [211–212].

From effectiveness to efficiency. While effectiveness metrics—such as task completion or accuracy—have traditionally been the focus of agent evaluation, efficiency is becoming an increasingly important consideration. How quickly and resource-efficiently an agent performs a task is essential. Benchmarks like Worfbench [119] and GOVSIM [155] already assess efficiency alongside effectiveness, and future benchmarks will need to balance both aspects to ensure that agents complete tasks efficiently, without overconsuming resources or time.

From accuracy-oriented to social-good. As AI agents are integrated into society, evaluating their alignment with societal values such as fairness, safety, and ethical behavior is becoming a critical focus. Current benchmarks have begun to incorporate safety metrics, such as those found in ST-WebAgentBench [50]. However, additional metrics, such as trustworthiness, robustness, and the ethical implications of agents' actions (e.g., making erroneous payments or generating harmful content), are still lacking and should be developed further in future benchmarks. We hope that agents will contribute to society in a socially beneficial way, rather than causing any harmful issues.

5.5 Benchmark selection methodology

Having detailed the evolution of agent evaluation from four distinct perspectives, we now return to address the core question posed at the beginning of our discussion section: *Given the rapid advancement of AI agents*, *how can we systematically evaluate LLM-based AI agents from an evolutionary perspective*? To tackle this question, we

propose a two-stage benchmark selection methodology containing present-focused selection and future-oriented selection, and illustrate it vividly through a practical use case.

Step 1: Present-focused selection. This step addresses the immediate evaluation needs that arise when developers have newly created an agent requiring immediate assessment. For example, consider the developer Z, who has developed an initial agent capable of booking flights and hotels. In this scenario, the developer can first consult Fig. 2 to identify relevant benchmark categories by considering the agent's external environment and internal capabilities. For this particular use case, the environment is web-based, and the primary capability involves interaction, specifically tool calling. Subsequently, the developer Z may proceed to the corresponding category sections to obtain an overview of each benchmark. Crucially, the developer could examine the associated category table provided in the Appendix A, which enumerates attributes we deem essential for informed benchmark selection such as modality, task domain, data source, and evaluation metrics. Based on these attributes, the developer Z can identify suitable benchmarks. In this use case scenario, benchmarks such as WebVoyager [41] (for web environments) and ComplexFuncBench [139] (for interaction capabilities) would be appropriate choices.

Step 2: Future-oriented selection. This second step considers the prospective evolution and ongoing improvement of agents, addressing the potential shifts in evaluation methodologies required as the agent matures. Developers can leverage Fig. 3, which provides a high-level prospective view on the evolution of agent evaluation. This figure assists developers in two critical areas: (1) identifying directions for further optimization of their agents, and (2) anticipating the future dimensions along which their agents might be assessed. Continuing our previous use case, the external environment might evolve dynamically, necessitating continuous attention to benchmarks adaptable to environmental changes exemplified by BFCL. Similarly, evaluation metrics may shift from purely task-oriented accuracy towards socially beneficial considerations like safety and robustness, reminding the developer Z of monitoring benchmarks like ST-WebAgentBench [50]. Additionally, as the developer Z's product moves towards commercialization, incorporating user history, greater attention to personalized benchmarks such as PeToolBench [142] will become increasingly important.

Through these two methodological steps, coupled with the illustrative use case provided, developers can systematically select suitable benchmarks for their agents. Furthermore, this approach equips developers with a high-level perspective on future evaluation trends, aiding in the continuous enhancement and comprehensive assessment of agent performance.

6 Conclusion

In this paper, we systematically explored the evolutionary transition from traditional LLM chatbots to advanced AI agents, highlighting critical evolution across five aspects: complex environment, multi-source instructor, dynamic feedback, multi-modal perception, and advanced capability. By adopting an evolutionary perspective, we addressed existing ambiguities in evaluation practices and offered a structured categorization of benchmarks aligned with external environments and internal capabilities. Our comprehensive attribute-based tables serve as practical resources for researchers aiming to select suitable evaluation frameworks. Furthermore, we discussed evolving trends and provided forward-looking insights on evaluation methodologies from four perspectives of environment, agent, evaluator, and metric. This work not only clarifies the evaluation landscape for AI agents but also sets the stage for future research directions, ultimately contributing to the ongoing advancement and refinement of AI agent systems.

Acknowledgements

The Shanghai Jiao Tong University team is partially supported by National Key RD Program of China (2022ZD0114804), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102) and National Natural Science Foundation of China (624B2096, 62322603, 62502310, 72595872, 72542012).

Competing interests

The authors declare that they have no competing interests or financial conflicts to disclose.

Open Access

This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

Appendices

To assist researchers in efficiently and accurately selecting appropriate evaluation benchmarks for various types of AI agents, we have systematically organized the majority of the categories identified in our taxonomy into tables. Specifically, for each benchmark category, we extract and summarize the most valuable attributes that characterize its intrinsic properties and suitability for different evaluation scenarios. These attributes are carefully selected to guide researchers in clearly understanding the strengths and limitations of each benchmark, thus providing structured guidance for evaluating AI agents and benchmark selection.

Below, we present the summarized tables for each benchmark category:

A.1 Coding environments

We categorize benchmarks of coding environments in Table 1. The

columns in Table 1 have the following meanings:

- Code environments/Datasets. Refers to the specific benchmarks or datasets used to evaluate AI agents or language models on coding-related tasks.
- Static/Interactive. Specifically classifies each benchmark as either static (isolated tasks without ongoing interaction) or interactive (tasks requiring continuous engagement with an environment or repository).
- Tasks. Describes the primary coding tasks evaluated by each benchmark, such as code generation, defect detection, retrieval, summarization, or software engineering tasks.
- Evaluation metric. Lists the metrics used to quantify model performance, including common metrics like Passk, Accuracy, Resolved Rate, Edit Similarity, and specialized indicators like Method/Field Dependencies.
- Evaluation dimensions. Clarifies the specific aspects or dimensions being assessed by each benchmark, for example, correctness of code, independency from context, accuracy in defect detection, or the overall completion quality of tasks.

A.2 Web environments

We categorize benchmarks of web environments in Table 2. The columns in Table 2 have the following meanings:

- Web environment. Identifies the specific benchmarks or datasets designed for evaluating AI agents in web-based tasks or scenarios.
- Realism. Describes how closely the evaluation environment replicates real-world conditions, ranging from synthetic (highly controlled or artificial), semi-real (partially realistic), to fully real environments.
- Offline/Online. Specifies whether the benchmark tasks are conducted offline (pre-recorded, simulated data or snapshot replay) or online (live, interactive connections with dynamic web content).
- Metrics. Lists the evaluation metrics used to quantify agent performance, including task success rate, trajectory likelihood, element accuracy, operation success, and other task-specific indicators.
- Evaluator. Indicates how evaluation results are obtained, whether through human annotators, automated systems, or combined human and automated evaluation methods, including similarity to LLM-generated judgments.
- Input modalities. Clarifies the types of input data provided to the agents, typically involving textual descriptions and images, or in some cases video and additional media.
- Main challenges. Summarizes the key difficulties or unique features presented by each benchmark, providing insight into the specific aspects that make these web-based evaluations challenging and relevant for assessing AI agents.

These columns collectively help researchers quickly understand the scope, complexity, realism, and evaluation approach of each web benchmark, guiding the selection of appropriate benchmarks for assessing web-based AI agents.

A.3 OS environments

We categorize benchmarks of OS environments in Table 3. The columns in Table 3 have the following meanings:

- Tasks. Lists the main types of tasks the AI agents are expected to perform within these OS benchmarks, such as office-related activities, system operations, web browsing, coding tasks, and daily workflow tasks.
- Observation type. Details the format and type of information provided to the agent for decision-making or interactions, such as screenshots, accessibility trees (a11y-tree), screen recordings, clipboard content, metadata, or textual descriptions.
- Action space. Defines the set of actions or interfaces available to the agent, including input devices like keyboards and mice, APIs, or specialized GUI automation libraries (e.g., PyAutoGUI).
- Platform. Indicates the specific operating systems or platforms supported by each benchmark, such as Ubuntu, Windows, Linux, macOS, or simulated desktop environments.
- Evaluation method. Describes the approach used to measure agent performance, including predefined scripts for automatic success rate calculation, action sequence matching, human evaluation, exact or fuzzy output matching, and penalties for inaccurate behaviors.

Collectively, these columns clearly outline the characteristics, operational details, and evaluation strategies of OS benchmarks, guiding researchers in selecting suitable platforms for assessing AI agents within realistic operating system contexts.

A.4 Mobile environments

We categorize benchmarks of mobile environments in Table 4. The columns in Table 4 have the following meanings:

- Static/Executable. Indicates whether the benchmark consists of static scenarios (tasks based on pre-defined and static interfaces or screens) or executable scenarios (dynamic, real-time interactions with mobile applications).
- Tasks. Specifies the core task evaluated by each benchmark, predominantly focusing on instruction-following capabilities within mobile interfaces.
- Observation type. Details the types of visual or textual information provided to the agent, such as screenshots, text view hierarchies, screen representations, HTML/XML content, and additional marks or metadata.
- Action space type. Defines the range of actions available to the agent, typically involving screen operations (e.g., tapping, swiping) and, in some benchmarks, macro functions or API calls.
- Platform. Denotes the mobile operating system for which the benchmark is intended, predominantly Android.
- Evaluation method. Describes how performance is measured,

using methods such as action sequence matching, LLM-assisted automatic judgments, rule-based evaluation, self-developed success detection, human validation, or UI state matching.

These columns collectively provide a clear overview of mobile benchmarks, highlighting how tasks are structured, how agent observations are presented, and how agent actions and performance are evaluated, thus guiding researchers in selecting appropriate mobile environment evaluation for AI agents.

A.5 Scientific environments

We categorize benchmarks of scientific environments in Table 5. The columns in Table 5 have the following meanings:

- Static/Interactive. Indicates whether the benchmark involves static scenarios (predefined, non-interactive tasks) or interactive environments (requiring agents to perform tasks dynamically, often in simulated or real workspaces).
- Environment. Describes the context or type of environment in which the benchmark tasks are set, such as research papers, scientific article pools, survey data, or simulated workspaces.
- Evaluation metric. Specifies the criteria and metrics used to assess agent performance, such as accuracy, F1 score, precision, recall, coverage, retrieval probability, area under the curve (AUC), success rate, error rate, and other task-specific indicators.
- Tasks. Details the main scientific or research-oriented tasks evaluated by each benchmark, including scientific question answering, research paper summarization, feedback generation, equation inference, data modeling, code generation, data analysis, and full-cycle scientific discovery.

Together, these columns provide a structured overview of scientific benchmarks, clarifying their operational format, application context, evaluation methodologies, and task focus, to guide researchers in choosing appropriate tools for assessing AI agents in scientific research scenarios.

A.6 Game environments

We categorize benchmarks of game environments in Table 6. The columns in Table 6 have the following meanings:

- Games. Specifies the types of games or specific game titles included within each benchmark, ranging from simple puzzles and board games to complex, strategic, or multiplayer games.
- Single-agent/Multi-agent. Indicates whether the benchmark focuses on evaluating single-agent scenarios (where an agent operates independently) or multi-agent scenarios (involving collaboration, competition, or interaction among multiple agents).
- Needed agent capabilities. Describes the key cognitive and behavioral abilities required from agents to succeed in the respective benchmark. These include skills such as reasoning, planning, decision-making, exploration, handling hidden information, cooperation, communication, and various forms of complex or multi-hop reasoning.

Together, these columns provide a comprehensive overview of game benchmarks, clarifying the context, interaction modality, and agent competencies needed, thereby assisting researchers in selecting appropriate gaming environments for evaluating diverse agent capabilities.

A.7 Planning capability

We categorize benchmarks of planning capabilities in Table 7. The columns in Table 7 have the following meanings:

- Domain category. Specifies the main subject area or reasoning type for each benchmark, such as mathematical reasoning, document navigation, or scientific knowledge.
- Task type. Describes the core task or problem-solving activity assessed by the benchmark, for example, algebraic word problem solving, multi-hop question answering, or implicit inference.
- Format. Indicates the format in which the benchmark is presented, most commonly as a static dataset containing predefined problems and questions.
- Key focus. Summarizes the main evaluation focus of the benchmark, highlighting aspects such as rationale generation, multi-step reasoning, performance on simple or complex variations, and the incorporation of charts or images.

A.8 Interaction capability

A.8.1 Static interaction

We categorize benchmarks of static interaction capabilities in Tables 8 and 9. The columns in these tables have the following meanings:

- Task domain. Specifies the application areas or domains each benchmark covers, such as daily life tasks, machine learning tasks, or mobile environment tasks.
- Task type. Describes the structure and complexity of the evaluated tasks, including single-turn or multi-turn API calls, chained or parallel functions, and multi-step tool use.
- API data source and authenticity. Details the origins and authenticity of the APIs or tools involved, indicating whether they are real, public, synthetic, handwritten, or from official repositories like RapidAPI or PyTorch Hub.
- Instruction data source and authenticity. Explains how the instructional or annotation data is generated for the benchmark, including whether it is LLM-generated, self-instructed, human-annotated, or derived from synthetic or real-world projects.
- Metric. Specifies the quantitative criteria or metrics used to evaluate agent performance, such as procedure accuracy, response accuracy, or hallucination rate.
- Evaluation label and method. Describes the way performance is measured and labeled, for example, by API name matching combined with automated evaluation procedures.
- Input modality. Indicates the type of input data provided to the agent, such as text.
- Main challenges. Summarizes the key difficulties or unique

features faced in each benchmark, such as generalizing to unseen APIs or handling large, dynamic, and overlapping API sets.

A.8.2 Interaction with humans

We categorize benchmarks of interaction with human capabilities in Table 10. The columns in the table have the following meanings:

- Dialogue task type. Describes the type or domain of dialogue tasks evaluated, such as specific user intents or multi-domain dialogues.
- Dialogue data source. Indicates how the dialogue data is collected or generated, e.g., via simulated users, crowd-sourcing, or automated processes.
- Metrics. Specifies the criteria or metrics used for evaluation, including task accuracy, action state tracking, belief tracking, API correctness, or dialogue success.
- Labeling and similarity evaluation. Explains the method for labeling responses and assessing similarity, which may involve human annotation, automated methods, or manual assessment.
- Modality. Indicates the type of input or interaction modality used, such as text.
- Explicit interaction evaluation beyond task accuracy. States whether the benchmark includes explicit evaluation of interaction quality in addition to basic task accuracy (typically yes/no).

A.8.3 Interaction with other agents

We categorize benchmarks of capabilities of interaction with other agents in Table 11. The columns in the table have the following meanings:

- Interaction paradigm. Specifies the nature of interactions among agents, such as cooperation, competition, or both.
- Modality. Indicates the format or channel of agent interactions, typically text-based.
- Metrics. Describes the performance metrics used for evaluation, such as rule understanding, collaboration quality, task completion, score, or resource efficiency.
- Application domain. States the primary domain or scenario for the benchmark, such as games, collaborative programming, social tasks, or resource management.
- Agent architecture. Indicates whether the agents in the benchmark are homogeneous (identical) or heterogeneous (diverse in capability or function).
- Key contributions. Summarizes the unique features, innovations, or evaluation focuses of each benchmark, such as multi-level difficulty, process-oriented metrics, simulation of social intelligence, or evaluation of planning and collaboration abilities.

A.9 Memory capability

We categorize benchmarks of the memory capability in Table 12. The columns in the table have the following meanings:

- Data source. Specifies the origins of the data used in each benchmark, such as books, scripts, real meeting transcripts, synthetic conversations, or manually created questions.
- Task domain. Describes the primary subject area or scenario being evaluated, including narrative understanding, product design, academic discussions, social dialogue, or programming.
- Task type. Outlines the main task structure, such as question answering (QA), query-based summarization, event summarization, memory synthesis, classification, or online streaming learning.
- Memory type. Indicates the type of memory evaluated, such as long-range, causal, temporal-spatial, semantic, episodic, foresight, multimodal, dynamic update, or incremental improvement.
- Metrics. Lists the evaluation metrics used to assess performance, for example, BLEU, ROUGE, accuracy, F1 score, NDCG@k, recall@k, MMRelevance, MAP, and hallucination rate.

References

[1] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser Ł, Polosukhin I. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017, 6000−6010

[2] Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, et al. Gpt-4 technical report. 2023, arXiv preprint arXiv: 2303.08774

[3] Touvron H, Lavril T, Izacard G, Martinet X, Lachaux M A, Lacroix T, Rozière B, Goyal N, Hambro E, Azhar F, Rodriguez A, Joulin A, Grave E, Lample G. LLaMA: open and efficient foundation language models. 2023, arXiv preprint arXiv: 2302.13971

[4] Anil R, Borgeaud S, Alayrac J B, Yu J, Soricut R, et al. Gemini: a family of highly capable multimodal models. 2023, arXiv preprint arXiv: 2312.11805

[5] Bai J, Bai S, Chu Y, Cui Z, Dang K, et al. Qwen technical report. 2023, arXiv preprint arXiv: 2309.16609

[6] Guo D, Yang D, Zhang H, Song J, Wang P, et al. DeepSeek-R1: incentivizing reasoning capability in LLMs via reinforcement learning. 2025, arXiv preprint arXiv: 2501.12948

[7] Liu W, Qin J, Huang X, Zeng X, Xi Y, Lin J, Wu C, Wang Y, Shang L, Tang R, Lian D, Yu Y, Zhang W. The real barrier to LLM agent usability is agentic ROI. 2025, arXiv preprint arXiv: 2505.17767

[8] Wang L, Ma C, Feng X, Zhang Z, Yang H, Zhang J, Chen Z, Tang J, Chen X, Lin Y, Zhao W X, Wei Z, Wen J. A survey on large language model based autonomous agents. Frontiers of Computer Science, 2024, 18(6): 186345

[9] Yang Y, Chai H, Song Y, Qi S, Wen M, Li N, Liao J, Hu H, Lin J, Chang G, Liu W, Wen Y, Yu Y, Zhang W. A survey of AI agent protocols. 2025, arXiv preprint arXiv: 2504.16736

[10] Akkil D, Azam R, Abuelsaad T, Dey P, Vempaty A, Jagmohan A, Kokku R. Agent-E: from autonomous web navigation to foundational design principles in agentic systems. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[11] Song Y, Xu F F, Zhou S, Neubig G. Beyond browsing: API-based web agents. In: Proceedings of Findings of the Association for Computational Linguistics. 2025, 11066−11085

[12] Shinn N, Cassano F, Gopinath A, Narasimhan K, Yao S. Reflexion: language agents with verbal reinforcement learning. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023, 377

[13] Chang Y, Wang X, Wang J, Wu Y, Yang L, Zhu K, Chen H, Yi X, Wang C, Wang Y, Ye W, Zhang Y, Chang Y, Yu P S, Yang Q, Xie X. A survey on evaluation of large language models. ACM Transactions on Intelligent Systems and Technology, 2024, 15(3): 39

[14] Laskar T R, Alqahtani S, Bari M S, Rahman M, Khan M AM, Khan H, Jahan I, Bhuiyan A, Tan C W, Parvez R, Hoque E, Joty S, Huang J. A systematic survey and critical review on evaluating large language models: challenges, limitations, and recommendations. In: Proceedings of 2024 Conference on Empirical Methods in Natural Language Processing. 2024, 13785−13816

[15] Guo Z, Jin R, Liu C, Huang Y, Shi D, Supryadi, Yu L, Liu Y, Li J, Xiong B, Xiong D. Evaluating large language models: a comprehensive survey. 2023, arXiv preprint arXiv: 2310.19736

[16] Chen M, Tworek J, Jun H, Yuan Q, de Oliveira Pinto H P, et al. Evaluating large language models trained on code. 2021, arXiv preprint arXiv: 2107.03374

[17] Du X, Liu M, Wang K, Wang H, Liu J, Chen Y, Feng J, Sha C, Peng X, Lou Y. ClassEval: a manually-crafted benchmark for evaluating LLMs on class-level code generation. 2023, arXiv preprint arXiv: 2308.01861

[18] Yu H, Shen B, Ran D, Zhang J, Zhang Q, Ma Y. CoderEval: a benchmark of pragmatic code generation with generative pretrained models. In: Proceedings of the 46th IEEE/ACM International Conference on Software Engineering. 2024, 428−439

[19] Yuan Z, Liu J, Zi Q, Liu M, Peng X, Lou Y. Evaluating instruction-tuned large language models on code comprehension and generation. 2023, arXiv preprint arXiv: 2308.01240

[20] Zhang A, Dong M, Liu J, Zhang W, Wang Y, Yang J, Zhang G, Liu T, Peng Z, Tan Y, Zhang Y, Wang Z, Wang W, He Y, Deng K, Zhou W, Huang W, Zhang Z. CodeCriticBench: a holistic code critique benchmark for large language models. 2025, arXiv preprint arXiv: 2502.16614

[21] Gu A, Rozière B, Leather H, Solar-Lezama A, Synnaeve G, Wang S I. CRUXEval: a benchmark for code reasoning, understanding and execution. In: Proceedings of the 41st International Conference on Machine Learning. 2024, 659

[22] Zhuo T Y, Chien V M, Chim J, Hu H, Yu W, et al. BigCodeBench: Benchmarking code generation with diverse function calls and complex instructions. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[23] Yin P, Li W D, Xiao K, Rao A, Wen Y, Shi K, Howland J, Bailey P, Catasta M, Michalewski H, Polozov O, Sutton C. Natural language to code generation in interactive data science notebooks. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics. 2023, 126−173

[24] Jimenez C E, Yang J, Wettig A, Yao S, Pei K, Press O, Narasimhan K R. SWE-bench: can language models resolve real-world GitHub issues?. In: Proceedings of the 12th International Conference on Learning Representations. 2024

[25] Yang J, Jimenez C E, Zhang A L, Lieret K, Yang J, Wu X, Press O, Muennighoff N, Synnaeve G, Narasimhan K R, Yang D, Wang S, Press O. SWE-bench multimodal: do AI systems generalize to visual software domains?. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[26] Antoniades A, Örwall A, Zhang K, Xie Y, Goyal A, Wang W Y. SWE-search: enhancing software agents with Monte Carlo tree search and iterative refinement. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[27] Zan D, Huang Z, Yu A, Lin S, Shi Y, Liu W, Chen D, Qi Z, Yu H, Yu L, Ran D, Zeng M, Shen B, Bian P, Liang G, Guan B, Huang P, Xie T, Wang Y, Wang Q. SWE-bench-java: a GitHub issue resolving benchmark for java. 2024, arXiv preprint arXiv: 2408.14354

[28] SWE arena: an open evaluation platform for automated software engineering. 2024

[29] Yadavally A, Nguyen H, Callot L, Guinet G. Large language model critics for execution-free evaluation of code changes. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[30] Aleithan R, Xue H, Mohajer M M, Nnorom E, Uddin G, Wang S. SWE-bench+: enhanced coding benchmark for LLMs. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[31] Liu T, Xu C, McAuley J. RepoBench: benchmarking repository-level code auto-completion systems. In: Proceedings of the 12th International Conference on Learning Representations. 2024

[32] Mündler N, Müller M N, He J, Vechev M. SWT-bench: testing and validating real-world bug-fixes with code agents. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. 2024, 2601

[33] Qian C, Liu W, Liu H, Chen N, Dang Y, Li J, Yang C, Chen W, Su Y, Cong X, Xu J, Li D, Liu Z, Sun M. ChatDev: communicative agents for software development. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics. 2024, 15174−15186

[34] Li B, Wu W, Tang Z, Shi L, Yang J, Li J, Yao S, Qian C, Hui B, Zhang Q, Yu Z, Du H, Yang P, Lin D, Peng C, Chen K. Prompting large language models to tackle the full software development lifecycle: a case study. In: Proceedings of the 31st International Conference on Computational Linguistics. 2025, 7511−7531

[35] Tang X, Liu Y, Cai Z, Shao D, Lu J, Zhang Y, Deng Z, Hu H, An K, Huang R, Si S, Sheng C, Zhao H, Chen L, Liu T, Qin Y, Zhou W, Zhao Y, Jiang Z, Chang B, Cohan A, Gerstein M. ML-bench: evaluating large language models and agents for machine learning tasks on repository-level code. 2023, arXiv preprint arXiv: 2311.09835

[36] Zhang Y, Pan Y, Wang Y, Cai J, Zheng Z, Zeng G, Liu Z. PyBench: evaluating LLM agent on various real-world coding tasks. 2024, arXiv preprint arXiv: 2407.16732v1

[37] Shi T, Karpathy A, Fan L, Hernandez J, Liang P. World of bits: an open-domain platform for web-based agents. In: Proceedings of the 34th International Conference on Machine Learning. 2017, 3135−3144

[38] Liu E Z, Guu K, Pasupat P, Shi T, Liang P. Reinforcement learning on web interfaces using workflow-guided exploration. In: Proceedings of the 6th International Conference on Learning Representations. 2018

[39] Yao S, Chen H, Yang J, Narasimhan K. WebShop: towards scalable real-world web interaction with grounded language agents. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 1508

[40] Deng X, Gu Y, Zheng B, Chen S, Stevens S, Wang B, Sun H, Su Y. Mind2Web: towards a generalist agent for the web. In: Proceedings of the 37th Conference on Neural Information Processing Systems Datasets and Benchmarks Track. 2023

[41] He H, Yao W, Ma K, Yu W, Dai Y, Zhang H, Lan Z, Yu D. WebVoyager: building an end-to-end web agent with large multimodal models. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics. 2024, 6864−6890

[42] Lù X H, Kasner Z, Reddy S. WEBLINX: real-world website navigation with multi-turn dialogue. In: Proceedings of the 41st International Conference on Machine Learning. 2024, 1340

[43] Zhou S, Xu F F, Zhu H, Zhou X, Lo R, Sridhar A, Cheng X, Ou T, Bisk Y, Fried D, Alon U, Neubig G. WebArena: a realistic web environment for building autonomous agents. In: Proceedings of 2023 Foundation Models for Decision Making Workshop. 2023

[44] Koh J Y, Lo R, Jang L, Duvvur V, Lim M, Huang P Y, Neubig G, Zhou S, Salakhutdinov R, Fried D. VisualWebarena: evaluating multimodal agents on realistic visual web tasks. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics. 2024, 881−905

[45] Drouin A, Gasse M, Caccia M, Laradji I H, Del Verme M, Marty T, Vazquez D, Chapados N, Lacoste A. WorkArena: how capable are web agents at solving common knowledge work tasks?. In: Proceedings of the 41st International Conference on Machine Learning. 2024, 462

[46] Boisvert L, Thakkar M, Gasse M, Caccia M, Le Sellier De Chezelles T, Cappart Q, Chapados N, Lacoste A, Drouin A. WorkArena++: towards compositional planning and reasoning-based common knowledge work tasks. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. 2024, 195

[47] Tian S, Zhang Z, Chen L, Liu Z. MMInA: benchmarking multihop multimodal internet agents. In: Proceedings of Findings of the Association for Computational Linguistics. 2025, 13682−13697

[48] Yoran O, Amouyal S J, Malaviya C, Bogin B, Press O, Berant J. AssistantBench: can web agents solve realistic and time-consuming tasks?. In: Proceedings of 2024 Conference on Empirical Methods in Natural Language Processing. 2024, 8938−8968

[49] Pan Y, Kong D, Zhou S, Cui C, Leng Y, Jiang B, Liu H, Shang Y, Zhou S, Wu T, Wu Z. WebCanvas: benchmarking web agents in online environments. In: Proceedings of Agentic Markets Workshop. 2024

[50] Levy I, Wiesel B, Marreed S, Oved A, Yaeli A, Shlomov S. St-WebAgentBench: a benchmark for evaluating safety and trustworthiness in web agents. In: Proceedings of Workshop on Computer Use Agents. 2025

[51] Jang L K, Li Y, Zhao D, Ding C, Lin J, Liang P P, Bonatti R, Koishida K. VideoWebArena: evaluating long context multimodal agents with video understanding web tasks. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[52] Xu K, Kordi Y, Nayak T, Asija A, Wang Y, Sanders K, Byerly A, Zhang J, Van Durme B, Khashabi D. Tur[k]ingBench: a challenge benchmark for web agents. In: Proceedings of 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies. 2025, 3694−3710

[53] Song Y, Thai K, Pham C M, Chang Y, Nadaf M, Iyyer M. BEARCUBS: a benchmark for computer-using web agents. In: Proceedings of the 2nd Conference on Language Modeling. 2025

[54] Xu F F, Song Y, Li B, Tang Y, Jain K, et al. TheAgentCompany: benchmarking LLM agents on consequential real world tasks. In: Proceedings of the 39th Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track. 2025

[55] Kara S, Faisal F, Nath S. WABER: evaluating reliability and efficiency of web agents with existing benchmarks. In: Proceedings of 2025 Workshop on Foundation Models in the Wild. 2025

[56] Liu X, Zhang T, Gu Y, Iong I L, Song X, et al. VisualAgentBench: towards large multimodal models as visual foundation agents. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[57] Xue T, Qi W, Shi T, Song C H, Gou B, Song D, Sun H, Su Y. An illusion of progress? Assessing the current state of web agents. In: Proceedings of the 2nd Conference on Language Modeling. 2025

[58] Garg D, Caples D, Draguns A, Ravi N, Putta P, Garg N, Hebbar P, Joo Y, Gu J, London C, de Witt C S, Motwani S R. REAL: benchmarking autonomous agents on deterministic simulations of real websites. In: Proceedings of the 39th Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track. 2025

[59] Kapoor R, Butala Y P, Russak M, Koh J Y, Kamble K, AlShikh W, Salakhutdinov R. OmniACT: a dataset and benchmark for enabling multimodal generalist autonomous agents for desktop and web. In: Proceedings of the 18th European Conference on Computer Vision. 2024, 161−178

[60] Chen S, Wiseman S, Dhingra B. ChatShop: interactive information seeking with language agents. 2024, arXiv preprint arXiv: 2404.09911

[61] Xie T, Zhang D, Chen J, Li X, Zhao S, Cao R, Hua T, Cheng Z, Shin D, Lei F, Liu Y, Xu Y, Zhou S, Savarese S, Xiong C, Zhong V, Yu T. OSWORLD: benchmarking multimodal agents for open-ended tasks in real computer environments. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. 2024, 1650

[62] Bonatti R, Zhao D, Bonacci F, Dupont D, Abdali S, Li Y, Lu Y, Wagle J, Koishida K, Bucker A, Jang L K, Hui Z. Windows agent arena: evaluating multi-modal OS agents at scale. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[63] Zheng L, Huang Z, Xue Z, Wang X, An B, Yan S. AgentStudio: a toolkit for building general virtual agents. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[64] Wang Z, Cui Y, Zhong L, Zhang Z, Yin D, Lin B Y, Shang J. OfficeBench: benchmarking language agents across multiple applications for office automation. 2024, arXiv preprint arXiv: 2407.19056

[65] Liu H, Zhang X, Xu H, Wanyan Y, Wang J, Yan M, Zhang J, Yuan C, Xu C, Hu W, Huang F. PC-agent: a hierarchical agentic framework for complex task automation on PC. In: Proceedings of Workshop on Reasoning and Planning for Large Language Models. 2025

[66] Li Y, He J, Zhou X, Zhang Y, Baldridge J. Mapping natural language instructions to mobile UI action sequences. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020, 8198−8210

[67] Venkatesh S G, Talukdar P, Narayanan S. UGIF: UI grounded instruction following. 2022, arXiv preprint arXiv: 2211.07615

[68] Burns A, Arsan D, Agrawal S, Kumar R, Saenko K, Plummer B A. Mobile app tasks with iterative feedback (MoTIF): addressing task feasibility in interactive visual environments. 2021, arXiv preprint arXiv: 2104.08560

[69] Rawles C, Li A, Rodriguez D, Riva O. Android in the wild: a large-scale dataset for android device control. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023, 2609

[70] Li W, Bishop W, Li A, Rawles C, Campbell-Ajala F, Tyamagundlu D, Riva O. On the effects of data scale on UI control agents. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. 2024, 2925

[71] Chai Y, Huang S, Niu Y, Xiao H, Liu L, Wang G, Zhang D, Ren S, Li H. AMEX: android multi-annotation expo dataset for mobile GUI agents. In: Proceedings of Findings of the Association for Computational Linguistics. 2025, 2138−2156

[72] Lee J, Min T, An M, Hahm D, Lee H, Kim C, Lee K. B-MoCA: benchmarking mobile device control agents across diverse configurations. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[73] Rawles C, Clinckemaillie S, Chang Y, Waltz J, Lau G, Fair M, Li A, Bishop W E, Li W, Campbell-Ajala F, Toyama D K, Berry R J, Tyamagundlu D, Lillicrap T P, Riva O. AndroidWorld: a dynamic benchmarking environment for autonomous agents. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[74] Deng S, Xu W, Sun H, Liu W, Tan T, Liu J, Li A, Luan J, Wang B, Yan R, Shang S. Mobile-bench: an evaluation benchmark for LLM-based mobile agents. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics. 2024, 8813−8831

[75] Chai Y, Tang S, Xiao H, Lin W, Liu L, Li H, Zhang J, Zhao P, Liu G, Han R, Wang G, Ren S, Huang S, Li H. A3: android agent arena for mobile GUI agents. In: Proceedings of the 14th International Conference on Learning Representations. 2026

[76] Zhang D, Shen Z, Xie R, Zhang S, Xie T, Zhao Z, Chen S, Chen L, Xu H, Cao R, Yu K. Mobile-Env: building qualified evaluation benchmarks for LLM-GUI interaction. 2023, arXiv preprint arXiv: 2305.08144

[77] Xing M, Zhang R, Xue H, Chen Q, Yang F, Xiao Z. Understanding the weakness of large language model agents within a complex android environment. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2024, 6061−6072

[78] Lee J, Hahm D, Choi J S, Knox W B, Lee K. MobileSafetyBench: evaluating safety of autonomous agents in mobile device control. 2024, arXiv preprint arXiv: 2410.17520

[79] Wang L, Deng Y, Zha Y, Mao G, Wang Q, Min T, Chen W, Chen S. MobileAgentBench: an efficient and user-friendly benchmark for mobile LLM agents. 2024, arXiv preprint arXiv: 2406.08184

[80] Chen J, Yuen D, Xie B, Yang Y, Chen G, Wu Z, Li Y, Zhou X, Liu W, Wang S, Zhou K, Shao R, Nie L, Wang Y, Hao J, Wang J, Shao K. SPA-BENCH: a comprehensive benchmark for smartphone agent evaluation. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[81] Zhang L, Wang S, Jia X, Zheng Z, Yan Y, Gao L, Li Y, Xu M. LlamaTouch: a faithful and scalable testbed for mobile UI task automation. In: Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology. 2024, 132

[82] Xu Y, Liu X, Sun X, Cheng S, Yu H, Lai H, Zhang S, Zhang D, Tang J, Dong Y. AndroidLab: training and systematic benchmarking of android autonomous agents. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics. 2025, 2144−2166

[83] Sun J, Hua Z, Xia Y. AutoEval: a practical framework for autonomous evaluation of mobile agents. 2025, arXiv preprint arXiv: 2503.02403

[84] Clark P, Cowhey I, Etzioni O, Khot T, Sabharwal A, Schoenick C, Tafjord O. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. 2018, arXiv preprint arXiv: 1803.05457

[85] Lu P, Mishra S, Xia T, Qiu L, Chang K W, Zhu S C, Tafjord O, Clark P, Kalyan A. Learn to explain: multimodal reasoning via thought chains for science question answering. In: Proceedings of the 36th Conference on Neural Information Processing Systems. 2022

[86] Dasigi P, Lo K, Beltagy I, Cohan A, Smith N A, Gardner M. A dataset of information-seeking questions and answers anchored in research papers. In: Proceedings of 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2021, 4599−4610

[87] Lee Y, Lee K, Park S, Hwang D, Kim J, Lee H I, Lee M. QASA: advanced question answering on scientific articles. In: Proceedings of the 40th International Conference on Machine Learning. 2023, 19036−19052

[88] Chamoun E, Schlichtkrull M, Vlachos A. Automated focused feedback generation for scientific writing assistance. In: Proceedings of Findings of the Association for Computational Linguistics. 2024, 9742−9763

[89] Lou R, Xu H, Wang S, Du J, Kamoi R, Lu X, Xie J, Sun Y, Zhang Y, Ahn J J, Fang H, Zou Z, Ma W, Li X, Zhang K, Xia C, Huang L, Yin W. AAAR-1.0: assessing AI's potential to assist research. In: Proceedings of the 42nd International Conference on Machine Learning. 2025

[90] DeYoung J, Beltagy I, van Zuylen M, Kuehl B, Wang L L. MS^2: multi-document summarization of medical studies. In: Proceedings of 2021 Conference on Empirical Methods in Natural Language Processing. 2021, 7494−7513

[91] Laurent J M, Janizek J D, Ruzo M, Hinks M M, Hammerling M J, Narayanan S, Ponnapati M, White A D, Rodriques S G. LAB-bench: measuring capabilities of language models for biology research. 2024, arXiv preprint arXiv: 2407.10362

[92] Kang H, Xiong C. ResearchArena: benchmarking large language models' ability to collect and organize information as research agents. In: Proceedings of Findings of the Association for Computational Linguistics. 2025, 5653−5671

[93] Si C, Yang D, Hashimoto T. Can LLMs generate novel research ideas? A large-scale human study with 100+ NLP researchers. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[94] Lála J, ODonoghue O, Shtedritski A, Cox S, Rodriques S G, White A. PaperQA: retrieval-augmented generative agent for scientific research. In: Proceedings of the 12th International Conference on Learning

Representations. 2024

[95] Wang R, Jansen P, Côté M A, Ammanabrolu P. ScienceWorld: is your agent smarter than a 5th grader?. In: Proceedings of 2022 Conference on Empirical Methods in Natural Language Processing. 2022, 11279−11298

[96] Jansen P, Côté M A, Khot T, Bransom E, Mishra B D, Majumder B P, Tafjord O, Clark P. DiscoveryWorld: a virtual environment for developing and evaluating automated scientific discovery agents. In: Proceedings of the 38th Conference on Neural Information Processing Systems Datasets and Benchmarks Track. 2024

[97] Chen Z, Chen S, Ning Y, Zhang Q, Wang B, Yu B, Li Y, Liao Z, Wei C, Lu Z, Dey V, Xue M, Baker F N, Burns B, Adu-Ampratwum D, Huang X, Ning X, Gao S, Su Y, Sun H. ScienceAgentBench: toward rigorous assessment of language agents for data-driven scientific discovery. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[98] Nathani D, Madaan L, Roberts N, Bashlykov N, Menon A, Moens V, Plekhanov M, Budhiraja A, Magka D, Vorotilov V, Chaurasia G, Hupkes D, Cabral R S, Shavrina T, Foerster J N, Bachrach Y, Wang W Y, Raileanu R. MLGym: a new framework and benchmark for advancing AI research agents. In: Proceedings of the 2nd Conference on Language Modeling. 2025

[99] Huang Q, Vora J, Liang P, Leskovec J. MLAgentBench: evaluating language agents on machine learning experimentation. In: Proceedings of the 41st International Conference on Machine Learning. 2024, 814

[100] Jing L, Huang Z, Wang X, Yao W, Yu W, Ma K, Zhang H, Du X, Yu D. DSBench: how far are data science agents from becoming data science experts?. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[101] Zhang Y, Jiang Q, Han X, Chen N, Yang Y, Ren K. Benchmarking data science agents. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics. 2024, 5677−5700

[102] Huang Y, Luo J, Yu Y, Zhang Y, Lei F, Wei Y, He S, Huang L, Liu X, Zhao J, Liu K. DA-code: agent data science code generation benchmark for large language models. In: Proceedings of 2024 Conference on Empirical Methods in Natural Language Processing. 2024, 13487−13521

[103] Cao R, Lei F, Wu H, Chen J, Fu Y, et al. Spider2-V: how far are multimodal agents from automating data science and engineering workflows?. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. 2024, 3421

[104] Castillo-Bolado D, Davidson J, Gray F, Rosa M. Beyond prompts: dynamic conversational benchmarking of large language models. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. 2024, 1347

[105] Paglieri D, Cupiał B, Coward S, Piterbarg U, Wolczyk M, Khan A, Pignatelli E, Kuciński Ł, Pinto L, Fergus R, Foerster J N, Parker-Holder J, Rocktäschel T. BALROG: benchmarking agentic LLM and VLM reasoning on games. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[106] Wu Y, Tang X, Mitchell T M, Li Y. SmartPlay: a benchmark for LLMs as intelligent agents. In: Proceedings of the 12th International Conference on Learning Representations. 2024

[107] Wang X, Zhuang B, Wu Q. Are large vision language models good game players?. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[108] Ren Y, Tertikas K, Maiti S, Han J, Zhang T, Süsstrunk S, Kokkinos F. VGRP-bench: visual grid reasoning puzzle benchmark for large vision-language models. 2025, arXiv preprint arXiv: 2503.23064

[109] Zhang H, Guo H, Guo S, Cao M, Huang W, Liu J, Zhang G. ING-VP: MLLMs cannot play easy vision-based games yet. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[110] Tang W, Zhou Y, Xu E, Cheng K, Li M, Xiao L. DSGBench: a diverse strategic game benchmark for evaluating LLM-based agents in complex decision-making environments. 2025, arXiv preprint arXiv: 2503.06047

[111] Costarelli A, Allen M, Hauksson R, Sodunke G, Hariharan S, Cheng C, Yadav A. GameBench: evaluating strategic reasoning abilities of LLM agents. In: Proceedings of the 38th Conference on Neural Information Processing Systems Datasets and Benchmarks Track. 2024

[112] Huang J T, Li E J, Lam M H, Liang T, Wang W, Yuan Y, Jiao W, Wang X, Tu Z, Lyu M. Competing large language models in multi-agent gaming environments. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[113] Duan J, Zhang R, Diffenderfer J, Kailkhura B, Sun L, Stengel-Eskin E, Bansal M, Chen T, Xu K. GTBENCH: uncovering the strategic reasoning limitations of LLMs via game-theoretic evaluations. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. 2024, 885

[114] Hu L, Li Q, Xie A, Jiang N, Stoica I, Jin H, Zhang H. GameArena: Evaluating LLM reasoning through live computer games. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[115] Valmeekam K, Marquez M, Olmo A, Sreedharan S, Kambhampati S. PlanBench: an extensible benchmark for evaluating large language models on planning and reasoning about change. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023, 1693

[116] Valmeekam K, Marquez M, Sreedharan S, Kambhampati S. On the planning abilities of large language models: a critical investigation. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023, 3320

[117] Stein K, Fišer D, Hoffmann J, Koller A. AutoPlanBench: automatically generating benchmarks for LLM planners from PDDL. 2023, arXiv preprint arXiv: 2311.09830

[118] Kokel H, Katz M, Srinivas K, Sohrabi S. ACPBench: reasoning about action, change, and planning. In: Proceedings of the 39th AAAI Conference on Artificial Intelligence. 2025, 26559−26568

[119] Qiao S, Fang R, Qiu Z, Wang X, Zhang N, Jiang Y, Xie P, Huang F, Chen H. Benchmarking agentic workflow generation. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[120] Xiao R, Ma W, Wang K, Wu Y, Zhao J, Wang H, Huang F, Li Y. FlowBench: revisiting and benchmarking workflow-guided planning for LLM-based agents. In: Proceedings of Findings of the Association for Computational Linguistics. 2024, 10883−10900

[121] Geng L, Chang E Y. REALM-bench: a real-world planning benchmark for LLMs and multi-agent systems. 2025, arXiv preprint

arXiv: 2502.18836

[122] Zhang L, Wang Y, Gu H, Hamidizadeh A, Zhang Z, Liu Y, Wang Y, Arcos Bravo D G, Dong J, Zhou S, Cao T, Quan X, Zhuang Y, Zhang Y, Hao J. Et-plan-bench: embodied task-level planning benchmark towards spatial-temporal cognition with foundation models. In: Proceedings of 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2025, 21566−21573

[123] Cheng C A, Kolobov A, Misra D, Nie A, Swaminathan A. LLF-bench: benchmark for interactive learning from language feedback. 2023, arXiv preprint arXiv: 2312.06853

[124] You J, Liu M, Prabhumoye S, Patwary M, Shoeybi M, Catanzaro B. LLM-evolve: evaluation for LLM's evolving capability on benchmarks. In: Proceedings of 2024 Conference on Empirical Methods in Natural Language Processing. 2024, 16937−16942

[125] Li L, Wang Y, Zhao H, Kong S, Teng Y, Li C, Wang Y. Reflection-bench: probing AI intelligence with reflection. 2024, arXiv preprint arXiv: 2410.16270v1

[126] Pan J, Shar R, Pfau J, Talwalkar A, He H, Chen V. When benchmarks talk: re-evaluating code LLMs with interactive feedback. In: Proceedings of Findings of the Association for Computational Linguistics. 2025, 24672−24700

[127] Tang Q, Deng Z, Lin H, Han X, Liang Q, Cao B, Sun L. ToolAlpaca: generalized tool learning for language models with 3000 simulated cases. 2023, arXiv preprint arXiv: 2306.05301

[128] Peng Y, Li S, Gu W, Li Y, Wang W, Gao C, Lyu M. Revisiting, benchmarking and exploring API recommendation: how far are we?. IEEE Transactions on Software Engineering, 2023, 49(4): 1876−1897

[129] Qin Y, Liang S, Ye Y, Zhu K, Yan L, Lu Y, Lin Y, Cong X, Tang X, Qian B, Zhao S, Hong L, Tian R, Xie R, Zhou J, Gerstein M, Li D, Liu Z, Sun M. ToolLLM: facilitating large language models to master 16000+ real-world APIs. In: Proceedings of the 12th International Conference on Learning Representations. 2024

[130] Team G. Berkeley function calling leaderboard v3 (aka berkeley tool calling leaderboard v3). Gorilla, 2024

[131] Lu J, Holleis T, Zhang Y, Aumayer B, Nan F, Bai H, Ma S, Ma S, Li M, Yin G, Wang Z, Pang R. ToolSandbox: a stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities. In: Proceedings of Findings of the Association for Computational Linguistics. 2025, 1160−1183

[132] Wu M, Zhu T, Han H, Tan C, Zhang X, Chen W. Seal-tools: self-instruct tool learning dataset for agent tuning and detailed benchmark. In: Proceedings of the 13th National CCF Conference on Natural Language Processing and Chinese Computing. 2024, 372−384

[133] Li M, Zhao Y, Yu B, Song F, Li H, Yu H, Li Z, Huang F, Li Y. API-bank: a comprehensive benchmark for tool-augmented LLMs. In: Proceedings of 2023 Conference on Empirical Methods in Natural Language Processing. 2023, 3102−3116

[134] Team N. Nexusraven-13b, a new sota open-source LLM for function calling. Nexusflow, 2024

[135] Basu K, Abdelaziz I, Chaudhury S, Dan S, Crouse M, Munawar A, Austel V, Kumaravel S, Muthusamy V, Kapanipathi P, Lastras L. API-BLEND: a comprehensive corpora for training and benchmarking API LLMs. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics. 2024, 12859−12870

[136] Song Y, Xiong W, Zhu D, Wu W, Qian H, Song M, Huang H, Li C, Wang K, Yao R, Tian Y, Li S. RestGPT: connecting large language models with real-world restful APIS. 2023, arXiv preprint arXiv: 2306.06624

[137] Liu Z, Hoang T, Zhang J, Zhu M, Lan T, Kokane S, Tan J, Yao W, Liu Z, Feng Y, Murthy R, Yang L, Savarese S, Niebles J C, Wang H, Heinecke S, Xiong C. APIGen: automated pipeline for generating verifiable and diverse function-calling datasets. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. 2024, 1725

[138] Guo Z, Cheng S, Wang H, Liang S, Qin Y, Li P, Liu Z, Sun M, Liu Y. StableToolBench: towards stable large-scale benchmarking on tool learning of large language models. In: Proceedings of Findings of the Association for Computational Linguistics. 2024, 11143−11156

[139] Zhong L, Du Z, Zhang X, Hu H, Tang J. ComplexFuncBench: exploring multi-step and constrained function calling under long-context scenario. 2025, arXiv preprint arXiv: 2501.10132

[140] Basu K, Abdelaziz I, Kate K, Agarwal M, Crouse M, Rizk Y, Bradford K, Munawar A, Kumaravel S, Goyal S, Wang X, Lastras L A, Kapanipathi P. NESTFUL: a benchmark for evaluating LLMs on nested sequences of API calls. In: Proceedings of 2025 Conference on Empirical Methods in Natural Language Processing. 2025, 33538−33547

[141] Ye J, Li G, Gao S Y, Huang C, Wu Y, Li S, Fan X, Dou S, Ji T, Zhang Q, Gui T, Huang X. ToolEyes: fine-grained evaluation for tool learning capabilities of large language models in real-world scenarios. In: Proceedings of the 31st International Conference on Computational Linguistics. 2025, 156−187

[142] Team P. Petoolbench: personalized evaluation benchmark for tool-augmented LLMs. 2024

[143] Yao S, Shinn N, Razavi P, Narasimhan K R. τ-bench: a benchmark for tool-agent-user interaction in real-world domains. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[144] Chen D, Chen H, Yang Y, Lin A, Yu Z. Action-based conversations dataset: a corpus for building more in-depth task-oriented dialogue systems. In: Proceedings of 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2021, 3002−3017

[145] Budzianowski P, Wen T H, Tseng B H, Casanueva I, Ultes S, Ramadan O, Gašić M. MultiWOZ − a large-scale multi-domain wizard-of-Oz dataset for task-oriented dialogue modelling. In: Proceedings of 2018 Conference on Empirical Methods in Natural Language Processing. 2018, 5016−5026

[146] Arcadinho S, Aparicio D O, Almeida M S C. Automated test generation to evaluate tool-augmented LLMs as conversational AI agents. In: Proceedings of the 2nd GenBench Workshop on Generalisation (Benchmarking) in NLP. 2024, 54−68

[147] Yao S, Shinn N, Razavi P, Narasimhan K. IntellAgent: a benchmark for evaluating tool-augmented agents in real-world domains, 2024

[148] Wang W, Zhang D, Feng T, Wang B, Tang J. BattleAgentBench: a benchmark for evaluating cooperation and competition capabilities of language models in multi-agent systems. 2024, arXiv preprint arXiv: 2408.15971

[149] Zhu K, Du H, Hong Z, Yang X, Guo S, Wang Z, Wang Z, Qian C, Tang X, Ji H, You J. MultiAgentBench: evaluating the collaboration and competition of LLM agents. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics. 2025, 8580−8622

[150] Sun H, Zhang S, Niu L, Ren L, Xu H, Fu H, Zhao F, Yuan C, Wang X. Collab-overcooked: benchmarking and evaluating large language models as collaborative agents. In: Proceedings of 2025 Conference on Empirical Methods in Natural Language Processing. 2025, 4922−4951

[151] Zhou X, Zhu H, Mathur L, Zhang R, Yu H, Qi Z, Morency L P, Bisk Y, Fried D, Neubig G, Sap M. SOTOPIA: interactive evaluation for social intelligence in language agents. In: Proceedings of the 12th International Conference on Learning Representations. 2024

[152] Zhao R, Zhang W, Chia Y K, Xu W, Zhao D, Bing L. Auto-arena: automating LLM evaluations with agent peer battles and committee discussions. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics. 2025, 4440−4463

[153] Alahi M. Smarteval: Evaluation system for descriptive answers in examinations using iot-enabled technologies and artificial intelligence. Sensors, 2023, 23(11): 5206

[154] Gong R, Huang Q, Ma X, Noda Y, Durante Z, Zheng Z, Terzopoulos D, Li F F, Gao J, Vo H. MindAgent: emergent gaming interaction. In: Proceedings of Findings of the Association for Computational Linguistics. 2024, 3154−3183

[155] Piatti G, Jin Z, Kleiman-Weiner M, Schölkopf B, Sachan M, Mihalcea R. Cooperate or collapse: emergence of sustainable cooperation in a society of LLM agents. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. 2024, 3548

[156] Kočiský T, Schwarz J, Blunsom P, Dyer C, Hermann K M, Melis G, Grefenstette E. The narrativeQA reading comprehension challenge. Transactions of the Association for Computational Linguistics, 2018, 6: 317−328

[157] Zhong M, Yin D, Yu T, Zaidi A, Mutuma M, Jha R, Awadallah A H, Celikyilmaz A, Liu Y, Qiu X, Radev D. QMSum: a new benchmark for query-based multi-domain meeting summarization. In: Proceedings of 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2021, 5905−5921

[158] Pang R Y, Parrish A, Joshi N, Nangia N, Phang J, Chen A, Padmakumar V, Ma J, Thompson J, He H, Bowman S. QuALITY: question answering with long input texts, yes!. In: Proceedings of 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2022, 5336−5358

[159] Kim J, Chay W, Hwang H, Kyung D, Chung H, Cho E, Kwon Y, Jo Y, Choi E. DialSim: a dialogue simulator for evaluating long-term multi-party dialogue understanding of conversational agents. In: Proceedings of the 14th International Conference on Learning Representations. 2026

[160] Maharana A, Lee D H, Tulyakov S, Bansal M, Barbieri F, Fang Y. Evaluating very long-term conversational memory of LLM agents. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics. 2024, 13851−13870

[161] Wu D, Wang H, Yu W, Zhang Y, Chang K W, Yu D. LongMemEval: benchmarking chat assistants on long-term interactive memory. In: Proceedings of Adaptive Foundation Models: Evolving AI for Personalized and Efficient Learning. 2024

[162] Huet A, Ben-Houidi Z, Rossi D. Episodic memories generation and evaluation benchmark for large language models. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[163] Du Y, Wang H, Zhao Z, Liang B, Wang B, Zhong W, Wang Z, Wong K F. PerLTQA: a personal long-term memory dataset for memory classification, retrieval, and fusion in question answering. In: Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing. 2024, 152−164

[164] Wu C K, Tam Z R, Lin C Y, Chen Y N, Lee H Y. StreamBench: towards benchmarking continuous improvement of language agents. In: Proceedings of the 38th International Conference on Neural Information Processing Systems. 2024, 3398

[165] Liu X, Yu H, Zhang H, Xu Y, Lei X, et al. AgentBench: evaluating LLMs as agents. In: Proceedings of the 12th International Conference on Learning Representations. 2024

[166] Mialon G, Fourrier C, Swift C, Wolf T, LeCun Y, Scialom T. GAIA: a benchmark for general AI assistants. 2023, arXiv preprint arXiv: 2311.12983

[167] Yin G, Bai H, Ma S, Nan F, Sun Y, Xu Z, Ma S, Lu J, Kong X, Zhang A, Yap D A, Zhang Y, Ahnert K, Kamath V, Berglund M, Walsh D, Gindele T, Wiest J, Lai Z, Wang X S, Shan J, Cao M, Pang R, Wang Z. MMAU: a holistic benchmark of agent capabilities across diverse domains. In: Proceedings of Findings of the Association for Computational Linguistics. 2025, 4737−4765

[168] Team G A. Galileo AI agent leaderboard, 2023

[169] Stroebl B, Kapoor S, Narayanan A. HAL: a holistic agent leaderboard for centralized and reproducible agent evaluation. See github.com/princeton-pli/hal-harness website, 2025

[170] Yekollu N, Bohra A, Chirumamilla A, Wen K, Kolasani S, Chiang W L, Angelopoulos A, Gonzalez G, Stoica I, Patil S G. Agent arena. 2024

[171] Yang Y, Huang B, Qi S, Feng C, Hu H, Zhu Y, Hu J, Zhao H, He Z, Liu X, Wang Z, Qiu L, Cao X, Cai X, Yu Y, Zhang W. Who's the MVP? A game-theoretic evaluation benchmark for modular attribution in LLM agents. 2025, arXiv preprint arXiv: 2502.00510

[172] Anthropic. Claude AI, 2025. Accessed: 2025-04-27

[173] Fu L, Chai H, Luo S, Du K, Zhang W, Fan L, Lei J, Rui R, Lin J, Fang Y, Liu Y, Wang J, Qi S, Zhang K, Zhang W, Yu Y. CodeApex: a bilingual programming evaluation benchmark for large language models. 2023, arXiv preprint arXiv: 2309.01940

[174] Zhang F, Chen B, Zhang Y, Keung J, Liu J, Zan D, Mao Y, Lou J G, Chen W. RepoCoder: repository-level code completion through iterative retrieval and generation. In: Proceedings of 2023 Conference on Empirical Methods in Natural Language Processing. 2023, 2471−2484

[175] Zhang W, Liao J, Li N, Du K, Lin J. Agentic information retrieval. 2024, arXiv preprint arXiv: 2410.09713

[176] Xi Y, Lin J, Xiao Y, Zhou Z, Shan R, Gao T, Zhu J, Liu W, Yu Y, Zhang W. A survey of LLM-based deep search agents: paradigm, optimization, evaluation, and challenges. 2025, arXiv preprint arXiv: 2508.05668

[177] Chan J S, Chowdhury N, Jaffe O, Aung J, Sherburn D, Mays E,

Starace G, Liu K, Maksin L, Patwardhan T, Madry A, Weng L. MLE-bench: evaluating machine learning agents on machine learning engineering. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[178] Carta T, Romac C, Wolf T, Lamprier S, Sigaud O, Oudeyer P Y. Grounding large language models in interactive environments with online reinforcement learning. In: Proceedings of the 40th International Conference on Machine Learning. 2023, 150

[179] Hafner D. Benchmarking the spectrum of agent capabilities. In: Proceedings of the 10th International Conference on Learning Representations. 2022

[180] Côté M A, Kádár Á, Yuan X, Kybartas B, Barnes T, Fine E, Moore J, Hausknecht M, El Asri L, Adada M, Tay W, Trischler A. TextWorld: a learning environment for text-based games. In: Proceedings of the 7th Workshop on Computer Games. 2018, 41−75

[181] Bradley R A, Terry M E. Rank analysis of incomplete block designs: I. The method of paired comparisons. Biometrika, 1952, 39(3-4): 324−345

[182] Ling W, Yogatama D, Dyer C, Blunsom P. 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. 2017, 158−167

[183] Cobbe K, Kosaraju V, Bavarian M, Chen M, Jun H, Kaiser L, Plappert M, Tworek J, Hilton J, Nakano R, Hesse C, Schulman J. Training verifiers to solve math word problems. 2021, arXiv preprint arXiv: 2110.14168

[184] Hendrycks D, Burns C, Kadavath S, Arora A, Basart S, Tang E, Song D, Steinhardt J. Measuring mathematical problem solving with the MATH dataset. In: Proceedings of the 35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track. 2021

[185] Zhang Y, Yang J, Yuan Y, Yao A C. Cumulative reasoning with large language models. In: Proceedings of the 12th International Conference on Learning Representations. 2024

[186] Patel A, Bhattamishra S, Goyal N. Are NLP models really able to solve simple math word problems?. In: Proceedings of 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2021, 2080−2094

[187] Yang Z, Qi P, Zhang S, Bengio Y, Cohen W, Salakhutdinov R, Manning C D. HotpotQA: a dataset for diverse, explainable multi-hop question answering. In: Proceedings of 2018 Conference on Empirical Methods in Natural Language Processing. 2018, 2369−2380

[188] Geva M, Khashabi D, Segal E, Khot T, Roth D, Berant J. *Did Aristotle use a laptop?* A question answering benchmark with implicit reasoning strategies. Transactions of the Association for Computational Linguistics, 2021, 9: 346−361

[189] Han S, Schoelkopf H, Zhao Y, Qi Z, Riddell M, et al. FOLIO: natural language reasoning with first-order logic. In: Proceedings of 2024 Conference on Empirical Methods in Natural Language Processing. 2024, 22017−22031

[190] Han S, Yu A, Shen R, Qi Z, Riddell M, Zhou W, Qiao Y, Zhao Y, Yavuz S, Liu Y, Joty S, Zhou Y, Xiong C, Radev D, Ying R, Cohan A. P-FOLIO: evaluating and improving logical reasoning with abundant human-written reasoning chains. In: Proceedings of Findings of the Association for Computational Linguistics. 2024, 16553−16565

[191] Bohnet B, Nova A, Parisi A T, Goshvadi K, Swersky K, Dai H, Schuurmans D, Fiedel N, Sedghi H. Exploring and benchmarking planning capabilities of large language models. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[192] Zheng H S, Mishra S, Zhang H, Chen X, Chen M, Nova A, Hou L, Cheng H T, Le Q V, Chi H, Zhou D. NATURAL plan: benchmarking LLMs on natural language planning. 2024, arXiv preprint arXiv: 2406.04520

[193] Wang X, Wang Z, Liu J, Chen Y, Yuan L, Peng H, Ji H. MINT: evaluating LLMs in multi-turn interaction with tools and language feedback. In: Proceedings of the 12th International Conference on Learning Representations. 2024

[194] Wei J, Wang X, Schuurmans D, Bosma M, Ichter B, Xia F, Chi H, Le Q V, Zhou D. Chain-of-thought prompting elicits reasoning in large language models. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 1800

[195] Yao S, Yu D, Zhao J, Shafran I, Griffiths T L, Cao Y, Narasimhan K. Tree of thoughts: deliberate problem solving with large language models. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023, 517

[196] Renze M, Guven E. The benefits of a concise chain of thought on problem-solving in large language models. In: Proceedings of the 2nd International Conference on Foundation and Large Language Models (FLLM). 2024, 476−483

[197] Liu F, AlDahoul N, Eady G, Zaki Y, Rahwan T. Self-reflection makes large language models safer, less biased, and ideologically neutral. 2025, arXiv preprint arXiv: 2406.10400v2

[198] Huang J, Chen X, Mishra S, Zheng H S, Yu A W, Song X, Zhou D. Large language models cannot self-correct reasoning yet. In: Proceedings of the 12th International Conference on Learning Representations. 2024

[199] Zhang W, Shen Y, Wu L, Peng Q, Wang J, Zhuang Y, Lu W. Self-contrast: better reflection through inconsistent solving perspectives. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics. 2024, 3602−3622

[200] Hendrycks D, Basart S, Kadavath S, Mazeika M, Arora A, Guo E, Burns C, Puranik S, He H, Song D, Steinhardt J. Measuring coding challenge competence with APPS. In: Proceedings of the 35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track. 2021

[201] Jain N, Han K, Gu A, Li W D, Yan F, Zhang T, Wang S, Solar-Lezama A, Sen K, Stoica I. LiveCodeBench: holistic and contamination free evaluation of large language models for code. In: Proceedings of the 13th International Conference on Learning Representations. 2025

[202] Chen C, Hao X, Liu W, Huang X, Zeng X, Yu S, Li D, Wang S, Gan W, Huang Y, Liu W, Wang X, Lian D, Yin B, Wang Y, Liu W. ACEBench: who wins the match point in tool usage?. 2025, arXiv preprint arXiv: 2501.12851

[203] Chiang W L, Zheng L, Sheng Y, Angelopoulos A N, Li T, Li D, Zhu B, Zhang H, Jordan M I, Gonzalez J E, Stoica I. Chatbot arena: an open platform for evaluating LLMs by human preference. In: Proceedings of the 41st International Conference on Machine Learning. 2024, 331

[204] Zhuge M, Zhao C, Ashley D R, Wang W, Khizbullin D, Xiong Y,

Liu Z, Chang E, Krishnamoorthi R, Tian Y, Shi Y, Chandra V, Schmidhuber J. Agent-as-a-judge: evaluate agents with agents. In: Proceedings of the 42nd International Conference on Machine Learning. 2025

[205] Xi Y, Lin J, Zhu M, Xiao Y, Ou Z, Liu J, Wan T, Chen B, Liu W, Wang Y, Tang R, Zhang W, Yu Y. InfoDeepSeek: benchmarking agentic information seeking for retrieval-augmented generation. 2025, arXiv preprint arXiv: 2505.15872

[206] Asterisk. DeepClaude: Combining deepseek r1's reasoning with claude's creativity and code generation, 2025. Accessed: 2025-04-27

[207] Chen W, Su Y, Zuo J, Yang C, Yuan C, Chan C M, Yu H, Lu Y, Hung Y H, Qian C, Qin Y, Cong X, Xie R, Liu Z, Sun M, Zhou J. AgentVerse: facilitating multi-agent collaboration and exploring emergent behaviors. In: Proceedings of the 12th International Conference on Learning Representations. 2024

[208] Lin J, Dai X, Xi Y, Liu W, Chen B, Zhang H, Liu Y, Wu C, Li X, Zhu C, Guo H, Yu Y, Tang R, Zhang W. How can recommender systems benefit from large language models: a survey. ACM Transactions on Information Systems, 2025, 43(2): 28

[209] Xi Y, Liu W, Lin J, Chen B, Tang R, Zhang W, Yu Y. MemoCRS: memory-enhanced sequential conversational recommender systems with large language models. In: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management. 2024, 2585−2595

[210] Rui R, Xi Y, Liu W, Lin J, Chen B, Tang R, Zhang W, Yu Y. Action first: leveraging preference-aware actions for more effective decision-making in interactive recommender systems. In: Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2025, 53−63

[211] Zheng C, Zhu J, Ou Z, Chen Y, Zhang K, Shan R, Zheng Z, Yang M, Lin J, Yu Y, Zhang W. A survey of process reward models: from outcome signals to process supervisions for large language models. 2025, arXiv preprint arXiv: 2510.08049

[212] Zhu J, Zheng C, Lin J, Du K, Wen Y, Yu Y, Wang J, Zhang W. Retrieval-augmented process reward model for generalizable mathematical reasoning. In: Proceedings of Findings of the Association for Computational Linguistics. 2025, 8453−8468



Jiachen ZHU is currently a PhD student of the Department of Computer Science, Shanghai Jiao Tong University (SJTU), China under the supervision of Prof. Weinan Zhang and Prof. Yong Yu. He received his Bachelor's degree in computer science also from SJTU in 2023. His research interest covers natural language processing and information retrieval. He has published innovative works in conferences such as ACL, Sigir, and WWW.



Menghui ZHU is currently a researcher with Recommendation and Search Lab, Huawei Noah's Ark Lab, China. He received his BS and MS degrees in computer science and engineering from Shanghai Jiao Tong University, China in 2020 and 2023, respectively. His research interests include recommender system and reinforcement learning. He has published multiple works in conferences, such as WWW, CIKM, and IJCAI.



Renting RUI is currently a master student of the Department of Computer Science, Shanghai Jiao Tong University (SJTU), China under the supervision of Prof. Weinan Zhang and Prof. Yong Yu. She received her Bachelor's degree in computer science from SJTU in 2023. Her research interest covers natural language processing and information retrieval. She has published innovative works in conferences such as SIGIR, WWW, and CIKM.



Rong SHAN is currently a master student of the Department of Computer Science, Shanghai Jiao Tong University (SJTU), China under the supervision of Prof. Weinan Zhang and Prof. Yong Yu. He received his Bachelor's degree in computer science from SJTU in 2024. His research interest covers natural language processing and information retrieval. He has published innovative works in conferences such as KDD and WWW.



Congmin ZHENG is currently a undergraduate student of the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China under the supervision of Prof. Weinan Zhang and Prof. Yong Yu. His research interest covers natural language processing and information retrieval. He has published innovative works in conferences such as ACL.



Bo CHEN is currently a senior researcher at Huawei Noah's Ark Lab since 2020. Before that, he got his M.S. in software engineering from Shanghai Jiao Tong University, China. His research interests include recommender systems, large language models, computational advertising, and AutoML. He was honored with the Best Paper Award at DLP-KDD 2021. He has published several papers in the areas of recommender systems and information retrieval, e.g., KDD, IJCAI, SIGIR, CIKM, WSDM, and RecSys.



Yunjia XI is currently a PhD student of the Department of Computer Science, Shanghai Jiao Tong University (SJTU), China under the supervision of Prof. Weinan Zhang and Prof. Yong Yu. She received her Bachelor's degree in computer science from SJTU in 2021. Her research interest covers large language models and information retrieval. He has published multiple works in conferences, such as SIGIR, KDD, WWW, and ICDE.



Jianghao LIN is currently a tenure-track assistant professor at Shanghai Jiao Tong University, China. He received his PhD in computer science and technology from Shanghai Jiao Tong University, China in 2025, under the supervision of Prof. Weinan Zhang. His research interests focus on large language models, AI agents, and their real-world applications to data science scenarios. He has published over 40

papers in top-tier international conferences and journals. He was honored with the Best Paper Awards at DLP-RecSys 2023 and RelKD-KDD 2024, respectively. He serves as a PC/reviewer for leading conferences and journals like TPAMI, TKDE, TOIS, WWW, KDD, and SIGIR.



Weiwen LIU is currently a tenure-track associate professor at Shanghai Jiao Tong University, China. From September 2020 to April 2025, she was a researcher at Huawei's Noah's Ark Lab. She received her PhD in computer science and engineering from The Chinese University of Hong Kong, China in 2020, under the supervision of Prof. Pheng Ann Heng and Prof. Shengyu Zhang. She was honored with the Best Paper Award at DLP-RecSys 2023. Her research interests include large language models and recommendation algorithms. She has published over 80 papers in top-tier international conferences and journals. She serves as the chair of several workshops, including GenRec-WWW, GenRec-CIKM, and DLP-RecSys, and has been a PC/SPC for leading conferences and journals such as ICLR, WWW, KDD, SIGIR, IJCAI, AAAI, and TKDE.



Ruiming TANG is currently a principal researcher at Huawei Noah's Ark Lab, where he serves as the director of the Recommendation and Search Lab. He leads cutting-edge research and drives the deployment of advanced technologies in recommender and retrieval systems. His research interests include machine learning, recommender systems, deep learning, reinforcement learning, AutoML, and graph neural networks. He received his Bachelor's degree from Northeastern University, China in 2009, and his PhD in computer science from the National University of Singapore, Singapore in 2014. He joined Huawei Noah's Ark Lab at the end of the same year. He has published several papers in top-tier international conferences and journals in the areas of data mining, machine learning, recommender systems, and information retrieval.



Yong YU is currently a professor with the Department of Computer Science, Shanghai JiaoTong University, China. He has published over 200 papers and served as a PC member of several conferences, including WWW, RecSys, and a dozen of other related conferences (e.g..NIPS, ICML, SIGIR, and IWSC) in these fields, His research interests include information systems, web search, data mining, and machine learning.



Weinan ZHANG is now a professor at the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. His research interests include reinforcement learning and data science with various real-world applications of robotic control, game AI, recommender systems, etc. He has published over 200 research papers at prestigious international conferences and journals, accumulating over 20k citations on Google Scholar, and has been selected as Elsevier China Highly Cited Researcher. He has been serving as an area chair at ICML, NeurIPS, ICLR, KDD, etc. and an associate editor at TPAMI.