A Survey on the Feedback Mechanism of LLM-based AI Agents

Zhipeng Liu[1,3], **Xuefeng Bai**[1*], **Kehai Chen**[1], **Xinyang Chen**[1], **Xiucheng Li**[1], **Yang Xiang**[2], **Jin Liu**[3], **Hong-Dong Li**[3], **Yaowei Wang**[1,2], **Liqiang Nie**[1], **Min Zhang**[1]

[1]School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China
[2]Peng Cheng Laboratory, China
[3]School of Computer Science and Engineering, Central South University, China

Abstract

Large language models (LLMs) are increasingly being adopted to develop general-purpose AI agents. However, it remains challenging for these LLM-based AI agents to efficiently learn from feedback and iteratively optimize their strategies. To address this challenge, tremendous efforts have been dedicated to designing diverse feedback mechanisms for LLM-based AI agents. To provide a comprehensive overview of this rapidly evolving field, this paper presents a systematic review of these studies, offering a holistic perspective on the feedback mechanisms in LLM-based AI agents. We begin by discussing the construction of LLM-based AI agents, introducing a generalized framework that encapsulates much of the existing work. Next, we delve into the exploration of feedback mechanisms, categorizing them into four distinct types: internal feedback, external feedback, multi-agent feedback, and human feedback. Additionally, we provide an overview of evaluation protocols and benchmarks specifically tailored for LLM-based AI agents. Finally, we highlight the significant challenges and identify potential directions for future studies. The relevant papers are summarized and will be consistently updated at https://github.com/kevinson7515/Agents-Feedback-Mechanisms.

1 Introduction

An artificial intelligence (AI) agent refers to an autonomous system that performs tasks, makes decisions, or interacts with its environment based on a set of rules, training, or learned knowledge [Wang *et al.*, 2024b]. Traditional feedback systems, such as symbolic [Xi *et al.*, 2025] or reinforcement learning-based agents [Xi *et al.*, 2025; Cheng *et al.*, 2024], are often limited by rigid predefined rules or heuristic policy functions, which markedly diverge from the human learning process, where individuals demonstrate the ability to adapt flexibly across diverse contexts and learn from unstructured feedback. Consequently, the agents developed in these studies often fall short of replicating human-level general-purpose decision-making abilities [Durante *et al.*, 2024].

Recently, large language models (LLMs) [Ouyang *et al.*, 2022; OpenAI, 2023] have shown remarkable success across diverse tasks, exhibiting human-like reasoning and decision-making capabilities [Xi *et al.*, 2025]. Building upon this success, there has been a surge of research interest [Cheng *et al.*, 2024] in employing LLMs as central controllers to construct general-purpose AI agents capable of processing input and executing actions in open-domain scenarios. However, a critical challenge remains: LLM-based AI agents struggle to learn quickly and efficiently from trial-and-error processes in a manner comparable to humans. Consequently, the feedback mechanism has emerged as an increasingly vital component in LLM-based AI agents. This mechanism analyzes and evaluates behavioral data, environmental responses, and historical trajectories to guide agents in refining their strategies and actions. Researchers have developed numerous promising feedback mechanisms, spanning from reinforcement learning [Akyürek *et al.*, 2023], self-refine [Madaan *et al.*, 2024] to multi-model debate [Du *et al.*, 2024a]. However, these models were developed independently, with limited efforts made to summarize and compare them holistically. While several surveys [Pan *et al.*, 2024; Madaan *et al.*, 2024] offer a general perspective of feedback, they focus on LLMs rather than the AI agents which operate within a broader ecosystem of interactions and decision-making processes. The feedback mechanism for LLM-based AI agents is more complex than for standalone LLMs, as it involves not only the refinement of language generation but also the agent's ability to interpret, learn from, and adapt to feedback continuously.

To fill this gap, we shed light on recent advances in feedback mechanisms for LLM-based AI agents. We start by introducing a generalized framework of LLM-based AI agents (§2). We then present a taxonomy of recent studies on feedback mechanisms for LLM-based AI agents (§3), categorized as internal feedback, external feedback, multi-agent feedback, and human feedback. After that, we summarize the evaluation method and benchmarks specifically tailored for assessing feedback mechanisms (§4). Finally, we compare the representative methods of each category on three benchmarks (§5) and analyze the limitations of existing work and highlight important directions for future research (§6). We advocate collaborative efforts within the community to

*corresponding author

pave the way toward the ultimate goal of Artificial General Intelligence (AGI). The contributions of this work can be summarized as: (1) we present a structured taxonomy that categorizes existing works into four categories; (2) we provide the first comprehensive survey of recent advancements in feedback mechanisms for LLM-based AI agents, encompassing methodologies, evaluation protocols, and benchmarks; (3) we discuss the remaining limitations of current research and point out potential directions for future work.

2 Preliminary: A Unified Framework for LLM-based AI agents

Recent advances in LLM-based AI agents have tended to be modularized to improve the agent's efficiency, adaptability, and scalability. Inspired by previous studies [Cheng *et al.*, 2024; Durante *et al.*, 2024], we propose a unified framework to summarize these modules. As shown in Figure 1, an LLM-based AI agents is composed of five modules: *perception, planning, feedback, memory,* and *action.* The core functions and roles of each module are described below:

Perception: The perception module enables AI agents to analyze and understand environmental inputs, identifying key patterns to support task execution.

Planning: The planning module helps AI agents break down complex tasks into simpler sub-tasks, generating plans and guiding actions to ensure logical and reliable behavior.

Memory: The memory module stores past experiences to support decision-making, with long-term memory preserving stable knowledge and short-term memory handling real-time task information, thus enabling AI agents to accumulate expertise and transfer knowledge between different scenarios.

Action: The action module translates decisions into operational outputs, interacting with the environment through tools, APIs, and embodied actions, bringing the agent's capabilities to real-world applications.

Feedback: The feedback module serves as a critical component in equipping AI agents with self-reflection and self-optimization capabilities. Leveraging the power of LLMs, this module facilitates the agent's ability to critically evaluate its prior decisions and actions, enabling dynamic adjustments and optimizations that enhance both intelligence and adaptability. This mechanism is particularly crucial for achieving continuous evolution in complex environments, ensuring more reliable and efficient task execution.

Moreover, the feedback module is tightly coupled with the memory and planning modules, collectively optimizing the agent's decision-making and execution. Taking Voyager [Wang *et al.*, 2023] iron-smelting task as an example: Check furnace inventory → Discover no furnace → Retrieve skills → Refer to chest crafting skill to craft a furnace → Place the furnace → Check raw iron inventory → Smelt 5 raw iron. When detecting the absence of a furnace in the inventory, the feedback module critiques this failure state, triggering a skill retrieval mechanism, thereby guiding the agent to learn furnace crafting.

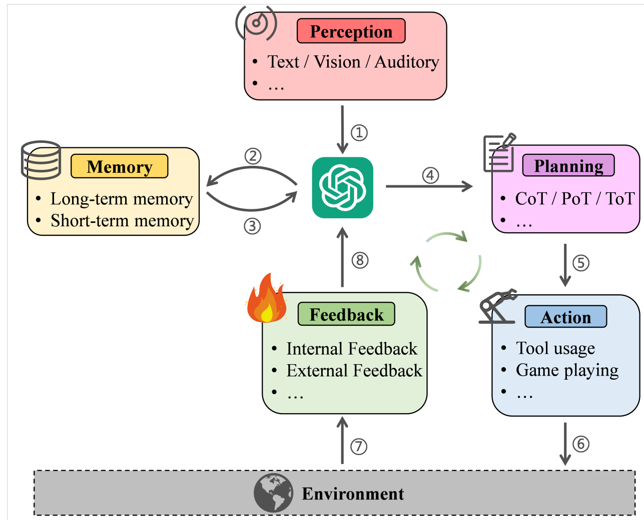

Figure 1: A unified framework for LLM-based AI agents.

3 Taxonomy of Feedback Mechanisms

As shown in Figure 2, we present a structured taxonomy that organizes existing research into four core categories: (1) *Internal Feedback*, where AI agents generate feedback internally to refine their strategies and actions; (2) *External Feedback*, where AI agents leverage external models or tools to enhance feedback quality; (3) *Multi-agent Feedback*, where multiple agents interact and provide feedback to one another; and (4) *Human Feedback*, where humans act as the primary source of feedback to guide agent behavior.

We illustrate the basic flow of each category in Figure 3 and summarize the representative works, feedback format, learning strategy, iteration (or not), and application tasks associated with each category in Table 1.

3.1 Internal Feedback

As shown in Figure 3(a), Internal feedback originates from the agent itself and serves to motivate proactive exploration, learning, and self-improvement. This type of feedback does not rely on explicit external goals; instead, it encourages the agent to discover latent patterns, generalize strategies, and develop adaptive behaviors [Liu *et al.*, 2025]. Internal feedback often manifests as signals generated and utilized by the model itself [Pan *et al.*, 2024]. For example, in internal feedback mechanisms based on Chain of Thought (CoT) reasoning, the model is prompted to generate answers with explanations. By identifying and reinforcing reasoning chains that lead to correct answers, the model's behavior can be fine-tuned, improving performance even without explicit external supervision. This process can be iterative, continuously enhancing the model's reasoning and generalization capabilities. Depending on whether cross-task knowledge is used, internal feedback can be categorized into *intra-task* and *inter-task feedback*.

Intra-task feedback refers to the feedback derived from an agent's historical steps during its trial-and-error interactions with the environment, serving as the most relevant and informative signals to guide the agent's future actions. For

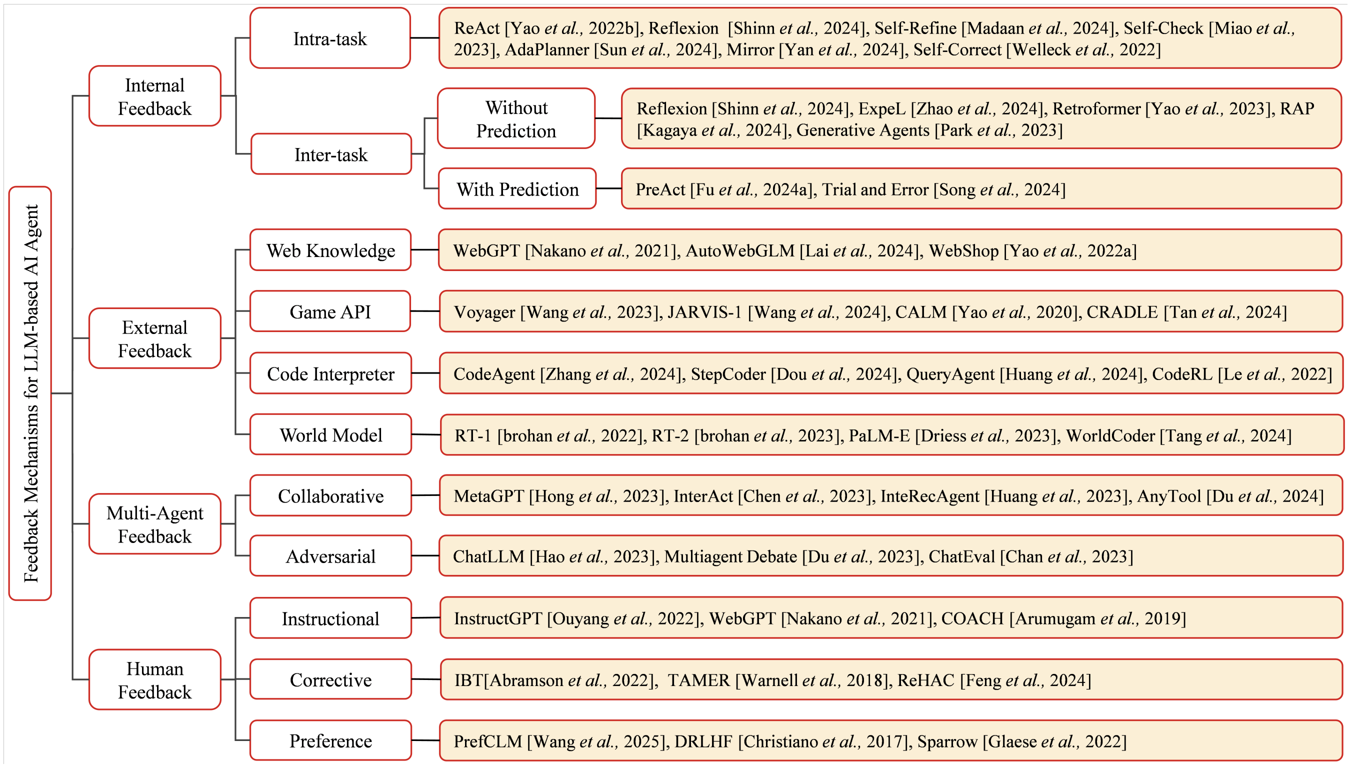

Figure 2: Taxonomy of feedback mechanisms for LLM-based AI agents.

example, ReAct [Yao *et al.*, 2023] uses LLMs to generate reasoning trajectories and task-specific actions in an interleaved manner, improving the ability to solve complex tasks. Self-Refine [Madaan *et al.*, 2024] improves the initial output of LLMs by iterative feedback and refinement during the same task, leading to higher quality and more accurate results. ***Inter-task feedback*** involves the transfer of knowledge and experience across tasks, enabling the agent to apply lessons learned from previous tasks to adapt to new or similar tasks. Reflexion [Shinn *et al.*, 2024] derives experiences from past tasks and applies them in subsequent tasks to improve performance in new scenarios. ExpeL [Zhao *et al.*, 2024] retrieves and analyzes similar past trajectories, identifying success patterns by contrasting positive and negative examples.

Discussion: Intra-task feedback enables rapid optimization for the current task, but its task-specific nature limits broader knowledge learning and accumulation. Conversely, inter-task feedback facilitates cross-task learning and long-term experience integration, while it demands substantial memory to store historical task experience. The fundamental challenge lies in effectively combining these complementary approaches to harness their respective strengths.

3.2 External Feedback

External feedback, as illustrated in Figure 3(b), comprises environment-defined signals that evaluate and guide an agent's behavior toward goal achievement. For example, in reinforcement learning, scores, task completion indicators, and success rates are all forms of external feedback. In AI agents, external feedback is usually provided by independent feedback models or external tools (such as compilers, search engines, or world models). Based on the external modules involved, external feedback can be categorized into *web knowledge*, *game API*, *code interpreter*, and *world model*.

Web knowledge is widely used to enhance AI agents' feedback capabilities. WebGPT [Nakano *et al.*, 2021] enables web-based question answering by searching and browsing online content, then refining answers through user feedback. ***Game API*** serves as a key resource for improving agent functionality and adaptability. Voyager [Wang *et al.*, 2023] introduces an iterative hinting mechanism that combines execution errors and self-validation to enhance program quality. Some approaches employ ***code interpreters*** to boost feedback effectiveness. StepCoder [Dou *et al.*, 2024] learns from compiler feedback, improving its ability to understand complex requirements and generate better code. ***World model*** facilitates feedback loops by allowing AI agents to incorporate and adapt to real-world sensory and environmental data. PaLM-E [Driess *et al.*, 2023] integrates real-world sensor feedback and visual feedback into a language model to establish a link between words and perceptions.

Discussion: External feedback offers LLM-based AI agents additional insights that they might not obtain through self-improvement alone. However, the effectiveness of such feedback is highly dependent on the quality and relevance of external information. Therefore, optimizing the acquisition and processing mechanism of external feedback is crucial to boosting the performance of LLM-based AI agents.

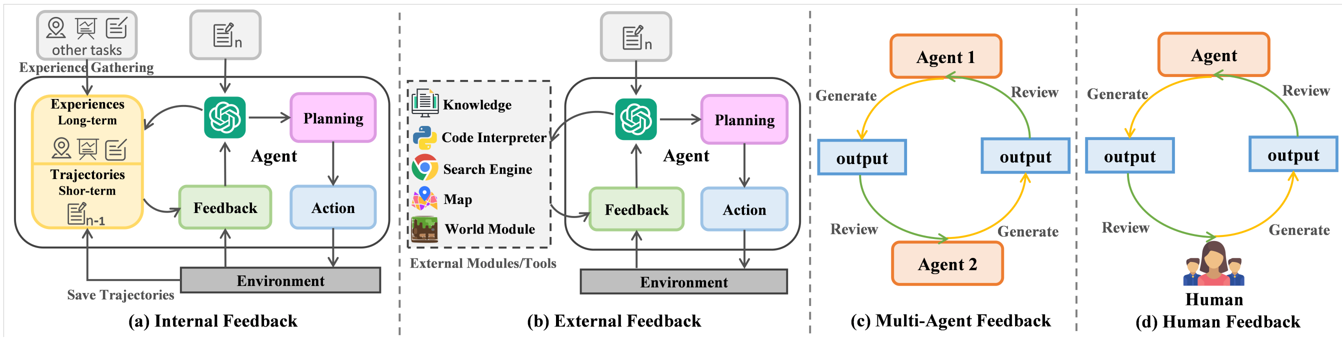

Figure 3: Illustrations of different feedback mechanisms for LLM-based AI agents.

3.3 Multi-Agent Feedback

Multi-agent feedback, shown in Figure 3(c), mimics multi-agent interactions where diverse perspectives converge to improve solutions. This process promotes knowledge sharing and stronger decision-making in either *collaborative* or *adversarial* ways.

Collaborative approaches encourage divergent thinking, where each agent presents and discusses its own points of view in multiple rounds, with the optimal solution as the final answer. MetaGPT [Hong *et al.*, 2024] employs an executable feedback loop for iterative code refinement, with agents collaboratively enhancing code quality using execution history and debug logs. InteRecAgent [Huang *et al.*, 2023] adopts a dual-role feedback mechanism, pairing a recommender agent with a reviewer agent to evaluate results and identify execution errors. ***Adversarial*** approaches aim to improve the quality of the output by employing multiple agents to debate the same viewpoints in multiple rounds until a consensus is reached. Du *et al.* [2024a] proposed a multi-agent debate framework, featuring multiple agents expressing their arguments in a "tit-for-tat" state, and a judge manages the debate process toward final resolutions.

Discussion: In a multi-agent system, agents take on distinct roles and responsibilities, coordinating their actions to achieve shared goals, which reduces the burden on individual agents and enhances task performance. However, challenges such as coordination, reward alignment, and learning stability persist. For coordination, when three or more agents freely express their views, discussions may become uncontrolled; Solutions like introducing a dedicated coordination agent (e.g., ChatLLM [Hao *et al.*, 2023]) can integrate responses to refine the final answer. Regarding reward alignment, misaligned goals and reward functions among agents may lead to conflicts or suboptimal outcomes. CollaQ [Zhang *et al.*, 2020] addresses this by decomposing reward allocation to provide an innovative solution for decentralized Q-functions. For learning stability, multi-agent negotiations may converge on incorrect consensuses, PRD [Li *et al.*, 2023] peer-ranking algorithm analyzes pairwise preferences to generate a final ranking, ensuring more accurate consensus answers.

3.4 Human Feedback

As shown in Figure 3(d), human feedback enables agents to iteratively refine their behavior according to human feedbacks while performing the task. Human feedback manifests in three primary forms: *instructional feedback*, *corrective feedback*, and *preference-based feedback*.

Instructional feedback provides direct task guidance through explicit human instructions. For instance, WebGPT [Nakano *et al.*, 2021] uses imitation learning to train models by setting tasks in a way that humans can perform them and then optimizes the quality of answers through user feedback. ***Corrective feedback*** is provided when the agent's behavior deviates or contains errors, and humans intervene to correct these mistakes or guide improvements. Abramson *et al.* [2022] developed a 3D simulation framework capturing human corrections when agents stray from goals. ReHAC [Feng *et al.*, 2024b] includes a policy model designed to determine the most opportune stages for human intervention within the task-solving process. ***Preference-based feedback*** shapes agent behavior through human preferences or choices. PrefCLM [Wang *et al.*, 2025] introduces a human-in-the-loop pipeline that facilitates collective refinements based on user comparative feedback to train robots.

Discussion: Human feedback is closely tied to human values, guiding an agent's behavior and learning through direct human input. However, relying solely on human feedback is often impractical due to limited resources or real-time constraints, as it often requires continuous monitoring, evaluation, and adjustment by experts to ensure the agent aligns with desired outcomes. Several solutions have been proposed, such as LLM-as-a-judge [Zheng *et al.*, 2023], which leverages the scalability and interpretability of LLMs to reduce the need for human intervention, providing scores and explanations, but it suffers from positional bias, verbosity bias, and limitations in evaluating mathematical problems. Agent-as-a-judge [Zhuge *et al.*, 2024] integrates agent capabilities to provide intermediate feedback during task resolution, achieving better consistency with human evaluators.

Ethical concerns about human feedback are growing. Specifically, human feedback mechanisms have the potential to introduce biases into the training data, particularly along lines of gender, race, and culture, which in turn can amplify the inherent biases within LLMs. To mitigate these risks, future research should prioritize the development of more diverse learning methodologies, such as the integration of culturally representative datasets and the enhancement of cultural awareness in model design.

Method	Subcategory	Representative Works	Format	Learning	Iteration	Domain
Internal Feedback	Intra-task	ReAct [Yao *et al.*, 2023]	NL	ICL	✗	QA, Fact Verification
		Reflexion [Shinn *et al.*, 2024]	NL	ICL	✓	QA, Code Generation
		Self-Refine [Madaan *et al.*, 2024]	NL	ICL	✓	Multiple Tasks
		Self-Check [Miao *et al.*, 2024]	NL	ICL	✓	Arithmetic Reasoning
		AdaPlanner [Sun *et al.*, 2023]	Code	ICL	✓	Embodied Action, Web Browsing
		Mirror [Yan *et al.*, 2024]	NL	ICL	✓	Reasoning, Fact Verification
		Self-Correct [Welleck *et al.*, 2023]	NL / Scalar	SL	✓	Reasoning, Generation, Toxicity
	Inter-task	Reflexion [Shinn *et al.*, 2024]	NL	ICL	✓	QA, Code Generation
		ExpeL [Zhao *et al.*, 2024]	NL	ICL	✓	Multiple Tasks
		Retroformer [Yao *et al.*, 2024]	NL	SL	✓	QA, Embodied Action, Web Browsing
		RAP [Kagaya *et al.*, 2024]	Scalar	ICL	✓	Planning, Reasoning
		Generative Agents [Park *et al.*, 2023]	NL	ICL	✗	Social Simulation
		PreAct [Fu *et al.*, 2025]	NL	ICL	✓	Multiple Tasks
		Trial and Error [Song *et al.*, 2024]	Scalar	IL & RL	✓	Web Navigation, Embodied Action
External Feedback	Web Knowledge	WebGPT [Nakano *et al.*, 2021]	NL	IL & RL	✓	QA
		AutoWebGLM [Lai *et al.*, 2024]	NL	IL & RL	✓	Web Browsing
		WebShop [Yao *et al.*, 2022]	Scalar	IL & RL	✓	Web Shopping
	Game API	Voyager [Wang *et al.*, 2023]	NL	ICL	✗	Build Tools
		JARVIS-1 [Wang *et al.*, 2024c]	NL	ICL	✗	Build Tools
		CALM [Yao *et al.*, 2020]	NL / Scalar	RL	✓	Text Games
		CRADLE [Tan *et al.*, 2024]	NL	ICL	✓	Computer Control, Game Playing
	Code Interpreter	CodeAgent [Zhang *et al.*, 2024]	NL	ICL	✗	Code Generation
		StepCoder [Dou *et al.*, 2024]	Scalar	RL	✓	Code Generation
		QueryAgent [Huang *et al.*, 2024]	NL / Code	ICL	✓	QA
		CodeRL [Le *et al.*, 2022]	Scalar	RL	✗	Code Generation
	World Model	RT-1 [Brohan *et al.*, 2022]	Scalar	IL & SL	✓	Embodied Action
		RT-2 [Zitkovich *et al.*, 2023]	Scalar	IL & SL	✓	Embodied Action
		PaLM-E [Driess *et al.*, 2023]	Scalar	IL & SL	✓	Embodied Action
		WorldCoder [Tang *et al.*, 2024]	Code	RL	✓	Embodied Action
Multi-Agent Feedback	Collaborative	MetaGPT [Hong *et al.*, 2024]	NL / Code	ICL	✗	Program
		InterAct [Chen and Chang, 2023]	NL	ICL & RL	✗	Embodied Action
		InteRecAgent [Huang *et al.*, 2023]	NL	RL	✓	Recommendation
		AnyTool [Du *et al.*, 2024b]	Scalar	RL	✗	Tool Usage
	Adversarial	ChatLLM [Hao *et al.*, 2023]	NL	ICL	✓	Classification, Sentiment Reversal
		Multiagent Debate [Du *et al.*, 2024a]	NL	ICL	✓	Reasoning, Factuality
		ChatEval [Chan *et al.*, 2023]	NL	ICL	✓	Text Evaluation
Human Feedback	Instructional	InstructGPT [Ouyang *et al.*, 2022]	NL	SL & RL	✗	Multiple Tasks
		WebGPT [Nakano *et al.*, 2021]	NL	IL & RL	✓	QA
		COACH [Arumugam *et al.*, 2019]	NL	RL	✗	Build Tools
	Corrective	IBT [Abramson *et al.*, 2022]	NL	IL & RL	✓	Scripted Probe
		TAMER [Warnell *et al.*, 2018]	Scalar	RL	✗	Embodied Action
		ReHAC [Feng *et al.*, 2024b]	NL	RL	✓	QA, Reasoning, Code Generation
	Preference	PrefCLM [Wang *et al.*, 2025]	Scalar	RL	✗	Control Suite, Embodied Action
		DRLHF [Christiano *et al.*, 2017]	Scalar	RL	✗	Embodied Action
		Sparrow [Glaese *et al.*, 2022]	NL	SL & RL	✓	Dialogue

Table 1: An overview of existing feedback mechanisms for LLM-based AI agents. NL refers to natural language, ICL refers to in-context learning, SL refers to supervised learning, RL refers to reinforcement learning, and IL refers to imitation learning.

4 Evaluation and Benchmark

4.1 Evaluation

Although LLM-based AI agents have demonstrated remarkable capabilities in downstream tasks, the question of how to effectively evaluate feedback mechanisms remains unresolved. Current studies can be broadly categorized into *Outcome-based evaluation* and *Process-based evaluation*.

Outcome-based evaluation: Presently, most evaluations are Outcome-based evaluation, relying on the assessment of feedback mechanisms through end-to-end agent task performance. These methods typically measure the effectiveness of feedback mechanisms based on task success rates. Olausson *et al.* [2023] analyze LLMs' self-repair capabilities on the APPS coding benchmark, where the feedback mechanism is evaluated by the pass rate of programming tasks. TravelPlanner [Xie *et al.*, 2024] assess feedback mechanisms by assigning agents constrained travel planning tasks and measuring tool usage error rates and planning failure rates. While providing valuable insights, these indirect evaluations often oversimplify the assessment process by focusing on aggregate outcomes rather than granular performance details.

Process-based evaluation: Traditional agent evaluation methods tend to focus solely on final outcomes, overlooking critical details during execution or relying heavily on manual assessment. Recently, DevAI [Zhuge *et al.*, 2024] enables agents to evaluate each other, assessing both the final outcomes and intermediate execution steps for richer feedback. AMOR [Guan *et al.*, 2024] solves problems through autonomous executions and transitions over disentangled modules, allowing to provide feedback to the individual modules, and thus naturally forms process supervision.

Discussion: Gaps remain in developing robust quantitative indicators for self-correction across diverse domains. Factors like learning stage, task complexity, and environmental diversity can significantly impact feedback effectiveness, highlighting the need for a more comprehensive evaluation system. Therefore, it is necessary to develop a more comprehensive feedback evaluation framework that integrates both Outcome-based and Process-based metrics.

Domain	Benchmark	Size	Major Modalities	Evaluation Method	Task
Reasoning	HotPotQA [Yang *et al.*, 2018]	113k Q&A pairs	text	Outcome-based	fact extraction and verification
	ScienceQA [Lu *et al.*, 2022]	21k Q&A pairs	text, image	Outcome-based	science QA
	StrategyQA [Geva *et al.*, 2021]	2.7k Q&A pairs	text	Outcome-based	multi-step QA
	FEVER [Thorne *et al.*, 2018]	185k claims	text	Outcome-based	fact extraction and verification
Virtual World	ALFWorld [Shridhar *et al.*, 2020]	3.1k examples	text, image	Outcome-based	low-level actuation
	IGLU [Mehta *et al.*, 2024]	8k data	text, 3D	Outcome-based	create structure
	Minecraft [Wang *et al.*, 2023]	58 items	3D	Outcome-based	build tools
Embodied Action	Franka-Kitchen [Gupta *et al.*, 2020]	400 demonstrations	3D	Outcome-based	household action
	Meta-World [Yu *et al.*, 2020]	50 tasks	3D	Outcome-based	manipulation action
	RT-X [Padalkar *et al.*, 2023]	1m trajectories	3D	Outcome-based	household action
Web Navigation	WebShop [Yao *et al.*, 2022]	1.8k examples, 12k instructions	text, image	Outcome-based	web shopping
	WebArena [Zhou *et al.*, 2023]	812 tasks	text, image	Outcome-based	web browsing
	Mind2Web [Deng *et al.*, 2024]	2k tasks	text, image	Outcome-based	web browsing
	WebVoyager [He *et al.*, 2024]	300 tasks	text, image	Outcome-based	web browsing
Code Generation	DevAI [Zhuge *et al.*, 2024]	55 tasks	text, iamge	Process-based	code generation
	MBPP [Austin *et al.*, 2021]	1k problems	text	Outcome-based	code generation
	HumanEval [Chen *et al.*, 2021a]	164 problems	text	Outcome-based	code generation
	SWE-Bench [Jimenez *et al.*, 2024]	2k problems	text, image	Outcome-based	code generation
Social Simulation	SocialBench [Chen *et al.*, 2024]	6k questions, 30k utterances	text	Outcome-based	role playing, social simulation
	SocKET [Choi *et al.*, 2023]	58 tasks	text	Outcome-based	social factors, trustworthiness
Tool Usage	ToolBench [Qin *et al.*, 2024]	16k API, 3k tools	text, API	Outcome-based	tool implementation
	TravelPlanner [Xie *et al.*, 2024]	1225 plans, 4m data records	text, API	Outcome-based	travel planning
	ToolEyes [Ye *et al.*, 2025]	600 tools	text, API	Outcome-based	tool learning
Multi-agent Collaboration	RocoBench [Mandi *et al.*, 2024]	6 tasks	3D	Outcome-based	path planning
	PARTNR [Chang *et al.*, 2024]	100k tasks, 5k objects	3D	Outcome-based	house-hold collaboration
	VillagerBench [Dong *et al.*, 2024]	3 tasks	3D	Outcome-based	construction cooporation
Machine Translation	WMT [Specia *et al.*, 2021]	14 languages pairs	text	Outcome-based	general machine translation
	FLORES-200 [Guzmán *et al.*, 2019]	842 articles, 3k sentences	text	Outcome-based	article machine translation
Financial	FiQA_SA [Cheng *et al.*, 2023]	1k documents	text, image	Outcome-based	financial document classification
	FinQA [Chen *et al.*, 2021b]	2.8k reports, 8k Q&A pairs	text	Outcome-based	financial numerical reasoning
Multidimensional evolution	AgentBench [Liu *et al.*, 2024]	1.4k samples	text, image	Outcome-based	multiple tasks

Table 2: Common benchmarks for LLM-based AI agents.

In the future, standard evaluation methods in the field of code generation may use datasets like DevAI for intermediate process optimization and MBPP [Austin *et al.*, 2021] for overall performance testing.

4.2 Benchmarks

As presented in Table 2, we summarize and categorize mainstream benchmarking approaches for evaluating LLM-based AI agents, with emphasis on their feedback mechanisms.

Reasoning: Reasoning tasks aim to test an agent's ability to handle complex reasoning by answering questions. Common benchmarks include HotPotQA [Yang *et al.*, 2018], ScienceQA [Lu *et al.*, 2022], FEVER [Thorne *et al.*, 2018], and StrategyQA [Geva *et al.*, 2021]. ScienceQA involves multiple-choice questions, but recent studies suggest agents may exhibit option biases (e.g., favoring option C) rather than truly understanding the problem. Moreover, this Q&A format overlooks the reasoning process, where an incorrect process might still yield a correct result.

Virtual World: Virtual world tasks evaluate an agent's decision-making and execution abilities in dynamic, complex environments. Common benchmarks include ALFWorld [Shridhar *et al.*, 2020], IGLU [Mehta *et al.*, 2024], and Minecraft [Wang *et al.*, 2023]. ALFWorld's grid-world simplicity may fail to capture real-world ambiguity. Minecraft, a widely adopted open-world simulation, offers diverse tasks and vast potential for evaluating simulated environments. However, this platform encounters difficulties concerning the transferability of skills from simulation to real-world application, i.e., the "sim-to-real" transfer problem.

Embodied Action: Embodied tasks test an agent's integrated abilities in perception, decision-making, and action execution, typically simulating real-world physical actions like moving a cup, pouring water, or cleaning an environment. Common benchmarks include Franka-Kitchen [Gupta *et al.*, 2020], Meta-World [Yu *et al.*, 2020], and RT-X [Padalkar *et al.*, 2023], with task success rate as the evaluation metric. Franka-Kitchen focuses on kitchen environments, limiting generalization to other physical scenarios. RT-X evaluates diverse physical settings.

Web Navigation: Web navigation tasks simulate human processes of solving problems through querying and filtering information. Common benchmarks include WebShop [Yao *et al.*, 2022], WebArena [Zhou *et al.*, 2023], Mind2Web [Deng *et al.*, 2024], and WebVoyager [He *et al.*, 2024]. Mind2Web interacts only with static website states, while WebArena creates realistic, dynamic, and reproducible web environments through simulated websites. WebVoyager goes further, using Selenium to directly interact with real web pages.

Code Generation: Code generation tasks test an agent's logical reasoning, planning, and interaction with compilers in programming-related problems. Typical benchmarks include MBPP [Austin *et al.*, 2021], HumanEval [Chen *et al.*, 2021a], and SWE-Bench [Jimenez *et al.*, 2024]. However, these rely heavily on success rates, which fail to provide specific feedback on each stage or capture the dynamic performance of agent systems. DevAI [Zhuge *et al.*, 2024] not only focuses on final outcomes but also tracks and evaluates each

Method	Success Rate			Feedback Round	Computational Cost	Stability	Multi-agent task transferability	Real-time
	HotpotQA	ALFWorld	WebShop					
Act	29%	28%	34%	Single-round	Low	Low stability	Transferable	✓
CoT	29%	/	/	Multi-round	Medium	Low stability	Transferable	✓
ReAct	28%	40%	35%	Multi-round	Medium	Low stability	Transferable	✓
Reflexion-R1	33%	48%	43%	One-round with experiment replay				
Reflexion-R2	40%	52%	46%	Two-round with experiment replay	High	Low stability in short-term, High stability in long-term	Require shared experience pool	✗
Reflexion-R3	40%	54%	48%	Three-round with experiment replay				
ExpeL	39%	59%	41%	Multi-round with experiment replay	High	Medium stability	Require shared experience pool	✗
AdaPlanner	/	63%	/	Multi-round	Medium	High stability	Transferable	✓
AutoGuide	/	79%	46%	Multi-round with experiment replay	High	High stability	Require shared experience pool	✗

Table 3: Comparison of different feedback mechanisms.

stage of task execution for more comprehensive feedback.

Social Simulation: Social simulation tasks involve agents playing roles and interacting with other virtual characters in simulated social environments. Common benchmarks include SocialBench [Chen *et al.*, 2024] and SocKET [Choi *et al.*, 2023]. However, these benchmarks often feature overly idealized social scenarios, lacking the complexity and unpredictability of real-world social interactions. Park *et al.* [2023]'s AI Town is a notable attempt to address this.

Tool Usage: Tool use tasks test an agent's ability to leverage external tools to achieve complex task goals. Common benchmarks include ToolBench [Qin *et al.*, 2024], TravelPlanner [Xie *et al.*, 2024], and ToolEyes [Ye *et al.*, 2025]. These tasks may overly focus on specific tools, limiting flexibility in tool selection and combination. HuggingGPT [Shen *et al.*, 2023] uses a broader range of tools. Beyond evaluating tool use, future assessments might explore an agent's ability to create tools tailored to specific tasks.

Multi-agent Collaboration: Multi-agent collaboration tasks examine the ability of multiple agents to work together in shared environments. Common benchmarks include RocoBench [Mandi *et al.*, 2024], PARTNR [Chang *et al.*, 2024], and VillagerBench [Dong *et al.*, 2024]. VillagerBench relies on Minecraft environments, while RocoBench assumes perfect perception (e.g., object detection, pose estimation, collision checking), which may fail in real-world scenarios like industrial production or medical collaboration.

Mathine Translation: Machine translation tasks assess an agent's ability to accurately and contextually translate text across languages, preserving meaning, tone, and cultural nuances. Common benchmarks include WMT [Specia *et al.*, 2021] and FLORES-200 [Guzmán *et al.*, 2019]. FLORES-200 covers multiple languages but focuses on document translation, falling short in general translation. WMT is widely used for general translation evaluation.

Financial: Financial tasks typically require understanding numerical data, interpreting market signals, and generating actionable insights. Common benchmarks include FiQA_SA [Cheng *et al.*, 2023] and FinQA [Chen *et al.*, 2021b], with evaluation based on prediction accuracy or financial calculation correctness. However, these tasks limit the ability to test agents in dynamic market scenarios like stock trading. Increasingly, financial agents are tested in real-world stock trading scenarios.

Multidimensional evolution: Multidimensional evolution evaluates an agent's general capabilities across multiple domains, including reasoning, decision-making, and execution. For example, AgentBench [Liu *et al.*, 2024] includes diverse scenarios covering operating systems, databases, knowledge graphs, card games, lateral thinking puzzles, household tasks, online shopping, and web browsing, assessing an agent's comprehensive abilities across eight distinct task scenarios.

5 Experimental Comparison

This section selects several representative feedback mechanisms for comparative experiments on HotpotQA [Yang *et al.*, 2018], ALFWorld [Shridhar *et al.*, 2020], and WebShop [Yao *et al.*, 2022] (some experimental results are borrowed from ExpeL [Zhao *et al.*, 2024] and AutoGuide [Fu *et al.*, 2024]), as shown in Table 3.

In terms of reasoning capability, ReAct [Yao *et al.*, 2023] demonstrates limited performance improvement on complex reasoning tasks such as HotpotQA and StrategyQA, whereas Reflexion [Shinn *et al.*, 2024] exhibits more significant enhancements on HotPotQA. For sequential decision-making tasks (e.g., ALFWorld and WebShop), ReAct shows outstanding performance, while Reflexion further improves ALFWorld's performance through its multi-step environmental navigation capability, with the most notable improvements observed between the first and second trials. ExpeL [Zhao *et al.*, 2024], AdaPlanner [Sun *et al.*, 2023], and AutoGuide [Fu *et al.*, 2024] also demonstrate applicability in such tasks.

From the stability perspective, non-learning-based methods like CoT, Act, and ReAct are suitable for immediate single-run tasks but face accuracy bottlenecks. In contrast, Reflexion, ExpeL, and AutoGuide exhibit greater stability in long-term dynamic tasks through iterative learning mechanisms. AdaPlanner mitigates hallucination issues by employing code-style prompts, demonstrating stability performance.

Compared to traditional reinforcement learning, feedback mechanisms like Reflexion offer significant advantages over traditional reinforcement learning (RL) across four dimensions: (1) *Trial-and-Error learning;* it improves performance by reflecting on errors and incorporating experience into subsequent decisions, suitable for tasks requiring trial-and-error learning, such as decision-making, reasoning, and programming; (2) *Efficiency;* it does not require fine-tuning language models, offering higher data and computational efficiency; (3) *Detailed feedback;* it uses linguistic feedback, which is more specific and detailed than the scalar rewards of traditional RL, helping agents understand errors and make targeted improvements; (4) *Interpretability and memory;* It provides explicit, interpretable staged memory, storing self-reflections to facilitate analysis of the learning process, outperforming traditional black-box RL.

6 Challenges and Future Directions

6.1 Real-Time Feedback and Multi-Agent System Implementation

Challenge: As the number of agents grows, computational demands rise sharply, increasing the need for efficient architecture and optimization. More complex communication networks slow information flow, make coordination harder, and lower collaboration efficiency, hindering the achievement of shared goals. Additionally, a lack of standardized protocols also complicates interactions, especially when agents come from different vendors and use different architectures.

Future Direction: Developing efficient and scalable frameworks, such as agent systems based on the MARL framework [Ma *et al.*, 2024], to reduce communication costs and computational complexity while enhancing adaptability to heterogeneous agents and systems. Additionally, promoting protocol standardization, such as MCP, A2A, ANP, and Agora, for building large-scale agent systems.

6.2 Integration of Multi-modal Feedback

Challenge: The core challenge lies in the complexity of aligning different modalities. Effective multimodal self-correction requires agents to process and combine diverse data—such as text, images, audio, and sensor inputs—where unique representations for each modality make integration difficult. Current research faces difficulties in establishing unified fusion or representation learning strategies to convert these diverse signals into cohesive features.

Future Direction: Developing unified frameworks for multimodal feedback. For instance, Tactical Rewind [Ke *et al.*, 2019] leverages self-correction strategies to optimize visual and language navigation. MM-React [Yang *et al.*, 2023] achieves multimodal reasoning and action by integrating ChatGPT with a pool of visual experts. A promising research avenue involves cross-modal representation learning to find shared feature space across modalities.

6.3 Meta-Learning Adaptive Feedback

Challenge: The primary challenge is creating a system that effectively balances learning from multiple feedback sources while maintaining stability and consistency in agent behavior. For instance, AutoGen [Wu *et al.*, 2023] highlights that agents must continuously adapt to evolving user needs and contexts, which can result in feedback conflicts or overfitting.

Future Direction: Future research should focus on developing more efficient and robust adaptive mechanisms to enhance agent performance in diverse applications. One potential direction is mutual learning, such as the two-stage mutual learning framework [Wang *et al.*, 2024a]. Another direction involves learning unified latent representations to integrate multi-source feedback and minimize conflicts.

6.4 Explainability of Feedback Mechanisms

Challenge: Current feedback mechanisms often lack transparency in how they make and adjust decisions, which undermines user trust—especially in critical areas like healthcare and transportation.

Future Direction: Developing explainable feedback models with visualization tools and transparency metrics to clarify how feedback affects agent behavior. For example, natural language explanation tools can generate easy-to-understand feedback [Feng *et al.*, 2024a], while AMOR [Guan *et al.*, 2024] improves explainability and safety by providing step-by-step feedback similar to a chain of thought.

6.5 Feedback in Embedded AI Agent

Challenge: In embedded AI, deploying models trained in simulated environments to the real world faces significant "sim-to-real" issues. Embodied agents trained with reinforcement learning might struggle to fully replicate real-world disturbances, lighting, gravity, and other physical properties, leading to poor model performance in reality. Approaches like domain randomization [Saito *et al.*, 2022], domain adaptation [Rao *et al.*, 2020], and simulation improvement [Martinez-Gonzalez *et al.*, 2020] are commonly solutions to address these gaps.

Future Direction: Future developments include designing adaptive learning algorithms that dynamically adjust in real-world settings, rapidly adapting to unseen physical properties through online learning or meta-learning; exploring hybrid training paradigms that combine simulated and real-world data to optimize generalization; developing high-fidelity simulators that integrate vision, touch, and mechanics to achieve more robust transfer to real environments.

6.6 Inherent Limitations of LLM Feedback

Challenge: LLMs and VLMs often produce hallucinations due to biases and spurious features in training data, such as incorrectly associating objects with visual cues or generating factually inaccurate text. Moreover, models may generate biased or incorrect outputs due to limitations in pre-trained knowledge or insufficient understanding of the dynamics of the deployment environment. Such issues are especially noticeable in AI agents with minimal fine-tuning, leading to unreliable feedback and reduced trust in the agent system.

Future Direction: Addressing hallucinations and biases might require the integration of retrieval-augmented generation (RAG) to cross-validate outputs with external sources, or integrating external databases to improve factual accuracy. In multi-agent systems, iterative interaction and debate among agents can help correct individual hallucinations and reasoning errors [Xi *et al.*, 2025]. Additionally, incorporating world models (environmental models) can also support systematic fact-checking of generated content.

7 Conclusion

In this survey, we provide a systematic review of feedback mechanisms in LLM-based agents. First, we introduce the overall framework of LLM-based agents to help readers understand their fundamental components and operational models. Next, we summarize the current research status of feedback mechanisms, evaluation protocols, and benchmarks in LLM-based agents, offering clear classifications and intuitive insights. Finally, we conduct an in-depth analysis of the main limitations of existing feedback mechanisms and highlight several research directions worth further exploration. Through this comprehensive review, we aspire to provide a comprehensive reference for readers interested in this rapidly evolving field and to inspire future research.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (62406091, 62276077, 62306085, U23B2055, 62350710797), Guangdong Basic and Applied Basic Research Foundation (2024A1515011205), Shenzhen Science and Technology Program (KQTD20240729102154066, ZDSYS20230626091203008), and Shenzhen College Stability Support Plan (GXWD20231130104007001, GXWD20220817123150002, GXWD20231130151329002, GXWD20220811170358002).

References

[Abramson *et al.*, 2022] Josh Abramson, Arun Ahuja, et al. Improving multimodal interactive agents with reinforcement learning from human feedback. *Arxiv*, 2022.

[Akyürek *et al.*, 2023] Afra Feyza Akyürek, Ekin Akyürek, et al. RL4F: Generating natural language feedback with reinforcement learning for repairing model outputs. In *ACL*, 2023.

[Arumugam *et al.*, 2019] Dilip Arumugam, Jun Ki Lee, et al. Deep reinforcement learning from policy-dependent human feedback. *Arxiv*, 2019.

[Austin *et al.*, 2021] Jacob Austin, Augustus Odena, et al. Program synthesis with large language models. *Arxiv*, 2021.

[Brohan *et al.*, 2022] Anthony Brohan, Noah Brown, et al. Rt-1: Robotics transformer for real-world control at scale. *Arxiv*, 2022.

[Chan *et al.*, 2023] Chi-Min Chan, Weize Chen, et al. Chateval: Towards better llm-based evaluators through multi-agent debate. In *ICLR*, 2023.

[Chang *et al.*, 2024] Matthew Chang, Gunjan Chhablani, et al. Partnr: A benchmark for planning and reasoning in embodied multi-agent tasks. *arXiv*, 2024.

[Chen and Chang, 2023] Po-Lin Chen and Cheng-Shang Chang. Interact: Exploring the potentials of chatgpt as a cooperative agent. *Arxiv*, 2023.

[Chen *et al.*, 2021a] Mark Chen, Jerry Tworek, et al. Evaluating large language models trained on code. *Arxiv*, 2021.

[Chen *et al.*, 2021b] Zhiyu Chen, Wenhu Chen, et al. Finqa: A dataset of numerical reasoning over financial data. *arXiv*, 2021.

[Chen *et al.*, 2024] Hongzhan Chen, Hehong Chen, et al. SocialBench: Sociality evaluation of role-playing conversational agents. In *ACL*, 2024.

[Cheng *et al.*, 2023] D Cheng, S Huang, et al. Adapting large language models via reading comprehension. arxiv 2023. *arXiv*, 2023.

[Cheng *et al.*, 2024] Yuheng Cheng, Ceyao Zhang, et al. Exploring large language model based intelligent agents: Definitions, methods, and prospects. *Arxiv*, 2024.

[Choi *et al.*, 2023] Minje Choi, Jiaxin Pei, et al. Do LLMs understand social knowledge? evaluating the sociability of large language models with SocKET benchmark. In *EMNLP*, 2023.

[Christiano *et al.*, 2017] Paul F Christiano, Jan Leike, et al. Deep reinforcement learning from human preferences. In *NeurIPS*, 2017.

[Deng *et al.*, 2024] Xiang Deng, Yu Gu, et al. Mind2web: Towards a generalist agent for the web. In *NeurIPS*, 2024.

[Dong *et al.*, 2024] Yubo Dong, Xukun Zhu, et al. Villagerbench: Benchmarking multi-agent collaboration in minecraft. *ACL*, 2024.

[Dou *et al.*, 2024] Shihan Dou, Yan Liu, et al. StepCoder: Improving code generation with reinforcement learning from compiler feedback. In *ACL*, 2024.

[Driess *et al.*, 2023] Danny Driess, Fei Xia, et al. Palm-e: An embodied multimodal language model. In *ICML*, 2023.

[Du *et al.*, 2024a] Yilun Du, Shuang Li, et al. Improving factuality and reasoning in language models through multiagent debate. In *ICML*, 2024.

[Du *et al.*, 2024b] Yu Du, Fangyun Wei, et al. AnyTool: Self-reflective, hierarchical agents for large-scale API calls. In *ICML*, 2024.

[Durante *et al.*, 2024] Zane Durante, Qiuyuan Huang, et al. Agent ai: Surveying the horizons of multimodal interaction. *Arxiv*, 2024.

[Feng *et al.*, 2024a] Xidong Feng, Ziyu Wan, et al. Natural language reinforcement learning. *arXiv*, 2024.

[Feng *et al.*, 2024b] Xueyang Feng, Zhi-Yuan Chen, et al. Large language model-based human-agent collaboration for complex task solving. In *ACL*, 2024.

[Fu *et al.*, 2024] Yao Fu, Dong-Ki Kim, et al. Autoguide: Automated generation and selection of context-aware guidelines for large language model agents. *arXiv*, 2024.

[Fu *et al.*, 2025] Dayuan Fu, Jianzhao Huang, et al. PreAct: Prediction enhances agent‘s planning ability. In *ICCL*, 2025.

[Geva *et al.*, 2021] Mor Geva, Daniel Khashabi, et al. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. In *ACL*, 2021.

[Glaese *et al.*, 2022] Amelia Glaese, Nat McAleese, et al. Improving alignment of dialogue agents via targeted human judgements. *Arxiv*, 2022.

[Guan *et al.*, 2024] Jian Guan, Wei Wu, et al. Amor: A recipe for building adaptable modular knowledge agents through process feedback. *NeurIPS*, 2024.

[Gupta *et al.*, 2020] Abhishek Gupta, Vikash Kumar, et al. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In *CoRL*, 2020.

[Guzmán *et al.*, 2019] Francisco Guzmán, Peng-Jen Chen, et al. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. In *EMNLP-IJCNLP*, 2019.

[Hao *et al.*, 2023] Rui Hao, Linmei Hu, et al. Chatllm network: More brains, more intelligence. *Arxiv*, 2023.

[He *et al.*, 2024] Hongliang He, Wenlin Yao, et al. Webvoyager: Building an end-to-end web agent with large multimodal models. *ArXiv*, 2024.

[Hong *et al.*, 2024] Sirui Hong, Xiawu Zheng, et al. MetaGPT: Meta programming for a multi-agent collaborative framework. In *ICLR*, 2024.

[Huang *et al.*, 2023] Xu Huang, Jianxun Lian, et al. Recommender ai agent: Integrating large language models for interactive recommendations. *Arxiv*, 2023.

[Huang *et al.*, 2024] Xiang Huang, Sitao Cheng, et al. QueryAgent: A reliable and efficient reasoning framework with environmental feedback based self-correction. In *ACL*, 2024.

[Jimenez *et al.*, 2024] Carlos E Jimenez, John Yang, et al. SWE-bench: Can language models resolve real-world github issues? In *ICLR*, 2024.

[Kagaya *et al.*, 2024] Tomoyuki Kagaya, Thong Jing Yuan, et al. RAP: Retrieval-augmented planning with contextual memory for multimodal LLM agents. In *NeurIPS*, 2024.

[Ke *et al.*, 2019] Liyiming Ke, Xiujun Li, et al. Tactical rewind: Self-correction via backtracking in vision-and-language navigation. In *CVPR*, 2019.

[Lai *et al.*, 2024] Hanyu Lai, Xiao Liu, et al. Autowebglm: Bootstrap and reinforce a large language model-based web navigating agent. *Arxiv*, 2024.

[Le *et al.*, 2022] Hung Le, Yue Wang, et al. CodeRL: Mastering code generation through pretrained models and deep reinforcement learning. In *NeurIPS*, 2022.

[Li *et al.*, 2023] Ruosen Li, Teerth Patel, et al. Prd: Peer rank and discussion improve large language model based evaluations. *arXiv*, 2023.

[Liu *et al.*, 2024] Xiao Liu, Hao Yu, et al. Agentbench: Evaluating LLMs as agents. In *ICLR*, 2024.

[Liu *et al.*, 2025] Bang Liu, Xinfeng Li, et al. Advances and challenges in foundation agents: From brain-inspired intelligence to evolutionary, collaborative, and safe systems. *arXiv*, 2025.

[Lu *et al.*, 2022] Pan Lu, Swaroop Mishra, et al. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *NeurIPS*, 2022.

[Ma *et al.*, 2024] Chengdong Ma, Aming Li, et al. Efficient and scalable reinforcement learning for large-scale network control. *Nature*, 2024.

[Madaan *et al.*, 2024] Aman Madaan, Niket Tandon, et al. Self-refine: Iterative refinement with self-feedback. In *NeurIPS*, 2024.

[Mandi *et al.*, 2024] Zhao Mandi, Shreeya Jain, et al. Roco: Dialectic multi-robot collaboration with large language models. In *ICRA*, 2024.

[Martinez-Gonzalez *et al.*, 2020] Pablo Martinez-Gonzalez, Sergiu Oprea, et al. Unrealrox: an extremely photorealistic virtual reality environment for robotics simulations and synthetic data generation. *Virtual Reality*, 2020.

[Mehta *et al.*, 2024] Nikhil Mehta, Milagro Teruel, et al. Improving grounded language understanding in a collaborative environment by interacting with agents through help feedback. In *ACL*, 2024.

[Miao *et al.*, 2024] Ning Miao, Yee Whye Teh, et al. Self-check: Using LLMs to zero-shot check their own step-by-step reasoning. In *ICLR*, 2024.

[Nakano *et al.*, 2021] Reiichiro Nakano, Jacob Hilton, et al. Webgpt: Browser-assisted question-answering with human feedback. *Arxiv*, 2021.

[Olausson *et al.*, 2023] Theo X Olausson, Jeevana Priya Inala, et al. Demystifying gpt self-repair for code generation. *Arxiv*, 2023.

[OpenAI, 2023] OpenAI. GPT-4 technical report. *Arxiv*, 2023.

[Ouyang *et al.*, 2022] Long Ouyang, Jeffrey Wu, et al. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.

[Padalkar *et al.*, 2023] Abhishek Padalkar, Acorn Pooley, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *ArXiv*, 2023.

[Pan *et al.*, 2024] Liangming Pan, Michael Saxon, et al. Automatically correcting large language models: Surveying the landscape of diverse automated correction strategies. *TACL*, 2024.

[Park *et al.*, 2023] Joon Sung Park, Joseph O'Brien, et al. Generative agents: Interactive simulacra of human behavior. In *UIST*, 2023.

[Qin *et al.*, 2024] Yujia Qin, Shengding Hu, et al. Tool learning with foundation models. *ACM Comput. Surv*, 2024.

[Rao *et al.*, 2020] Kanishka Rao, Chris Harris, et al. Rl-cyclegan: Reinforcement learning aware simulation-to-real. In *CVPR*, 2020.

[Saito *et al.*, 2022] Daichi Saito, Kazuhiro Sasabuchi, et al. Task-grasping from a demonstrated human strategy. In *Humanoids*, 2022.

[Shen *et al.*, 2023] Yongliang Shen, Kaitao Song, et al. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *NeurIPS*, 2023.

[Shinn *et al.*, 2024] Noah Shinn, Federico Cassano, et al. Reflexion: Language agents with verbal reinforcement learning. In *NeurIPS*, 2024.

[Shridhar *et al.*, 2020] Mohit Shridhar, Xingdi Yuan, et al. Alfworld: Aligning text and embodied environments for interactive learning. In *ICLR*, 2020.

[Song *et al.*, 2024] Yifan Song, Da Yin, et al. Trial and error: Exploration-based trajectory optimization of LLM agents. In *ACL*, 2024.

[Specia *et al.*, 2021] Lucia Specia, Frédéric Blain, et al. Findings of the wmt 2021 shared task on quality estimation. In *ACL*, 2021.

[Sun *et al.*, 2023] Haotian Sun, Yuchen Zhuang, et al. Adaplanner: Adaptive planning from feedback with language models. In *NeurIPS*, 2023.

[Tan *et al.*, 2024] Weihao Tan, Wentao Zhang, et al. Cradle: Empowering foundation agents towards general computer control. In *NeurIPS*, 2024.

[Tang *et al.*, 2024] Hao Tang, Darren Key, et al. Worldcoder, a model-based llm agent: Building world models by writing code and interacting with the environment. *ArXiv*, 2024.

[Thorne *et al.*, 2018] James Thorne, Andreas Vlachos, et al. Fever: a large-scale dataset for fact extraction and verification. In *NAACL*, 2018.

[Wang *et al.*, 2023] Guanzhi Wang, Yuqi Xie, et al. Voyager: An open-ended embodied agent with large language models. In *NeurIPS*, 2023.

[Wang *et al.*, 2024a] Dingmin Wang, Qiuyuan Huang, et al. Retrieve what you need: A mutual learning framework for open-domain question answering. *ACL*, 2024.

[Wang *et al.*, 2024b] Lei Wang, Chen Ma, et al. A survey on large language model based autonomous agents. *Front. Comput. Sci*, 2024.

[Wang *et al.*, 2024c] Zihao Wang, Shaofei Cai, et al. Jarvis-1: Open-world multi-task agents with memory-augmented multimodal language models. *TPAMI*, 2024.

[Wang *et al.*, 2025] Ruiqi Wang, Dezhong Zhao, et al. Prefclm: Enhancing preference-based reinforcement learning with crowdsourced large language models. *IEEE Rob. Autom. Lett.*, 2025.

[Warnell *et al.*, 2018] Garrett Warnell, Nicholas Waytowich, et al. Deep tamer: Interactive agent shaping in high-dimensional state spaces. In *AAAI*, 2018.

[Welleck *et al.*, 2023] Sean Welleck, Ximing Lu, et al. Generating sequences by learning to self-correct. *ICLR*, 2023.

[Wu *et al.*, 2023] Qingyun Wu, Gagan Bansal, et al. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv*, 2023.

[Xi *et al.*, 2025] Zhiheng Xi, Wenxiang Chen, et al. The rise and potential of large language model based agents: A survey. *Sci. China Inf. Sci*, 2025.

[Xie *et al.*, 2024] Jian Xie, Kai Zhang, et al. TravelPlanner: A benchmark for real-world planning with language agents. In *ICML*, 2024.

[Yan *et al.*, 2024] Hanqi Yan, Qinglin Zhu, et al. Mirror: Multiple-perspective self-reflection method for knowledge-rich reasoning. In *ACL*, 2024.

[Yang *et al.*, 2018] Zhilin Yang, Peng Qi, et al. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*, 2018.

[Yang *et al.*, 2023] Zhengyuan Yang, Linjie Li, et al. Mmreact: Prompting chatgpt for multimodal reasoning and action. *arXiv*, 2023.

[Yao *et al.*, 2020] Shunyu Yao, Rohan Rao, et al. Keep CALM and explore: Language models for action generation in text-based games. In *EMNLP*, 2020.

[Yao *et al.*, 2022] Shunyu Yao, Howard Chen, et al. Webshop: Towards scalable real-world web interaction with grounded language agents. In *NeurIPS*, 2022.

[Yao *et al.*, 2023] Shunyu Yao, Jeffrey Zhao, et al. React: Synergizing reasoning and acting in language models. In *ICLR*, 2023.

[Yao *et al.*, 2024] Weiran Yao, Shelby Heinecke, et al. Retroformer: Retrospective large language agents with policy gradient optimization. In *ICLR*, 2024.

[Ye *et al.*, 2025] Junjie Ye, Guanyu Li, et al. ToolEyes: Fine-grained evaluation for tool learning capabilities of large language models in real-world scenarios. In *ICCL*, 2025.

[Yu *et al.*, 2020] Tianhe Yu, Deirdre Quillen, et al. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *CoRL*, 2020.

[Zhang *et al.*, 2020] Tianjun Zhang, Huazhe Xu, et al. Multi-agent collaboration via reward attribution decomposition. *arXiv*, 2020.

[Zhang *et al.*, 2024] Kechi Zhang, Jia Li, et al. CodeAgent: Enhancing code generation with tool-integrated agent systems for real-world repo-level coding challenges. In *ACL*, 2024.

[Zhao *et al.*, 2024] Andrew Zhao, Daniel Huang, et al. Expel: Llm agents are experiential learners. In *AAAI*, 2024.

[Zheng *et al.*, 2023] Lianmin Zheng, Wei-Lin Chiang, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *NeurIPS*, 2023.

[Zhou *et al.*, 2023] Shuyan Zhou, Frank F. Xu, et al. Webarena: A realistic web environment for building autonomous agents. *ArXiv*, 2023.

[Zhuge *et al.*, 2024] Mingchen Zhuge, Changsheng Zhao, et al. Agent-as-a-judge: Evaluate agents with agents. *Arxiv*, 2024.

[Zitkovich *et al.*, 2023] Brianna Zitkovich, Tianhe Yu, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *ACRL*, 2023.