Scaling Environments for LLM Agents in the Era of Learning from Interaction: A Survey

Yuchen Huang[♡] **Sijia Li**[♡] **Minghao Liu**[♡] **Wei Liu**[♠] **Zhiyuan Fan**[♡]
Yi R. (May) Fung[♡]
[♡]Hong Kong University of Science and Technology
[♠]King's College London
{yhuanggn, yrfung}@cse.ust.hk

Abstract

LLM-based agents can autonomously accomplish complex tasks across various domains. However, to further cultivate capabilities such as adaptive behavior and long-term decision-making, training on static datasets built from human-level knowledge is insufficient. These datasets are costly to construct and lack both dynamism and realism. A growing consensus is that agents should instead interact directly with environments and learn from experience through reinforcement learning. We formalize this iterative process as the Generation-Execution-Feedback (GEF) loop, where environments generate tasks to challenge agents, return observations in response to agents' actions during task execution, and provide evaluative feedback on rollouts for subsequent learning. Under this paradigm, environments function as indispensable producers of experiential data, highlighting the need to scale them toward greater complexity, realism, and interactivity. In this survey, we first systematically review representative methods for environment scaling from a pioneering environment-centric perspective and organize them along the stages of the GEF loop. We further analyze benchmarks, implementation frameworks, and applications, consolidating fragmented advances and outlining future research directions for agent intelligence. [1]

1 Introduction

The rapid progress of large language models (LLMs) has catalyzed a transformative shift in artificial intelligence, precipitating a surge of research on LLM-based agents [Luo et al., 2025a, Xi et al., 2025]. Such agents inherit strong reasoning and task-decomposition capabilities from their base models and, when augmented with modules for tool use and memory, can execute actions, interact with real or simulated environments, accumulate experience over time, and progressively improve their own behavior. This design has achieved remarkable progress across diverse domains, including automated coding [Qwen Team, 2025, Anthropic, 2025], interactive web navigation [OpenAI, 2025a, He et al., 2025], tool use [Zhang et al., 2025a, Anthropic, 2024], and deep research [Tongyi DeepResearch Team, 2025, OpenAI, 2025b, Google DeepMind, 2024].

However, as agent capabilities continue to evolve, it is infeasible to attain intelligence beyond the human-level merely by supervised fine-tuning (SFT) pretrained models on static datasets [Huang et al., 2025a, Su et al., 2025a, Zhao et al., 2025]. Such datasets are typically manually annotated or curated under human oversight, which makes them costly and labor-intensive to produce at scale,

[1]We provide a GitHub repository with real-time updates on this topic: https://github.com/lukahhcm/Awesome_Scaling_Environments.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Scaling Environments for Agents (SEA).

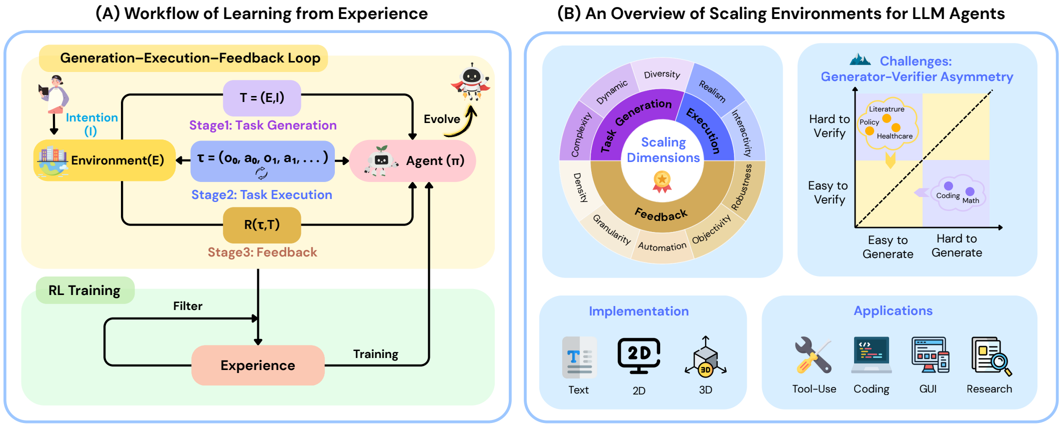

Figure 1: (A) Experience arises from the Generation-Execution-Feedback (GEF) loop, where environments generate tasks, agents execute them, and environments evaluate and filter useful experience for RL training. (B) Overview of environment scaling: a GEF-aligned taxonomy of environment-scaling methods, alongside implementation, applications, and the unique challenge of Generator-Verifier asymmetry.

intrinsically bounded by human-level knowledge, and lacking realism and adaptability. By contrast, reinforcement learning provides a more aligned training paradigm [Tao et al., 2024, Zhang et al., 2025b], where agents can explore in the environment, accumulate experiences, and finally acquire new knowledge or skills. We formalize this interactive process as the Generation-Execution-Feedback (GEF) loop, illustrated in Figure 1 (A). In each iteration, the environment first generates diverse tasks, then the agent executes them within the environment, producing action-observation trajectories. The environment subsequently evaluates these rollouts and retains useful experience for subsequent training. Repeated iterations progressively refine the policy and expand the agent's capabilities. Notably, unlike prior work [Gao et al., 2025], we adopt a broad view of the environment: everything external to the current agent, including the state space, the executable action space, the design of feedback for interaction and evaluation, and the activities of users and other agents, is considered part of it. In this setting, the environment is no longer a mere container for agents' activities; it has become an active producer of experiential data, underscoring the growing need for scaling environments to create a more complex, realistic, and richly interactive world [CAMEL-AI, 2025].

Recent research has embraced this trend of scaling the environment from different perspectives. For instance, systems like AgentGen [Hu et al., 2025a], AgentGym [Xi et al., 2024], and GEM [Liu et al., 2025a] devise heterogeneous environments to increase the diversity of the generated tasks. R-Zero [Huang et al., 2025a] proposes a challenger-solver framework that autonomously generates increasingly difficult tasks. RandomWorld [Sullivan et al., 2025] scales up the interactivity by procedural generation of diverse tools for agents to access. ARE [Andrews et al., 2025] develops an event-driven environment that supports asynchronous interactions between the environment and agents, scaling up the environmental dynamics that conform to realistic settings. However, a systematic analysis that connects these research directions remains absent.

Therefore, we comprehensively investigate current environment scaling methods and propose a unified taxonomy aligned with the stages of the GEF loop, adopting a pioneering environment-centric perspective. In the **task generation** stage, we categorize scaling methods into *complexity scaling*, *dynamic scaling*, and *diversity scaling*, which together characterize an environment's ability to generate challenging, adaptive, and diverse tasks continuously. In the **task execution** stage, we highlight interactivity and realism, since these properties determine the richness and fidelity of the interaction data from which agents learn. In the **feedback** stage, we categorize the scaling of evaluative signals along *density*, *granularity*, *automation*, *objectivity*, and *robustness*. Beyond this taxonomy, we also analyze current evaluation benchmarks, implementation frameworks, applications, and future research directions. Figure 1 (B) shows a high-level overview of environment scaling, and representative works are listed in Figure 2.

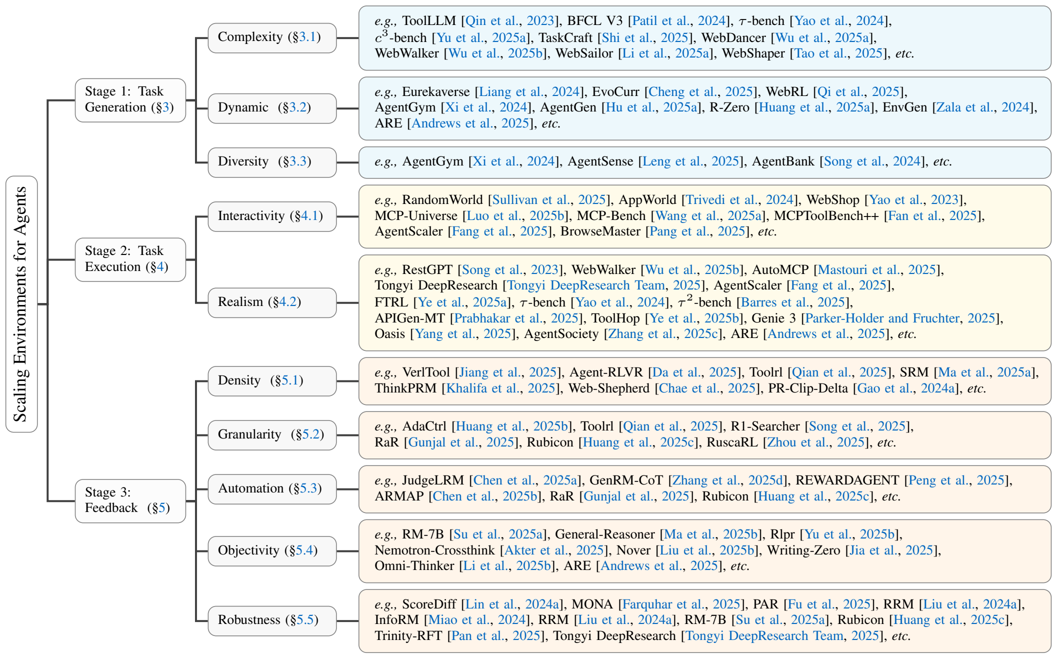

Figure 2: GEF-aligned taxonomy of environment scaling with dimensions for *Task Generation*, *Task Execution*, and *Feedback*. Representative works are illustrated as leaves on the branches.

The survey is organized as follows. We first introduce the background and conceptual framework in §2 and §A. We then categorize representative environment scaling methods along the three-stage taxonomy: task generation (§3), task execution (§4), and feedback (§5). Next, we discuss evaluation benchmarks in §B, implementation frameworks in §6, and applications in §C. Finally, we outline future research directions (§7).

2 Background

Scaling Laws for LLM Agents Just as large language models exhibit predictable performance scaling with increases in the number of parameters, the volume of training data, and the compute budget, agent systems likewise display scaling regularities along three axes: (i) expanding the agent population and identifying properties that emerge as interactions increase; (ii) increasing environmental complexity and assessing how realistic, dynamic settings shape learning and adaptation; and (iii) extending the horizons of evolution and memory to study how agents generalize and improve through accumulated experience [CAMEL-AI, 2025]. While most existing surveys on LLM agents adopt an agent-centric view [Luo et al., 2025a, Xi et al., 2025, Yehudai et al., 2025, Gao et al., 2025], covering topics from multi-agent interaction [Qian et al., 2024, Tran et al., 2025] to self-evolution [Gao et al., 2025, Tao et al., 2024], environment scaling remains underexplored and has not been systematically organized. In this work, we take an environment-centric perspective on scaling environments and examine how dynamic, richly interactive, high-fidelity worlds can accelerate agent development and evolution.

Generator-Verifier Asymmetry Challenge A fundamental characteristic in many real-world tasks is the inherent *Generator-Verifier Asymmetry* [Wei, 2025], namely the mismatch between the intelligence required for generator, which generates (§3) or executes (§4) tasks, and that required for verifier, which provides feedback (§5). These two kinds of intelligence naturally form two axes critical to next-generation Agentic AI, as illustrated in Figure 1 (B). From this perspective, scaling up environments essentially corresponds to scaling intelligence along the x-axis and the y-axis. Current progress in RL largely exploits the regime on the easy-to-verify side of this asymmetry. These **Easy-to-Verify, Hard-to-Generate Domains** include fields such as mathematics and programming [Wei et al., 2025, Jimenez et al., 2023, Phan et al., 2025]. For these domains, generating and solving a continual stream of high-quality, non-trivial tasks is challenging. In contrast, verification is objective

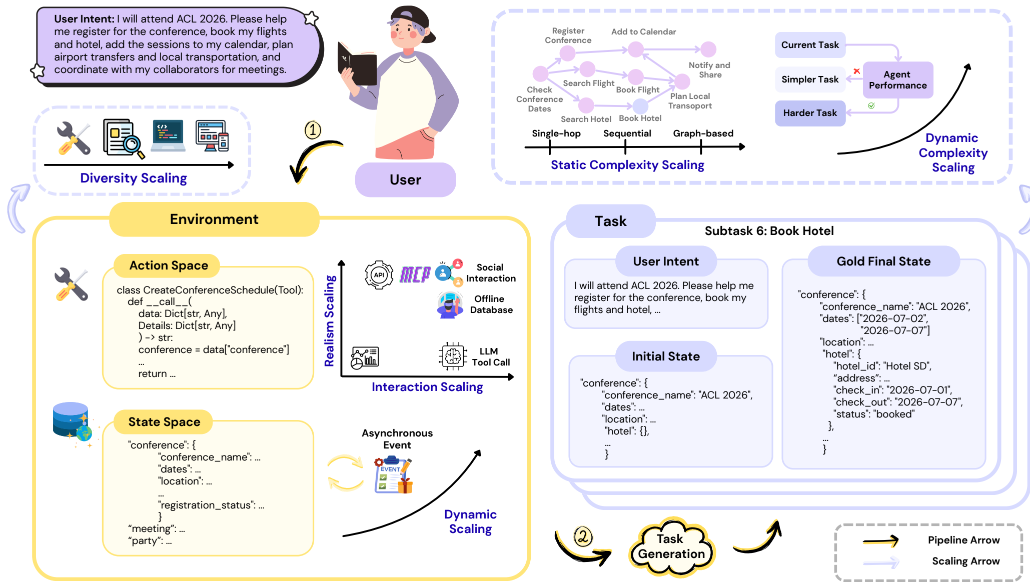

Figure 3: Illustration of environment scaling in the **task generation** and **task execution** stages, using the example of conference scheduling. Given a user intent, the environment produces a set of tasks for the agent to complete. Scaling in the task generation stage covers *complexity scaling*, *dynamic scaling*, and *diversity scaling*, while in the task execution stage scaling encompasses *interactivity scaling* and *realism scaling*.

and computationally inexpensive (e.g., via unit tests or exact match on mathematical results). This enables weak-to-strong supervision, where a simple verifier can provide accurate feedback to train a much stronger agent for solving hard tasks. On the contrary, the **Hard-to-Verify, Easy-to-Generate Domains** include areas such as creative writing, policy-making, or healthcare [Lin et al., 2024b, Arora et al., 2025]. For these easy-to-propose, open-ended tasks, verification is subjective, requires substantial expert judgment, or unfolds over long horizons, making high-quality feedback scarce and expensive. This bottleneck, corresponding to the upper-left region of coordinate system, poses more difficulty in modeling the environment, and rendering environment scaling more challenging yet offering greater potential for advancing agent capabilities. Notably, the asymmetry also presents an opportunity: if the generator's stronger intelligence can be systematically leveraged to strengthen the verifier, so that it can supervise an even stronger generator, then such asymmetric property can be exploited to drive agents' self-evolution [Huang et al., 2025a, Hong et al., 2025, Lu et al., 2025a, Chen et al., 2025c, Wang et al., 2025b].

3 Stage 1: Task Generation

In the task generation stage, the environment is required to propose challenging tasks that push the agent toward its capability boundary. Scaling at this stage targets three aspects of the task design: increasing difficulty (*complexity scaling* § 3.1), introducing dynamics (*dynamic scaling* § 3.2), and expanding diversity (*diversity scaling* § 3.3). An illustrative example is shown in Figure 3. For clarity, in *complexity scaling*, we only consider the intrinsic difficulty of a task (i.e., static complexity). We group the temporal evolution of task difficulty (dynamic complexity) together with changes in the environment itself under the *dynamic scaling* subsection.

3.1 Complexity Scaling

Static complexity increases a task's inherent structural intricacy, moving beyond single-step commands to challenges defined by dependencies, logical flows, and hierarchical relationships. A typical example is LLM tool use, from early single-step tasks to multi-turn, multi-step scenarios, where complexity scales up as the number of turns and steps increases [Qin et al., 2023, Patil et al., 2024, Yao et al., 2024, Yu et al., 2025a]. More sophisticated tasks exhibit hierarchical or compositional structure, decomposing high-level objectives into nested sub-goals and thereby testing compositional generalization, namely an agent's ability to solve novel problems by recombining known skills [Shao et al., 2023]. TaskCraft [Shi et al., 2025] operationalizes this by expanding tasks both in depth (longer sequences of tool executions) and in width (multiple sub-goals per objective), enhancing hierarchical reasoning. At the highest level, conditional and graph-based tasks involve non-linear structures with branching logic, where planning must adapt dynamically to intermediate outcomes. Recent efforts in information-seeking agents extend linear sequences to complex graph-based information chains [Wu et al., 2025a, Tao et al., 2025, Li et al., 2025a, Wu et al., 2025b], and multi-agent settings further amplify this complexity, as optimal plans become contingent on the actions of other agents, producing intrinsically interdependent, graph-structured challenges.

3.2 Dynamic Scaling

Task Difficulty Dynamics Scaling the task's difficulty dynamically helps agents generalize, which makes the targets non-stationary and changes the actions and states, and we can tune this either on a predetermined schedule or based on how the agent performs. A common strategy is performance-driven scheduling, where the task difficulty is regulated by the success rate (SR), as in Eurekaverse [Liang et al., 2024]. Other approaches target newly acquired or weaker skills, as in EvoCurr [Cheng et al., 2025] and EnvGen [Zala et al., 2024]. AgentGen's BI-EVAL mechanism [Hu et al., 2025a] introduces a bidirectional variation, which adjusts complexity upward or downward to match agent capability, in contrast to earlier methods that mostly increased difficulty [Xu et al., 2025, Luo et al., 2025c]. Beyond these, WebRL [Qi et al., 2025] implements self-adjusting curricula across complex web settings, and AgentGym [Xi et al., 2024] generalizes performance-adaptive scheduling to diverse benchmarks. R-Zero [Huang et al., 2025a] formalizes a challenger-solver paradigm in which a challenger proposes near-boundary tasks based on the solver's uncertainty, and the solver improves by training on filtered task sets, yielding iterative and targeted curricula.

Environmental Dynamics Environmental dynamics provide more realistic scenarios for agents. For example, the Meta Agents Research Environments (ARE) platform [Andrews et al., 2025] pushes this paradigm further by introducing a more realistic and dynamic environment. In most setups, if the agent is not interacting, the environment typically remains frozen. By contrast, ARE allows the environment to change dynamically through random or scheduled events at all times, allowing it to evolve asynchronously and independently of the agent. As shown in ARE's Mobile and Gaia2 benchmarks [Andrews et al., 2025], agents in such dynamic settings need to balance cognitive depth and temporal responsiveness to manage interaction latency and uncertainty as the environment's state continually changes. This design shifts the focus from static interaction to more continuous and proactive engagement.

3.3 Diversity Scaling

Scaling the diversity of data is key to building more robust and generalizable agents. Managing diversity at the task level (e.g., task difficulty, task objectives) helps agents acquire broader skills rather than overfitting to specific patterns [Hu et al., 2025a, Huang et al., 2025a]. At the environment level, exposing agents to a wide range of scenarios (e.g., different domains or tool suites) can further enhance their adaptability to novel situations. For example, representative works like AgentGen [Hu et al., 2025a] and AgentGym [Xi et al., 2024] synthesize a wide range of heterogeneous settings that broaden the training signals agents receive. Beyond these, AgentSense [Leng et al., 2025] generates diverse virtual sensor data by simulating different human personas and routines, and AgentBank [Song et al., 2024] shows that training on tens of thousands of heterogeneous interaction trajectories will substantially improves generalization. Collectively, these approaches demonstrate that diversity across tasks and environments is foundational for training capable, adaptable agents.

4 Stage 2: Task Execution

In the task execution stage, after the agent takes an action, it receives an observation from the environment. Consequently, whether the agent can interact with the environment in real time (*interactivity* § 4.1) and whether the returned observations are consistent with real-world scenarios (*realism* § 4.2) are both critical to the quality of the resulting experience. Accordingly, we organize environment scaling in this stage into two directions: *interactivity scaling* and *realism scaling*, as shown next to the action space in Figure 3.

4.1 Interactivity Scaling

Despite the advent of standard protocols such as the Model Context Protocol (MCP) [Anthropic, 2024, Luo et al., 2025b, Wang et al., 2025a, Fan et al., 2025] integrates heterogeneous data sources and tools into a unified form of context and thus greatly improves the efficiency and controllability of tool use, many datasets [Fan et al., 2025, Liu et al., 2024b, Qin et al., 2023] still consist of predefined, carefully curated sequences of tool calls, even some of them include real API or MCP calls. Under this non-interactive settings, each task has a single predefined solution path. Agents are blind to intermediate tool outputs and cannot adapt subsequent tool selection based on the returned results. Consequently, agents trained on this static supervision exhibit poor generalization to novel tasks and limited diversity in solution paths [Sullivan et al., 2025]. Recent methods [Pang et al., 2025, Wang et al., 2024, Yao et al., 2023, Trivedi et al., 2024, Sullivan et al., 2025] start to allow agents to interactively invoke real world APIs or leverage tools via function calling or code generation, where they can adjust subsequent tool selections based on current output. Among these approaches, BrowseMaster [Pang et al., 2025] further supports parallel tool calling, expanding the typical one-tool-call-per-invocation pattern to an average of 12.11 calls per invocation, which further increases interactivity. Another promising direction uses an offline real database as the interaction environment [Tongyi DeepResearch Team, 2025, Fang et al., 2025, Ye et al., 2025a, Yao et al., 2024, Barres et al., 2025, Prabhakar et al., 2025, Ye et al., 2025b], where the agent interactively calls functions to read and write the state database. This approach strikes a practical balance between interactivity and realism and helps agents accumulate meaningful experience efficiently.

4.2 Realism Scaling

To ensure that large language model (LLM) agents can generalize effectively to complex, real-world scenarios, the training data derived from these environments should maintain real-world consistency. In tool-use environments, earlier works [Qin et al., 2023, Lu et al., 2025b, Sun et al., 2025] utilize LLMs to generate the results of tool calls as simulations, avoiding monetary cost and unexpected implementation errors. However, to better ensure real-world consistency, more recent approaches start to use real APIs [Song et al., 2023, Wu et al., 2025b, Mastouri et al., 2025] or execute tasks in simulated environments backed by offline, real-world databases [Tongyi DeepResearch Team, 2025, Fang et al., 2025, Ye et al., 2025a, Yao et al., 2024, Barres et al., 2025, Prabhakar et al., 2025, Ye et al., 2025b]. Specifically, Tongyi DeepResearch Team [2025] builds a custom tool suite and a simulated environment based on an offline Wikipedia database, where agents execute actual tool calls to directly read and write to the database, achieving lower cost and higher efficiency. Beyond tool-use scenarios, where interactions are mainly text-based, the emerging paradigm of *thinking with images* [Su et al., 2025b] advocates using visual information as a dynamic, manipulable workspace for intermediate reasoning. Genie 3 [Parker-Holder and Fruchter, 2025] further extends realism scaling to 3D scenarios by implementing a physically grounded, real-time interactive 3D world. It preserves realistic physical properties while improving long-horizon consistency, offering a practical framework and motivating the development of more realistic environments for agents to explore.

Another path to improving realism of environments is the simulation of multi-agent settings. In such contexts, agents may coordinate or compete with each other [Tran et al., 2025, Qian et al., 2024, Li et al., 2024, Zhang et al., 2024, Kim et al., 2024], therefore each agent's behavior naturally becomes part of the environment for the others. As the number of agents scales, these interactions can produce emergent social and economic phenomena such as information diffusion, opinion polarization, and herding effects [Yang et al., 2025, Zhang et al., 2025c]. To reproduce such societal dynamics and improve data fidelity, multi-agent frameworks like Oasis [Yang et al., 2025] leverage real world social media data stored in a relational database to simulate interactive environments, enabling more realistic modeling of social processes. Meanwhile, Zhang et al. [2025c] move beyond earlier frameworks [Li et al., 2023, Gao et al., 2024b] that enforce execution order via message passing SOPs by adopting the MQTT communication protocol to support asynchronous decision making among autonomous agents, more closely simulating real-world workflows. Similarly, ARE [Andrews et al., 2025] decouples agent and environment clocks so that world state evolves asynchronously, and it simulates other agents' activities by treating them as independent events in the environment's event stream. Such

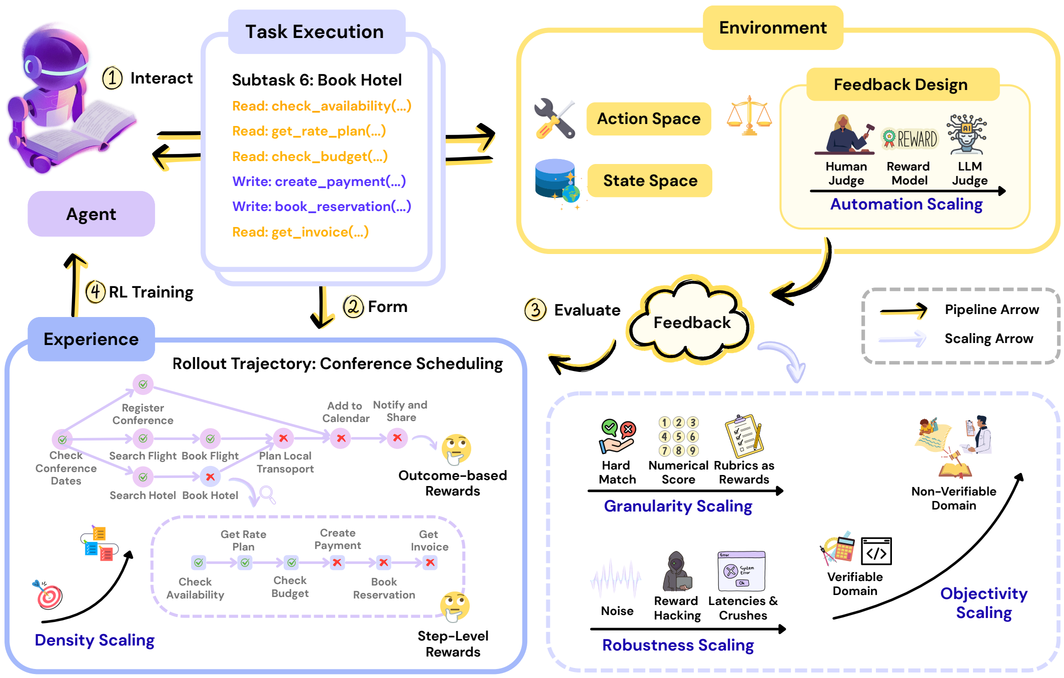

Figure 4: Illustration of environment scaling in the **feedback** stage using a conference-scheduling example. The agent first executes tasks in the environment and produces action-observation trajectories. The environment then evaluates these trajectories and returns feedback, yielding the experience used to train the agent. Scaling in the feedback stage covers *density*, *granularity*, *automation*, *objectivity*, and *robustness*.

environments enable the simulation of real-world societal processes and support the collection of more realistic experiential data.

5 Stage 3: Feedback

In the feedback stage, the environment assesses the trajectories collected during task execution and generates feedback signals for subsequent RL training. Scaling at this stage focuses on how feedback is provided, including its frequency and richness (*density* § 5.1 and *granularity* § 5.2), its level of automation (*automation* § 5.3), as well as how objectively and reliably it is delivered (*objectivity* § 5.4 and *robustness* § 5.5). Accordingly, we categorize representative scaling approaches along five dimensions: *density*, *granularity*, *automation*, *objectivity*, and *robustness*, as shown in Figure 4.

5.1 Density Scaling

The density of feedback refers to how frequently the environment provides evaluative feedback signals. Early environments typically provided trajectory-level rewards based on the final outcome (success or failure) [Jiang et al., 2025, Da et al., 2025, Qian et al., 2025, Wang et al., 2025c]. Despite such sparse signals leading to more stable training, especially in tasks like mathematics or code generation [DeepSeek-AI et al., 2025], they are insufficient for more complex multi-step tasks. In these tasks, it is challenging to verify whether an intermediate step contributes to or hinders task completion. To address this issue and scale feedback density, more recent studies [Khalifa et al., 2025, Chae et al., 2025, Gao et al., 2024a, Ma et al., 2025a, Park et al., 2025] have introduced step-level process-based rewards, often combined with traditional trajectory-level outcome-based rewards, thus providing denser supervision and more detailed guidance for agent improvement. However, the mechanism behind such reward designs remains underexplored, though some interesting phenomena have been observed in existing experiments. For instance, Ma et al. [2025a] shows that step-level reward models can perform well in logical-coherent tasks such as mathematical reasoning, but they are not suitable for some natural language tasks (e.g., creative writing, policy making). Gao et al.

[2024a] shows that naively combining step-level rewards with trajectory-level rewards can lead to reward hacking. That is, trivial actions may get high rewards, resulting in repetitive reasoning behavior that ultimately harms the agent's training process. They further use reward differences between adjacent steps as rewards and clip them if they exceed a predefined threshold to mitigate this problem. Apart from these, Park et al. [2025] finds that even state-of-the-art reward models can be poorly calibrated, assigning overly optimistic scores to some intermediate steps. These studies collectively showcase both the potential and bottlenecks of current dense reward designs. In the future, designing reward models that are more accurately calibrated, stable in training, and interpretable will be the focus of the reward density scaling.

5.2 Granularity Scaling

Granularity scaling refers to increasing the level of detail in feedback as well as enriching the forms of feedback provided. In earlier stages, evaluative feedback typically consisted of binary signals [Christiano et al., 2017a, Ibarz et al., 2018] or a single scalar score [Stiennon et al., 2022, Ouyang et al., 2022]. At finer granularities, the evaluative feedback on agents' performances is broken down into structured components, such as a set of scores across multiple criteria, offering more informative guidance [Huang et al., 2025b]. Building on this idea, some works use manually designed, multi-faceted rewards to improve correctness and format [Qian et al., 2025, Song et al., 2025]. The Rubrics as Rewards (RaR) framework [Gunjal et al., 2025, Huang et al., 2025c, Zhou et al., 2025, Zhang et al., 2025e, Viswanathan et al., 2025] further decomposes task requirements into tangible, human-interpretable criteria. By designing rewards as checklist-style, instance-specific rubrics, it provides a middle ground between binary correctness signals and broad preference rankings.

5.3 Automation Scaling

Automation scaling refers to the process where the feedback mechanism shifts from human supervision to automated evaluations generated by artificial intelligence. Reinforcement Learning from Human Feedback (RLHF) encompasses a wide range of frameworks, with the goal of aligning large language models (LLMs) with human preferences [Sheng et al., 2025, Hu et al., 2025b, Christiano et al., 2017b]. Although this method is effective, it heavily relies on human annotators, which results in a slow, costly and difficult-to-scale feedback process. In order to achieve automated feedback and reduce the reliance on labor-intensive preference annotations, an increasing number of studies have begun to use LLMs with evaluation capabilities as automated judges to replace human evaluators, thereby forming a new paradigm called "Reinforcement Learning from AI Feedback" (RLAIF) [Zhang et al., 2025d, Su et al., 2025a, Chen et al., 2025a, Lee et al., 2024, Bai et al., 2022]. The pioneering work REWARDAGENT [Peng et al., 2025] introduced a verification agent to simultaneously assess the factual correctness of model responses and their compliance with instructions, and combine these assessment results with basic human feedback to effectively guide model training. In contrast, ARMAP [Chen et al., 2025b] innovatively bypassed the need for a more powerful LLM to act as a judge, instead using the positive and negative trajectories generated by the LLM to train the classification model, thereby constructing an automatic and efficient reward mechanism. Rubric-based methods [Gunjal et al., 2025, Huang et al., 2025c], on the other hand, decompose open-ended tasks into multi-dimensional, interpretable criteria, improving the quality of rewards generated by LLMs. By integrating this automated assessment into the training environment, agents can gradually acquire higher-level capabilities without the need for continuous human supervision. However, this paradigm also magnifies risks such as the propagation of LLM's own biases in the process of adjudication, as well as the phenomenon of rewarding hacking, which refers to the situation where intelligent agents exploit the loopholes in the reward model to strive for high scores, thereby deviating from the intended goals and even causing the model to crash. This reveals the crucial trade-off between scalability and security, and emphasizes the need for research on robust risk mitigation strategies.

5.4 Objectivity Scaling

The RLVR paradigm has achieved great success in objective and easy-to-verify domains such as mathematical reasoning and code generation [Shao et al., 2024, DeepSeek-AI et al., 2025]. However, in many real-world scenarios such as creative writing, medical consultation, and policy making, the objectivity is subjective, open, and hard to verify. In these real environments, the feedback collected often contains many biases and noise. An environment for training agents should be able to extract accurate and high-quality feedback from such kinds of objectivity, and gradually scale from simple and verifiable tasks to difficult and hard-to-verify ones. Some recent studies have made early attempts in this direction [Andrews et al., 2025, Ma et al., 2025b, Akter et al., 2025, Su et al., 2025a, Yu et al., 2025b, Liu et al., 2025b, Gunjal et al., 2025, Huang et al., 2025c]. For example, Writing-Zero [Jia et al., 2025] uses a pairwise generative model to extract reliable and verifiable signals from subjective evaluations. Omni-Thinker [Li et al., 2025b] combines rule-based verifiable rewards with generative preference signals to form a unified multi-task RL training loop. ARE's verifier [Andrews et al., 2025] applies RLVR by using both hard checks (exact parameter matches) and soft checks (LLM-based semantic judgments) under strict causal order. Although these works show the feasibility of structured and verifiable rewards in open-ended environments, scaling environments to handle harder forms of objectivity still faces many challenges.

5.5 Robustness Scaling

Feedback robustness scaling requires the environment to provide more stable and reliable reward signals. Regarding the robustness of the reward itself, signals may be noisy, or face the challenge of reward hacking, which refers to agents learning undesirable and tricky behavioral patterns that obtain high reward without achieving the intended goals [Miao et al., 2024, Liu et al., 2024a, Farquhar et al., 2025, Fu et al., 2025, Tarek and Beheshti, 2025]. For the former issue, some approaches generate soft probabilistic rewards with generative verifiers to mitigate noise in rewards [Lin et al., 2024a, Su et al., 2025a]. And as for the latter, frameworks such as MONA [Farquhar et al., 2025] try to mitigate reward hacking problem by evaluating the future utility of actions through an overseer, thus constraining unstable behaviors while preserving explainability. Huang et al. [2025c] develops a reward hacking defense rubric that penalizes sycophantic praise towards user prompts and overly flattering self-assessments in responses, encouraging the model to produce more substantive content.

Apart from reward robustness, interaction robustness at environment-level is also crucial, since it is common for environmental instabilities (e.g., delays, crashes, corrupted tool outputs) to undermine feedback reliability and degrade the training process. To handle such failures, Trinity-RFT [Pan et al., 2025] proposes asynchronous inference and retry mechanisms, while Tongyi DeepResearch Team [2025] employs caching, retrying failed calls, and switching to similar providers to prevent corrupted trajectories. Looking ahead, future reward design should prioritize both efficacy and the prevention of hacking through tricky patterns, and greater attention should be paid to the design of more robust system.

6 Implementation Frameworks

There exist diverse implementations of simulated environments that vary in modality and complexity.

Visual Environments For 2D scenarios, tasks in grid-based environments can be symbolic or pixel-level, often associated with game-playing [Chevalier-Boisvert et al., 2023, Bellemare et al., 2013]. Works such as WebArena [Zhou et al., 2023], Mind2Web [Deng et al., 2023], and Visual-WebArena [Koh et al., 2024a] extend agent capabilities to more realistic, web-based environments. ARE [Andrews et al., 2025] further builds upon this line by introducing a mobile-style setting that integrates multiple apps and tools, incorporating time control and a structured verifier to support scalable evaluation. In 3D scenarios, environments emphasize more about embodied interaction, realistic physics and visual perception. More general simulators such as Habitat [Savva et al., 2019] and ThreeDWorld [Gan et al., 2020] support more actions such as navigation and manipulation, while domain-specific worlds develop the specific aspect. For example, MineRL [Guss et al., 2019] and MineDojo [Fan et al., 2022] exploit the flexibility of Minecraft. Besides, Household benchmarks such as ALFRED [Shridhar et al., 2020a] and EmbodiedQA [Das et al., 2018] further evaluate the grounding function of language in photo-realistic 3D spaces, requiring agents to follow instructions, answer questions, or perform multi-step tasks.

Text-based Environments Text-based environments emphasize more on reasoning and decision-making from natural language descriptions. TextWorld [Côté et al., 2018] proposes a framework to generate the text games with different goal-driven challenges. Reddit-RL-simulator [He et al., 2016, Chan and King, 2018] implements an RL environment for iteratively tracking and recommending popular discussion threads on Reddit. ALFWorld [Shridhar et al., 2020b] adapts ALFRED tasks into textual form, and JerichoAgentBench [Liu et al., 2023] extends the Jericho suite [Hausknecht et al., 2020] with annotated benchmarks and concrete objectives. Such text environments stress the agent's ability to infer latent dynamics under partial observability, making them a natural testbed for language-based decision-making.

7 Future Directions

Co-Evolution via Embedded External Tools As the complexity of task increases, feedback mechanisms must scale in order to avoid sparse, noisy, or misaligned learning signals. Future environments can achieve this by including external tools or modules to serve as verifiers, simulators, compilers, or executable systems into the learning loop. These tools would evolve together with the task generation, providing structured, verifiable feedback and enabling agents to interact with increasingly sophisticated challenges. By integrating embedded tools with automated evaluators such as LLM-based formative signals, environments can better support multi-step, open-ended, and creative problem solving, while also helping to mitigate the Generator-Verifier Asymmetry.

Scaling Through Generator-Verifier Synergy Future environments can encourage stronger generators to have the ability to decompose complex tasks into smaller subproblems with intermediate solutions, making them tractable for weaker verifiers. This enables scalable supervision in domains where holistic verification is difficult such as creative cultural production and policy-making. In contrast, weaker generators could provide diverse candidate solutions that could be filtered, ranked, or refined by stronger verifiers. By incorporating these co-evolving dynamics into the environment, the asymmetry between generators and verifiers can serve as a catalyst for continuous self-improvement.

Open-Ended, Multi-Agent Environments Future environments can scale to support large-scale multi-agent interactions, emergent social dynamics, and economic or organizational level simulations, providing rich contexts for studying both collaborative and non-cooperative behaviors in complex environments. In particular, scaling to massively multi-cultural and multi-lingual settings requires scaling environment construction for agents to navigate the subtle semantics of concepts and values that vary across different societies, not only in the textual but also in the multi-modal domain [Koh et al., 2024b, Huang et al., 2025d]. Such open-ended and interactive scenarios environments foster generalization and strategic planning abilities of agents, equipping LLM agents to better handle complex, real-world challenges beyond isolated task execution.

8 Conclusion

The era of experience makes environments central to the development of LLM agents, positioning them as active producers of experiential data and underscoring the growing need for scaling environments to create a more complex, realistic, and richly interactive world. From a pioneering environment-centric perspective, our survey proposes a unified taxonomy that organizes representative work across the GEF loop (task generation, task execution, and feedback), together with evaluation, implementation, and applications. Besides, we surface key challenges for advancing agent intelligence, including the asymmetry between generators and verifiers and the construction of more open, large-scale multi-agent environments, thereby providing insights for future research on agentic systems.

Limitations

As scaling environments remains an emerging research topic, relatively few studies have explicitly adopted this framing. Thus, we take a broad view of environment and organize representative studies along the GEF loop (task generation, task execution, and feedback) from a pioneering environment-centric perspective. While this broader lens brings in some adjacent lines of work that may not have been explicitly designed from the environment side, our taxonomy is both comprehensive and insightful. Given the rapid pace of agentic research, some of the most recent papers may fall outside this snapshot. We will continue to update this survey as the literature evolves.

Acknowledgments

We thank the anonymous reviewers for their valuable suggestions and comments. Yuchen Huang was supported by the Hong Kong Ph.D. Fellowship Scheme (HKPFS).

References

Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi Chen, Ziyue Qiao, Qingqing Long, Rongcheng Tu, Xiao Luo, Wei Ju, Zhiping Xiao, Yifan Wang, Meng Xiao, Chenwu Liu, Jingyang Yuan, Shichang Zhang, Yiqiao Jin, Fan Zhang, Xian Wu, Hanqing Zhao, Dacheng Tao, Philip S. Yu, and Ming Zhang. Large language model agent: A survey on methodology, applications and challenges, 2025a. URL https://arxiv.org/abs/2503.21460.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 68(2):121101, 2025.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Anthropic. Introducing claude 4. https://www.anthropic.com/news/claude-4, 2025.

OpenAI. Introducing ChatGPT agent: bridging research and action. https://openai.com/index/introducing-chatgpt-agent/, 2025a.

Zhitao He, Zijun Liu, Peng Li, Yi R. Fung, Ming Yan, Ji Zhang, Fei Huang, and Yang Liu. Advancing language multi-agent learning with credit re-assignment for interactive environment generalization. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=SoEmgM1ioC.

Wentao Zhang, Liang Zeng, Yuzhen Xiao, Yongcong Li, Ce Cui, Yilei Zhao, Rui Hu, Yang Liu, Yahui Zhou, and Bo An. Agentorchestra: A hierarchical multi-agent framework for general-purpose task solving. *arXiv preprint arXiv:2506.12508*, 2025a.

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

Tongyi DeepResearch Team. Tongyi-deepresearch. https://github.com/Alibaba-NLP/DeepResearch, 2025.

OpenAI. Introducing Deep Research. https://openai.com/index/introducing-deep-research/, 2025b.

Google DeepMind. Gemini Deep Research. https://blog.google/products/gemini/google-gemini-deep-research/, 2024.

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-zero: Self-evolving reasoning llm from zero data. *arXiv preprint arXiv:2508.05004*, 2025a.

Yi Su, Dian Yu, Linfeng Song, Juntao Li, Haitao Mi, Zhaopeng Tu, Min Zhang, and Dong Yu. Crossing the reward bridge: Expanding rl with verifiable rewards across diverse domains, 2025a. URL https://arxiv.org/abs/2503.23829.

Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data, 2025. URL https://arxiv.org/abs/2505.03335.

Zhengwei Tao, Ting-En Lin, Xiancai Chen, Hangyu Li, Yuchuan Wu, Yongbin Li, Zhi Jin, Fei Huang, Dacheng Tao, and Jingren Zhou. A survey on self-evolution of large language models, 2024. URL https://arxiv.org/abs/2404.14387.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, Yifan Zhou, Yang Chen, Chen Zhang, Yutao Fan, Zihu Wang, Songtao Huang, Yue Liao, Hongru Wang, Mengyue Yang, Heng Ji, Michael Littman, Jun Wang, Shuicheng Yan, Philip Torr, and Lei Bai. The landscape of agentic reinforcement learning for llms: A survey, 2025b. URL https://arxiv.org/abs/2509.02547.

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

CAMEL-AI. Scaling environments for agents. https://www.camel-ai.org/blogs/scaling-environments-for-agents, 2025.

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

Zhiheng Xi, Yiwen Ding, Wenxiang Chen, Boyang Hong, Honglin Guo, Junzhe Wang, Dingwen Yang, Chenyang Liao, Xin Guo, Wei He, et al. Agentgym: Evolving large language model-based agents across diverse environments. *CoRR*, 2024.

Zichen Liu, Anya Sims, Keyu Duan, Changyu Chen, Simon Yu, Xiangxin Zhou, Haotian Xu, Shaopan Xiong, Bo Liu, Chenmien Tan, Chuen Yang Beh, Weixun Wang, Hao Zhu, Weiyan Shi, Diyi Yang, Michael Shieh, Yee Whye Teh, Wee Sun Lee, and Min Lin. Gem: A gym for agentic llms, 2025a. URL https://arxiv.org/abs/2510.01051.

Michael Sullivan, Mareike Hartmann, and Alexander Koller. Procedural environment generation for tool-use agents. *arXiv preprint arXiv:2506.11045*, 2025.

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

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

Shishir G. Patil, Huanzhi Mao, Charlie Cheng-Jie Ji, Fanjia Yan, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (bfcl): From tool use to agentic evaluation of large language models. In *Advances in Neural Information Processing Systems*, 2024.

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

Peijie Yu, Yifan Yang, Jinjian Li, Zelong Zhang, Haorui Wang, Xiao Feng, and Feng Zhang. c^3-bench: The things real disturbing llm based agent in multi-tasking. *arXiv preprint arXiv:2505.18746*, 2025a.

Dingfeng Shi, Jingyi Cao, Qianben Chen, Weichen Sun, Weizhen Li, Hongxuan Lu, Fangchen Dong, Tianrui Qin, King Zhu, Minghao Liu, et al. Taskcraft: Automated generation of agentic tasks. *arXiv preprint arXiv:2506.10055*, 2025.

Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Gang Fu, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Webdancer: Towards autonomous information seeking agency, 2025a. URL https://arxiv.org/abs/2505.22648.

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and Fei Huang. Webwalker: Benchmarking llms in web traversal, 2025b. URL https://arxiv.org/abs/2501.07572.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, Weizhou Shen, Junkai Zhang, Dingchu Zhang, Xixi Wu, Yong Jiang, Ming Yan, Pengjun Xie, Fei Huang, and Jingren Zhou. Websailor: Navigating super-human reasoning for web agent, 2025a. URL https://arxiv.org/abs/2507.02592.

Zhengwei Tao, Jialong Wu, Wenbiao Yin, Junkai Zhang, Baixuan Li, Haiyang Shen, Kuan Li, Liwen Zhang, Xinyu Wang, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Webshaper: Agentically data synthesizing via information-seeking formalization, 2025. URL https://arxiv.org/abs/2507.15061.

William Liang, Sam Wang, Hung-Ju Wang, Osbert Bastani, Dinesh Jayaraman, and Yecheng Jason Ma. Environment curriculum generation via large language models. In *8th Annual Conference on Robot Learning*, 2024.

Yang Cheng, Zilai Wang, Weiyu Ma, Wenhui Zhu, Yue Deng, and Jian Zhao. Evocurr: Self-evolving curriculum with behavior code generation for complex decision-making. *arXiv preprint arXiv:2508.09586*, 2025.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Jiadai Sun, Xinyue Yang, Yu Yang, Shuntian Yao, Wei Xu, et al. Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Abhay Zala, Jaemin Cho, Han Lin, Jaehong Yoon, and Mohit Bansal. Envgen: Generating and adapting environments via llms for training embodied agents. *arXiv preprint arXiv:2403.12014*, 2024.

Zikang Leng, Megha Thukral, Yaqi Liu, Hrudhai Rajasekhar, Shruthi K Hiremath, and Thomas Plötz. Agentsense: Virtual sensor data generation using llm agent in simulated home environments. *arXiv preprint arXiv:2506.11773*, 2025.

Yifan Song, Weimin Xiong, Xiutian Zhao, Dawei Zhu, Wenhao Wu, Ke Wang, Cheng Li, Wei Peng, and Sujian Li. Agentbank: Towards generalized llm agents via fine-tuning on 50000+ interaction trajectories. In *EMNLP (Findings)*, 2024.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. Appworld: A controllable world of apps and people for benchmarking interactive coding agents, 2024. URL https://arxiv.org/abs/2407.18901.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents, 2023. URL https://arxiv.org/abs/2207.01206.

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

Zhenting Wang, Qi Chang, Hemani Patel, Shashank Biju, Cheng-En Wu, Quan Liu, Aolin Ding, Alireza Rezazadeh, Ankit Shah, Yujia Bao, et al. Mcp-bench: Benchmarking tool-using llm agents with complex real-world tasks via mcp servers. *arXiv preprint arXiv:2508.20453*, 2025a.

Shiqing Fan, Xichen Ding, Liang Zhang, and Linjian Mo. Mcptoolbench++: A large scale ai agent model context protocol mcp tool use benchmark. *arXiv preprint arXiv:2508.07575*, 2025.

Runnan Fang, Shihao Cai, Baixuan Li, Jialong Wu, Guangyu Li, Wenbiao Yin, Xinyu Wang, Xiaobin Wang, Liangcai Su, Zhen Zhang, Shibin Wu, Zhengwei Tao, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Towards general agentic intelligence via environment scaling, 2025. URL https://arxiv.org/abs/2509.13311.

Xianghe Pang, Shuo Tang, Rui Ye, Yuwen Du, Yaxin Du, and Siheng Chen. Browsemaster: Towards scalable web browsing via tool-augmented programmatic agent pair, 2025. URL https://arxiv.org/abs/2508.09129.

Yifan Song, Weimin Xiong, Dawei Zhu, Wenhao Wu, Han Qian, Mingbo Song, Hailiang Huang, Cheng Li, Ke Wang, Rong Yao, Ye Tian, and Sujian Li. Restgpt: Connecting large language models with real-world restful apis, 2023. URL https://arxiv.org/abs/2306.06624.

Meriem Mastouri, Emna Ksontini, and Wael Kessentini. Making rest apis agent-ready: From openapi to mcp servers for tool-augmented llms, 2025. URL https://arxiv.org/abs/2507.16044.

Junjie Ye, Changhao Jiang, Zhengyin Du, Yufei Xu, Xuesong Yao, Zhiheng Xi, Xiaoran Fan, Qi Zhang, Tao Gui, Xuanjing Huang, and Jiecao Chen. Feedback-driven tool-use improvements in large language models via automated build environments, 2025a. URL https://arxiv.org/abs/2508.08791.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment, 2025. URL https://arxiv.org/abs/2506.07982.

Akshara Prabhakar, Zuxin Liu, Ming Zhu, Jianguo Zhang, Tulika Awalgaonkar, Shiyu Wang, Zhiwei Liu, Haolin Chen, Thai Hoang, Juan Carlos Niebles, Shelby Heinecke, Weiran Yao, Huan Wang, Silvio Savarese, and Caiming Xiong. Apigen-mt: Agentic pipeline for multi-turn data generation via simulated agent-human interplay, 2025. URL https://arxiv.org/abs/2504.03601.

Junjie Ye, Zhengyin Du, Xuesong Yao, Weijian Lin, Yufei Xu, Zehui Chen, Zaiyuan Wang, Sining Zhu, Zhiheng Xi, Siyu Yuan, Tao Gui, Qi Zhang, Xuanjing Huang, and Jiecao Chen. Toolhop: A query-driven benchmark for evaluating large language models in multi-hop tool use, 2025b. URL https://arxiv.org/abs/2501.02506.

Jack Parker-Holder and Shlomi Fruchter. Genie 3: A new frontier for world models, 2025. URL https://deepmind.com/discover/blog/genie-3-a-new-frontier-for-world-models/.

Ziyi Yang, Zaibin Zhang, Zirui Zheng, Yuxian Jiang, Ziyue Gan, Zhiyu Wang, Zijian Ling, Jinsong Chen, Martz Ma, Bowen Dong, Prateek Gupta, Shuyue Hu, Zhenfei Yin, Guohao Li, Xu Jia, Lijun Wang, Bernard Ghanem, Huchuan Lu, Chaochao Lu, Wanli Ouyang, Yu Qiao, Philip Torr, and Jing Shao. Oasis: Open agent social interaction simulations with one million agents, 2025. URL https://arxiv.org/abs/2411.11581.

Jun Zhang, Yuwei Yan, Junbo Yan, Zhiheng Zheng, Jinghua Piao, Depeng Jin, and Yong Li. A parallelized framework for simulating large-scale llm agents with realistic environments and interactions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 1339–1349, 2025c.

Dongfu Jiang, Yi Lu, Zhuofeng Li, Zhiheng Lyu, Ping Nie, Haozhe Wang, Alex Su, Hui Chen, Kai Zou, Chao Du, et al. Verltool: Towards holistic agentic reinforcement learning with tool use. *arXiv preprint arXiv:2509.01055*, 2025.

Jeff Da, Clinton Wang, Xiang Deng, Yuntao Ma, Nikhil Barhate, and Sean Hendryx. Agent-rlvr: Training software engineering agents via guidance and environment rewards. *arXiv preprint arXiv:2506.11425*, 2025.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*, 2025.

Yiran Ma, Zui Chen, Tianqiao Liu, Mi Tian, Zhuo Liu, Zitao Liu, and Weiqi Luo. What are step-level reward models rewarding? counterintuitive findings from mcts-boosted mathematical reasoning, 2025a. URL https://arxiv.org/abs/2412.15904.

Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. Process reward models that think, 2025. URL https://arxiv.org/abs/2504.16828.

Hyungjoo Chae, Sunghwan Kim, Junhee Cho, Seungone Kim, Seungjun Moon, Gyeom Hwangbo, Dongha Lim, Minjin Kim, Yeonjun Hwang, Minju Gwak, Dongwook Choi, Minseok Kang, Gwanhoon Im, ByeongUng Cho, Hyojun Kim, Jun Hee Han, Taeyoon Kwon, Minju Kim, Beong woo Kwak, Dongjin Kang, and Jinyoung Yeo. Web-shepherd: Advancing prms for reinforcing web agents, 2025. URL https://arxiv.org/abs/2505.15277.

Jiaxuan Gao, Shusheng Xu, Wenjie Ye, Weilin Liu, Chuyi He, Wei Fu, Zhiyu Mei, Guangju Wang, and Yi Wu. On designing effective rl reward at training time for llm reasoning. *arXiv preprint arXiv:2410.15115*, 2024a.

Shijue Huang, Hongru Wang, Wanjun Zhong, Zhaochen Su, Jiazhan Feng, Bowen Cao, and Yi R Fung. Adactrl: Towards adaptive and controllable reasoning via difficulty-aware budgeting. *arXiv preprint arXiv:2505.18822*, 2025b.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*, 2025.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains, 2025. URL https://arxiv.org/abs/2507.17746.

Zenan Huang, Yihong Zhuang, Guoshan Lu, Zeyu Qin, Haokai Xu, Tianyu Zhao, Ru Peng, Jiaqi Hu, Zhanming Shen, Xiaomeng Hu, et al. Reinforcement learning with rubric anchors. *arXiv preprint arXiv:2508.12790*, 2025c.

Yang Zhou, Sunzhu Li, Shunyu Liu, Wenkai Fang, Kongcheng Zhang, Jiale Zhao, Jingwen Yang, Yihe Zhou, Jianwei Lv, Tongya Zheng, Hengtong Lu, Wei Chen, Yan Xie, and Mingli Song. Breaking the exploration bottleneck: Rubric-scaffolded reinforcement learning for general llm reasoning, 2025. URL https://arxiv.org/abs/2508.16949.

Nuo Chen, Zhiyuan Hu, Qingyun Zou, Jiaying Wu, Qian Wang, Bryan Hooi, and Bingsheng He. Judgelrm: Large reasoning models as a judge, 2025a. URL https://arxiv.org/abs/2504.00050.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction, 2025d. URL https://arxiv.org/abs/2408.15240.

Hao Peng, Yunjia Qi, Xiaozhi Wang, Zijun Yao, Bin Xu, Lei Hou, and Juanzi Li. Agentic reward modeling: Integrating human preferences with verifiable correctness signals for reliable reward systems, 2025. URL https://arxiv.org/abs/2502.19328.

Zhenfang Chen, Delin Chen, Rui Sun, Wenjun Liu, and Chuang Gan. Scaling autonomous agents via automatic reward modeling and planning, 2025b. URL https://arxiv.org/abs/2502.12130.

Xueguang Ma, Qian Liu, Dongfu Jiang, Ge Zhang, Zejun Ma, and Wenhu Chen. General-reasoner: Advancing llm reasoning across all domains. *arXiv preprint arXiv:2505.14652*, 2025b.

Tianyu Yu, Bo Ji, Shouli Wang, Shu Yao, Zefan Wang, Ganqu Cui, Lifan Yuan, Ning Ding, Yuan Yao, Zhiyuan Liu, et al. Rlpr: Extrapolating rlvr to general domains without verifiers. *arXiv preprint arXiv:2506.18254*, 2025b.

Syeda Nahida Akter, Shrimai Prabhumoye, Matvei Novikov, Seungju Han, Ying Lin, Evelina Bakhturina, Eric Nyberg, Yejin Choi, Mostofa Patwary, Mohammad Shoeybi, et al. Nemotron-crossthink: Scaling self-learning beyond math reasoning. *arXiv preprint arXiv:2504.13941*, 2025.

Wei Liu, Siya Qi, Xinyu Wang, Chen Qian, Yali Du, and Yulan He. Nover: Incentive training for language models via verifier-free reinforcement learning. *arXiv preprint arXiv:2505.16022*, 2025b.

Ruipeng Jia, Yunyi Yang, Yongbo Gai, Kai Luo, Shihao Huang, Jianhe Lin, Xiaoxi Jiang, and Guanjun Jiang. Writing-zero: Bridge the gap between non-verifiable tasks and verifiable rewards, 2025. URL https://arxiv.org/abs/2506.00103.

Derek Li, Jiaming Zhou, Amirreza Kazemi, Qianyi Sun, Abbas Ghaddar, Mohammad Ali Alomrani, Liheng Ma, Yu Luo, Dong Li, Feng Wen, et al. Omni-think: Scaling cross-domain generalization in llms via multi-task rl with hybrid rewards. *arXiv preprint arXiv:2507.14783*, 2025b.

Muhan Lin, Shuyang Shi, Yue Guo, Behdad Chalaki, Vaishnav Tadiparthi, Ehsan Moradi Pari, Simon Stepputtis, Joseph Campbell, and Katia Sycara. Navigating noisy feedback: Enhancing reinforcement learning with error-prone language models. *arXiv preprint arXiv:2410.17389*, 2024a.

Sebastian Farquhar, Vikrant Varma, David Lindner, David Elson, Caleb Biddulph, Ian Goodfellow, and Rohin Shah. Mona: Myopic optimization with non-myopic approval can mitigate multi-step reward hacking, 2025. URL https://arxiv.org/abs/2501.13011.

Jiayi Fu, Xuandong Zhao, Chengyuan Yao, Heng Wang, Qi Han, and Yanghua Xiao. Reward shaping to mitigate reward hacking in rlhf. *arXiv preprint arXiv:2502.18770*, 2025.

Tianqi Liu, Wei Xiong, Jie Ren, Lichang Chen, Junru Wu, Rishabh Joshi, Yang Gao, Jiaming Shen, Zhen Qin, Tianhe Yu, et al. Rrm: Robust reward model training mitigates reward hacking. *arXiv preprint arXiv:2409.13156*, 2024a.

Yuchun Miao, Sen Zhang, Liang Ding, Rong Bao, Lefei Zhang, and Dacheng Tao. Inform: Mitigating reward hacking in rlhf via information-theoretic reward modeling, 2024. URL https://arxiv.org/abs/2402.09345.

Xuchen Pan, Yanxi Chen, Yushuo Chen, Yuchang Sun, Daoyuan Chen, Wenhao Zhang, Yuexiang Xie, Yilun Huang, Yilei Zhang, Dawei Gao, Weijie Shi, Yaliang Li, Bolin Ding, and Jingren Zhou. Trinity-rft: A general-purpose and unified framework for reinforcement fine-tuning of large language models, 2025. URL https://arxiv.org/abs/2505.17826.

Asaf Yehudai, Lilach Eden, Alan Li, Guy Uziel, Yilun Zhao, Roy Bar-Haim, Arman Cohan, and Michal Shmueli-Scheuer. Survey on evaluation of llm-based agents. *arXiv preprint arXiv:2503.16416*, 2025.

Chen Qian, Zihao Xie, Yifei Wang, Wei Liu, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Scaling large-language-model-based multi-agent collaboration. *CoRR*, 2024.

Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D Nguyen. Multi-agent collaboration mechanisms: A survey of llms. *arXiv preprint arXiv:2501.06322*, 2025.

Jason Wei. Asymmetry of verification and verifier's rule. Blog post, July 2025. https://www.jasonwei.net/blog/asymmetry-of-verification-and-verifiers-law.

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

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

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Faeze Brahman, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. Wildbench: Benchmarking llms with challenging tasks from real users in the wild. *arXiv preprint arXiv:2406.04770*, 2024b.

Rahul K Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, et al. Healthbench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*, 2025.

Haitao Hong, Yuchen Yan, Xingyu Wu, Guiyang Hou, Wenqi Zhang, Weiming Lu, Yongliang Shen, and Jun Xiao. Cooper: Co-optimizing policy and reward models in reinforcement learning for large language models. *arXiv preprint arXiv:2508.05613*, 2025.

Songshuo Lu, Hua Wang, Zhi Chen, and Yaohua Tang. Urpo: A unified reward & policy optimization framework for large language models. *arXiv preprint arXiv:2507.17515*, 2025a.

Lili Chen, Mihir Prabhudesai, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. Self-questioning language models. *arXiv preprint arXiv:2508.03682*, 2025c.

Shaobo Wang, Zhengbo Jiao, Zifan Zhang, Yilang Peng, Xu Ze, Boyu Yang, Wei Wang, Hu Wei, and Linfeng Zhang. Socratic-zero : Bootstrapping reasoning via data-free agent co-evolution, 2025b. URL https://arxiv.org/abs/2509.24726.

Nan Shao, Zefan Cai, Hanwei Xu, Chonghua Liao, Yanan Zheng, and Zhilin Yang. Compositional task representations for large language models. In *ICLR*, 2023.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. Wizardlm: Empowering large pre-trained language models to follow complex instructions, 2025. URL https://arxiv.org/abs/2304.12244.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct, 2025c. URL https://arxiv.org/abs/2308.09583.

Zuxin Liu, Thai Hoang, Jianguo Zhang, Ming Zhu, Tian Lan, Shirley Kokane, Juntao Tan, Weiran Yao, Zhiwei Liu, Yihao Feng, Rithesh Murthy, Liangwei Yang, Silvio Savarese, Juan Carlos Niebles, Huan Wang, Shelby Heinecke, and Caiming Xiong. Apigen: Automated pipeline for generating verifiable and diverse function-calling datasets, 2024b. URL https://arxiv.org/abs/2406.18518.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. Executable code actions elicit better llm agents. In *Forty-first International Conference on Machine Learning*, 2024.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Felix Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. Toolsandbox: A stateful, conversational, interactive evaluation benchmark for llm tool use capabilities, 2025b. URL https://arxiv.org/abs/2408.04682.

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. Zerosearch: Incentivize the search capability of llms without searching, 2025. URL https://arxiv.org/abs/2505.04588.

Zhaochen Su, Peng Xia, Hangyu Guo, Zhenhua Liu, Yan Ma, Xiaoye Qu, Jiaqi Liu, Yanshu Li, Kaide Zeng, Zhengyuan Yang, et al. Thinking with images for multimodal reasoning: Foundations, methods, and future frontiers. *arXiv preprint arXiv:2506.23918*, 2025b.

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

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Arik. Chain of agents: Large language models collaborating on long-context tasks. *Advances in Neural Information Processing Systems*, 37:132208–132237, 2024.

Kyungha Kim, Sangyun Lee, Kung-Hsiang Huang, Hou Pong Chan, Manling Li, and Heng Ji. Can llms produce faithful explanations for fact-checking? towards faithful explainable fact-checking via multi-agent debate, 2024. URL https://arxiv.org/abs/2402.07401.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for "mind" exploration of large language model society, 2023. URL https://arxiv.org/abs/2303.17760.

Dawei Gao, Zitao Li, Xuchen Pan, Weirui Kuang, Zhijian Ma, Bingchen Qian, Fei Wei, Wenhao Zhang, Yuexiang Xie, Daoyuan Chen, Liuyi Yao, Hongyi Peng, Zeyu Zhang, Lin Zhu, Chen Cheng, Hongzhu Shi, Yaliang Li, Bolin Ding, and Jingren Zhou. Agentscope: A flexible yet robust multi-agent platform, 2024b. URL https://arxiv.org/abs/2402.14034.

Hongru Wang, Cheng Qian, Wanjun Zhong, Xiusi Chen, Jiahao Qiu, Shijue Huang, Bowen Jin, Mengdi Wang, Kam-Fai Wong, and Heng Ji. Otc: Optimal tool calls via reinforcement learning. *arXiv e-prints*, pages arXiv–2504, 2025c.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Young-Jin Park, Kristjan Greenewald, Kaveh Alim, Hao Wang, and Navid Azizan. Know what you don't know: Uncertainty calibration of process reward models, 2025. URL https://arxiv.org/abs/2506.09338.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017a. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf.

Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in atari, 2018. URL https://arxiv.org/abs/1811.06521.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback, 2022. URL https://arxiv.org/abs/2009.01325.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Junkai Zhang, Zihao Wang, Lin Gui, Swarnashree Mysore Sathyendra, Jaehwan Jeong, Victor Veitch, Wei Wang, Yunzhong He, Bing Liu, and Lifeng Jin. Chasing the tail: Effective rubric-based reward modeling for large language model post-training, 2025e. URL https://arxiv.org/abs/2509.21500.

Vijay Viswanathan, Yanchao Sun, Shuang Ma, Xiang Kong, Meng Cao, Graham Neubig, and Tongshuang Wu. Checklists are better than reward models for aligning language models, 2025. URL https://arxiv.org/abs/2507.18624.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, page 1279–1297. ACM, March 2025. doi: 10.1145/3689031.3696075. URL http://dx.doi.org/10.1145/3689031.3696075.

Jian Hu, Xibin Wu, Wei Shen, Jason Klein Liu, Zilin Zhu, Weixun Wang, Songlin Jiang, Haoran Wang, Hao Chen, Bin Chen, Weikai Fang, Xianyu, Yu Cao, Haotian Xu, and Yiming Liu. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework, 2025b. URL https://arxiv.org/abs/2405.11143.

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

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback, 2024. URL https://arxiv.org/abs/2309.00267.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022. URL https://arxiv.org/abs/2212.08073.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Mirza Farhan Bin Tarek and Rahmatollah Beheshti. Reward hacking mitigation using verifiable composite rewards. *arXiv preprint arXiv:2509.15557*, 2025.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo Perez-Vicente, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *Advances in Neural Information Processing Systems*, 36:73383–73394, 2023.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of artificial intelligence research*, 47: 253–279, 2013.

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

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

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

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347, 2019.

Chuang Gan, Jeremy Schwartz, Seth Alter, Damian Mrowca, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, et al. Threedworld: A platform for interactive multi-modal physical simulation. *arXiv preprint arXiv:2007.04954*, 2020.

William H Guss, Brandon Houghton, Nicholay Topin, Phillip Wang, Cayden Codel, Manuela Veloso, and Ruslan Salakhutdinov. Minerl: A large-scale dataset of minecraft demonstrations. *arXiv preprint arXiv:1907.13440*, 2019.

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

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749, 2020a.

Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–10, 2018.

Marc-Alexandre Côté, Akos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, et al. Textworld: A learning environment for text-based games. In *Workshop on Computer Games*, pages 41–75. Springer, 2018.

Ji He, Mari Ostendorf, Xiaodong He, Jianshu Chen, Jianfeng Gao, Lihong Li, and Li Deng. Deep reinforcement learning with a combinatorial action space for predicting popular Reddit threads. In Jian Su, Kevin Duh, and Xavier Carreras, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1838–1848, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1189. URL https://aclanthology.org/D16-1189/.

Hou Pong Chan and Irwin King. Thread popularity prediction and tracking with a permutation-invariant model. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3392–3401, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1376. URL https://aclanthology.org/D18-1376/.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020b.

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

Matthew Hausknecht, Prithviraj Ammanabrolu, Marc-Alexandre Côté, and Xingdi Yuan. Interactive fiction games: A colossal adventure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7903–7910, 2020.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 881–905, Bangkok, Thailand, August 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.50. URL https://aclanthology.org/2024.acl-long.50/.

Yuchen Huang, Zhiyuan Fan, Zhitao He, Sandeep Polisetty, Wenyan Li, and Yi R Fung. Cultureclip: Empowering clip with cultural awareness through synthetic images and contextualized captions. In *Second Conference on Language Modeling*, 2025d.

Ruoyao Wang, Graham Todd, Eric Yuan, Ziang Xiao, Marc-Alexandre Côté, and Peter Jansen. Bytesized32: A corpus and challenge task for generating task-specific world models expressed as text games. *arXiv preprint arXiv:2305.14879*, 2023.

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

Zihui Xue, Kumar Ashutosh, and Kristen Grauman. Learning object state changes in videos: An open-world perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18493–18503, 2024.

Haoyi Duan, Hong-Xing Yu, Sirui Chen, Li Fei-Fei, and Jiajun Wu. Worldscore: A unified evaluation benchmark for world generation. *arXiv preprint arXiv:2504.00983*, 2025.

Delong Chen, Willy Chung, Yejin Bang, Ziwei Ji, and Pascale Fung. Worldprediction: A benchmark for high-level world modeling and long-horizon procedural planning. *arXiv preprint arXiv:2506.04363*, 2025d.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

OpenAI. Function calling. https://platform.openai.com/docs/guides/function-calling, 2023.

Qwen Team. Qwen3-coder technical blog. https://qwenlm.github.io/blog/qwen3-coder/, 2024. URL https://qwenlm.github.io/blog/qwen3-coder/. Accessed: 2024-07-23.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in llms, 2025. URL https://arxiv.org/abs/2504.11536.

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

Fosowl. AgenticSeek: A fully local and open-source AI agent for autonomous task execution. GitHub repository, 2024. URL https://github.com/Fosowl/AgenticSeek. Accessed: 2024-09-02.

Significant-Gravitas. Auto-GPT: An Autonomous GPT-4 Experiment. GitHub repository, 2023. URL https://github.com/Significant-Gravitas/Auto-GPT. Accessed: 2023-09-02.

Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, Wanjun Zhong, Kuanye Li, Jiale Yang, Yu Miao, Woyu Lin, Longxiang Liu, Xu Jiang, Qianli Ma, Jingyu Li, Xiaojun Xiao, Kai Cai, Chuang Li, Yaowei Zheng, Chaolin Jin, Chen Li, Xiao Zhou, Minchao Wang, Haoli Chen, Zhaojian Li, Haihua Yang, Haifeng Liu, Feng Lin, Tao Peng, Xin Liu, and Guang Shi. Ui-tars: Pioneering automated gui interaction with native agents, 2025. URL https://arxiv.org/abs/2501.12326.

Xinyuan Wang, Bowen Wang, Dunjie Lu, Junlin Yang, Tianbao Xie, Junli Wang, Jiaqi Deng, Xiaole Guo, Yiheng Xu, Chen Henry Wu, Zhennan Shen, Zhuokai Li, Ryan Li, Xiaochuan Li, Junda Chen, Boyuan Zheng, Peihang Li, Fangyu Lei, Ruisheng Cao, Yeqiao Fu, Dongchan Shin, Martin Shin, Jiarui Hu, Yuyan Wang, Jixuan Chen, Yuxiao Ye, Danyang Zhang, Dikang Du, Hao Hu, Huarong Chen, Zaida Zhou, Haotian Yao, Ziwei Chen, Qizheng Gu, Yipu Wang, Heng Wang, Diyi Yang, Victor Zhong, Flood Sung, Y. Charles, Zhilin Yang, and Tao Yu. Opencua: Open foundations for computer-use agents, 2025d. URL https://arxiv.org/abs/2508.09123.

Jiabo Ye, Xi Zhang, Haiyang Xu, Haowei Liu, Junyang Wang, Zhaoqing Zhu, Ziwei Zheng, Feiyu Gao, Junjie Cao, Zhengxi Lu, Jitong Liao, Qi Zheng, Fei Huang, Jingren Zhou, and Ming Yan. Mobile-agent-v3: Fundamental agents for gui automation, 2025c. URL https://arxiv.org/abs/2508.15144.

Quanfeng Lu, Wenqi Shao, Zitao Liu, Lingxiao Du, Fanqing Meng, Boxuan Li, Botong Chen, Siyuan Huang, Kaipeng Zhang, and Ping Luo. Guiodyssey: A comprehensive dataset for cross-app gui navigation on mobile devices, 2025c. URL https://arxiv.org/abs/2406.08451.

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

Liangtai Sun, Xingyu Chen, Lu Chen, Tianle Dai, Zichen Zhu, and Kai Yu. Meta-gui: Towards multi-modal conversational agents on mobile gui, 2022. URL https://arxiv.org/abs/2205.11029.

Kung-Hsiang Huang, Haoyi Qiu, Yutong Dai, Caiming Xiong, and Chien-Sheng Wu. Gui-kv: Efficient gui agents via kv cache with spatio-temporal awareness, 2025e. URL https://arxiv.org/abs/2510.00536.

Haoming Wang, Haoyang Zou, Huatong Song, Jiazhan Feng, Junjie Fang, Junting Lu, Longxiang Liu, Qinyu Luo, Shihao Liang, Shijue Huang, Wanjun Zhong, Yining Ye, Yujia Qin, Yuwen Xiong, Yuxin Song, Zhiyong Wu, Aoyan Li, Bo Li, Chen Dun, Chong Liu, Daoguang Zan, Fuxing Leng, Hanbin Wang, Hao Yu, Haobin Chen, Hongyi Guo, Jing Su, Jingjia Huang, Kai Shen, Kaiyu Shi, Lin Yan, Peiyao Zhao, Pengfei Liu, Qinghao Ye, Renjie Zheng, Shulin Xin, Wayne Xin Zhao, Wen Heng, Wenhao Huang, Wenqian Wang, Xiaobo Qin, Yi Lin, Youbin Wu, Zehui Chen, Zihao Wang, Baoquan Zhong, Xinchun Zhang, Xujing Li, Yuanfan Li, Zhongkai Zhao, Chengquan Jiang, Faming Wu, Haotian Zhou, Jinlin Pang, Li Han, Qi Liu, Qianli Ma, Siyao Liu, Songhua Cai, Wenqi Fu, Xin Liu, Yaohui Wang, Zhi Zhang, Bo Zhou, Guoliang Li, Jiajun Shi, Jiale Yang, Jie Tang, Li Li, Qihua Han, Taoran Lu, Woyu Lin, Xiaokang Tong, Xinyao Li, Yichi Zhang, Yu Miao, Zhengxuan Jiang, Zili Li, Ziyuan Zhao, Chenxin Li, Dehua Ma, Feng Lin, Ge Zhang, Haihua Yang, Hangyu Guo, Hongda Zhu, Jiaheng Liu, Junda Du, Kai Cai, Kuanye Li, Lichen Yuan, Meilan Han, Minchao Wang, Shuyue Guo, Tianhao Cheng, Xiaobo Ma, Xiaojun Xiao, Xiaolong Huang, Xinjie Chen, Yidi Du, Yilin Chen, Yiwen Wang, Zhaojian Li, Zhenzhu Yang, Zhiyuan Zeng, Chaolin Jin, Chen Li, Hao Chen, Haoli Chen, Jian Chen, Qinghao Zhao, and Guang Shi. Ui-tars-2 technical report: Advancing gui agent with multi-turn reinforcement learning, 2025e. URL https://arxiv.org/abs/2509.02544.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments, 2024. URL https://arxiv.org/abs/2404.07972.

OpenAI. Computer-using agent: Introducing a universal interface for ai to interact with the digital world. https://openai.com/index/computer-using-agent/, 2025. URL https://openai.com/index/computer-using-agent/. Accessed: 2025-09-16.

A Conceptual Framework

Following the formalization by Gao et al. [2025], we model the environment E as a partially observable Markov decision process (POMDP). At the beginning of each episode, the environment generates a task $T = (E, I)$, where $I \in \mathcal{I}$ represents user intention drawn from intention space $\mathcal{I}$. The agent π interacts with the environment over horizon $T \in \mathbb{N}$, producing an interleaved observation–action trajectory $\tau = (o_0, a_0, o_1, a_1, \ldots, o_T) \in (\mathcal{O} \times \mathcal{A})^{T+1}$, where $\mathcal{O}$ and $\mathcal{A}$ denote the observation and action spaces respectively. The environment then evaluates performance and provides feedback $r \in \mathbb{R}^k$, which may take the form of step-level signals $r_{\text{step}}^{(t)} = R_{\text{step}}(s_t, a_t, T) \in \mathbb{R}$ for $t \in \{0, \ldots, T\}$, where $s_t \in \mathcal{S}$ denotes the state of the environment at step t, trajectory-level signals $r_{\text{traj}} = R_{\text{traj}}(\tau, T) \in \mathbb{R}^m$, or a combination of both $r = f(r_{\text{step}}^{(0:T)}, r_{\text{traj}})$ where $f : \mathbb{R}^{T+1} \times \mathbb{R}^m \to \mathbb{R}^k$. These signals need not be limited to sparse scalar rewards but can encode structured or adaptive assessments reflecting correctness, efficiency, reasoning depth, or long-term outcomes. This Generation-Execution-Feedback (GEF) Loop $\mathcal{L} = (T_{\text{gen}}, \text{Exec}, \text{Eval}) : \mathcal{I} \to \mathcal{T} \times (\mathcal{O} \times \mathcal{A})^* \times \mathbb{R}^k$, which encompasses task generation, task execution, and feedback, defines the essential mechanics of environments. Repeated iterations drive the accumulation of experience and the progressive evolution of the agent π.

B Evaluation Benchmarks

Previous evaluation studies have typically focused on the intelligence of the agents themselves, but there is a lack of direct measurement indicators for aspects such as adaptability to the environment, interactivity, realism, and robustness. Therefore, most environmental assessments are conducted indirectly, usually by observing the performance of intelligent agents to reflect the quality of the environment. For instance, studies such as TaskCraft [Shi et al., 2025] and AgentScaler [Fang et al., 2025] train the agents through the trajectories generated by the interaction between the environment and the agents, thereby evaluating the environment. The stronger performance of the agents is regarded as an indirect indication of higher environmental quality. Initially, direct measurements of the environment are mainly limited to symbolic or textual environments. Bytesized32 [Wang et al., 2023] proposes specific-task text games and evaluates them using automated metrics in terms of fidelity, validity, specification adherence, and winnability. Text2World [Hu et al., 2025c] benchmarks the generation of symbolic world models, using structural similarity for overall evaluation, and capturing more granular features such as action dynamics through component-level F1 scores. Recent studies have begun to extend the direct assessment to more modalities. VidOSC [Xue et al., 2024] explores the dynamic characteristics of open-world environments. WorldScore [Duan et al., 2025] proposes a unified framework for evaluating world generation. While WorldPrediction [Chen et al., 2025d] focuses on advanced visual reasoning, emphasizing long-term procedural planning and semantic-time abstraction capabilities. Despite these advancements, comprehensive and universal assessment protocols are still scarce, highlighting the need for more generalized and domain-independent methodologies to rigorously and directly evaluate environmental quality beyond the performance metrics of intelligent agents.

C Key Applications

Recent progress in agentic systems, typically built on state-of-the-art LLM families such as GPT [OpenAI, 2025a], Claude [Anthropic, 2025], Gemini [Team et al., 2024], LLaMA [Touvron et al., 2023], and Qwen [Bai et al., 2023], is increasingly driven by interactions with dynamic and multifaceted environments.

Tool-use Environments Tool-use environments expose APIs and function calls as structured action spaces, and many LLMs now natively support function invocation, thereby extending reasoning with external tools [OpenAI, 2023, Anthropic, 2024, Wu et al., 2025b, Mastouri et al., 2025, Luo et al., 2025b, Wang et al., 2025a, Fan et al., 2025].

Coding Environments Coding environments leverage repositories, test frameworks, and IDE integration to support long-horizon programming. Within these settings, systems such as Qwen3-Coder [Qwen Team, 2024] and Claude 4 [Anthropic, 2025] demonstrate reliability in code editing and debugging. ReTool [Feng et al., 2025] further integrates code-interpreter execution into the reasoning loop, enabling agents to exhibit code self-correction and adaptive tool selection.

GUI Environments Web navigation (browser control) environments build on HTML/DOM structures to support tasks such as browsing, form filling, and transactions [Lai et al., 2024, Fosowl, 2024, Significant-Gravitas, 2023]. GUI environments extend these to graphical user interfaces on desktops and mobile devices [Qin et al., 2025, Wang et al., 2025d, Ye et al., 2025c, Lu et al., 2025c, Liu et al., 2024c, Sun et al., 2022, Huang et al., 2025e]. Beyond these, more comprehensive platforms [Wang et al., 2025e, Xie et al., 2024, OpenAI, 2025] integrate terminals, operating systems, applications, and APIs to create more interactive, open-ended, and realistic environments, fostering the development of more advanced computer-use agents (CUAs).

Deep Research Environments To nurture the next generation of more powerful research agents, deep research environments demand stronger long-context reasoning and more robust retrieval capabilities. Systems such as Gemini 1.5 Pro [Google DeepMind, 2024] and OpenAI's deep research agents [OpenAI, 2025b] demonstrate that extended context windows enable sustained, in-depth analysis, while effective retrieval pipelines help mitigate distraction and context dilution.