🔮 Environments Matter: A Survey of Scaling Interactive Experience Collection for LLM Agents

Anonymous ACL submission

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 dynamism, failing to capture the interactive nature of real-world scenarios. A growing consensus is that agents should instead interact directly with environments and learn from experience through reinforcement learning. We formalize this 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 feedback on trajectories, yielding experience for subsequent training. Under this paradigm, environments become the primary engines for experience generation, making their scaling in complexity, dynamism, and interactivity a prerequisite for collecting high-quality trajectories at scale. In this survey, we adopt an environment-centric lens and organize existing advances into a unified taxonomy along the *GEF loop*. Beyond taxonomy, we further analyze underlying implementation frameworks, identify open problems, and outline critical future directions, arguing that the deliberate engineering of interactive environments will be a decisive frontier in the pursuit of generalist agents.

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 (Wei et al., 2025; Qi et al., 2025), tool use (Zhang et al., 2025f; Anthropic, 2024), and deep research (OpenAI, 2025b; Google DeepMind, 2024).

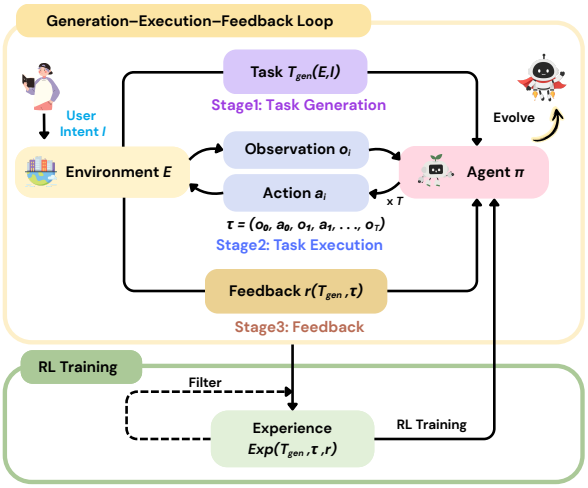

Figure 1: We formally define the *Generation-Execution-Feedback (GEF) loop* in Appendix A.3, where the environment generates tasks from user intent, supports interactive execution, and provides trajectory feedback to create valuable RL experience. This reframes environments from passive containers for agent activity into active generators of adaptive tasks, realistic interactions, and informative feedback.

However, as agent capabilities continue to evolve, achieving intelligence beyond human-level merely through supervised fine-tuning (SFT) pre-trained models on static datasets (Li et al., 2025c; Lu et al., 2025a) becomes increasingly challenging. Such datasets are typically manually annotated or curated under human oversight, which makes them costly and labor-intensive to produce at scale, intrinsically bounded by human-level knowledge, and lacking realism and adaptability. A growing

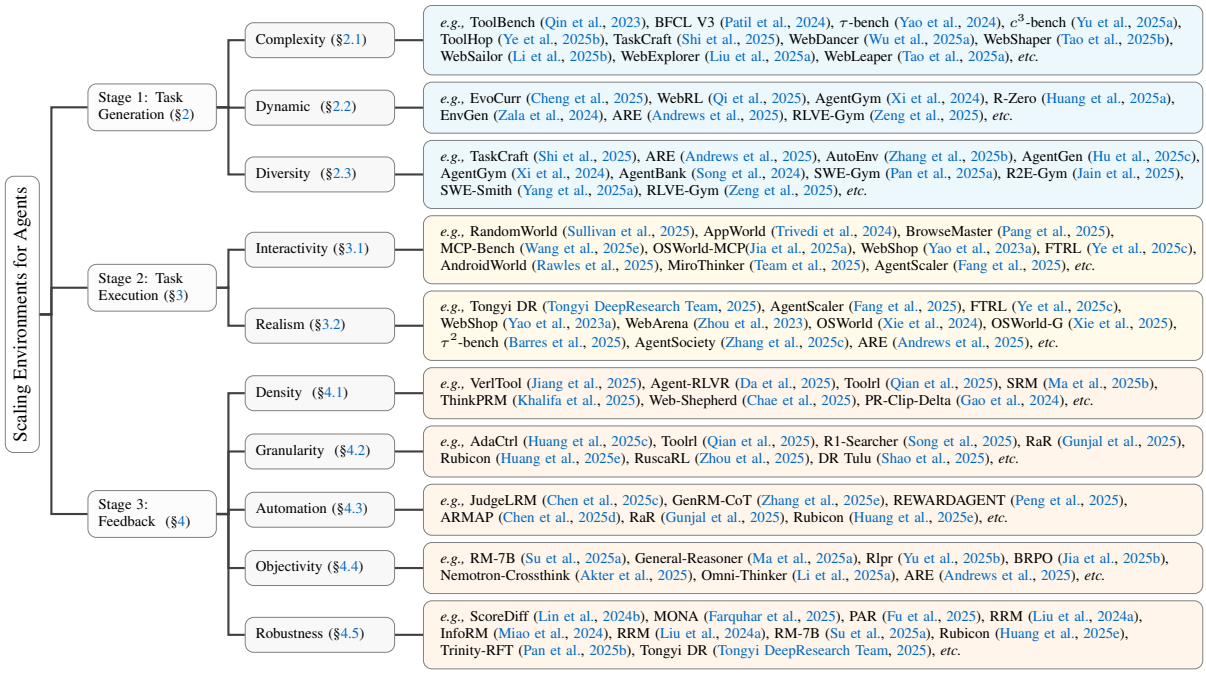

Figure 2: Taxonomy of environment scaling, with dimensions for *Task Generation*, *Task Execution*, and *Feedback* (GEF). Representative works are illustrated as leaves on the branches.

consensus suggests reinforcement learning as a more aligned training paradigm (Tao et al., 2024; Zhang et al., 2025a), where, instead of relying on pre-curated data, agents are encouraged to explore directly within the environment, learn from their accumulated experience, and ultimately awaken new capabilities. In this process, the environment is no longer a mere container for the agent's activities but an active producer of experience data.

We formalize this process as the *Generation-Execution-Feedback (GEF) loop*, illustrated in Figure 1. In each iteration, the environment generates diverse tasks, and the agent takes actions to obtain observations, forming an execution trajectory. The environment then evaluates these trajectories, providing valuable feedback that guides the agent's learning. Over repeated iterations, this loop refines the agent's policy and expands its capabilities, underscoring the growing need to design more complex, realistic, and richly interactive environments that can efficiently generate and collect qualitative experience (CAMEL-AI, 2025).

Recent advances recognize the environment's key role in scaling experience collection and have begun scaling environments from various angles. For instance, AgentGen (Hu et al., 2025c) and GEM (Liu et al., 2025c) design heterogeneous environments to enhance task diversity. R-Zero (Huang et al., 2025a) proposes a co-evolving challenger-solver framework to adaptively increase task difficulty, while ARE (Andrews et al., 2025) introduces an asynchronous event system and large-scale states to support dynamic and realistic exploration. Despite these advancements, a systematic framework that categorizes and analyzes these approaches in a unified way remains absent.

Therefore, we comprehensively investigate current methods and propose a unified taxonomy (Figure 2) aligned with the stages of the *GEF loop*, adopting a pioneering environment-centric perspective. In the **task generation** stage (§2), we emphasize *complexity*, *dynamic*, and *diversity*, which together characterize an environment's ability to generate challenging, adaptive, and diverse tasks continuously. Moving into the **task execution** stage (§3), we highlight *interactivity* and *realism*, since these properties determine the richness and reliability of the collected experience. Finally, in the **feedback** stage (§4), which plays a crucial role in shaping agent behavior and providing guidance, we categorize reward designs along *density*, *granularity*, *automation*, *objectivity*, and *robustness*.

Beyond taxonomy, we surface key recurring trade-offs across stages, like realism vs. cost and automation vs. reliability. We also discuss domain-specific frameworks and implementation challenges in §5. Overall, our survey emphasizes that environments are key to experience collection and offers a unified, environment-centric view of this emerging research area.

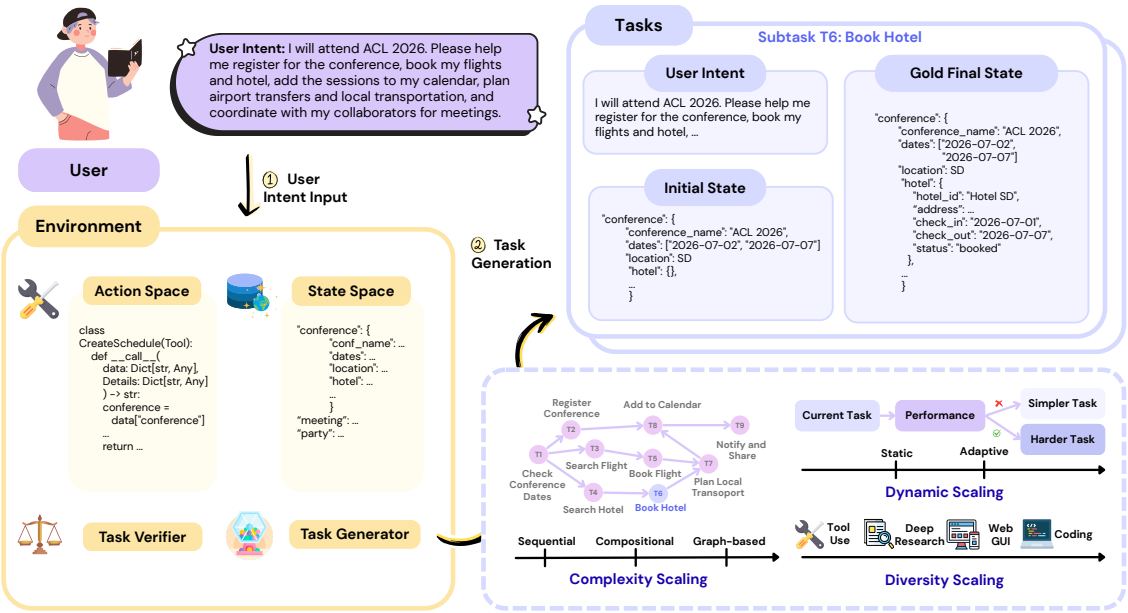

Figure 3: Illustration of scaling in the **task generation** stage. Given a user intent, the environment, equipped with infrastructure (e.g., tools, state space) and a task generator (generation mechanism), automatically creates tasks that are executable by agents. Scaling in this stage covers *complexity scaling*, *dynamic scaling*, and *diversity scaling*.

2 Stage 1: Task Generation

As illustrated by Figure 3, task generation encompasses the process where an environment provides infrastructure (e.g., tools, state space) and a generator (generation mechanism), enabling the automated creation of tasks based on the user intent. To create more challenging tasks that push agents toward their capability boundary, scaling methods in this stage focus on three key areas: *complexity scaling* increases the structural difficulty of individual tasks (§ 2.1), *dynamic scaling* adapts task difficulty over the course of training (§ 2.2), and *diversity scaling* expands the range of task types and environments that agents encounter (§ 2.3).

2.1 Complexity Scaling

Static task 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 scenario is 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; Ye et al., 2025b). More sophisticated tasks exhibit hierarchical or compositional structure, where high-level objectives are decomposed into nested sub-goals, forcing agents to engage in genuine planning rather than surface pattern matching (Shao et al., 2023; Yu et al., 2025a; Shi et al., 2025; Wu et al., 2025b,a). TaskCraft (Shi et al., 2025) operationalizes this by expanding task structure along both depth, through longer sequences of tool executions, and width, through multiple sub-goals per objective. Conditional and graph-based tasks push further still, introducing non-linear branching logic where planning must adapt to intermediate outcomes (Tao et al., 2025b; Li et al., 2025b; Tao et al., 2025a; Liu et al., 2025a; Geng et al., 2025), as reflected in the performance gains shown in Table 2.

2.2 Dynamic Scaling

A natural extension of complexity scaling is dynamic scaling, which adjusts task difficulty based on the agent's evolving performance rather than fixing it statically. A common approach is curriculum learning, which schedules tasks of varying difficulty based on agent performance metrics such as success rate (SR) or progress rate (PR), or targets newly acquired or weaker skills for focused training (Zeng et al., 2025; Liang et al., 2024; Zala et al., 2024; Qi et al., 2025; Xi et al., 2024; Cheng et al., 2025). For instance, EnvGen (Zala et al., 2024) uses an LLM to iteratively generate and adapt embodied environments by feeding back the agent's performance, continuously focusing training on the skills the agent is weakest at. More recent advances introduce a challenger-solver co-evolving pipeline, where the challenger is trained to propose tasks that maximize the solver's uncertainty, ensuring

problems remain at the frontier of the solver's current capability (Huang et al., 2025a). RLVE (Zeng et al., 2025) follows a similar principle by automatically generating verifiable environments and adaptively shifting a sliding window of difficulty to ensure problems remain appropriately challenging throughout training.

🔮 **Future Direction I** Despite the benefits of adaptively changing environments or task difficulties, most current methods are limited to verifiable tasks and require manual intervention to ensure high-quality task generation (Huang et al., 2025a; Zeng et al., 2025). Future designs should develop efficient pipelines for non-verifiable tasks while ensuring the reliability of automated task generation.

2.3 Diversity Scaling

While complexity scaling concerns how hard a single task is structurally and dynamic scaling governs how difficulty evolves over training, diversity scaling addresses what kinds of tasks and environments agents encounter, ensuring the training distribution spans sufficiently varied domains to prevent overfitting to specific action patterns (Hu et al., 2025c; Huang et al., 2025a). This diversity can be pursued at two levels. *Task-level Diversity Scaling* generates tasks with different objectives and complexity within a fixed environment (Huang et al., 2025a), but Andrews et al. (2025) argues that simple tasks in complex environments may be more effective than complex tasks in simple ones, suggesting greater emphasis on *Environment-level Diversity*. As shown in Table 3, training across representative domains (e.g., web, tool use, and coding) enables AgentGym (Xi et al., 2024) and AgentBank (Song et al., 2024) to achieve stronger performance. Recent work therefore explores automatically constructing diverse environments from either pre-defined (Shi et al., 2025; Yang et al., 2025a; Jain et al., 2025; Song et al., 2026) or continuously refined tool and application suites (Hu et al., 2025c; Zhang et al., 2025b). Empirically, scaling the number of environment domains yields consistent generalization gains on held-out benchmarks (Tu et al., 2026), though no single fixed learning strategy generalizes effectively across all environments (Zhang et al., 2025b), suggesting that diversity scaling must be accompanied by adaptive learning algorithms.

3 Stage 2: Task Execution

After task generation, the next crucial phase is task execution, where the agent engages in the interaction loop with the environment, receiving observations and taking actions iteratively. We organize works related into two dimensions: interactivity and realism (Figure 4). *Interactivity* (§ 3.1) refers to how the agent interacts with the environment, while *Realism* (§ 3.2) focuses on making observations more realistic to enhance interaction quality.

3.1 Interactivity

Interactivity concerns whether the environment is executable and whether the agent is blind to the returns of intermediate steps. In non-interactive settings, agents are evaluated against a single static ground-truth solution path, even though there may in fact exist diverse paths to successful task completion (Deng et al., 2023; Kapoor et al., 2024; Rawles et al., 2023; Qin et al., 2023). Consequently, such static supervision fails to credit alternative correct solutions and limits agents' ability to generalize to novel tasks (Sullivan et al., 2025; Xie et al., 2024; Rawles et al., 2025). Recent environment designs start to allow agents to interactively invoke real-world APIs, call functions, generate codes, or interact through GUI (Wang et al., 2024; Yao et al., 2023a; Trivedi et al., 2024; Sullivan et al., 2025) and to adapt subsequent actions based on intermediate results dynamically. *Width scaling* methods focus on enabling parallel tool invocation (Pang et al., 2025) and broadening the interaction dimension (e.g., facilitating flexible switching between GUI operations and tool invocations (Jia et al., 2025a)), while *depth scaling* methods focus on efficient tool-use context management. For example, MiroThinker (Team et al., 2025) removes weakly related tool calls from the context, thereby freeing up capacity to scale up tool-use depth and yielding consistent performance gains across various benchmarks. A bottleneck for such scaling lies in the cost of API calls. To alleviate this issue, several recent advances implement an offline realistic database as the environment (Tongyi DeepResearch Team, 2025; Fang et al., 2025; Ye et al., 2025c), using function calls to read from and write to the database state, thereby keeping the cost of scaling up interactivity within a reasonable budget while preserving some realism.

🔮 **Future Direction II** While scaling interactivity enables richer tool use, it also introduces

problems such as overlong responses, diminishing marginal returns from additional tool calls, and escalating costs (Team et al., 2025). Moreover, efficacy diminishes in scenarios that require combining multiple tools (Jia et al., 2025a). As the emerging paradigm of *thinking with images* (Su et al., 2025b) yields multimodal tool-call returns, future environments must prioritize the efficiency and quality of heterogeneous tool invocation.

3.2 Realism

Realism aims to ensure that observations obtained from the environment remain consistent with the real world. In Tool-Use scenarios, despite the benefits brought by the Model Context Protocol (MCP) (Anthropic, 2024; Luo et al., 2025b; Wang et al., 2025e; Fan et al., 2025), which integrates heterogeneous tools into a unified context and greatly improves the efficiency of tool use, directly calling APIs in real environments still faces challenges of monetary cost and lack of robustness (Song et al., 2023; Wu et al., 2025b; Mastouri et al., 2025; Prabhakar et al., 2025). Previous *Model-based* environment designs use LLMs as a world model (Hao et al., 2023; Ge et al., 2024; Tang et al., 2024) or to simulate the results of tool calls (Qin et al., 2023; Lu et al., 2025b; Sun et al., 2025; Lyu et al., 2026), reducing overhead and avoiding implementation errors but inevitably sacrificing realism. Facing this dilemma, *Offline-Database* methods (Tongyi DeepResearch Team, 2025; Fang et al., 2025; Ye et al., 2025c) make a trade-off by using a snapshot of a real-world database as the interaction environment, thereby preserving relatively realistic simulation. This approach has been scaled further by automatically generating large collections of code-driven, database-backed synthetic environments, yielding more reliable state transitions than LLM simulation while remaining fully scalable (Wang et al., 2026b).

In Computer-Use scenarios, earlier works mainly construct on Q&A over static webpage screenshots (Mialon et al., 2023; Deng et al., 2023; Sun et al., 2022; Kapoor et al., 2024), rather than actually operating in real executable environments. This idealized setting makes agents brittle and prone to failure in real-world conditions (e.g. similar buttons, intrusive popup windows, and heterogeneous user interfaces) that demand context awareness and fine-grained control (Xie et al., 2025). To mitigate these, recent advances instead adopt real executable environments that involve coding (Yang et al., 2023), web browser navigation (Yao et al., 2023a; Zhou et al., 2023), and application control across operating systems (Xie et al., 2024).

Future Direction III Most existing environments follow a ReAct-style paradigm (Yao et al., 2023b), assuming that the environment remains static between an agent's observation and action. However, information arrives as overlapping streams in real world so perception and action should occur concurrently (Andrews et al., 2025). Future designs should place greater emphasis on such asynchronous settings, while also exploring ways to unlock richer scenarios and modalities.

4 Stage 3: Feedback

After the trajectories are collected during the execution stage, the environment evaluates them in the feedback stage. The task, trajectory, and feedback together form the experience for subsequent training. Scaling at this stage focuses on how feedback is provided, including its frequency and richness (*density* § 4.1 and *granularity* § 4.2), its level of automation (*automation* § 4.3), as well as how objectively and reliably it is delivered (*objectivity* § 4.4 and *robustness* § 4.5), as shown in Figure 4. These five dimensions are not independent. Automation enables scalability but risks propagating LLM biases, while denser signals improve guidance yet can amplify noise and invite reward hacking, and granularity often reintroduces the manual bottlenecks that automation aims to eliminate. Navigating these tensions is the central challenge in scaling feedback.

4.1 Density

Feedback density refers to how frequently evaluative signals are provided. *Trajectory-level Outcome-based Rewards* (Jiang et al., 2025; Da et al., 2025; Qian et al., 2025; Wang et al., 2025b) are relatively sparse, contributing to more stable training in tasks like mathematical reasoning and code generation (DeepSeek-AI et al., 2025). However, crediting whether an intermediate step helps or hinders task completion in multi-step tasks remains challenging. To alleviate this issue, *Step-level Process-based Rewards* (Khalifa et al., 2025; Chae et al., 2025; Gao et al., 2024; Ma et al., 2025b; Park et al., 2025) provide denser supervision and more detailed guidance. Nevertheless, scaling up feedback frequency does not necessarily improve performance consistently. Gains are often smaller

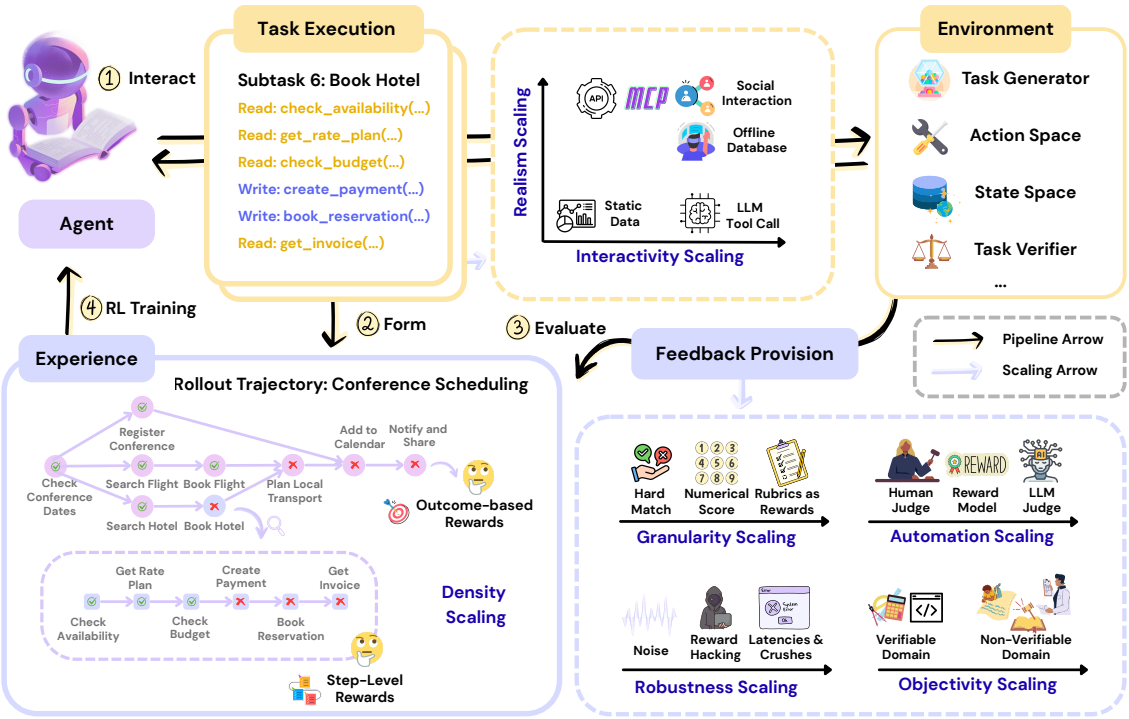

Figure 4: Illustration of scaling in the **task execution** and **feedback** stages. The agent interacts with the environment during task execution, producing action-observation trajectories, which are then evaluated by the environment, forming the experience for subsequent training. Scaling in the task execution stage involves *interactivity* and *realism*, while scaling in the feedback stage encompasses *density*, *granularity*, *automation*, *objectivity*, and *robustness*.

for natural language tasks (e.g., creative writing) than for logically coherent ones (e.g., coding), and it may exacerbate reward hacking, where trivial patterns obtain high rewards (Ma et al., 2025b; Gao et al., 2024). Efforts like PR-Clip-Delta (Gao et al., 2024) use reward differences between adjacent steps, clipping those that exceed a predefined threshold, which helps mitigate repetitive behavior and training instability associated with step-level rewards.

4.2 Granularity

Granularity scaling refers to increasing the level of detail and richness of feedback. Early feedback typically consisted of binary signals (Christiano et al., 2017a; Ibarz et al., 2018) or a single numeric score (Stiennon et al., 2022; Ouyang et al., 2022). At finer granularities, feedback is broken down into structured components, such as scores across multiple criteria like correctness and format, offering more informative guidance (Huang et al., 2025c; Qian et al., 2025; Song et al., 2025). The *Rubrics as Rewards* framework (Gunjal et al., 2025; Huang et al., 2025e; Zhou et al., 2025; Zhang et al., 2025d; Viswanathan et al., 2025) further decomposes task requirements into checklist-style, instance-specific criteria, providing a middle ground between binary correctness signals and broad preference rankings.

Future Direction IV Despite rubrics providing more detailed information than a simple numeric reward, their predefined nature by humans and reliance on internal parametric knowledge may lead agents to exploit internal patterns rather than explore external knowledge (Shao et al., 2025). Future work should support dynamically adjusting rubrics as the boundaries of knowledge evolve.

4.3 Automation

Automation scaling involves the shift of the feedback mechanism from costly and labor-intensive human feedback to automated evaluations. Previous works rely heavily on human annotators (see A.6 for a broader discussion of human roles in agent environments), which poses a major scalability bottleneck (Sheng et al., 2025; Hu et al., 2025a; Christiano et al., 2017b). Recent works adopt LLM-as-a-Judge, train generative reward models, or utilize agentic workflows with external tools to facilitate the feedback process. *LLM-as-a-Judge* (Chen et al., 2025c; Gunjal et al., 2025; Huang et al., 2025e) prompts off-the-shelf LLMs to provide preferences or numeric scores across multiple aspects. *Training-based* methods (Chen et al., 2025d; Zhang et al., 2025e) train a reward model

with synthesized rationales or trajectories to judge whether user intent is fulfilled, significantly outperforming simple prompt engineering but at the cost of limited generalization and higher training costs. *Agentic* methods (Peng et al., 2025) further incorporate external searching tools to help verify factual accuracy and assess instruction-following ability, offering more reliable guidance, though there is still room for improvement in workflow efficiency and reward accuracy.

🔮 **Future Direction V** Automation magnifies risks such as verbosity bias (favoring longer responses), position bias (favoring answers in certain positions), and egocentric bias (preferring self-generated responses) (Zheng et al., 2023). It also introduces reward hacking, where agents exploit loopholes in the reward model to achieve high scores without fulfilling intended goals. Future work should therefore prioritize balancing the trade-off between scalability and reliability as automation scales.

4.4 Objectivity

Objectivity scaling aims to reduce the biases introduced by automated LLM verifiers and the subjectivity inherent in human evaluation. Despite the *Reinforcement Learning with Verifiable Rewards (RLVR)* paradigm (Lambert et al., 2025) achieving great success in some easy-to-verify domains (e.g., math, coding (Shao et al., 2024; DeepSeek-AI et al., 2025)), it still falls short in creative domains (e.g., writing, medical consultation, and policy making), where ground truth is often missing, and verification remains open, making it difficult to reach a unified standard. Recent advances aim to extend RLVR to these hard-to-verify domains (Andrews et al., 2025; Ma et al., 2025a; Akter et al., 2025; Su et al., 2025a; Yu et al., 2025b; Liu et al., 2025b; Gunjal et al., 2025; Huang et al., 2025e; Li et al., 2025a). For instance, BRPO (Jia et al., 2025b) introduces a pairwise generative reward model designed to offer comparative quality scores, avoiding the direct scoring of creativity in writing materials. Works like ARE (Andrews et al., 2025) combine parameter-matching hard checks (e.g., verifying email id) with generative rubric-based soft checks to ensure relatively fair evaluation.

4.5 Robustness

Feedback robustness requires the environment to provide stable and reliable reward signals. *Reward-level Robustness* concerns the stability of the reward itself, since such signals may be inherently noisy or susceptible to hacking (agents learning undesirable and tricky behavioral patterns that obtain high rewards without achieving the intended goals). To mitigate noise, Lin et al. (2024b) and Su et al. (2025a) generate soft probabilistic rewards with generative verifiers. To prevent reward hacking, Farquhar et al. (2025) evaluates the future utility of actions through an overseer, constraining unstable behaviors while preserving explainability. Huang et al. (2025e) develops a defense rubric that penalizes sycophantic praise toward user prompts and overly flattering self-assessments in responses, encouraging the model to produce more substantive content.

Environment-level Robustness is also crucial for agents, as environmental instabilities (e.g., delays, crashes, corrupted tool outputs) may also undermine feedback reliability and degrade the training process. To handle such failures, Trinity-RFT (Pan et al., 2025b) proposes asynchronous inference and retry mechanisms, while Tongyi Deep-Research Team (2025) employs caching, retrying failed calls, and switching to similar providers to prevent corrupted trajectories. These efforts take an encouraging step toward establishing the underlying frameworks that ensure stable feedback.

5 Implementation Considerations

5.1 Domain-Specific Design Considerations

While the GEF stages provide a unified analytical lens, the dominant design bottleneck shifts substantially across domains.

Tool Use and Deep Research Tool-use environments prioritize expanding the width and depth of *interactivity*, which involves supporting deeper tool-call sequences to solve more complex tasks and incorporating more general, versatile tools to handle tasks across a broader range of domains. Deep research scenarios similarly focus on enabling agents to orchestrate *diverse* search and browsing APIs during task generation, as well as *robust* retrieval and reasoning through long contexts. For example, DR Tulu (Shao et al., 2025) optimizes a unified MCP-based tool backend through global caching and an asynchronous process lock, supporting high-concurrency operations. ARE (Andrews et al., 2025) further transforms tool-use benchmarks into separate applications, integrating mobile use and tool use into a unified environment.

Web Navigation and Computer Use Web navigation scenarios center on *realism* and *interactivity*. WebArena (Zhou et al., 2023) delivers websites in self-contained Docker images, avoiding unexpected content or configuration changes during evaluation. WebShop (Yao et al., 2023a) provides a diverse semantic action space covering search queries and button clicks, with a modular architecture that facilitates scaling to new web tasks and domains. VisualWebArena (Koh et al., 2024b) further introduces a Classifieds website built on an open-source CMS, supporting visually grounded tasks such as posting, searching, and commenting. Computer-use scenarios share these priorities while additionally making data collection efficiency and *diversity* integration key concerns, since agents must handle cross-OS tasks and heterogeneous tools. OpenCUA (Wang et al., 2025d) and OS-World (Xie et al., 2024) provide infrastructure that supports efficient task annotation and real-time capture of user interactions, while UI-TARS-2 (Wang et al., 2025a) unifies GUI operations and SDK function calls into a single versatile platform, supporting a variety of task types while maintaining long-lived states across intricate multi-step interactions.

Software Engineering SWE environments place the greatest emphasis on *Realism* and *Robustness*, namely whether agents can actually execute code and whether that execution process can be faithfully reproduced. Docker (Merkel et al., 2014) has become the standard execution substrate in practice, as it ensures cross-machine consistency through standard Dockerfiles while enabling safe, isolated code execution that prevents potential safety issues (Yang et al., 2023; Jimenez et al., 2023; Yang et al., 2025a).

Embodied AI Embodied environments require agents to ground objects in photo-realistic 3D spaces, answer questions, and perform multi-step physical actions, demanding that environments maintain *realism* in grounding and collision avoidance (Weihs et al., 2021; Das et al., 2018). ALFWorld (Shridhar et al., 2020) provides the necessary infrastructure for connecting abstract reasoning with concrete physical execution. More recently, Genie 3 (Parker-Holder and Fruchter, 2025) has advanced real-time *interactivity* and long-horizon consistency in generative world models, pointing toward environments that are dynamically synthesized rather than statically authored.

Cross-Domain Pattern Across these domains, *realism* and *interactivity* emerge as universal priorities, yet the dominant bottleneck differs by modality. Text-grounded domains such as SWE and tool use enjoy cheap, reliable verification via unit tests or exact match, but generating diverse non-trivial tasks at scale without manual curation remains hard. Visually grounded domains face the inverse: rich task diversity is easier to obtain, but constructing realistic and reproducible environments grows increasingly costly. This recurring tension between generation ease and verification cost points to a shared underlying challenge (Wei, 2025).

5.2 Generator-Verifier Asymmetry

The domain patterns above converge on a fundamental cross-cutting challenge (Wei, 2025), namely the mismatch between the intelligence required to *generate or solve* a task and that required to *verify* it. In **easy-to-verify** domains such as mathematics and coding (Wei et al., 2025; Jimenez et al., 2023; Phan et al., 2025), generating high-quality non-trivial tasks is hard while verification via unit tests is cheap, enabling weak-to-strong supervision. In **hard-to-verify** domains such as creative writing and medical consultation (Lin et al., 2024a; Arora et al., 2025), tasks are trivially proposed but verification is subjective and expert-dependent. Crucially, this asymmetry can become a catalyst, because a stronger generator that improves the verifier, which then supervises an even stronger generator, drives self-improving agents (Huang et al., 2025a; Hong et al., 2025; Lu et al., 2025d; Chen et al., 2025b; Wang et al., 2025c, 2026a).

6 Conclusion

In the era of experience, environments, positioned as active experience generators, are central to the evolution of LLM agents. Scaling them toward more complex, dynamic, realistic, and richly interactive worlds is essential for high-quality experience collection. From an environment-centric perspective, our survey is the first to propose a unified taxonomy that organizes representative advances across the three-stage *GEF loop* (task generation, task execution, and feedback), together with implementation frameworks, evaluation benchmarks, and applications. Additionally, we surface key challenges and highlight future frontiers in environmental design, offering insights to guide future research toward advancing agent intelligence.

Limitations

Scaling environments for LLM agents is still an emerging area with unsettled terminology and rapidly changing benchmarks. We therefore adopt a broad environment-centric scope that includes adjacent works on task generation, executable interaction, and feedback design, even when those works do not explicitly use the language of environment scaling. This scope improves coverage but also means that the GEF taxonomy should be interpreted as an analytical lens rather than a set of mutually exclusive classes. We assign works according to their primary bottleneck and discuss representative overlaps in Appendix A.5.

This survey clarifies its selection criteria and thematic boundaries, but it is not intended as a systematic review or meta-analysis with exhaustive search statistics or quantitative effect estimates. The evidence base is heterogeneous: papers differ in target domains, model backbones, training budgets, interaction protocols, and evaluation settings, and some recent systems are reported primarily through preprints, technical reports, or project pages. As a result, our comparison tables are meant to summarize design choices, reported strengths, and stated limitations rather than establish definitive rankings across methods. Given the pace of agentic research, some recent or concurrently released papers may fall outside this snapshot; we will continue to update the survey as the literature evolves.

References

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

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

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

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

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. 2025. Healthbench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*.

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

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

Jonathan Bragg, Mike D'Arcy, Nishant Balepur, Dan Bareket, Bhavana Dalvi, Sergey Feldman, Dany Haddad, Jena D. Hwang, Peter Jansen, Varsha Kishore, Bodhisattwa Prasad Majumder, Aakanksha Naik, Sigal Rahamimov, Kyle Richardson, Amanpreet Singh, Harshit Surana, Aryeh Tiktinsky, Rosni Vasu, Guy Wiener, Chloe Anastasiades, Stefan Candra, Jason Dunkelberger, Dan Emery, Rob Evans, Malachi Hamada, Regan Huff, Rodney Kinney, Matt Latzke, Jaron Lochner, Ruben Lozano-Aguilera, Cecile Nguyen, Smita Rao, Amber Tanaka, Brooke Vlahos, Peter Clark, Doug Downey, Yoav Goldberg, Ashish Sabharwal, and Daniel S. Weld. 2025. Astabench: Rigorous benchmarking of ai agents with a scientific research suite. *Preprint*, arXiv:2510.21652.

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

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. 2025. Webshepherd: Advancing prms for reinforcing web agents. *Preprint*, arXiv:2505.15277.

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

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

Nuo Chen, Zhiyuan Hu, Qingyun Zou, Jiaying Wu, Qian Wang, Bryan Hooi, and Bingsheng He. 2025c. Judgelrm: Large reasoning models as a judge. *Preprint*, arXiv:2504.00050.

Zhenfang Chen, Delin Chen, Rui Sun, Wenjun Liu, and Chuang Gan. 2025d. Scaling autonomous agents via automatic reward modeling and planning. *Preprint*, arXiv:2502.12130.

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

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017a. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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

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

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

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. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

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

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

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

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

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. 2025. Towards general agentic intelligence via environment scaling. *Preprint*, arXiv:2509.13311.

Sebastian Farquhar, Vikrant Varma, David Lindner, David Elson, Caleb Biddulph, Ian Goodfellow, and Rohin Shah. 2025. Mona: Myopic optimization with non-myopic approval can mitigate multi-step reward hacking. *Preprint*, arXiv:2501.13011.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. 2025. Retool: Reinforcement learning for strategic tool use in llms. *Preprint*, arXiv:2504.11536.

Fosowl. 2024. AgenticSeek: A fully local and open-source AI agent for autonomous task execution. GitHub repository. Accessed: 2024-09-02.

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

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

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

Zhiqi Ge, Hongzhe Huang, Mingze Zhou, Juncheng Li, Guoming Wang, Siliang Tang, and Yueting Zhuang. 2024. Worldgpt: Empowering llm as multimodal world model. *Preprint*, arXiv:2404.18202.

Xinyu Geng, Peng Xia, Zhen Zhang, Xinyu Wang, Qiuchen Wang, Ruixue Ding, Chenxi Wang, Jialong Wu, Yida Zhao, Kuan Li, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. 2025. Webwatcher: Breaking new frontier of vision-language deep research agent. *Preprint*, arXiv:2508.05748.

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

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Bing Liu, and Sean Hendryx. 2025. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *Preprint*, arXiv:2507.17746.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. *Preprint*, arXiv:2305.14992.

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

John Howell and Stephanie Ifayemi. 2024. Policy alignment on ai transparency. *Partnership on AI*, 9.

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. 2025a. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *Preprint*, arXiv:2405.11143.

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

Mengkang Hu, Pu Zhao, Can Xu, Qingfeng Sun, Jian-Guang Lou, Qingwei Lin, Ping Luo, and Saravan Rajmohan. 2025c. 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.

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

Kung-Hsiang Huang, Haoyi Qiu, Yutong Dai, Caiming Xiong, and Chien-Sheng Wu. 2025b. Gui-kv: Efficient gui agents via kv cache with spatio-temporal awareness. *Preprint*, arXiv:2510.00536.

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

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

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

Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. 2018. Reward learning from human preferences and demonstrations in atari. *Preprint*, arXiv:1811.06521.

Naman Jain, Jaskirat Singh, Manish Shetty, Liang Zheng, Koushik Sen, and Ion Stoica. 2025. R2e-gym: Procedural environments and hybrid verifiers for scaling open-weights swe agents. *Preprint*, arXiv:2504.07164.

Hongrui Jia, Jitong Liao, Xi Zhang, Haiyang Xu, Tianbao Xie, Chaoya Jiang, Ming Yan, Si Liu, Wei Ye, and Fei Huang. 2025a. Osworld-mcp: Benchmarking mcp tool invocation in computer-use agents. *Preprint*, arXiv:2510.24563.

Ruipeng Jia, Yunyi Yang, Yongbo Gai, Kai Luo, Shihao Huang, Jianhe Lin, Xiaoxi Jiang, and Guanjun Jiang. 2025b. Writing-zero: Bridge the gap between non-verifiable tasks and verifiable rewards. *Preprint*, arXiv:2506.00103.

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

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

Raghav Kapoor, Yash Parag Butala, Melisa Russak, Jing Yu Koh, Kiran Kamble, Waseem Alshikh, and Ruslan Salakhutdinov. 2024. Omniact: A dataset and benchmark for enabling multimodal generalist autonomous agents for desktop and web. *Preprint*, arXiv:2402.17553.

Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. 2025. Process reward models that think. *Preprint*, arXiv:2504.16828.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. 2024a. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 881–905, Bangkok, Thailand. Association for Computational Linguistics.

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

Hanyu Lai, Xiao Liu, Iat Long Iong, Shuntian Yao, Yuxuan Chen, Pengbo Shen, Hao Yu, Hanchen Zhang, Xiaohan Zhang, Yuxiao Dong, et al. 2024. 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.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. 2025. Tulu 3: Pushing frontiers in open language model post-training. *Preprint*, arXiv:2411.15124.

Dawei Li, Zhen Tan, Peijia Qian, Yifan Li, Kumar Satvik Chaudhary, Lijie Hu, and Jiayi Shen. 2024. Smoa: Improving multi-agent large language models with sparse mixture-of-agents. *Preprint*, arXiv:2411.03284.

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

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. 2025b. Websailor: Navigating super-human reasoning for web agent. *Preprint*, arXiv:2507.02592.

Yuetai Li, Huseyin A Inan, Xiang Yue, Wei-Ning Chen, Lukas Wutschitz, Janardhan Kulkarni, Radha Poovendran, Robert Sim, and Saravan Rajmohan. 2025c. Simulating environments with reasoning models for agent training. *Preprint*, arXiv:2511.01824.

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

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

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

Junteng Liu, Yunji Li, Chi Zhang, Jingyang Li, Aili Chen, Ke Ji, Weiyu Cheng, Zijia Wu, Chengyu Du, Qidi Xu, Jiayuan Song, Zhengmao Zhu, Wenhu Chen, Pengyu Zhao, and Junxian He. 2025a. Webexplorer: Explore and evolve for training long-horizon web agents. *Preprint*, arXiv:2509.06501.

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

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

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. 2024b. Autoglm: Autonomous foundation agents for guis. *Preprint*, arXiv:2411.00820.

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. 2025c. Gem: A gym for agentic llms. *Preprint*, arXiv:2510.01051.

Fanbin Lu, Zhisheng Zhong, Ziqin Wei, Shu Liu, Chi-Wing Fu, and Jiaya Jia. 2025a. Steve: A step verification pipeline for computer-use agent training. *Preprint*, arXiv:2503.12532.

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. 2025b. Toolsandbox: A stateful, conversational, interactive evaluation benchmark for llm tool use capabilities. *Preprint*, arXiv:2408.04682.

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

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

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. 2025a. Large language model agent: A survey on methodology, applications and challenges. *Preprint*, arXiv:2503.21460.

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

Yuanjie Lyu, Chengyu Wang, Lei Shen, Jun Huang, and Tong Xu. 2026. Mock worlds, real skills: Building small agentic language models with synthetic tasks, simulated environments, and rubric-based rewards. *Preprint*, arXiv:2601.22511.

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

Yiran Ma, Zui Chen, Tianqiao Liu, Mi Tian, Zhuo Liu, Zitao Liu, and Weiqi Luo. 2025b. What are step-level reward models rewarding? counterintuitive findings from mcts-boosted mathematical reasoning. *Preprint*, arXiv:2412.15904.

Meriem Mastouri, Emna Ksontini, and Wael Kessentini. 2025. Making rest apis agent-ready: From openapi to mcp servers for tool-augmented llms. *Preprint*, arXiv:2507.16044.

Dirk Merkel et al. 2014. Docker: lightweight linux containers for consistent development and deployment. *Linux j*, 239(2):2.

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

Yuchun Miao, Sen Zhang, Liang Ding, Rong Bao, Lefei Zhang, and Dacheng Tao. 2024. Inform: Mitigating reward hacking in rlhf via information-theoretic reward modeling. *Preprint*, arXiv:2402.09345.

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

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

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

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

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. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Jiayi Pan, Xingyao Wang, Graham Neubig, Navdeep Jaitly, Heng Ji, Alane Suhr, and Yizhe Zhang. 2025a. Training software engineering agents and verifiers with swe-gym. *Preprint*, arXiv:2412.21139.

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. 2025b. Trinity-rft: A general-purpose and unified framework for reinforcement fine-tuning of large language models. *Preprint*, arXiv:2505.17826.

Xianghe Pang, Shuo Tang, Rui Ye, Yuwen Du, Yaxin Du, and Siheng Chen. 2025. Browsemaster: Towards scalable web browsing via tool-augmented programmatic agent pair. *Preprint*, arXiv:2508.09129.

Young-Jin Park, Kristjan Greenewald, Kaveh Alim, Hao Wang, and Navid Azizan. 2025. Know what you don't know: Uncertainty calibration of process reward models. *Preprint*, arXiv:2506.09338.

Jack Parker-Holder and Shlomi Fruchter. 2025. Genie 3: A new frontier for world models.

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

Hao Peng, Yunjia Qi, Xiaozhi Wang, Zijun Yao, Bin Xu, Lei Hou, and Juanzi Li. 2025. Agentic reward modeling: Integrating human preferences with verifiable correctness signals for reliable reward systems. *Preprint*, arXiv:2502.19328.

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

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. 2025. Apigen-mt: Agentic pipeline for multi-turn data generation via simulated agent-human interplay. *Preprint*, arXiv:2504.03601.

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

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

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

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

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. 2025. Ui-tars: Pioneering automated gui interaction with native agents. *Preprint*, arXiv:2501.12326.

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

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, Daniel Toyama, Robert Berry, Divya Tyamagundlu, Timothy Lillicrap, and Oriana Riva. 2025. Androidworld: A dynamic benchmarking environment for autonomous agents. *Preprint*, arXiv:2405.14573.

Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. 2023. Android in the wild: A large-scale dataset for android device control. *Preprint*, arXiv:2307.10088.

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

Rulin Shao, Akari Asai, Shannon Zejiang Shen, Hamish Ivison, Varsha Kishore, Jingming Zhuo, Xinran Zhao, Molly Park, Samuel G. Finlayson, David Sontag, Tyler Murray, Sewon Min, Pradeep Dasigi, Luca Soldaini, Faeze Brahman, Wen tau Yih, Tongshuang Wu, Luke Zettlemoyer, Yoon Kim, Hannaneh Hajishirzi, and Pang Wei Koh. 2025. Dr tulu: Reinforcement learning with evolving rubrics for deep research. *Preprint*, arXiv:2511.19399.

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

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

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

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

Significant-Gravitas. 2023. Auto-GPT: An Autonomous GPT-4 Experiment. GitHub repository. Accessed: 2023-09-02.

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

Xiaoshuai Song, Haofei Chang, Guanting Dong, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. 2026. Envscaler: Scaling tool-interactive environments for llm agent via programmatic synthesis. *Preprint*, arXiv:2601.05808.

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

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. 2023. Restgpt: Connecting large language models with real-world restful apis. *Preprint*, arXiv:2306.06624.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2022. Learning to summarize from human feedback. *Preprint*, arXiv:2009.01325.

Yi Su, Dian Yu, Linfeng Song, Juntao Li, Haitao Mi, Zhaopeng Tu, Min Zhang, and Dong Yu. 2025a. Crossing the reward bridge: Expanding rl with verifiable rewards across diverse domains. *Preprint*, arXiv:2503.23829.

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

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

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. 2025. Zerosearch: Incentivize the search capability of llms without searching. *Preprint*, arXiv:2505.04588.

Liangtai Sun, Xingyu Chen, Lu Chen, Tianle Dai, Zichen Zhu, and Kai Yu. 2022. Meta-gui: Towards multi-modal conversational agents on mobile gui. *Preprint*, arXiv:2205.11029.

Hao Tang, Darren Key, and Kevin Ellis. 2024. Worldcoder, a model-based llm agent: Building world models by writing code and interacting with the environment. *Advances in Neural Information Processing Systems*, 37:70148–70212.

Xiangru Tang, Qiao Jin, Kunlun Zhu, Tongxin Yuan, Yichi Zhang, Wangchunshu Zhou, Meng Qu, Yilun Zhao, Jian Tang, Zhuosheng Zhang, Arman Cohan, Zhiyong Lu, and Mark Gerstein. 2025. Risks of ai scientists: Prioritizing safeguarding over autonomy. *Preprint*, arXiv:2402.04247.

Zhengwei Tao, Ting-En Lin, Xiancai Chen, Hangyu Li, Yuchuan Wu, Yongbin Li, Zhi Jin, Fei Huang, Dacheng Tao, and Jingren Zhou. 2024. A survey on self-evolution of large language models. *Preprint*, arXiv:2404.14387.

Zhengwei Tao, Haiyang Shen, Baixuan Li, Wenbiao Yin, Jialong Wu, Kuan Li, Zhongwang Zhang, Huifeng Yin, Rui Ye, Liwen Zhang, Xinyu Wang, Pengjun Xie, Jingren Zhou, and Yong Jiang. 2025a. Webleaper: Empowering efficiency and efficacy in webagent via enabling info-rich seeking. *Preprint*, arXiv:2510.24697.

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. 2025b. Webshaper: Agentically data synthesizing via information-seeking formalization. *Preprint*, arXiv:2507.15061.

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

MiroMind Team, Song Bai, Lidong Bing, Carson Chen, Guanzheng Chen, Yuntao Chen, Zhe Chen, Ziyi Chen, Jifeng Dai, Xuan Dong, Wenhan Dou, Yue Deng, Yunjie Fu, Junqi Ge, Chenxia Han, Tammy Huang, Zhenhang Huang, Jerry Jiao, Shilei Jiang, Tianyu Jiao, Xiaoqi Jian, Lei Lei, Ruilin Li, Ryan Luo, Tiantong Li, Xiang Lin, Ziyuan Liu, Zhiqi Li, Jie Ni, Qiang Ren, Pax Sun, Shiqian Su, Chenxin Tao, Bin Wang, Hellen Wang, Haonan Wang, James Wang, Jin Wang, Jojo Wang, Letian Wang, Shizun Wang, Weizhi Wang, Zixuan Wang, Jinfan Xu, Sen Xing, Chenyu Yang, Hai Ye, Jiaheng Yu, Yue Yu, Muyan Zhong, Tianchen Zhao, Xizhou Zhu, Yanpeng Zhou, Yifan Zhang, and Zhi Zhu. 2025. Mirothinker: Pushing the performance boundaries of open-source research agents via model, context, and interactive scaling. *Preprint*, arXiv:2511.11793.

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

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

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta,

Ashish Sabharwal, and Niranjan Balasubramanian. 2024. Appworld: A controllable world of apps and people for benchmarking interactive coding agents. *Preprint*, arXiv:2407.18901.

Dunwei Tu, Hongyan Hao, Hansi Yang, Yihao Chen, Yi-Kai Zhang, Zhikang Xia, Yu Yang, Yueqing Sun, Xingchen Liu, Furao Shen, Qi Gu, Hui Su, and Xunliang Cai. 2026. Scaleenv: Scaling environment synthesis from scratch for generalist interactive tool-use agent training. *Preprint*, arXiv:2602.06820.

Vijay Viswanathan, Yanchao Sun, Shuang Ma, Xiang Kong, Meng Cao, Graham Neubig, and Tongshuang Wu. 2025. Checklists are better than reward models for aligning language models. *Preprint*, arXiv:2507.18624.

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. 2025a. Ui-tars-2 technical report: Advancing gui agent with multi-turn reinforcement learning. *Preprint*, arXiv:2509.02544.

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

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

Shaobo Wang, Zhengbo Jiao, Zifan Zhang, Yilang Peng, Xu Ze, Boyu Yang, Wei Wang, Hu Wei, and Linfeng Zhang. 2025c. Socratic-zero : Bootstrapping reasoning via data-free agent co-evolution. *Preprint*, arXiv:2509.24726.

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

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. 2025d. Opencua: Open foundations for computer-use agents. *Preprint*, arXiv:2508.09123.

Yinjie Wang, Tianbao Xie, Ke Shen, Mengdi Wang, and Ling Yang. 2026a. Rlanything: Forge environment, policy, and reward model in completely dynamic rl system. *Preprint*, arXiv:2602.02488.

Zhaoyang Wang, Canwen Xu, Boyi Liu, Yite Wang, Siwei Han, Zhewei Yao, Huaxiu Yao, and Yuxiong He. 2026b. Agent world model: Infinity synthetic environments for agentic reinforcement learning. *Preprint*, arXiv:2602.10090.

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

Jason Wei. 2025. Asymmetry of verification and verifier's rule. Blog post. 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. 2025. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*.

Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. 2021. Visual room rearrangement. *Preprint*, arXiv:2103.16544.

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. 2025a. Webdancer: Towards autonomous information seeking agency. *Preprint*, arXiv: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. 2025b.

Webwalker: Benchmarking llms in web traversal. *Preprint*, arXiv:2501.07572.

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

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

Tianbao Xie, Jiaqi Deng, Xiaochuan Li, Junlin Yang, Haoyuan Wu, Jixuan Chen, Wenjing Hu, Xinyuan Wang, Yuhui Xu, Zekun Wang, Yiheng Xu, Junli Wang, Doyen Sahoo, Tao Yu, and Caiming Xiong. 2025. Scaling computer-use grounding via user interface decomposition and synthesis. *Preprint*, arXiv:2505.13227.

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

Zihui Xue, Kumar Ashutosh, and Kristen Grauman. 2024. 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.

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

John Yang, Akshara Prabhakar, Karthik Narasimhan, and Shunyu Yao. 2023. Intercode: Standardizing and benchmarking interactive coding with execution feedback. *Preprint*, arXiv:2306.14898.

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. 2025b. Oasis: Open agent social interaction simulations with one million agents. *Preprint*, arXiv:2411.11581.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2023a. Webshop: Towards scalable real-world web interaction with grounded language agents. *Preprint*, arXiv:2207.01206.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

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. 2025a. Mobile-agent-v3: Fundamental agents for gui automation. *Preprint*, arXiv:2508.15144.

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. 2025b. Toolhop: A query-driven benchmark for evaluating large language models in multi-hop tool use. *Preprint*, arXiv:2501.02506.

Junjie Ye, Changhao Jiang, Zhengyin Du, Yufei Xu, Xuesong Yao, Zhiheng Xi, Xiaoran Fan, Qi Zhang, Tao Gui, Xuanjing Huang, and Jiecao Chen. 2025c. Feedback-driven tool-use improvements in large language models via automated build environments. *Preprint*, arXiv:2508.08791.

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

Li S. Yifei, Allen Chang, Chaitanya Malaviya, and Mark Yatskar. 2025. Researchqa: Evaluating scholarly question answering at scale across 75 fields with survey-mined questions and rubrics. *Preprint*, arXiv:2509.00496.

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

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

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

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. 2023. Agenttuning: Enabling generalized agent abilities for llms. *Preprint*, arXiv:2310.12823.

Zhiyuan Zeng, Hamish Ivison, Yiping Wang, Lifan Yuan, Shuyue Stella Li, Zhuorui Ye, Siting Li, Jacqueline He, Runlong Zhou, Tong Chen, Chenyang Zhao, Yulia Tsvetkov, Simon Shaolei Du, Natasha Jaques, Hao Peng, Pang Wei Koh, and Hannaneh Hajishirzi. 2025. Rlve: Scaling up reinforcement learning for

language models with adaptive verifiable environments. *Preprint*, arXiv:2511.07317.

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. 2025a. The landscape of agentic reinforcement learning for llms: A survey. *Preprint*, arXiv:2509.02547.

Jiayi Zhang, Yiran Peng, Fanqi Kong, Yang Cheng, Yifan Wu, Zhaoyang Yu, Jinyu Xiang, Jianhao Ruan, Jinlin Wang, Maojia Song, HongZhang Liu, Xiangru Tang, Bang Liu, Chenglin Wu, and Yuyu Luo. 2025b. Autoenv: Automated environments for measuring cross-environment agent learning. *Preprint*, arXiv:2511.19304.

Jun Zhang, Yuwei Yan, Junbo Yan, Zhiheng Zheng, Jinghua Piao, Depeng Jin, and Yong Li. 2025c. 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.

Junkai Zhang, Zihao Wang, Lin Gui, Swarnashree Mysore Sathyendra, Jaehwan Jeong, Victor Veitch, Wei Wang, Yunzhong He, Bing Liu, and Lifeng Jin. 2025d. Chasing the tail: Effective rubric-based reward modeling for large language model post-training. *Preprint*, arXiv:2509.21500.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2025e. Generative verifiers: Reward modeling as next-token prediction. *Preprint*, arXiv:2408.15240.

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

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

Jingchen Zhao and Beatriz Gómez Fariñas. 2023. Artificial intelligence and sustainable decisions. *European Business Organization Law Review*, 24(1):1–39.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

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

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. 2025. Breaking the exploration bottleneck: Rubric-scaffolded reinforcement learning for general llm reasoning. *Preprint*, arXiv:2508.16949.

A Conceptual and Analytical Framework

A.1 Scaling Laws for LLM Agents

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.

A.2 Generator-Verifier Asymmetry

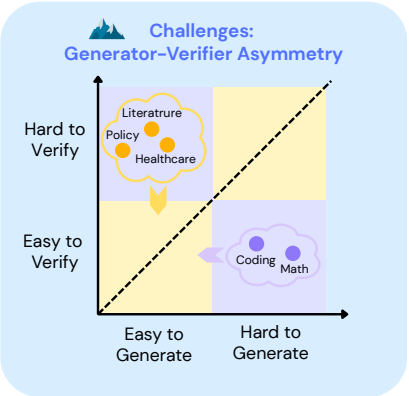

Figure 5: Generator-verifier asymmetry challenge.

Many real-world tasks exhibit *Generator-Verifier Asymmetry* (Wei, 2025). The core issue is that generating or executing a task may require a different level of intelligence from verifying its outcome. This distinction maps naturally to the GEF loop. Task generation and task execution rely on the generator side, while feedback relies on the verifier side. From this view, scaling environments requires progress on both axes, as illustrated in Figure 5.

Current RL progress mainly exploits domains where verification is easier than generation. Mathematics and programming are representative cases (Wei et al., 2025; Jimenez et al., 2023; Phan et al., 2025). In these settings, producing a continual stream of high-quality tasks is difficult, but checking an answer can often be done with exact match or unit tests. This enables weak-to-strong supervision because a simple verifier can train a stronger solver. The opposite regime appears in creative writing, policy-making, and healthcare (Lin et al., 2024a; Arora et al., 2025). These tasks are easy to propose but difficult to judge, since feedback may require expert knowledge, subjective preference, or long-term outcome assessment. This harder-to-verify regime makes environment scaling more difficult, but it also points to an important direction. Future systems may use strong generators to improve verifiers, allowing the two sides to co-evolve and support agent self-evolution (Huang et al., 2025a; Hong et al., 2025; Lu et al., 2025d; Chen et al., 2025b; Wang et al., 2025c).

A.3 Conceptual Framework

Following the formalization by Gao et al. (2025), we model the environment E as a partially observable Markov decision process (POMDP)[1]. At the beginning of each episode, the environment generates a task $T_{\text{gen}} = (E, I)$, 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 $r_{\text{step}}^{(t)} = R_{\text{step}}(s_t, a_t, T_{\text{gen}}) \in \mathbb{R}$ for $t \in \{0, \ldots, T\}$, where $s_t \in \mathcal{S}$ denotes the state of the environment at step t, trajectory-level signals $r_{\text{traj}} = R_{\text{traj}}(\tau, T_{gen}) \in \mathbb{R}^m$, or a combination of both $r_{\text{comb}} = f(r_{\text{step}}^{(0:T)}, r_{\text{traj}})$ 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* encompasses task generation, task execution, and feedback. Repeated iterations drive the accumulation of experience, denoted as $Exp = (T_{\text{gen}}, \tau, r) : \mathcal{I} \to \mathcal{T} \times (\mathcal{O} \times \mathcal{A})^* \times \mathbb{R}^k$, and the progressive evolution of the agent π.

[1]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.

A.4 Survey Methodology

This survey focuses on the role of environments in collecting interactive experience for LLM-based agents. We considered papers and technical reports from 2023 onward, gathered from arXiv, ACL Anthology, OpenReview, major NLP and ML venues, and references cited by representative works. Search terms included combinations of "LLM agent", "interactive environment", "tool use", "web agent", "GUI agent", "computer use", "software engineering agent", "process reward", "verifiable reward", "RLVR", and "self-evolution".

Inclusion requires that a work contributes at least one GEF component: generating agent-executable tasks, supporting action-observation interaction, or providing feedback for trajectories. We excluded static supervised datasets, pure agent-architecture papers, and traditional RL environments not used for LLM-based agents. For works spanning multiple GEF dimensions, we assign them to the component they primarily scale and note secondary dimensions where relevant.

A.5 Interpreting Overlapping GEF Dimensions

Many environment-scaling methods affect more than one part of the GEF loop. The taxonomy in Figure 2 should therefore be read as an analytical lens rather than a set of exclusive classes. We assign each work according to the bottleneck it addresses most directly, while noting secondary connections when they matter.

R-Zero (Huang et al., 2025a) produces increasingly difficult tasks, so it is related to task complexity. Its central mechanism is the challenger-solver

process that adapts task difficulty to the solver's capability, so we discuss it under dynamic scaling. WebArena (Zhou et al., 2023) and VisualWebArena (Koh et al., 2024b) support interactive web navigation. Their main contribution is the construction of self-contained web environments with persistent states and realistic observations, so they are discussed under realism. AppWorld (Trivedi et al., 2024) combines executable API interaction with database-state evaluation. We place it under executable environments because its primary contribution is a controllable app-world backend for agent interaction. Web-Shepherd (Chae et al., 2025) operates on web trajectories, but it does not introduce a new web environment. Its main contribution is process-level supervision for long-horizon web tasks, so it is best understood as feedback scaling.

These cases show that overlap is expected. Task-generation dimensions mainly modify T_{gen}, task-execution dimensions mainly shape the trajectory τ, and feedback dimensions mainly modify the evaluator or reward signal r.

A.6 Human-in-the-loop Environments

Human involvement in agent environments is broader than post-hoc annotation. In the generation stage, humans specify goals, constraints, and preferences that may be underspecified or change over time. In the execution stage, humans may act as users or collaborators whose responses affect the agent's next actions and the environment state. For example, τ^2-bench (Barres et al., 2025) models a dual-control service environment in which both the user and the agent can interact with tools and update the backend state. This setting makes human participation part of the environment dynamics rather than an external labeling step.

The value of human involvement depends on the task setting. Fully automated environments are more suitable when success can be checked by tests, database states, exact answers, or executable rules, since they are cheaper, reproducible, and easier to scale for RL rollouts. Systems such as AppWorld (Trivedi et al., 2024) and SWE-Gym (Pan et al., 2025a) illustrate this case by using database states or tests as scalable checks for task completion. Human-in-the-loop designs are more useful when goals are ambiguous, preferences are subjective, policies are safety-sensitive, or social interaction changes the task itself. Feedback design often benefits from combining both sources. Agentic Reward Modeling (Peng et al., 2025) illustrates this hybrid direction by integrating human preference signals with verifiable correctness checks. Scalable environments should therefore use automated verifiers for frequent objective checks and reserve human input for intent specification, policy boundaries, preference judgment, and audits of ambiguous trajectories.

B Cross-Method Comparisons

We provide three complementary comparisons rather than a single leaderboard. Table 1 compares representative works by design motivation, affected GEF component, domain, environment substrate, and feedback or evaluation signal; Tables 2 and 3 further summarize narrower reported attributes for task complexity and environment diversity under each paper's original setting.

The *Domain*, *Substrate*, and *Signal* columns also make modality-specific infrastructure differences explicit: tool-use systems rely on API schemas and tool-call validation, SWE systems on repositories and tests, web/GUI systems on persistent browser or OS states, and app/mobile systems on stateful backends and executable checks. This comparison therefore treats environment scaling as a question of state representation, action interface, and verification mechanism, rather than only task quantity or benchmark size.

C Additional Environment Design Topics

C.1 Multi-Agent Realism

When extending to the **multi-agent setting**, where agents may coordinate or compete, each agent's behavior naturally becomes part of others' environment (Tran et al., 2025; Qian et al., 2024; Li et al., 2024; Zhang et al., 2024; Barres et al., 2025). Realism in this regime therefore requires environment designs that support agent–agent interactions in addition to agent–tool interactions. This highlights the need to build robust underlying infrastructure and communication mechanisms, which provide the necessary foundation for complex social and economic phenomena (e.g., information diffusion, opinion polarization, and herding effects (Yang et al., 2025b; Zhang et al., 2025c)) to emerge and be meaningfully simulated as the number of agents scales. For instance, Zhang et al. (2025c) adopts the MQTT communication protocol to support asynchronous decision making among autonomous agents, while Andrews et al. (2025) decouples agent and environment clocks

Work	Motive	GEF	Domain	Substrate	Signal
ToolBench (Qin et al., 2023)	Task/Data	G	Tool	APIs	ToolEval
TaskCraft (Shi et al., 2025)	Task/Data	G	Tool	Synth. Tools	Verifier
SYNTHAGENT (Lyu et al., 2026)	Task/Data	G+E+F	Tool	Mock Tools	Rubric
SWE-smith (Yang et al., 2025a)	Task/Data	G	SWE	Repos	Tests
WebShaper (Tao et al., 2025b)	Task/Data	G	Web	Web Graph	QA Bench
WebSailor (Li et al., 2025b)	Task/Data	G	Web	Web Graph	QA Bench
AgentGym (Xi et al., 2024)	Task/Data	G+E	Multi	Env Suite	Rewards
EnvGen (Zala et al., 2024)	Task/Data	G	Embodied	Sim	Success
R-Zero (Huang et al., 2025a)	Task/Data	G	Reasoning	Self-play	Verifier
EnvScaler (Song et al., 2026)	Exec. Env.	G+E+F	Tool	Prog. Env.	State
ScaleEnv (Tu et al., 2026)	Exec. Env.	G+E+F	Tool	Tool DB	Rule
Agent World Model (Wang et al., 2026b)	Exec. Env.	G+E+F	Tool	SQL/MCP	Judge
WebArena (Zhou et al., 2023)	Exec. Env.	E+F	Web	Web Apps	State
VisualWebArena (Koh et al., 2024b)	Exec. Env.	E+F	Web	Web Apps	State
OSWorld (Xie et al., 2024)	Exec. Env.	E+F	GUI	VM	Script
AndroidWorld (Rawles et al., 2025)	Exec. Env.	E+F	Mobile	Android	State
AppWorld (Trivedi et al., 2024)	Exec. Env.	E+F	App	App DB	Tests
τ-bench (Yao et al., 2024)	Exec. Env.	E+F	Service	User Sim	State
SWE-Gym (Pan et al., 2025a)	Exec. Env.	E+F	SWE	Repos	Tests
ARE (Andrews et al., 2025)	Exec. Env.	E+F	Mobile	Async Apps	Verifier
Web-Shepherd (Chae et al., 2025)	Feedback	F	Web	Traj.	PRM
ToolRL (Qian et al., 2025)	Feedback	F	Tool	Traj.	Rule/RM
RLAnything (Wang et al., 2026a)	Feedback	E+F	Multi	Adaptive Env.	RM
Agent-RLVR (Da et al., 2025)	Feedback	F	SWE	Repos	Tests
Agentic RM (Peng et al., 2025)	Feedback	F	Open	Agent Tools	RM/Judge

Table 1: Cross-method comparison using compact categorical labels. *Motive* indicates the primary design motivation: *Task/Data* = task or data scaling, *Exec. Env.* = construction of executable domain environments, and *Feedback* = feedback or reward design. *GEF* denotes the main or strongly affected loop components, where G means generation, E means execution, and F means feedback. *Domain* identifies the task setting; *Substrate* summarizes the main environment carrier, such as APIs, repositories, web apps, virtual machines, trajectories, or agent tools; and *Signal* summarizes the primary evaluation, verifier, reward, or supervision signal reported by the original work.

and treats other agents' activities as independent events, thereby allowing the world state to evolve asynchronously and supporting the collection of more realistic experiential data.

C.2 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., 2025b) 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

Method	Skill Dimension	Task Structure	Task Complexity		Multi-Turn	Performance
			Width	Depth		
ToolBench (Qin et al., 2023)	Tool Use	Sequential	1	-	×	-
τ-bench (Yao et al., 2024)	Tool Use	Sequential	1	-, 30	✓	-
BFCL V3 (Patil et al., 2024)	Tool Use	Sequential	1	-, 20	✓	-
c^3-bench (Yu et al., 2025a)	Tool Use	Compositional	1∼9	1∼4, 10	✓	-
TaskCraft (Shi et al., 2025)	Tool Use	Compositional	2	1∼4, 7	×	-
WebRL (Qi et al., 2025)	Web	Compositional	1∼4	-, >10	✓	-
WebDancer-32B (Wu et al., 2025a)	Web	Compositional	-	2∼4, 10	×	40.7
WebShaper-32B (Tao et al., 2025b)	Web	Graph-Based	-	3∼7, 30	×	52.4
WebSailor-32B (Li et al., 2025b)	Web	Graph-Based	-	2∼8, 30	×	53.2
WebExplorer-8B (Li et al., 2025b)	Web	Graph-Based	-	0∼6, 100	×	50.0

Table 2: Comparative Analysis of Task Complexity and Performance Across Methods. *Width* denotes the number of requirements that must be satisfied within a single task. *Depth* denotes the number of tool calls/steps per task, where the first value indicates the range of steps taken by the majority of tasks and the second value gives the maximum steps. *Performance* reports GAIA (Mialon et al., 2023) scores (if available). *Multi-Turn* indicates whether interaction between the user and the agent is multi-turn. A dash indicates that the quantity is not reported by the original paper, not applicable to the method, or not directly comparable under the table definition. Unless otherwise specified, performance numbers are taken from the cited papers rather than re-run experiments or live leaderboards.

Method	Modality	Skill Dimension	# of Envs	# of Tasks	# of Trajs	Source	Performance		
							SWE	ALF	WS
SWE-Gym-32B (Pan et al., 2025a)	Text	Code;	11	2,438	491	Real	20.6	-	-
R2E-Gym-32B (Jain et al., 2025)	Text	Code;	13	8,135	3,321	Synth	34.4	-	-
SWE-Smith-32B (Yang et al., 2025a)	Text	Code;	128	50,137	5,016	Both	40.2	-	-
AgentLM-13B (Zeng et al., 2023)	Text + Visual	Web; Embodied; Code; Tool Use;	6	6	1,866	Both	-	55.2	65.0
AgentBank-13B (Song et al., 2024)	Text + Visual	Web; Embodied; Code; Tool Use;	16	16	51,287	Both	-	72.4	63.1
AgentGym-7B (Xi et al., 2024)	Text + Visual	Web; Embodied; Code; Tool Use;	14	89	6,130	Both	-	88.0	76.5

Table 3: Comparative Analysis of Environment and Task Diversity across Methods. *SWE* refers to SWE-bench (Jimenez et al., 2023). *ALF* refers to ALFWorld (Shridhar et al., 2020). *WS* refers to WebShop (Yao et al., 2023a). A dash (–) indicates that the quantity is not reported by the original paper, not applicable to the method, or not directly comparable under the table definition. Unless otherwise specified, performance numbers are taken from the cited papers rather than re-run experiments or live leaderboards.

unified framework for evaluating world generation. While WorldPrediction (Chen et al., 2025a) 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.3 Applications Across Domains

Driven by interactions with dynamic and diverse environments, recent agentic systems built on state-of-the-art LLM families such as GPT (OpenAI, 2025a), Claude (Anthropic, 2025), Gemini (Team et al., 2024), and Qwen (Bai et al., 2023) have become increasingly powerful and widely applicable.

Tool Use In tool-use scenarios, environments expose APIs and function calls as structured action spaces for agents, enabling them to reason with external tools through function calling or code generation (OpenAI, 2023; Anthropic, 2024; Wu et al., 2025b; Mastouri et al., 2025; Luo et al., 2025b; Wang et al., 2025e; Fan et al., 2025).

Software Engineering In code generation or software engineering scenarios, environments leverage repositories, test frameworks, and IDE integration to cultivate agents' long-horizon programming capabilities. Systems such as Qwen3-Coder (Qwen Team, 2024) and Claude 4 (Anthropic, 2025) demonstrate reliability in code edit-

ing and debugging. ReTool (Feng et al., 2025) novelly integrates code-interpreter execution into the reasoning loop, enabling agents to exhibit code self-correction and adaptive tool selection. SWE-smith (Yang et al., 2025a) and R2E-Gym (Jain et al., 2025) develop scalable, executable code environments based on GitHub repositories, enabling the further evolution of SWE agents.

Web Navigation In web navigation scenarios, environments are typically built on HTML/DOM structures to support tasks such as browsing, form filling, and transactions (Lai et al., 2024; Fosowl, 2024; Significant-Gravitas, 2023), requiring agents to click buttons and navigate between web pages. Graphical user interface (GUI) environments extend these tasks to desktops and mobile devices, further complicating task complexity (Qin et al., 2025; Ye et al., 2025a; Lu et al., 2025c; Liu et al., 2024b; Sun et al., 2022; Huang et al., 2025b).

Computer Use In more recent computer-use scenarios, environments integrate terminals, different operating systems, various applications, and APIs, making tasks much more interactive, open-ended, and realistic (Wang et al., 2025d,a; Xie et al., 2024; OpenAI, 2025; Jia et al., 2025a; Xie et al., 2025).

Deep Research In deep research scenarios, environments demand stronger long-context reasoning and more robust retrieval capabilities, as demonstrated by prominent long-form tasks such as AstaBench (Bragg et al., 2025), DeepResearch-Bench (Du et al., 2025), and ResearchQA (Yifei et al., 2025). By learning from the interactive experiences collected from these environments, agentic systems such as Tongyi DR (Tongyi DeepResearch Team, 2025), Gemini DR (Team et al., 2024), OpenAI DR (OpenAI, 2025b), and DR Tulu (Shao et al., 2025) are able to conduct sustained, in-depth analysis with mitigated context dilution.

D More 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 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., 2024a; 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.

E Ethics and Safety Implications

Agent systems require robust ethical safeguards, interoperable standards, and alignment with emerging regulatory frameworks to enable responsible integration into the global technology ecosystem (Howell and Ifayemi, 2024). Some prior work focuses specifically on safety, such as mitigating inherent biases that produce discriminatory outputs or defending against malicious inputs that circumvent safety-alignment mechanisms (Zhao

and Gómez Fariñas, 2023). However, new challenges are likely to emerge as agents accumulate experience through interactions with their environments and with other agents, and as their capabilities for autonomous task planning, external data access, and tool use continue to grow. For example, prompt injection or tool injection attacks on a single agent may lead to privacy breaches or system-wide contamination, while agents empowered to autonomously operate on external data can, in the absence of strict safeguards, cause irreversible damage by initiating fund transfers or deleting critical files (Tang et al., 2025). Therefore, when developing scaled environments for future LLM-based agents, safety considerations must be treated as a central and indispensable element of system design.