Agentic Environment Engineering for Large Language Models: A Survey of Environment Modeling, Synthesis, Evaluation, and Application

Jiachun Li*, Zhuoran Jin*, Tianyi Men*, Yupu Hao*, Kejian Zhu*, Lingshuai Wang*, Dongqi Huang*, Longxiang Wang*, Shengjia Hua*, Lu Wang*, Jinshan Gao*, Hongbang Yuan, Ruilin Xu, Kang Liu, Jun Zhao†

Abstract—Environments serve as interactive systems for large language model (LLM) based agents across diverse scenarios and play a crucial role in driving the continual evolution of model capabilities. Despite this importance, existing work lacks a systematic categorization and deep analysis. This paper systematically studies current researches on agentic environments from the perspective of the environment engineering lifecycle, covering their modeling, synthesis, evaluation and application. Specifically, the paper first introduces representative environments from the perspectives of eight attributes and eight domains, providing detailed analyses of their development paths and highlighting their core capabilities. Second, for automated environment synthesis, two paradigms are introduced, such as symbolic synthesis and neural synthesis. This paper also shows different environment evaluation methods in each paradigm. Thirdly, the corresponding environment applications from the perspective of agent-environment co-evolution are discussed. In specific, the paper characterizes the primary pathways for agent evolution in dynamic environments from four complementary perspectives: memory-centric experience evolution, orchestration-centric workflow evolution, trajectory-centric offline evolution, and exploration-centric online evolution. And three paradigms of environment evolution are identified, namely neural-driven, difficulty-driven, and scaling-driven approaches. At last, several promising future directions are discussed, including Environment-as-a-Service, Multi-agent Environments, and Neural-Symbolic Environments.

Index Terms—Agentic Environment Engineering, Large Language Model, Reinforcement Learning, World Model, Agent Evolution

1 Introduction

Large language models (LLMs) have demonstrated remarkable performance across a wide range of tasks, from basic factual question answering to complex reasoning tasks, with their capabilities continuing to evolve [1]–[4]. Recently, representative advanced models such as GPT-5.4 [5], Gemini-3.1-Pro [6], and Kimi K2.5 [7] have demonstrated strong agentic capabilities, particularly in complex tool calling, long-horizon planning, and self-improving [8]–[10]. To rigorously benchmark and further unlock the potential of these models, researchers draw inspiration from how humans continuously acquire experience and develop capabilities through interaction with environments, and accordingly design **agentic environments** for LLM agents.

An agentic environment is a dynamic system that simulates a real-world scenario that models can interact with. While models require interaction with the real world to acquire human-level skills, such interactions are often infeasible in practice due to multiple limitations, including high costs, safety risks, and data privacy concerns [11]–[13]. Furthermore, real-world environments are inherently irreproducible, rendering operational failures exceptionally costly. Conversely, manually engineering simulated alternatives is not only resource-intensive but also suffers from constrained scenario coverage. For example, allowing an autonomous driving system to directly control a vehicle on real roads is not only highly inefficient but also poses significant risks of accidents. Agentic environments effectively address these challenges by incorporating sophisticated tools and reward signals that closely approximate real-world conditions. In addition, by interacting with simulated environments, large-scale trajectory data can be generated. In this way, continual evolution could be thereby enabled. Consequently, the agentic environment has become an inseparable twin of the agent throughout different stages, such as capability evaluation [14]–[16], inference-time reasoning enhancement [17]–[19], and reinforcement learning training [20]–[23].

Nevertheless, agentic environments have not yet been systematically organized and analyzed in the recent literature [24]–[26]. To bridge this gap, this paper systematically reviews the whole lifecycle of **agentic environment engineering**, including environment modeling, automated environment synthesis, evaluation of environment quality, and environment applications. Specifically, our survey is organized around the following three research questions:

- **RQ1:** What are the key characteristics and categories of agentic environments? (See Sec. 3 and Sec. 4)
- **RQ2:** How can agentic environments be systematically

Version: v1 (major update on May 19, 2026)
†*Corresponding author: Jun Zhao.*
**These authors contributed equally to this work.*
Jiachun Li, Zhuoran Jin, Tianyi Men, Yupu Hao, Kejian Zhu, Lingshuai Wang, Dongqi Huang, Longxiang Wang, Shengjia Hua, Lu Wang, Jinshan Gao, Hongbang Yuan, Ruilin Xu, Kang Liu, Jun Zhao are with the Key Laboratory of Cognition and Decision Intelligence for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100191, China (E-mail: jiachun.li@nlpr.ia.ac.cn, zhuoran.jin@nlpr.ia.ac.cn, tianyi.men@nlpr.ia.ac.cn, kliu@nlpr.ia.ac.cn, jzhao@nlpr.ia.ac.cn).

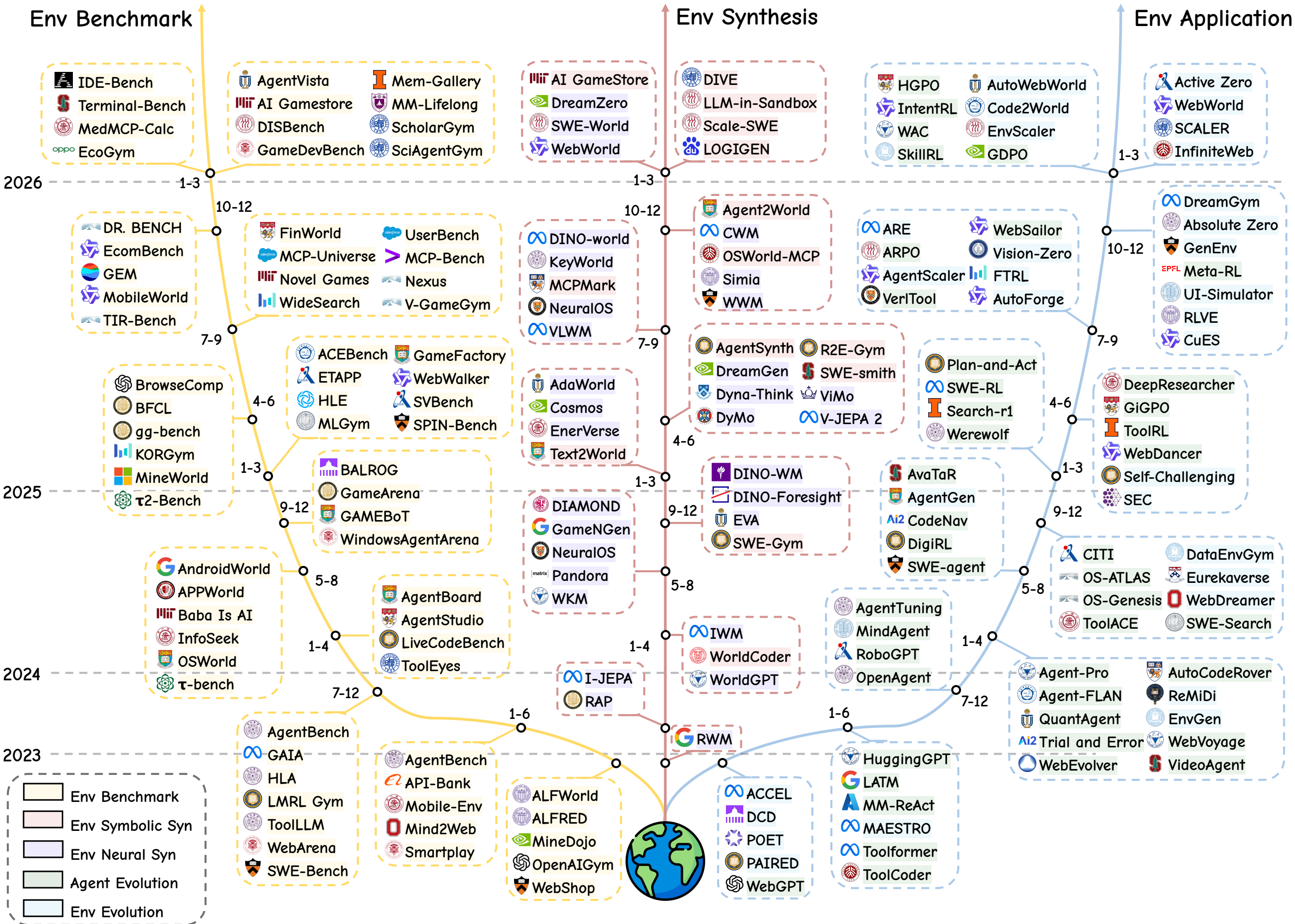

Fig. 1. An overview of agentic environment engineering.

constructed and evaluated? (See Sec. 5)

- **RQ3:** How do agentic environments facilitate the closed-loop co-evolution process of agent and environment? (See Sec. 6 and Sec. 7)

To answer RQ1, this paper begins by formally defining key concepts related to environments, clarifying their core components and mechanisms, and introducing the main focus of this survey. Then, existing environments are analyzed from the perspective of their attributes, including representation, feedback, timing, observability, stochasticity, continuity, modality, and cardinality. Through comparative analysis, we observe that existing environments are inadequately suited for multi-agent settings. Moreover, there is significant potential for improvement in striking a balance between the engineering reliability of symbolic systems and the infinite generative scalability of neural models.

To fully demonstrate the types of existing environments, the paper further introduces current benchmarks from the task domain. Specifically, we classify all environments into eight representative domains: GUI [15], [27], [28], Deep Research [29]–[31], Embodied [32]–[34], Game [35]–[37], Tool [38]–[40], Code [41]–[43], Domain-Specific [44]–[46], and Cross-Domain [3], [47], [48]. Related works are categorized into each domain, and detailed descriptions of their development trajectories and core agent capabilities are provided.

To answer RQ2, the paper emphasizes the environment synthesis methods and quality evaluation. **On environment synthesis**, all existing works are divided into two types based on the environment synthesis forms:

- *Symbolic Synthesis* uses symbolic rules with different forms (*e.g.*, code) to synthesize environments, employing verifiable rubrics to ensure reliable environmental feedback. Following this approach, this survey identifies an evolution of scalability. Specifically, the transition from task-driven to real-world–driven synthesis paradigms, and ultimately to *de novo* synthesis methods, signifies a continuous expansion in both synthesis flexibility and the breadth of the environmental space.
- *Neural Synthesis* parameterizes the environment using a neural network (particularly a world model), and constructs function mappings based on the model's parameters to enable dynamic interactions with the agent. This paper systematically separates neural synthesis into three fine-grained paradigms based on the form of environment representation: Pixel-level, Word-level, and Latent-level Modeling.

On environment evaluation, the paper discusses commonly used quality control techniques in four perspectives: correctness, diversity, complexity, and fidelity. We highlight that, while correctness has been well-established and there exist several evaluation frameworks, the study of evaluating diversity, complexity, and fidelity remains under-researched [49]–[51]. Actually, these dimensions are crucial not only for assessing the environmental reliability but also for ensuring the environments' effectiveness in training agents across a wide range of tasks.

To answer RQ3, this survey discusses the application of environments from two complementary perspectives: agent evolution and environment evolution. **On agent evolution**, the agent's capabilities advance through continuous interaction with the environment. Existing agent evolutionary methods are classified into four categories:

- *Memory-Centric Experience Evolution* enhances task-processing capabilities by accumulating and leveraging experiences from past interactions.
- *Orchestration-Centric Workflow Evolution* focuses on the dynamic coordination and adaptation of task sequences and multi-agent interactions within a workflow, thereby optimizing performance and efficiency across diverse environments.
- *Trajectory-Centric Offline Evolution* generates high-quality task interaction trajectories, which subsequently facilitate refining agent behavior by simulating high-fidelity and multifaceted environments.
- *Exploration-Centric Online Evolution* leverages reinforcement learning to refine agent capabilities dynamically, enabling the agent to continuously adapt and optimize its performance.

On environment evolution, this survey focuses on how environments are progressively evolved to support the continuous improvement of agent capability. We categorize this paradigm into three types based on the aspects of environmental changes:

- *Neural-Driven Evolution* adjusts the environment's internal parameters to simulate a variety of potential states and outcomes, helping the agent to prepare for diverse scenarios.
- *Difficulty-Driven Evolution* adapts the complexity of tasks in the environment to match the agent's evolving capabilities, typically through curriculum learning.
- *Scaling-Driven Evolution* aims to expand the range of environments by increasing scenario diversity or introducing entirely new structures, which pushes agents towards broader generalization across different contexts.

Finally, current challenges and future research directions are highlighted. Specifically, this paper emphasizes Environment-as-a-Service as a potential paradigm for standardized, scalable, and reproducible environment deployment. We then discuss the ongoing evolution of environments toward dynamic, long-horizon, open-ended, multimodal, and multi-agent settings, which impose fundamentally new requirements on agent capabilities. Moreover, emerging directions such as Neural-Symbolic Environments, sim-to-real alignment, and agent–environment co-evolution are introduced for developing more reliable and adaptive agent systems. Looking ahead, the paper argues that establishing a scientific foundation for environment engineering, including environment scaling laws and environment–capability relationships, will be crucial for advancing the next generation of LLM agents.

In summary, this survey is organized as follows:

- Sec. 2 introduces the necessary preliminaries and highlights the gap between environment engineering and traditional data engineering.
- Sec. 3 introduces the key properties of environments, provides their formal definitions, and offers detailed explanations with concrete examples.
- Sec. 4 categorizes existing environments from the perspective of tasks, and provides a detailed overview of the specific tasks and representative environments in domains such as GUI, Deep Research, and Embodied.
- Sec. 5 introduces existing methods of automatic environment synthesis in both symbolic and neural perspectives, and analyzes their respective development pathways and evaluating solutions.
- Sec. 6 presents how environments facilitate agent evolution, including Experience Utilization, Agentic Workflow Design, Synthetic Data Generation, and Reinforcement Learning Optimization.
- Sec. 7 focuses on the main paradigms of environment evolution, explaining how environments adapt to the agent's training from three perspectives (neural, difficulty, and scaling).
- Sec. 8 summarizes the main challenges of existing environment-related research and discusses potential future directions.

2 PRELIMINARIES

2.1 Definition of Key Concepts

2.1.1 Environment

We define the **Environment** $\mathcal{E}$ as a stochastic dynamical system with which the agent π interacts. Formally, we represent $\mathcal{E}$ as a Partially Observable Markov Decision Process (POMDP), defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \Omega, \mathcal{O}, \gamma \rangle$:

- **State Space** $\mathcal{S}$: The set of all possible latent states of the environment.
- **Action Space** $\mathcal{A}$: The set of all possible actions available to the agent.
- **Transition Function** $\mathcal{P}$: Defined as $\mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, this probabilistic kernel governs the environment's physical dynamics, where $\mathcal{P}(s_{t+1}|s_t, a_t)$ denotes the probability of transitioning to state s_{t+1}.
- **Reward Function** $\mathcal{R}$: Defined as $\mathcal{S} \times \mathcal{A} \to \mathbb{R}$, given the current state s_t and action a_t, the mechanism providing scalar feedback r_t to the agent.
- **Observation Space** Ω and **Observation Function** $\mathcal{O}$: Defined as $\mathcal{S} \times \mathcal{A} \to \Delta(\Omega)$, the interface denotes the agent's perception of the latent state, where $\mathcal{O}(o_t|s_t, a_t)$ denotes the probability of receiving observation o_t given the current state and action.
- **Discount Factor** γ: A constant $\in [0, 1)$ that weights the importance of future rewards, ensuring the mathematical convergence of the expected total return in infinite-horizon tasks.

In this work, we focus on **agentic environments**, defined as a class of dynamic, interactive systems designed for evaluating and training LLM agents. Unlike traditional reinforcement learning simulators which operate over predefined state-action spaces and fixed transition dynamics, agentic

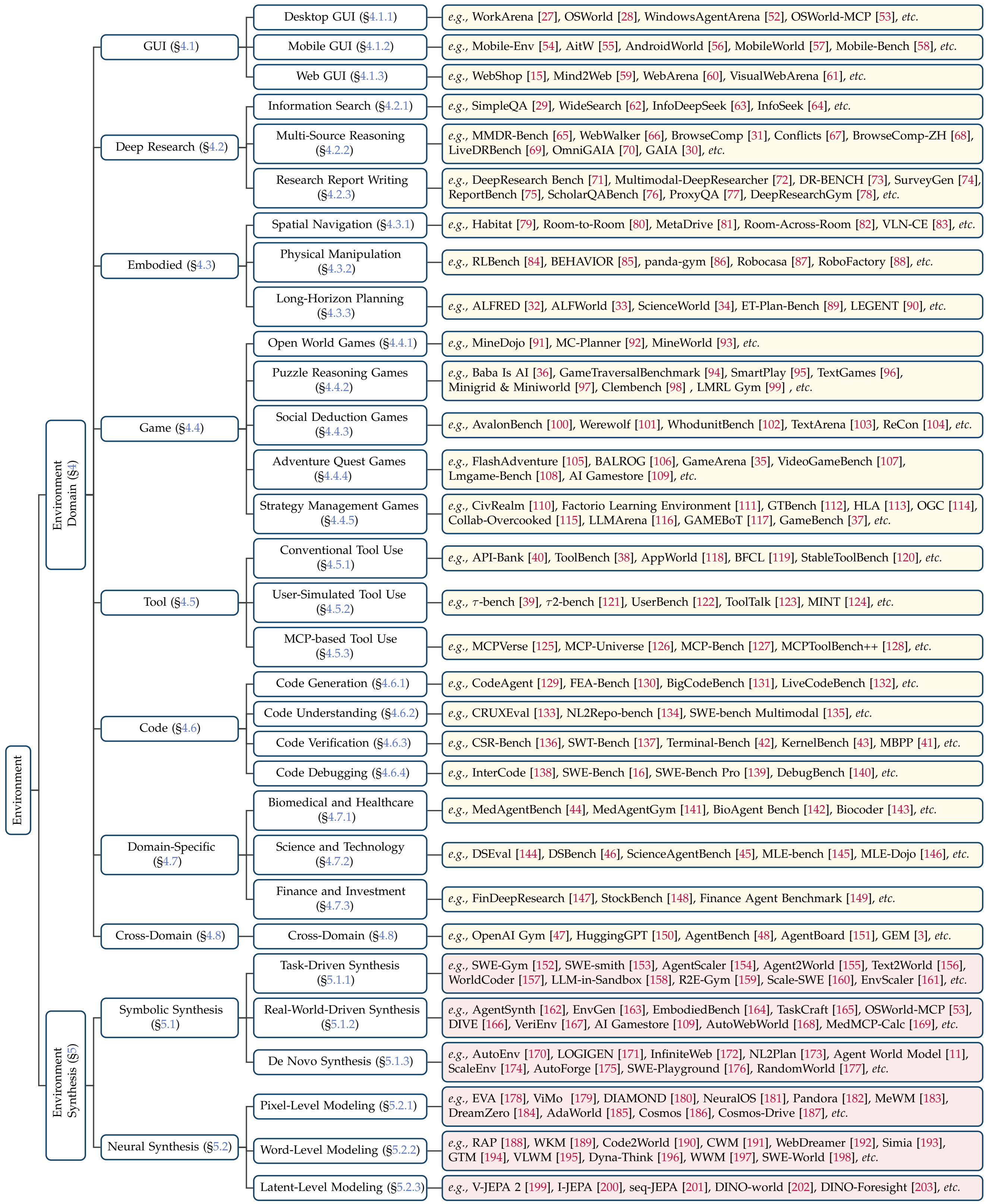

Fig. 2. Taxonomy of the main content of this survey.

environments enable more open-ended, language-centric, and tool-augmented interactions. Such environments define a set of permissible actions that the LLM can execute (e.g., generating natural language responses [33], [47], [145] or invoking external tools [8], [265], [321]), and return corresponding observations (e.g., code execution results [16], [42], [132] or HTML pages [15], [60], [227]) and rewards (e.g., answer correctness or format correctness [4], [30], [116]), thereby enabling iterative interaction with the model.

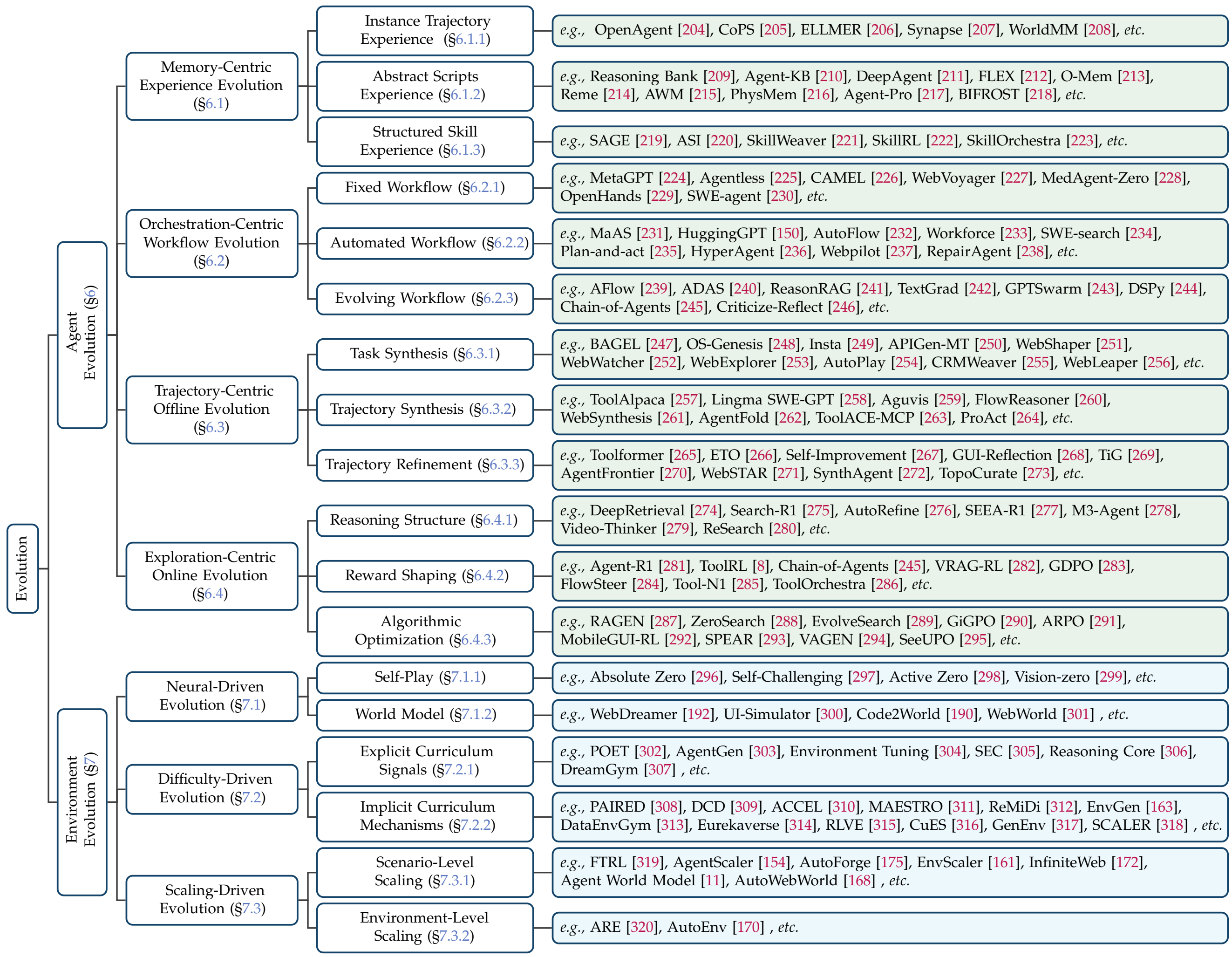

Fig. 3. Taxonomy of the main content of this survey.

2.1.2 Agent

The **Agent** is defined as an autonomous decision-making entity that interacts with $\mathcal{E}$. In a POMDP setting, the underlying state s_t is not directly observable. Therefore, the agent conditions its next action on the **History** $h_t \in \mathcal{H}$, which represents the sequence of all previous observations, actions, and rewards:

$$h_t = (o_0, a_0, r_0, \ldots, a_{t-1}, r_{t-1}, o_t) \in \mathcal{H} \tag{1}$$

The agent's behavior is governed by a **Policy** $\pi : \mathcal{H} \to \Delta(\mathcal{A})$, which maps the history to a probability distribution over actions. The probability of selecting action a_t is given by:

$$\pi(a_t|h_t) = P(A_t = a_t|H_t = h_t). \tag{2}$$

The primary objective of the agent is to find an optimal policy π^* that maximizes the **Expected Discounted Return** $J(\pi)$:

$$J(\pi) = \mathbb{E}_{\pi,\mathcal{P},\mathcal{O}}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k}\right] \tag{3}$$

where the expectation is taken over the trajectories induced by the interaction between the policy π and the environment's transition and observation dynamics.

As this survey focuses on agents built upon LLMs, the policy π corresponds to the internal parameters of the model, while h_t represents the interaction history between the model and the environment. Unlike traditional reinforcement learning, where actions are typically discrete, the action a_t in this setting primarily consists of natural language tokens generated by the LLM. As a result, trajectories are largely formed by chains of thought produced during the interaction between the model and the environment.

2.1.3 Environment-Agent Alignment

In the previous subsection, we note that the primary objective of environment-agent interaction is to optimize the policy π for maximizing expected returns, which we term **Environment–Agent Alignment**. How to achieve this objective is also one of the main focuses of this survey.

Traditional approaches widely adopt teacher models to synthesize high-quality offline trajectories, which are then used for supervised fine-tuning (SFT) to enable imitation learning for the agent's policy π [38], [265], [267]. However, recent efforts have increasingly focused on a new paradigm known as Agentic Reinforcement Learning (RL). This approach is particularly instrumental for developing reasoning models (e.g., DeepSeek-R1 [2]), which refine multi-turn cognitive trajectories or tool-mediated interactions through RL [275], [290], [291]. Here, we introduce several representative Agentic RL algorithms as preliminary background:

PPO. Proximal Policy Optimization (PPO) [322] has long been the de facto standard for LLM alignment and reinforcement learning. PPO typically employs an actor-critic framework, maintaining a value network V_ϕ to estimate the expected return, which serves as a baseline for variance reduction. The training objective involves a clipped surrogate loss to ensure stable updates by constraining the policy shift between iterations. For a given state-action pair, the PPO objective is defined as:

$$\mathcal{L}^{\text{PPO}}(\theta) = \mathbb{E}_t\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right] \quad (4)$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio, with $\pi_{\theta_{\text{old}}}$ representing the reference policy from the previous iteration. The advantage $\hat{A}_t$ is typically computed using Generalized Advantage Estimation (GAE) based on the rewards and the learned value function:

$$\hat{A}_t = \sum_{k=0}^{\infty}(\gamma\lambda)^k\delta_{t+k}, \quad \delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t) \quad (5)$$

where γ and λ are hyperparameters for discounting and smoothing. While effective, the requirement of a dedicated critic network introduces significant memory and computational overhead, posing challenges for scaling in resource-constrained environments.

GRPO. While PPO serves as a foundational baseline, **Group Relative Policy Optimization (GRPO)** [20] has recently emerged as a highly efficient alternative for agentic RL works. In contrast to the standard actor-critic architecture, GRPO eliminates the need for an explicit value network by estimating the advantage from a sample-based baseline. Specifically, for a set of G outputs $\{y_i\}_{i=1}^G$ sampled from the same prompt q, the GRPO objective is formulated as:

$$\mathcal{L}^{\text{GRPO}}(\theta) = \mathbb{E}_q\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\left(\mathcal{L}_{i,t}^{\text{CLIP}}(\theta) - \beta\mathbb{D}_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}})\right)\right] \quad (6)$$

where the surrogate clipping objective is defined as:

$$\mathcal{L}_{i,t}^{\text{CLIP}}(\theta) = \min\left(r_{i,t}(\theta)\hat{A}_i, \text{ clip}(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_i\right) \quad (7)$$

with $r_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t}|q, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|q, y_{i,<t})}$ denoting the importance sampling ratio. The group-normalized advantage $\hat{A}_i$ is computed as:

$$\hat{A}_i = \frac{r_i - \mu}{\sigma + \delta}, \quad \mu = \frac{1}{G}\sum_{j=1}^{G} r_j, \quad \sigma = \sqrt{\frac{1}{G}\sum_{j=1}^{G}(r_j - \mu)^2} \quad (8)$$

where $\delta > 0$ is a small constant for numerical stability. This relative reward mechanism effectively reduces variance without the computational overhead of a critic network, facilitating more stable optimization in high-dimensional discrete action spaces.

DAPO. To further enhance the stability and efficiency of group-relative methods, **Decoupled Clip and Dynamic Sampling Policy Optimization (DAPO)** [21] builds upon the GRPO framework by introducing several key architectural refinements. Addressing the entropy collapse and gradient vanishing issues in standard GRPO, DAPO decouples the clipping mechanism into separate upper and lower bounds, allowing for more aggressive updates when moving away from suboptimal states while maintaining strict constraints on positive shifts. Furthermore, DAPO shifts from sample-level to token-level policy gradient loss to ensure a more fine-grained credit assignment. The DAPO objective is refined as:

$$\mathcal{L}^{\text{DAPO}}(\theta) = \mathbb{E}_q\left[\frac{1}{\sum_{i=1}^{G}|y_i|}\sum_{i=1}^{G}\sum_{t=1}^{|y_i|}\mathcal{L}_{i,t}^{\text{D-CLIP}}(\theta)\right] \quad (9)$$

where the decoupled surrogate clipping objective is defined as:

$$\mathcal{L}_{i,t}^{\text{D-CLIP}}(\theta) = \min\left(r_{i,t}(\theta)\hat{A}_i, \text{clip}(r_{i,t}(\theta), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}})\hat{A}_i\right) \quad (10)$$

The $\hat{A}_i$ and $r_{i,t}(\theta)$ follows the definition in GRPO. By integrating these refinements, DAPO effectively mitigates gradient vanishing and overshooting in complex reasoning tasks, ensuring more robust convergence in large-scale agentic training.

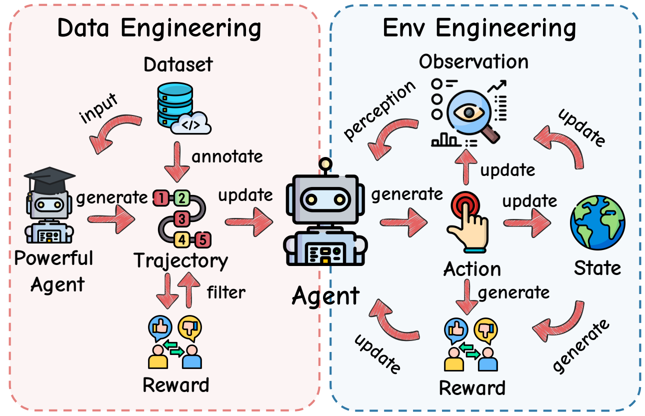

Fig. 4. A comparison between data engineering and environment engineering.

2.2 From Data Engineering to Environment Engineering

In traditional tasks, we train and test LLMs based on datasets, and data-centric engineering has already matured significantly. So why, at this point in time, do we need environments to replace the role that data previously played? In this section, we aim to further introduce the characteristics of the environment and explain its value by comparing it with traditional data engineering.

2.2.1 *From Passive Learning to Collaborative Evolution*

In the paradigm of traditional data engineering, the training process is characterized by a unidirectional transfer of information from a static corpus to the model. The data acts as a fixed boundary of knowledge which forces the model to act as a passive recipient of pre-collected trajectories (see Fig. 4, left). This static nature creates a significant mismatch between the fixed difficulty of the dataset and the evolving proficiency of the learner. If the provided data is too advanced for the current state of the model, the optimization process often becomes unstable or fails to converge. Conversely, when the model encounters trajectories that are relatively simple, it quickly reaches a point of diminishing returns where additional training iterations no longer yield performance gains because the dataset lacks the necessary complexity to drive further improvement.

The introduction of an environment shifts this relationship toward a collaborative evolution, where the data distribution is dynamically adjusted based on the real-time performance of the agent. Instead of being restricted to a fixed set of samples, the environment acts as a generative engine that can modulate the complexity of tasks to match the growing capabilities of the model. For instance, some environments can regulate their complexity to match the agent's progressively enhancing capability during the training phases [303], [307], [318]. This synchronization ensures that the learning signal remains within an optimal range of difficulty, allowing the model to continuously transcend its previous performance limits through a process of mutual progression between the learner and the simulated world.

2.2.2 From Single-turn Q&A to Multi-turn Interaction

In traditional data engineering, most samples interact with the model in a single-turn question-and-answer format, such as in GSM8K [323], MMLU [324], MMMU [325], and others [326]. However, this format inherently overlooks the procedural complexity required to navigate open-ended problems that cannot be solved in a single step. By isolating the question from a broader context of trial and error, traditional datasets encourage the model to find a direct mapping from the prompt to the answer. This often results in a model that is proficient at pattern matching within limited contexts but lacks the resilience to handle tasks that require long-term reasoning or error correction.

In contrast, an environment provides a rich interactive framework that supports multi-turn engagement and external tool integration. Within this ecosystem, the agent does not work in isolation but rather collaborates with various functional roles such as code executors [16], [42], [327], knowledge retrievers [30], [31], [62], and other specialized agents [88], [116], [125], [328]. These multi-turn interactions allow the agent to decompose a complex objective into a series of manageable sub-tasks while receiving immediate feedback at each juncture. By interacting with an executor to verify a logic step or querying a search engine to fill a knowledge gap, the agent learns to manage a dynamic workflow [232], [329], [330]. This transition enables the model to move beyond the rote memorization of static solutions and instead master the general strategies of problem-solving and strategic planning that are essential for real-world applications.

2.2.3 From Open-loop System to Closed-loop System

Traditional data-driven training is essentially an open-loop system where the learning process is decoupled from the consequences of the model's outputs. In this paradigm, the model generates a response based on a static prompt and the loss is calculated against a fixed gold standard. This lack of a feedback loop means that the model cannot perceive how its predicted tokens influence the subsequent state of a task. Consequently, any minor error in the initial steps will accumulate without a mechanism for correction because the static data cannot provide a reactive signal to guide the model back to a successful path.

An environment transforms this process into a closed-loop control system by establishing a continuous coupling between the agent's actions and the resulting state changes. Every decision made by the model triggers a response from the environment which then serves as the new input for the next iteration. This reciprocal interaction forces the model to internalize the transition dynamics of the task and learn robust recovery strategies. By experiencing the objective outcomes of its own errors, the model moves beyond the simple memorization of surface patterns and develops a functional understanding of goal-oriented logic. This closed-loop mechanism is the fundamental prerequisite for achieving long-term stability and autonomous self-improvement in complex reasoning chains.

2.3 Positioning Our Survey

Consistent with the notion that *man is shaped by his environment*, an agent and its environment are intrinsically coupled and cannot be considered in isolation. However, despite the increasing attention in recent surveys to the evolution of agents within their environments, existing studies largely adopt an agent-centric view [24], [331]–[336]. Some studies focus on agent reasoning methods [24], [332], [334], investigating how algorithms such as reinforcement learning can be applied to effectively improve agent performance across different environments. Other studies take a system-level perspective [331], [333], [336], examining the evolution of agents by investigating how different components in agentic systems, such as context, tools, and algorithms, can be coordinated effectively. Our survey is organized around the lifecycle of environments, systematically covering the full pipeline of environment modeling, construction, evaluation and application, and providing perspectives complementary to those absent in prior work.

3 Environment Attribute

As shown in Fig. 5, the landscape of agent evaluation environments is characterized by a multidimensional taxonomy of attributes. These underlying attributes change how agents perceive, interact, and make decisions with the environments. To capture the fundamental differences among these environments, it is necessary to analyze them based on their basic mechanics and mathematical formulations. Therefore, we break down the basic attributes of these environments into several key pairs, including **Symbolic vs. Neural** (Sec. 3.1), **Open-Loop vs. Closed-Loop** (Sec. 3.2), **Online vs. Offline** (Sec. 3.3), **MDP vs. POMDP** (Sec. 3.4), **Deterministic vs. Nondeterministic** (Sec. 3.5), **Discrete vs. Continuous** (Sec. 3.6), **Unimodal vs. Multimodal** (Sec. 3.7), and **Single-Agent vs. Multi-Agent** (Sec. 3.8). This taxonomy establishes a systematic framework for analyzing these environments, highlighting the diverse complexities and fundamental attributes that shape agent-environment interactions.

3.1 Symbolic vs. Neural

The core distinction between these paradigms lies in the underlying implementation of the transition function $\mathcal{P}$

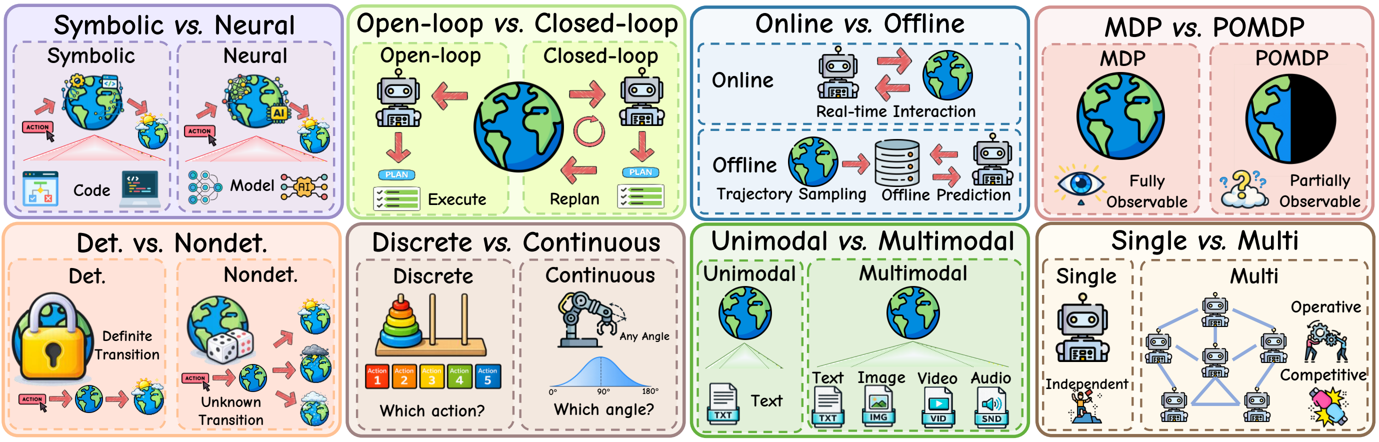

Fig. 5. An overview of environment attributes.

and how the environment dynamics are generated. This determines whether the system relies on programmed code to update the states or uses a neural model to predict future states.

3.1.1 Symbolic Environment

In a symbolic environment, the transition dynamics are governed by explicit programmed logic and predefined rules. The transition function $\mathcal{P}(s_{t+1}|s_t, a_t)$ is realized through executing source code, software scripts, or physics engines. As a foundational standard in this paradigm, the planning domain definition language (PDDL) governs transition dynamics through explicit symbolic action schemas, utilizing logical preconditions and deterministic effects to update environment states. While PDDL provides a strict logical foundation, modern environments predominantly realize these transitions through flexible programming code. Agent World Model [11] synthesizes large executable environments using code-driven backends and SQL databases to ensure reliable state transitions and consistency for agent training. EnvScaler [161] programmatically synthesizes executable tool-interactive environments where transition dynamics and state updates are governed by explicit Python code and predefined rule-based logic. AutoEnv [170] utilizes domain-specific languages and code abstractions to automate the synthesis of executable environments with explicit, programmed transition and reward logic.

3.1.2 Neural Environment

In a neural environment, the transition dynamics are approximated by a model parameterized by θ. The transition function $\mathcal{P}_\theta(s_{t+1}|s_t, a_t)$ is realized through the forward propagation of these learned parameters, typically utilizing neural networks. This approach often serves as a surrogate simulator to generate state transitions where the true underlying physical rules are unknown or overly complex. WebDreamer [192] uses LLMs as learned world models to simulate web state transitions, approximating environment dynamics through neural network parameters rather than explicit code. DreamGen [337] utilizes neural video world models to approximate environment transitions and generate synthetic trajectories for robot training through the forward propagation of learned parameters. WebEvolver [338] employs concurrently optimized LLMs to function as virtual web servers, forecasting future webpage representations based on present conditions and executed actions to synthesize multi-step interaction trajectories.

3.2 Open-Loop vs. Closed-Loop

The core distinction between the two paradigms lies in how the agent utilizes the observation sequence to formulate its actions over time. This determines whether the agent simply executes a fixed plan based on the starting conditions, or continuously adjusts its next steps based on new feedback from the environment.

3.2.1 Open-Loop Environment

In an open-loop system, the agent receives only a single initial observation o_0 and executes a predetermined sequence of actions. It does not incorporate subsequent feedback from the environment. The policy simplifies to depend on a sequence predetermined from the initial observation $\pi(a_t|o_0)$ in a non-adaptive manner. TaskLAMA [339] decomposes complex tasks into structured task graphs from initial goal prompts without incorporating real-time environmental observations or adaptive feedback mechanisms. HuggingGPT [150] adopts a global planning strategy to predetermine a complete task sequence from the initial request without incorporating subsequent feedback from the environment during execution. TaskBench [340] assesses task automation capabilities by requiring models to generate a tool invocation graph directly from the initial user instruction, omitting interactive feedback from actual tool execution.

3.2.2 Closed-Loop Environment

In a closed-loop system, the agent receives new observations o_t at each time step and adapts its actions accordingly. The policy utilizes the dynamically updating history h_t to make reactive decisions, allowing the agent to respond to environmental stochasticity and correct deviations. Mobile-Bench [58] implements an iterative execution framework where agents dynamically update their thoughts and actions based on real-time UI feedback and historical environment observations. MCPVerse [125] utilizes an automated agentic loop where models interact with executable tools and leverage real-time feedback to iteratively refine their planning and multi-step actions. WebWalker [66] employs an explore-and-critic framework that iteratively adapts its navigation based on real-time observations and dynamic memory to resolve complex web traversal tasks.

3.3 Online vs. Offline

The core distinction between the two paradigms lies entirely in the interaction mechanism. This mechanism dictates whether an agent actively drives state transitions through sequential exploration with the environment, or is evaluated solely on its ability to mimic expert behavior from prerecorded trajectories.

3.3.1 Online Environment

In an online environment, the agent actively interacts with the true dynamical system $\mathcal{E}$. At each time step t, the agent executes an action a_t, and the environment sequentially yields the subsequent observation o_{t+1}. The agent evaluates and refines its behavior through real-time feedback. For example, WebArena [60] provides a suite of fully functional websites where agents interact dynamically and receive real-time feedback to achieve complex tasks. EmbodiedBench [164] assesses vision-driven agents through real-time interaction across diverse simulated environments , where models must actively process sequential feedback and visual observations to refine their task execution. Terminal-Bench [42] benchmarks agents on interactive, long-horizon tasks within real terminal environments, requiring models to autonomously manipulate containerized systems and refine actions based on sequential execution feedback.

3.3.2 Offline Environment

In an offline environment, direct interaction with the system $\mathcal{E}$ is strictly prohibited. The agent relies instead on a static dataset of previously sampled trajectories $\mathcal{D}$. Evaluation is conducted statically by comparing the predicted single step action $\pi(a_t|h_t)$ against the expert action a_t^* from human or strong models, severing the standard sequential feedback loop. For example, Mind2Web [59] evaluates agents on cached snapshots of real-world websites by comparing predicted actions against human-annotated trajectories without real-time sequential feedback. ALFRED [32] uses a static dataset of expert demonstrations to evaluate models, comparing predicted actions against human-annotated ground truth without real-time environmental feedback. Android-Control [341] evaluates agents on a static dataset across apps, measuring single-step action accuracy against expert ground truth rather than through real-time environmental interaction.

3.4 MDP vs. POMDP

The core distinction between these two paradigms lies in the mapping relationship between the environment's state space $\mathcal{S}$ and the agent's observation space Ω. Essentially, this determines whether the agent has full access to everything happening in the environment, or if it must rely on incomplete observations to understand the true situation.

3.4.1 Fully Observable Environment

In a fully observable environment, typically formalized as a Markov Decision Process (MDP), the agent has complete and transparent access to the true dynamics. The observation space perfectly aligns with the state space, formally $\Omega = \mathcal{S}$. Here, the observation o_t is entirely identical to the latent state s_t, allowing the agent to base its policy strictly on the current state without needing historical context, formulated as $\pi(a_t|s_t)$. For example, KOR-Bench [342] evaluates reasoning capabilities through tasks where all logical rules and complete problem states are explicitly provided in the text prompt. GVGAI-LLM [343] directly presents the evolving game status and foundational principles through formatted text descriptions during every single step. V-GameGym [344] evaluates LLMs on synthesizing complete visual games from explicit text instructions where the sandbox execution provides transparent access to all underlying game states.

3.4.2 Partially Observable Environment

In a partially observable environment, modeled as a POMDP, the agent lacks direct access to the complete latent state. The observation $o_t \in \Omega$ serves as an incomplete or noisy projection of s_t, strictly governed by the observation function $\mathcal{O}(o_t|s_t, a_t)$. Consequently, the agent is required to aggregate information across its entire interactive trajectory h_t to infer the underlying state distribution and execute optimal decisions. For example, WebArena [60] evaluates web agents that observe only the focused tab or a limited viewport. Consequently, agents must rely on their interaction history to infer the full underlying website state. GAIA [30] evaluates general AI assistants on complex real-world questions where agents must iteratively gather information from a diverse environment. EmbodiedBench [164] evaluates multimodal agents in 3D simulated tasks where the complete environment state is hidden , requiring models to make decisions relying entirely on egocentric visual observations and interaction history.

3.5 Deterministic vs. Nondeterministic

The distinction between deterministic and nondeterministic environments is characterized by the properties of the transition function $\mathcal{P}$ and the reward function $\mathcal{R}$. Essentially, this determines whether taking a specific action in a given situation always produces the exact same, predictable states, or if the outcome involves randomness and uncertainty.

3.5.1 Deterministic Environment

In a deterministic environment, executing a specific action $a_t \in \mathcal{A}$ in a given state $s_t \in \mathcal{S}$ invariably results in a single, predictable next state s_{t+1} and reward r_t. Formally, the transition kernel collapses into a deterministic mapping, such that $\mathcal{P}(s_{t+1}|s_t, a_t) = 1$ for exactly one target state. GameTraversalBenchmark [94] evaluates planning on static 2D maps where discrete directional actions predictably determine the next state. Baba Is AI [36] introduces a gridworld puzzle game where manipulating objects and textual rules results in predictable and exact state transitions. VSP [345] features classical maze navigation and block-moving scenarios, which serve as deterministic environments because they are governed by strict, predefined rules where every allowed action yields a guaranteed, predictable outcome.

3.5.2 Nondeterministic Environment

In a nondeterministic environment, the environmental dynamics are governed by probability distributions. Taking an action a_t in state s_t results in a next state s_{t+1} that is sampled from the distribution defined by the transition kernel $\mathcal{P}(\cdot|s_t, a_t) \in \Delta(\mathcal{S})$. Consequently, the identical state-action pair (s_t, a_t) can yield different resulting states and rewards across independent interactions. Furthermore, environments in which the agent's state transition function is fundamentally difficult to obtain or intractable to model are also practically classified as nondeterministic. Frozen Lake [47] in the slippery ice setting introduces inherent stochasticity through its slippery grid-world mechanics, where intended directional movements have a probability of resulting in transitions to unintended adjacent states. BrowseComp [31] requires agents to persistently navigate the open web to find obscure information, operating within a vast internet environment where state transitions are fundamentally intractable to model. WebVoyager [227] navigates live real-world websites where dynamic elements like constant updates and pop up windows make state transitions hard to predict.

3.6 Discrete vs. Continuous

The core distinction between the two paradigms lies entirely in the action space $\mathcal{A}$. This determines whether the agent chooses from a fixed, limited set of options or generates exact numerical values across an infinite range.

3.6.1 Discrete Action Space

In an environment with a discrete action space, the available actions form a finite and countable set like $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$. The agent's policy typically outputs a categorical probability distribution over these specific choices. ALFWorld [33] evaluates agents on embodied household tasks by restricting interactions to a finite set of text commands and physical action primitives. WebShop [15] simulates online shopping where agents operate by selecting from a finite set of textual search and button click actions. APPWorld [118] evaluates interactive coding agents on everyday digital tasks where the agent must select from a finite set of predefined API calls.

3.6.2 Continuous Action Space

In an environment with a continuous action space, the available actions span a real-valued vector space: $\mathcal{A} \subseteq \mathbb{R}^d$. The agent cannot enumerate the actions and must instead rely on its policy to output a continuous probability density function to select an action vector. RoboFactory [88] evaluates collaborative manipulation tasks where multiple embodied agents must output continuous joint control signals across eight dimensions to execute precise physical interactions. Scenario Dreamer [346] evaluates autonomous vehicle planners that navigate using a continuous control space of steering and acceleration commands applied to a realistic bicycle model. EmbodiedBench [164] evaluates vision-driven agents on low-level navigation and manipulation tasks that require precise control over real-valued vectors representing translational and rotational displacements.

3.7 Unimodal vs. Multimodal

The core distinction between the two paradigms lies in the modality of the observation space Ω. This determines whether the agent perceives the environment through a single modality or combine several different modalities to understand the environment.

3.7.1 Unimodal Environment

In a unimodal environment, the observation space Ω consists of a single modality, predominantly textual or visual data. The observation o_t is presented in a homogeneous linguistic format, requiring the agent to process and infer the state s_t entirely through text. MetaDrive [81] provides diverse driving scenarios where agents perceive the visual environment through a homogeneous observation modality to benchmark generalizability across unseen traffic scenes. API-Bank [40] evaluates tool-augmented capabilities by requiring models to plan and execute API calls through a textual interface consisting of natural language dialogues and structured documentation. OSWorld-MCP [53] evaluates computer-use agents by standardizing tool invocation through the text-based Model Context Protocol (MCP) to facilitate structured context ingestion and interaction.

3.7.2 Multimodal Environment

In a multimodal environment, the observation space Ω spans across heterogeneous sensory modalities, typically integrating text, images, and videos. The observation becomes a composite representation, formally $o_t = \langle o_t^{\text{text}}, o_t^{\text{image}}, o_t^{\text{video}} \rangle$. Consequently, the agent is required to align and fuse these diverse observational streams to accurately comprehend the environment. VisualWebArena [61] evaluates multimodal agents on realistic web tasks requiring the integration of images and html to navigate and execute actions on complex websites. AgentStudio [347] offers a realistic environment with universal observation spaces that integrate text, image, and video modalities to evaluate general virtual agents. MLE-bench [145] evaluates AI agents on machine learning engineering tasks requiring the processing of diverse data modalities including text, images, and signal data.

3.8 Single-Agent vs. Multi-Agent

The key difference is the number of agents interacting with the environment. This determines whether the environment's state transitions and rewards depend on a single action or a combination of actions from multiple agents.

3.8.1 Single-Agent Environment

In a single-agent environment, there exists only one decision-making entity π interacting with the system. The environment's physical dynamics and the resulting state transitions $\mathcal{P}(s_{t+1}|s_t, a_t)$ are exclusively influenced by the single agent's action $a_t \in \mathcal{A}$. For example, ScienceWorld [34] evaluates the scientific reasoning of a single autonomous agent completing elementary science experiments within an interactive text environment. SWE-Bench [16] tasks a single LLM with autonomously editing large codebases to resolve real-world GitHub issues. OSWorld-MCP [53] assesses a single agent that autonomously selects between graphical actions and tool invocations to complete real-world computer tasks.

3.8.2 Multi-Agent Environment

In a multi-agent environment, a set of N autonomous agents, indexed by $i \in \{1, \ldots, N\}$, interact simultaneously within the shared environment $\mathcal{E}$. The action space expands to a joint action space formed by the Cartesian product of individual action spaces, $\boldsymbol{\mathcal{A}} = \mathcal{A}^1 \times \cdots \times \mathcal{A}^N$. Consequently, the environment's transition function is driven by the joint action $\boldsymbol{a}_t = \langle a_t^1, \ldots, a_t^N \rangle \in \boldsymbol{\mathcal{A}}$, formulated as $\mathcal{P}(s_{t+1}|s_t, \boldsymbol{a}_t)$. Each agent i is governed by its own policy π^i and receives a reward $\mathcal{R}^i(s_t, \boldsymbol{a}_t)$, which may be competitive, cooperative, or mixed, depending on the multi-agent formulation. For example, Generative agents [348] presents a simulated village containing twenty five unique characters powered by LLMs. It operates as an environment with multiple agents because these digital entities autonomously converse, form relationships, and organize collective activities. Collab-overcooked [349] introduces an operative cooking simulation where distinct LLMs manage separated resources and uneven information. To succeed, these isolated entities must actively converse and deliver items to achieve shared goals. AvalonBench [100] proposes a social deduction environment where LLMs are assigned hidden roles with asymmetrical knowledge. To succeed in this competitive game, agents must engage in strategic dialogue, utilizing deception and logical reasoning to either cooperate on tasks or actively sabotage the group's missions.

Takeaway 3

- **Multi-Agent Environment:** While most existing frameworks are limited to single-agent scenarios, future designs should incorporate multi-agent cooperative and competitive environments to realize collective intelligence and enhance parallel processing efficiency.
- **Symbolic-Neural Environment Joint Modeling:** Current agents face a deployment bottleneck, caught between rigid, schema-bound traditional frameworks (stable but inflexible) and fully generative world models (open but uncontrollable and inconsistent). Future infrastructures must bridge this gap by providing an intermediate environment that combines the engineering reliability of symbolic systems with the infinite generative scalability of neural models.

4 Environment Domain

As shown in Fig. 6, agent evaluation spans a diverse set of environments, with different domains placing substantially different demands on agents. Some environments focus on grounded perception and interface manipulation, some emphasize long-horizon planning and interactive decision making, and others require evidence synthesis, tool use, or domain-specific professional reasoning. Consequently, existing evaluation environments can be broadly organized into several domains, including **GUI** (Sec. 4.1), **Deep Research** (Sec. 4.2), **Embodied** (Sec. 4.3), **Game** (Sec. 4.4), **Tool** (Sec. 4.5), **Code** (Sec. 4.6), **Domain-Specific** (Sec. 4.7), and **Cross-Domain** (Sec. 4.8). This taxonomy reflects a broader shift in agent evaluation from narrow, task-specific benchmarks toward more diverse, realistic, and capability-compositional environments.

4.1 GUI

GUI environments denote settings in which agents complete tasks through the perception and manipulation of graphical user interfaces [15], [28], [52], [54]–[56], [59]–[61], [347], [355]. These environments emphasize the ability to comprehend, localize and interact with interface elements, as well as to make sequential decisions and execute actions based on interface feedback. Based on the primary interaction platforms, existing work can be broadly categorized into three groups: **Desktop GUI** (Sec. 4.1.1), **Mobile GUI** (Sec. 4.1.2), and **Web GUI** (Sec. 4.1.3).

4.1.1 Desktop GUI

Desktop GUI concerns tasks carried out in operating systems, where agents need to interact with files, windows, applications, and system tools. Compared with other GUI settings, this domain places greater demands on long-horizon planning, cross application coordination, and stable grounding in complex visual interfaces. OSWorld [28] evaluates open-ended tasks in realistic desktop environments and covers a wide range of everyday computer operations. WindowsAgentArena [52] further focuses on practical Windows workflows and highlights the challenges of interacting with native system components and desktop software. AgentStudio [347] complements these environments by providing a more general platform for building and evaluating GUI agents, which supports broader research on desktop interaction. Overall, Desktop GUI is moving toward more realistic and more general computer use, making it an important domain for evaluating general purpose agents.

4.1.2 Mobile GUI

Mobile GUI focuses on agent interaction with smartphone interfaces, where tasks are completed through tapping, typing, scrolling, and swiping on small screens. Compared with desktop settings, mobile environments are more constrained by limited display space, deeper page transitions, and stronger dependence on app specific layouts. AitW [55] studies Android control from human demonstrations and natural language instructions, establishing mobile GUI as an important grounded interaction problem. Mobile-Env [54] moves this line of research toward controllable evaluation by providing an interactive Android environment for task construction and testing. More recent benchmarks such as AndroidWorld [56] and MobileAgentBench [355] further emphasize executable and reproducible evaluation, pushing

TABLE 1
An overview of various environments categorized into **GUI** and **Deep Research** domains. Within the Modality column, specific icons are utilized to denote the supported data types where T represents text, I represents images, and V represents videos. Additionally, the ✓and ✗icons indicate the presence or absence of specific attributes across the columns, while the resource icons provide direct links to the respective GitHub or Hugging Face 🤗 repositories.

Domain	Name	Size	Modality	Observability	Multi-agent	Continuity	Online	Resource
GUI (Sec. 4.1)	WebShop [15]	500	T I	Partially	✗	Discrete	✓	GitHub
	Mind2Web [59]	1,341	T I	Partially	✗	Discrete	✗	GitHub
	Mind2Web2 [350]	120	T I	Partially	✗	Discrete	✓	GitHub
	WebArena [60]	812	T I	Partially	✗	Discrete	✓	GitHub
	VisualWebArena [61]	910	T I	Partially	✗	Discrete	✓	GitHub
	WebVoyager [227]	643	T I	Partially	✗	Discrete	✓	GitHub
	Mobile-Env [54]	74	T I	Partially	✗	Mixed	✓	GitHub
	AitW [55]	715,142	T I	Partially	✗	Mixed	✗	GitHub
	AndroidWorld [56]	116	T I	Partially	✗	Mixed	✓	GitHub
	AndroidControl [341]	721	T I	Partially	✗	Mixed	✗	GitHub
	MobileWorld [57]	201	T I	Partially	✗	Mixed	✓	GitHub
	AgentStudio [347]	205	T I V	Partially	✗	Mixed	✓	GitHub
	WorkArena [27]	19,912	T I	Partially	✗	Mixed	✓	GitHub
	OSWorld [28]	369	T I	Partially	✗	Mixed	✓	GitHub
	WindowsAgentArena [52]	154	T I	Partially	✗	Mixed	✓	GitHub
	OSWorld-MCP [53]	361	T	Partially	✗	Mixed	✓	GitHub
	Mobile-Bench [58]	832	T I	Partially	✗	Discrete	✓	GitHub
	MT-Mind2Web [351]	120	T	Partially	✗	Discrete	✗	Hugging Face
	VideoWebArena [352]	2,021	T I V	Partially	✗	Mixed	✓	GitHub
	Mind2Web-Live [353]	104	T I	Partially	✗	Discrete	✓	Hugging Face
	Online-Mind2Web [354]	300	T I	Partially	✗	Discrete	✓	Hugging Face
	MobileAgentBench [355]	100	T I	Partially	✗	Mixed	✓	GitHub
Deep Research (Sec. 4.2)	HLE [4]	2,500	T I	Partially	✗	Discrete	✓	Hugging Face
	DeepResearch Bench [71]	100	T	Partially	✗	Discrete	✓	GitHub
	GAIA [30]	466	T	Partially	✗	Discrete	✓	Hugging Face
	WideSearch [62]	200	T	Partially	✗	Discrete	✓	Hugging Face
	WebWalker [66]	680	T	Partially	✗	Discrete	✓	GitHub
	BrowseComp [31]	1,266	T	Partially	✗	Discrete	✓	GitHub
	Conflicts [67]	458	T	Partially	✗	Discrete	✓	GitHub
	DeepResearchGym [78]	1,000	T	Partially	✗	Discrete	✓	GitHub
	InfoDeepSeek [63]	245	T	Partially	✗	Discrete	✓	GitHub
	InfoSeek [64]	830	T	Partially	✗	Discrete	✓	GitHub
	BrowseComp-ZH [68]	289	T	Partially	✗	Discrete	✓	GitHub
	SurveyGen [74]	4,200	T	Partially	✗	Discrete	✗	GitHub
	ReportBench [75]	100	T	Partially	✗	Discrete	✓	GitHub
	LiveDRBench [69]	100	T	Partially	✗	Discrete	✓	GitHub
	ScholarQABench [76]	2,967	T	Partially	✗	Discrete	✗	GitHub
	ResearcherBench [356]	65	T	Partially	✗	Discrete	✓	GitHub
	ProxyQA [77]	100	T	Partially	✗	Discrete	✗	GitHub
	AI Idea Bench 2025 [357]	3,495	T	Partially	✗	Discrete	✗	GitHub
	DeepReview [358]	13,378	T	Partially	✗	Discrete	✗	GitHub
	PaperBench [359]	20	T	Partially	✗	Discrete	✗	GitHub
	Multimodal-DeepResearcher [72]	100	T I	Partially	✗	Discrete	✓	GitHub
	Vision-DeepResearch [360]	2,000	T I	Partially	✗	Discrete	✓	GitHub
	MMDR-Bench [65]	140	T I	Partially	✗	Discrete	✓	GitHub
	OmniGAIA [70]	360	T I V	Partially	✗	Discrete	✓	GitHub

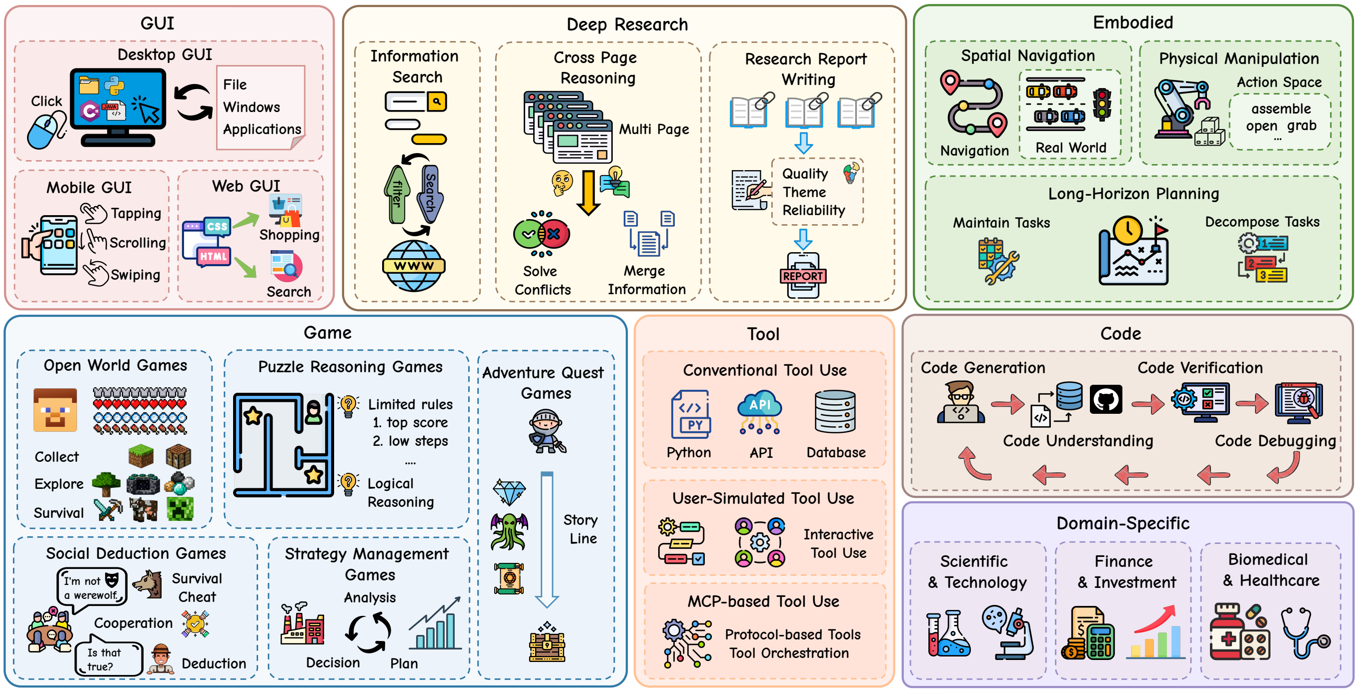

Fig. 6. An overview of environment domains, including GUI, Deep Research, Embodied, Game, Tool, Code, and Domain-Specific.

the field from static demonstration data toward realistic mobile task completion. Overall, Mobile GUI has gradually evolved from static demonstration-based settings toward more interactive and realistic mobile environments, making it a key domain for evaluating end-to-end mobile agents.

4.1.3 Web GUI

Web GUI centers on agent interaction with browser interfaces to perform tasks such as navigation, information retrieval, form filling, and shopping. Compared with other GUI settings, this domain combines textual content, visual layout, structured page elements, and dynamic interaction logic, which makes it one of the main testbeds for GUI agents. WebShop [15] frames online shopping as sequential decision making over webpages. Mind2Web [59] extends this setting to real websites from diverse domains and highlights generalization to unseen sites and tasks. WebArena [60] further advances the field by providing executable web environments for realistic long-horizon workflows, while VisualWebArena [61] shows that rendered visual information is often necessary for successful web interaction. Web GUI now serves as a central benchmark for studying agent performance in dynamic, information-rich, and workflow-oriented online environments.

4.2 Deep Research

Deep Research refers to the process where an agent continuously retrieves information, screens evidence, and synthesizes findings from diverse sources to address a research goal, ultimately generating answers grounded in evidence. Based on the major bottlenecks involved, existing work can be broadly divided into three categories (sections 4.2.1 to 4.2.3). The first [62], [63], [66] centers on finding relevant information, referred to as **Information Search**; the second [30], [31], [67], [68] emphasizes synthesizing information across sources, referred to as **Multi-Source Reasoning**; and the third [71], [73], [78] targets the production of structured research outputs, referred to as **Research Report Writing**.

4.2.1 Information Search

Information Search examines whether an agent can effectively search the open web and progressively extend its exploration to uncover high quality evidence. SimpleQA [29] evaluates the ability to retrieve precise factual information for question answering, capturing the most basic form of web information seeking. WideSearch [62] focuses on broad information collection across sources, highlighting the gathering and organization of large numbers of atomic facts. InfoDeepSeek [63] evaluates information gathering in Agentic RAG as a standalone capability, focusing on how agents decide whether to search, what to search for, and which evidence to retain. Overall, this line of work shifts Deep Research from single retrieval to persistent exploration and from finding a single answer to covering a body of valuable evidence.

4.2.2 Multi-Source Reasoning

Multi-Source Reasoning concerns whether an agent can synthesize information from multiple sources by comparing evidence, aligning claims, drawing inferences, and resolving conflicts. GAIA [30] requires models to combine reasoning, web browsing, and tool use to solve real world problems that often cannot be answered from a single page. WebWalker [66] also relates to this setting, as solving its tasks often requires agents to move through hierarchical website structures and connect evidence distributed across multiple linked pages. BrowseComp [31] further highlights this ability by requiring sustained navigation and the integration of scattered clues, making it well suited for evaluating cross page evidence synthesis. Conflicts [67] focuses explicitly on contradictions across sources, testing whether models can detect conflicts and distinguish their types. With benchmarks such as BrowseComp-ZH [68], this line of work has also expanded from English webpages to more complex Chinese information environments. Overall, the key capability in Deep Research is not reproducing a single source but relating multiple pieces of evidence to form a grounded conclusion.

4.2.3 Research Report Writing

Research Report Writing concerns whether an agent can transform collected evidence into structured, verifiable research outputs that are aligned with user needs. DeepResearch Bench [71] evaluates deep research agents with an emphasis on iterative web exploration, targeted retrieval, advanced synthesis, and report quality. DeepResearchGym [78] highlights reproducible evaluation by providing stable search interfaces and protocols for more controlled comparison of report generation. DR.BENCH [73] focuses more directly on long research reports, using a multidimensional framework that evaluates not only content quality but also topical focus and retrieval reliability. Overall, while earlier work mainly evaluates key stages of the research process, such as information search and multisource reasoning, this line of work shifts the focus toward the quality of the final deliverable.

4.3 Embodied

Embodied environments refer to settings in which an agent acts as a robot or virtual character situated in a 3D environment. In these settings, the agent completes tasks through perception, movement, and interaction [32], [33], [79]–[81], [84], [85], [87], [371], [372]. According to the primary task forms and capability requirements, existing work can be broadly categorized into three types: **Spatial Navigation** (Sec. 4.3.1), **Physical Manipulation** (Sec. 4.3.2), and **Long-Horizon Planning** (Sec. 4.3.3).

4.3.1 Spatial Navigation

Spatial Navigation environments center on tasks that require agents to move, explore, localize, and reach target positions in 3D spaces. This line of research mainly examines whether an agent can build usable spatial representations from egocentric observations and act reliably in unseen environments. Habitat [79] provides efficient photorealistic 3D simulation and supports standard embodied tasks such as point goal navigation. Room-to-Room [80] further extends this setting to language guided navigation in real building scale environments, highlighting the need to ground natural language instructions in perception and action. MetaDrive [81] broadens navigation oriented embodied evaluation to driving scenarios by generating diverse road environments and testing generalization under dynamic traffic conditions. Spatial navigation has become a core setting for evaluating spatial perception and generalization in embodied agents.

4.3.2 Physical Manipulation

Physical Manipulation environments focus on embodied interaction between agents and objects, including tasks such as grasping, placing, assembling and tool use. Compared with navigation tasks, this setting places greater emphasis on precise control, complex contact-rich interaction, and reasoning about object states. RLBench [84] provides a large scale manipulation benchmark with diverse vision-guided tasks and supports research on multi-task and imitation-based robot learning. Robocasa [87] extends this direction toward realistic everyday household tasks and language-conditioned mobile manipulation. BEHAVIOR [85] further improves realism by modeling a wide range of everyday household activities, while also showing that current embodied agents still exhibit clear limitations in complex physical interaction. Physical manipulation has expanded from narrow skill evaluation to more general and realistic object-centric interaction settings.

4.3.3 Long-Horizon Planning

Long-Horizon Planning environments center on multi step and compositional tasks whose success depends on maintaining task state, decomposing goals, and adapting to changing environments. In this setting, the main challenge is not only to execute individual actions, but also to organize them into coherent plans over long decision horizons. ALFRED [32] is a representative benchmark in this area, as it combines egocentric vision and natural language instructions with long household tasks that involve irreversible state changes. ALFWorld [33] complements this direction by aligning text based and embodied environments, making it possible to study abstract planning together with grounded execution. TEACh [371] further extends long-horizon embodied tasks to dialogue-enabled settings, where agents must complete household goals through both interaction and clarification. ReALFRED [372] pushes this line toward more realistic environments by reducing the visual and scene gap between simulation and real-world deployment. Long-horizon planning has become an important setting for evaluating planning consistency and long-term task execution in embodied agents.

4.4 Game

Game refers to scenarios in which an agent operates in a game world with explicit rules [36], [94], [95], goal constraints [91]–[93], and evolving states, continuously perceiving the environment, selecting actions, and adjusting its strategy based on feedback to accomplish interactive decision tasks. Based on differences in game mechanics and primary capabilities, existing work can be broadly categorized into five types: **Open World Games** (Sec. 4.4.1), **Puzzle Reasoning Games** (Sec. 4.4.2), **Social Deduction Games** (Sec. 4.4.3), **Adventure Quest Games** (Sec. 4.4.4), and **Strategy Management Games** (Sec. 4.4.5).

4.4.1 Open World Games

Open World Games examine whether an agent can sustain exploration and accomplish goals in highly open worlds with large state spaces and sparse feedback. MineDojo [91] is a representative benchmark in this direction, building a

TABLE 2
An overview of various environments categorized into **Embodied** and **Game** domains.

Domain	Name	Size	Modality	Observability	Multi-agent	Continuity	Online	Resource
Embodied (Sec. 4.3)	Habitat [79]	–	I	Partially	✗	Discrete	✓	GitHub
	Room-to-Room [80]	4173	T I	Partially	✗	Discrete	✓	GitHub
	RLBench [84]	100	T I	Partially	✗	Mixed	✓	GitHub
	ALFRED [32]	1,529	T I	Partially	✗	Discrete	✗	GitHub
	ALFWorld [33]	134	T	Partially	✗	Discrete	✓	GitHub
	Robocasa [87]	100	I	Partially	✗	Continuous	✓	GitHub
	BEHAVIOR [85]	100	I	Partially	✗	Continuous	✓	GitHub
	panda-gym [86]	80	T	Completely	✗	Continuous	✓	GitHub
	MetaDrive [81]	100	I	Partially	✓	Continuous	✓	GitHub
	ScienceWorld [34]	1,800	T	Partially	✗	Discrete	✓	GitHub
	ET-Plan-Bench [89]	11,838	T I	Partially	✗	Discrete	✓	GitHub
	LEGENT [90]	100	T I	Partially	✗	Mixed	✓	GitHub
	EmbodiedBench [164]	1,128	T I	Partially	✗	Mixed	✓	GitHub
	RoboFactory [88]	1,100	I	Partially	✓	Continuous	✓	GitHub
	Scenario Dreamer [346]	50,000	I	Partially	✗	Continuous	✓	GitHub
	SimuHome [361]	600	T	Partially	✗	Discrete	✓	GitHub
	Sari Sandbox [362]	100	T I	Partially	✗	Mixed	✓	GitHub
	Nexus [363]	1,090	I	Completely	✓	Continuous	✗	GitHub
Game (Sec. 4.4)	MineDojo [91]	3,142	T I	Partially	✗	Discrete	✓	GitHub
	KORGym [364]	55,200	T I	Partially	✗	Discrete	✓	GitHub
	TextArena [103]	74	T	Partially	✓	Discrete	✓	GitHub
	V-GameGym [344]	2,219	T	Completely	✗	Discrete	✗	🤗
	Kor-Bench [342]	1,250	T	Completely	✗	Discrete	✗	GitHub
	AI Gamestore [109]	2,100	I	Partially	✗	Mixed	✓	GitHub
	ING-VP [365]	300	I	Partially	✗	Discrete	✓	GitHub
	VideoGameBench [107]	50	I	Partially	✗	Mixed	✓	GitHub
	GVGAI-LLM [343]	540	T	Completely	✗	Discrete	✓	GitHub
	BALROG [106]	425	T I	Partially	✗	Discrete	✓	GitHub
	Smartplay [95]	931,900	T	Partially	✗	Discrete	✓	GitHub
	DSGBench [366]	160	I	Partially	✓	Discrete	✓	GitHub
	GTBench [112]	500	T	Partially	✓	Discrete	✓	GitHub
	PokéLLMon [367]	1,909	T	Partially	✓	Discrete	✓	GitHub
	MC-Planner [92]	2,130	T I	Partially	✗	Mixed	✓	GitHub
	GameTraversalBenchmark [94]	150	T	Completely	✗	Discrete	✓	GitHub
	LMRL Gym [99]	6,541	T	Partially	✗	Discrete	✓	GitHub
	Clembench [98]	250	T	Partially	✓	Discrete	✓	GitHub
	Baba Is AI [36]	1,275	I	Completely	✗	Discrete	✓	GitHub
	HLA [113]	330	T I	Completely	✓	Discrete	✓	GitHub
	LLMArena [116]	5,600	T	Partially	✓	Discrete	✓	GitHub
	TextGames [96]	24,000	T	Completely	✗	Discrete	✓	GitHub
	GameArena [35]	2,240	T	Partially	✓	Discrete	✓	GitHub
	SPIN-Bench [368]	1,280	T	Partially	✓	Discrete	✓	GitHub
	GAMEBoT [117]	2,720	T	Partially	✓	Discrete	✓	GitHub
	GameBench [37]	277	T	Partially	✓	Discrete	✓	GitHub
	MineWorld [93]	1,000	I	Partially	✗	Mixed	✓	GitHub
	AvalonBench [100]	1,140	T	Partially	✓	Discrete	✓	GitHub
	Werewolf [101]	1,200	T	Partially	✓	Discrete	✓	GitHub
	GameFactory [369]	7,407	I V	Partially	✗	Mixed	✗	GitHub
	GameNGen [370]	8,912	I V	Partially	✗	Mixed	✓	GitHub
	OGC [114]	37	I	Partially	✓	Discrete	✓	GitHub
	Minigrid & Miniworld [97]	35	T I	Partially	✗	Discrete	✓	GitHub

TABLE 3
An overview of various environments categorized into **Tool** and **Code** domains.

Domain	Name	Size	Modality	Observability	Multi-agent	Continuity	Online	Resource
Tool (Sec. 4.5)	API-Bank [40]	314	T	Completely	✗	Mixed	✓	GitHub
	ToolBench [38]	600	T	Completely	✗	Mixed	✓	GitHub
	ToolEyes [373]	382	T	Partially	✗	Discrete	✓	GitHub
	AppWorld [118]	585	T	Partially	✗	Discrete	✓	GitHub
	τ-bench [39]	165	T	Partially	✗	Mixed	✓	GitHub
	ACEBench [374]	2,000	T	Partially	✗	Discrete	✓	GitHub
	FlowBench [375]	536	T	Partially	✗	Discrete	✓	GitHub
	TRAJECT-Bench [376]	200	T	Partially	✗	Discrete	✗	🤗
	ETAPP [377]	800	T	Partially	✗	Mixed	✓	GitHub
	BFCL [119]	5,551	T	Partially	✗	Discrete	✓	GitHub
	UserBench [122]	471	T	Partially	✗	Discrete	✓	GitHub
	τ2-bench [121]	279	T	Partially	✓	Discrete	✓	GitHub
	MCPVerse [125]	250	T	Partially	✗	Mixed	✓	GitHub
	MCPToolBench++ [128]	1,509	T	Partially	✗	Discrete	✓	GitHub
	MCP-Universe [126]	231	T	Partially	✗	Mixed	✓	🤗
	MCP-Bench [127]	104	T	Partially	✗	Discrete	✓	GitHub
	MCPMark [378]	127	T	Partially	✗	Mixed	✓	GitHub
	M^3-Bench [379]	208	T I	Partially	✗	Discrete	✓	GitHub
Code (Sec. 4.6)	SWE-Bench [16]	2,294	T	Partially	✗	Discrete	✓	🤗
	InterCode [138]	1,351	T	Partially	✗	Discrete	✓	GitHub
	CodeAgent [129]	101	T	Partially	✗	Discrete	✓	GitHub
	BigCodeBench [131]	1,140	T	Completely	✗	Discrete	✓	🤗
	CodeElo [380]	387	T	Completely	✗	Discrete	✓	🤗
	LiveCodeBench [132]	511	T	Completely	✗	Discrete	✓	🤗
	CSR-Bench [136]	100	T	Partially	✗	Discrete	✓	GitHub
	SWE-Bench Pro [139]	1,865	T	Partially	✗	Discrete	✓	🤗
	Terminal-Bench [42]	89	T	Partially	✗	Discrete	✓	GitHub
	SWE-bench Multimodal [135]	517	T I	Partially	✗	Discrete	✓	🤗
	KernelBench [43]	250	T	Completely	✗	Discrete	✓	🤗
	IDE-Bench [327]	6,000	T	Partially	✗	Discrete	✓	GitHub
	MBPP [41]	2,383	T	Completely	✗	Discrete	✓	🤗
	CRUXEval [133]	800	T	Completely	✗	Discrete	✓	GitHub
	DebugBench [140]	4,253	T	Completely	✗	Discrete	✓	GitHub
	CodeRAG-Bench [381]	9,000	T	Partially	✗	Discrete	✓	GitHub
	SWT-Bench [137]	1,983	T	Partially	✗	Discrete	✓	GitHub
	FEA-Bench [130]	1,401	T	Partially	✗	Discrete	✓	GitHub
	Multi-SWE-bench [382]	1,632	T	Partially	✗	Discrete	✓	GitHub
	NL2Repo-Bench [134]	104	T	Partially	✗	Discrete	✓	GitHub

large suite of open world tasks on top of Minecraft that covers exploration, collection, crafting, and survival. MC Planner [92] further emphasizes multiple steps planning in Minecraft, highlighting the need for more detailed subgoal ordering and feedback correction in long-horizon tasks. MineWorld [93] shows that open world games can further evolve into persistent real time environments, where agents are no longer evaluated only on fixed task sets but instead make long term decisions in more continuous open worlds. Overall, open world games provide a testbed that more closely resembles complex real world environments.

4.4.2 Puzzle Reasoning Games

Puzzle Reasoning Games focus on whether an agent can reason effectively in tasks with constrained rules but high logical demands. Baba Is AI [36] is especially suited for evaluating rule rewriting and compositional reasoning. Agents must manipulate not only objects but also the rules themselves. GameTraversalBenchmark [94] abstracts planning as a two dimensional grid traversal problem, emphasizing whether an agent can reach the goal in as few steps as possible, and is therefore well suited for assessing path planning and state tracking. SmartPlay [95] decomposes multiple capabilities through mini games, enabling finer evaluation of abilities such as object based reasoning, spatial reasoning, and planning ahead. Overall, these environments make reasoning weaknesses easier to identify by placing complex capabilities within clearer rule structures.

4.4.3 Social Deduction Games

Social Deduction Games focus on communication, strategic interaction, and identity inference among multiple agents under incomplete information. AvalonBench [100] uses hidden roles, negotiation, and deception to evaluate multi agent language gaming abilities. Werewolf [101] further illustrates the importance of reasoning in environments where hidden roles and public discussion coexist. These social game settings require both logical analysis and effective dialogue generation. WhodunitBench [102] extends this line to multimodal murder mystery games, unifying perception, interaction, and cognition in one dynamic environment. In addition, more general competitive text game platforms such as TextArena [103] make it possible to compare negotiation, theory of mind, and deception capabilities across a broader range of games in a unified way. Overall, these environments make games an important testbed for collective reasoning and social interaction.

4.4.4 Adventure Quest Games

Adventure Quest Games focus on agents completing full task chains in scenarios involving story progression, clue collection, and multi stage objectives. FlashAdventure [105] directly evaluates whether an agent can complete an entire storyline, emphasizing that it must not only perform isolated actions but also sustain progress throughout the whole adventure progression. BALROG [106] takes a broader view of long-horizon interactive game tasks, integrating spatial reasoning, long term planning, and continued exploration across a range of challenging games. GameArena [35] further shows that interactive games can serve not only as task environments but also as dynamic evaluation platforms, capturing models' step by step reasoning during real gameplay. Overall, this direction shifts evaluation from one-shot problem solving to continuous task completion, thereby placing greater emphasis on capabilities such as retaining information, integrating clues, and planning over multiple stages.

4.4.5 Strategy Management Games

Strategy Management Games focus on high level decision making under resource constraints, long term returns, and complex competition. CivRealm [110] is built on a Civilization style environment. It unifies resource management, diplomacy, warfare, and long-term societal development within a single platform. This makes it well suited for studying macro level decision making. The Factorio Learning Environment [111] emphasizes long term planning in open factory systems through automation, resource optimization, and scalable production chains. While PokeLLMon [367] is set in a more strongly competitive tactical game scenario, it likewise highlights knowledge use, online feedback, and long term decision adjustment. Overall, these environments extend games from local rule solving to high level platforms for global utility optimization.

4.5 Tool

Tool Use refers to settings in which agents invoke external tools, such as functions, APIs, databases, or execution engines, to acquire information or execute actions to complete the task [38]–[40], [119], [121], [122], [125], [127], [128], [374]. Unlike GUI environments, tool use does not require visual interface manipulation. Instead, the interaction proceeds through structured tool calls and their corresponding outputs. According to whether tool use involves simulated users or is built upon MCP servers, existing work can be broadly categorized into three types: **Conventional Tool Use** (Sec. 4.5.1), **User-Simulated Tool Use** (Sec. 4.5.2), and **MCP-based Tool Use** (Sec. 4.5.3).

4.5.1 Conventional Tool Use

Conventional Tool Use focuses on settings where agents invoke external tools without interacting with explicit simulated users. In this setting, the main difficulty lies in selecting the correct tool, producing valid arguments, and executing tool calls in a reliable manner. API-Bank [40] evaluates whether models can select appropriate APIs and invoke them correctly under a fixed tool inventory. BFCL [119] further strengthens this setting by introducing a more systematic evaluation of function calls, covering serial and parallel calls, as well as structured verification of output correctness. ToolBench [38] expands the scale of tool space by exposing models to a large collection of real-world APIs and evaluating both single-tool and multi-tool problem solving. AppWorld [118] further advances tool-use evaluation toward executable app-based environments, where agents need to interact with multiple application APIs and complete tasks under state-based verification. In general, conventional tool use has become a core setting for evaluating whether agents can accurately use tools in standard tool call environments.

4.5.2 User-Simulated Tool Use

User-Simulated Tool Use studies settings in which tool use is embedded in interactions with simulated users. Compared with conventional tool use, these tasks place greater emphasis on responding to evolving user intentions, clarifying underspecified requests, and adjusting tool-use behavior according to conversational feedback. τ-bench [39] highlights this setting by embedding tool use into dynamic user interactions, where agents must follow policies while updating their actions according to conversation progress. UserBench [122] further pushes this setting toward more user-centered workflows, as agents are required to clarify underspecified goals and refine their decisions as user preferences are gradually revealed. τ^2-bench [121] further extends this paradigm by introducing a dual-control interactive environment, where both agents and simulated users can influence the environment through tool use, requiring more adaptive and coordinated decision-making. Overall, this class of benchmarks shifts evaluation from isolated tool use to interactive, user-centered settings, emphasizing adaptability and alignment with user intent.

4.5.3 MCP-based Tool Use

MCP-based Tool Use concerns settings in which agents interact with tools provided through MCP servers. The central challenge in this setting is not only correct tool invocation, but also effective tool discovery, task decomposition, and coordination across a large collection of MCP tools. MCPVerse [125] significantly expands the scale of tool space by incorporating hundreds of executable tools, making tool retrieval and orchestration substantially more demanding. MCPToolBench++ [128] further evaluates model performance in MCP-based environments with diverse tools and execution settings. MCP-Bench [127] emphasizes realistic multi-step tasks over live MCP servers, requiring agents to identify relevant tools, plan execution sequences, and integrate outputs across different sources. Overall, these works extend tool-use evaluation from general tool calling to standardized MCP-based tool ecosystems.

4.6 Code

Code refers to scenarios in which an agent interacts continuously with code, repositories [129], [135], [136], test harnesses [136], [138], and other related programming artifacts in order to accomplish a programming objective. Existing work covers different stages of the software development process, and the boundaries between benchmarks are often not strict. Based on the primary bottlenecks involved, they can be broadly categorized into four types: **Code Generation** (Sec. 4.6.1), **Code Understanding** (Sec. 4.6.2), **Code Verification** (Sec. 4.6.3), and **Code Debugging** (Sec. 4.6.4).

4.6.1 Code Generation

Code Generation focuses on producing new code from natural language requirements or problem descriptions. MBPP [41] is an early benchmark for function level program synthesis and serves as a basic reference for code generation ability. BigCodeBench [131] increases task complexity by moving beyond short algorithmic problems to emphasize complex instruction understanding and multi-function calling, making evaluation closer to real development needs. CodeAgent [129] further extends this setting to repository scale code generation, where models must generate solutions under complex project structures and dependencies. LiveCodeBench [132] highlights continual updates and contamination control, helping distinguish whether models solve tasks through memorization or genuine coding ability. Overall, this direction has evolved from simple static code generation to more complex, dynamic, and realistic settings.

4.6.2 Code Understanding

Code Understanding focuses on whether agents can understand repository structure, interface relationships, dependency paths, and task context. NL2Repo-bench [134] evaluates whether models can understand natural language requirements, locate relevant files and modules, and reason over project structure rather than isolated code snippets. CSR-Bench [136] highlights repository understanding in deployment tasks by requiring agents to read documentation, analyze directory structures, and generate executable commands. SWE-bench-Multimodal [135] further shows that software context may also include interfaces, screenshots, and other visual information. Overall, code understanding is expanding from reading functions to interpreting repositories, documentation, and multimodal software context.

4.6.3 Code Verification

Code Verification focuses on how agents use tests, compilation, execution results, and performance metrics to verify whether a candidate solution is correct. MBPP [41] and BigCodeBench [131] rely on executable tests for stable requirement checking. LiveCodeBench [132] further integrates execution, self-repair, and test evaluation in a dynamic setting, making verification part of the full coding process rather than just a final scoring step. KernelBench [43] extends this idea to performance oriented scenarios by evaluating both correctness and efficiency gains. Overall, code verification provides a valuable closed loop through objective execution feedback.

4.6.4 Code Debugging

Code Debugging involves locating and fixing problems based on error messages, test failures, execution anomalies, and environmental feedback. InterCode [138] makes debugging an interactive repair process through execution feedback. SWE-Bench [16] requires root cause analysis and multi-step fixes in real software contexts. CSR-Bench [136] further extends debugging to repository deployment scenarios involving dependencies, scripts, and experimental environments. This shows that **Code Debugging** has expanded from single file bug fixing to system level and environment level problem solving.

4.7 Domain-Specific

Domain-specific environments evaluate whether agents can effectively operate in specialized disciplinary or industrial settings [44], [45], [141]–[143], [145], [147]–[149], [384]. They require agents to understand domain-specific knowledge and operate in accordance with professional practices. In this section, we broadly divide existing work into three categories: **Biomedicine and Healthcare** (Sec. 4.7.1), **Science and Technology** (Sec. 4.7.2), and **Finance and Investment** (Sec. 4.7.3).

4.7.1 Biomedical and Healthcare

Biomedical and Healthcare environments evaluate whether agents can operate effectively in clinical and biomedical settings. Agents in these environments need to understand clinical terminology, biomedical data, and bioinformatics workflows, while acting reliably under professional constraints. MedAgentBench [44] studies agent behavior in clinical tasks derived from electronic health records and physician workflows. MedAgentGym [141] provides an executable environment for training and evaluating medical agents at scale. Beyond clinical settings, BioAgent Bench [142] focuses on bioinformatics workflows, while BioCoder [143] evaluates coding ability on biomedical programming tasks.

4.7.2 Science and Technology

Science and Technology environments study whether agents can support research activities such as literature understanding, experiment design, and result analysis. DiscoveryWorld [384] is a representative benchmark that places agents in interactive scientific tasks requiring multi-step reasoning and experimentation. ScienceAgentBench [45]

TABLE 4
An overview of various environments categorized into **Domain-Specific** and **Cross-Domain**.

Domain	Name	Size	Modality	Observability	Multi-agent	Continuity	Online	Resource
Domain-Specific (Sec. 4.7)	Biocoder [143]	460	T	Completely	✗	Discrete	✓	GitHub
	TravelPlanner [9]	1,225	T	Partially	✗	Discrete	✓	🤗
	DSEval [144]	825	T	Partially	✗	Discrete	✓	GitHub
	Natural Plan [383]	3,600	T	Completely	✗	Discrete	✗	GitHub
	ScienceAgentBench [45]	102	T	Completely	✗	Discrete	✓	🤗
	DSBench [46]	540	T I	Partially	✗	Discrete	✓	🤗
	MLE-bench [145]	75	T I V	Partially	✗	Discrete	✓	GitHub
	DiscoveryWorld [384]	120	T I	Partially	✗	Discrete	✓	GitHub
	MedAgentBench [44]	300	T	Partially	✗	Discrete	✓	🤗
	StockBench [148]	82	T	Partially	✗	Mixed	✓	GitHub
	MLE-Dojo [146]	50	T	Partially	✗	Discrete	✓	GitHub
	FinDeepResearch [147]	64	T	Partially	✗	Mixed	✓	🤗
	PaperArena [385]	784	T I	Partially	✗	Discrete	✓	GitHub
	BixBench [386]	205	T I	Partially	✗	Discrete	✓	🤗
	CRMArena-Pro [387]	8,614	T	Partially	✗	Discrete	✓	🤗
	MedAgentGym [141]	13,238	T	Partially	✗	Discrete	✓	GitHub
	Finance Agent Benchmark [149]	337	T	Partially	✗	Discrete	✓	🤗
	EcomBench [388]	100	T	Partially	✗	Discrete	✗	🤗
	MedAgentBench v2 [389]	300	T	Partially	✗	Discrete	✓	GitHub
	MedMCP-Calc [169]	118	T	Partially	✗	Mixed	✓	GitHub
	ESG [390]	291	T V	Partially	✓	Discrete	✓	GitHub
	BioAgent Bench [142]	–	T	Partially	✗	Discrete	✓	GitHub
	WoW-bench [391]	234	T	Partially	✗	Discrete	✓	GitHub
	EnterpriseOps-Gym [392]	1,150	T	Partially	✗	Discrete	✓	🤗
	MetaClaw [393]	934	T	Partially	✗	Discrete	✓	GitHub
	Claw-Eval [394]	300	T I V	Partially	✗	Discrete	✓	🤗
	ClawArena [395]	1,879	T	Partially	✗	Discrete	✓	GitHub
Cross-Domain (Sec. 4.8)	OpenAI Gym [47]	–	T I	Partially	✗	Mixed	✓	GitHub
	HuggingGPT [150]	3,497	T I	Completely	✗	Discrete	✓	GitHub
	AgentBench [48]	1,014	T	Partially	✗	Mixed	✓	GitHub
	TaskLAMA [339]	478	T	Completely	✗	Discrete	✓	🤗
	TaskBench [340]	17,331	T I V	Completely	✗	Discrete	✗	GitHub
	AgentBoard [151]	1,012	T	Partially	✗	Discrete	✓	🤗
	AgentGym [396]	20,509	T	Partially	✗	Discrete	✓	GitHub
	WorfBench [397]	2,146	T	Completely	✗	Discrete	✗	GitHub
	GEM [3]	–	T I	Partially	✗	Discrete	✓	GitHub
	MLGym [398]	–	T	Partially	✗	Discrete	✓	GitHub
	MedBrowseComp [399]	1,139	T	Partially	✗	Mixed	✓	🤗
	WebMMU [400]	4,240	T I	Completely	✗	Discrete	✗	🤗
	TPS-Bench [401]	200	T	Partially	✗	Discrete	✓	GitHub
	AutoEnv [170]	–	T	Partially	✗	Discrete	✓	GitHub
	AgencyBench [402]	138	T	Partially	✗	Discrete	✓	🤗
	AgentVista [403]	209	T	Partially	✗	Discrete	✓	🤗

evaluates agents on realistic research-oriented problems grounded in scientific practice. MLE-bench [145] further extends this setting to machine learning research and engineering, where agents must handle data, models, and iterative experimentation.

4.7.3 *Finance and Investment*

Finance and Investment environments evaluate whether agents can analyze financial information and make decisions in dynamic and risk sensitive settings. StockBench [148] focuses on stock trading tasks and tests whether agents can act on market signals, news, and financial indicators. FinDeepResearch [147] studies deep financial analysis and emphasizes evidence collection and structured reasoning. Finance Agent Benchmark [149] further evaluates research oriented financial tasks built around realistic analyst style workflows.

4.8 Cross-Domain

Cross-Domain environments evaluate whether agents can generalize across heterogeneous tasks and settings, rather than operate only within a single domain. OpenAI Gym [47] established a unified interface for benchmarking agent behavior across diverse environments. AgentBench [48] eval-

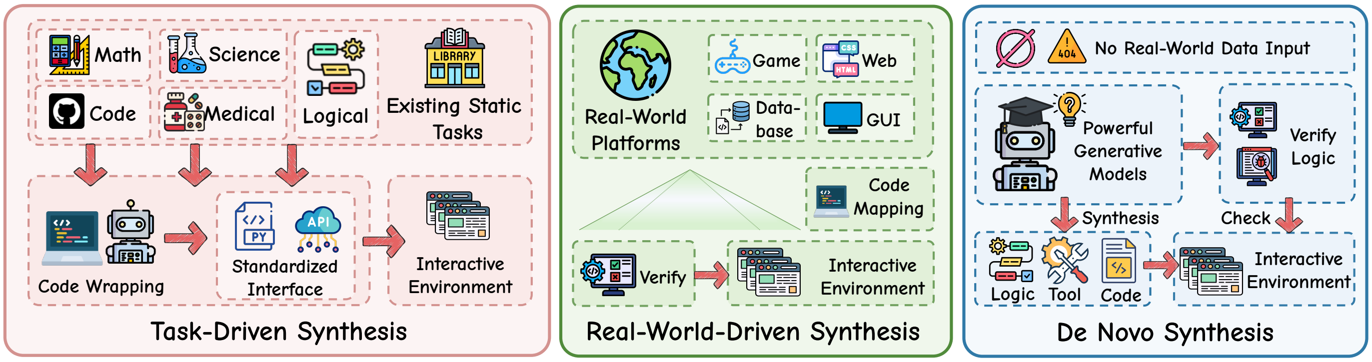

Fig. 7. Three symbolic environment synthesis methods are presented: Task-Driven Synthesis, Real-World-Driven Synthesis, and De Novo Synthesis. From left to right, the methods offer increasing degrees of freedom and require more verification logic.

uates reasoning and decision making in multiple interactive environments under a common evaluation setting. AgentBoard [151] further extends this direction by providing a unified analytical framework with finer-grained process-level metrics, making it easier to compare agent performance beyond final success rate. Recent work also begins to study broader transfer and learning across environments. GEM [3] provides a Gym-style framework for training and evaluating agentic language models in diverse environments, while AutoEnv [170] moves a step further by automatically generating heterogeneous worlds for measuring cross-environment adaptation. Overall, this line of research shifts evaluation from single-domain competence to cross-environment robustness, transferability, and scalable learning.

4.9 Summary

Takeaway 4

- **Capability Focus Across Domains:** GUI, Tool, and Code domains stress grounded external operation and workflow execution; Embodied and Game domains emphasize perception, control, and planning; Deep Research and Domain-Specific domains require evidence synthesis, knowledge-grounded reasoning, and reliable decision making; while Cross-Domain settings evaluate transfer across heterogeneous environments.
- **Evolution of Environment Design:** Agent environments are evolving from static and narrow benchmarks toward executable, multimodal, and long-horizon settings. Future environment design should better balance realism, diversity, controllability and verifiability to support both robust training and reliable evaluation of general agents.

5 Environment Synthesis

Large-scale, high-quality training environments are crucial for improving the capabilities of agents. However, relying on manual construction severely limits the scalability of both environment quantity and diversity. Consequently, recent research increasingly explores automated approaches to synthesize training environments at scale. Based on the form of the resulting environments, we categorize these methods into **Symbolic Synthesis** (Sec. 5.1) and **Neural Synthesis** (Sec. 5.2). Symbolic synthesis constructs environments using symbolic rules such as code, while the environment in neural synthesis is represented by a neural model (e.g., a world model).

5.1 Symbolic Synthesis

Formally, a symbolic environment can be defined as a tuple $\mathcal{E} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$. Here $\mathcal{S}$ and $\mathcal{A}$ represent the state and action spaces. The core characteristic of symbolic synthesis is that the state transition $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ and the feedback mechanism $\mathcal{R}$ are explicitly controlled by a set of symbolic rules or executable code $\mathcal{C}$. These environments are driven by underlying rules to ensure reliable feedback. To facilitate agent training, symbolic synthesis is widely used to obtain diverse, high-quality, and verifiable environments at scale [404]–[406]. As shown in Fig. 7, based on the source of synthesis logic and the degree of design freedom, we categorize these methods into three types: **(1) Task-Driven Synthesis** (Sec. 5.1.1), which transforms static datasets such as mathematics and coding problems into interactive environments through code encapsulation; **(2) Real-World-Driven Synthesis** (Sec. 5.1.2), which constructs simplified virtual mappings of real-world complex interactive environments like web pages and operating systems; **(3) De Novo Synthesis** (Sec. 5.1.3), which leverages generative models to autonomously build environments equipped with tool and verification logic without relying on predefined instances.

5.1.1 Task-Driven Synthesis

Task-driven synthesis denotes transforming existing static tasks [156], [157], [407]–[409] and data such as tool-calling [154], [161], [410] or mathematical data [158], [411] into interactive environments by wrapping them with programmatic rules, as shown in Fig. 7. This approach leverages the abundance of high-quality static data to ensure the correctness and scalability of the environments. Next, we introduce in detail the developmental trajectory of these methods.

A significant portion of task-driven synthesis methods focus on coding agent environments [155], [412]. Early work includes SWE-Gym [152] which packages issues and pull requests (PRs) from 11 Python repositories sourced from GitHub into executable Python environments using Docker. To overcome the limited scale of early work, recent research explores fully automated scaling [159], [411]. For example, Scale-SWE [160] and SWE-Hub [413] propose an automated environment synthesis paradigm based on multi-agent collaboration. Scale-SWE [160] decomposes the synthesis task into three agents: Environment Builder, Unit-test Creator, and Problem Statement Writer. SWE-Hub [413] proposes a similar concept (e.g., Env Agent, Bug Agent) and further emphasizes the automatic generation of more realistic and complex bug tasks at the system level. SCALER [318] utilizes competitive programming data to synthesize verifiable environments with adjustable difficulty, providing stable reward signals for RL. As environment volumes grow, environment configuration costs escalate, prompting efficiency optimizations. MEnvAgent [414] employs an incremental patching mechanism to enable environment reuse, while SWE-smith [153] proposes sharing a single environment image across thousands of tasks within the same repository. Conversely, DockSmith [415] trains specialized models to automate the generation and repair of Dockerfiles to improve construction efficiency. Beyond standard software engineering, CLI-Gym [416] shifts focus to command line interfaces through an agentic environment inversion paradigm where agents intentionally break environments to generate historical error data.

Beyond coding tasks, other studies explore broader tool usage environments. AgentScaler [154] focuses on the number and breadth of environments by using 30,000 heterogeneous APIs gathered from existing datasets to construct diverse environments. EnvScaler [161] focuses on synthesizing complex state dependencies to increase the difficulty and depth of environments through programmatic synthesis and dual agent filtering strategies. Additionally, Agentic CPT [410] extends this paradigm by transforming unstructured corpora from Wikipedia or CommonCrawl into interactive environments for continual pre-training.

Further research investigates knowledge intensive environments in vertical domains. MedAgentGym [141] standardizes static biomedical datasets into code centric reasoning tasks. SciAgentGym [417] introduces execution grounded synthesis by equipping existing scientific benchmarks with real tools and verification mechanisms. PaperArena [385] constructs literature analysis environments for complex cross document reasoning. FinMTM [418] synthesizes financial environments based on industrial reports for interaction through MCP. V-GameGym [344] incorporates visual rendering as feedback, marking the evolution of environment synthesis from text-only to multimodal streams.

Finally, recent efforts integrate multi domain tasks to explore idealized environment synthesis. LLM-in-Sandbox [158] presents a unified perspective by utilizing standard operating systems as universal sandboxes for all tasks. This demonstrates that non-coding tasks like mathematics can also achieve substantial performance gains through interaction within encapsulated environments.

5.1.2 Real-World-Driven Synthesis

Real-world-driven synthesis projects real-world complex interaction mediums (e.g., Web and Game) into simplified virtual environments for training and evaluation [108], [109], [162], [166]–[168], [406], [419]–[421]. Given the immense volume and vast action spaces of real world systems, this paradigm achieves significantly greater scalability than task driven approaches. Illustrated by Fig. 7, these methods basically project complex, diverse real-world environments into simplified forms.

Traditional real world computing systems include GUI, OS, and complex software tools. Some frameworks leverage these authentic applications to construct massive scale environments [155], [422] designed to generate high quality and long horizon trajectory data. To automate scaling, AgentSynth [162] exploits information asymmetry where progressively generating and executing simple subtasks is significantly easier than solving complex long horizon problems in a single attempt. This mechanism enables the low cost synthesis of extensive planning tasks with controllable difficulty. TaskCraft [165] similarly automates the synthesis of tool invocation environments with adjustable difficulty levels grounded in external web environments. Recently, the emergence of the MCP addresses the fragmentation of real-world interfaces. OSWorld-MCP [53] pioneers the integration of GUI with standardized protocol tool invocations within synthetic environments to bridge the gap between pure visual interactions and text based tool usage. VeriEnv [167] utilizes generative models to directly clone real-world websites for virtual environments. To stress test agent capabilities in realistic settings, MCPMark [378] synthesizes complex tasks with profound operational depth directly upon authentic protocol servers (e.g., GitHub and Notion). Applying this technology to vertical domains, MedMCP-Calc [169] constructs highly rigorous healthcare environments by integrating real electronic medical record databases and clinical guideline retrieval systems.

Recent research also simulates environments with visual feedback and physical constraints. AI Gamestore [109] proposes a multiverse of human games as a novel standard for evaluating models by synthesizing open-ended evaluation environments from digital platforms like Steam. lmgame-Bench [108] encapsulates diverse video games into standardized Gymnasium style APIs through modular harnesses. Shifting focus to game development, GameDevBench [419] leverages video tutorials and testing frameworks inherent to game engines. This approach translates hard to quantify visual feedback into executable testing tasks to evaluate agent proficiency in multimodal code generation. Finally, EmbodiedBench [164] achieves the systematic synthesis ranging from high level semantic planning to low level micro control (e.g., manipulation of robotic arms).

5.1.3 De Novo Synthesis

Emerging de novo synthesis methods aim to directly generate diverse interactive environments utilizing minimal seed examples or zero shot approaches, as demonstrated in Fig. 7. Compared to previous paradigms, this method offers the highest degree of freedom and the most expansive environmental space [170]–[173], [177], [404]–[406]. It closely approximates the ideal scenario of freely scalable environments.

The primary challenge of de novo synthesis involves ensuring rigorous internal logic within generated systems. AutoForge [175] addresses the instability of previous reverse synthesis by prompting generative models to first construct scalable state structures and model underlying logic graphs based on tool invocation relationships before producing code. LOGIGEN [171] advances this by introducing logic driven forward deductive synthesis. This framework utilizes an architect agent to compile rules into physical environments supported by SQLite databases, fundamentally altering traditional reverse synthesis pathways. AutoEnv [170] focuses on cross domain generalized generation by hierarchically decoupling the underlying logic of the environment from the surface interaction, successfully yielding datasets for numerous highly heterogeneous environments entirely from scratch.

Automated verification is equally crucial since generated environments inevitably suffer from model hallucinations. Within SWE domains, ScaleEnv [174] proposes methodologies based on procedural testing and graph expansion to guarantee code correctness and task diversity without human intervention. Agent World Model [11] ensures quality stability during massive scaling by accurately restoring the software development process and embedding execution level self correction mechanisms. Furthermore, SWE-Playground [176] constructs a pioneering purely synthetic training pipeline that completely removes the dependency on external data sources like GitHub.

Fully automated synthesis extends broadly to other domains. Endless Terminals [423] synthesizes thousands of terminal environments from scratch by sampling dimensions such as file operations and network configurations. To mitigate data contamination, gg-bench [424] employs random sampling to invent entirely novel two-player games. These games are then automatically encapsulated into standard Gymnasium interfaces to establish dynamic benchmarks.

5.1.4 Summary

Takeaway 5.1

- **Evolution of Scalability:** The progression from task driven to real world driven and to de novo synthesis methods represents a continuous expansion in synthesis freedom and overall environment space. This trajectory closely aligns with the core vision and ultimate objective of environment scaling in contemporary agent research [425].
- **Future Directions:** As paradigms gradually reduce dependence on real-world data, synthetic environments also encounter critical challenges. To ensure robust training, critical issues concerning **task diversity**, environmental **complexity**, execution **correctness**, generation **quality**, and **distributional bias** relative to real world scenarios must be concurrently resolved.

5.2 Neural Synthesis

Neural synthesis refers to modeling the environment using a neural network. Such approaches are typically centered around world models, which are neural models that learn to simulate the environment's state transitions and observations, enabling agents to interact with a learned environment instead of the real one. Depending on the form of representation used for environment modeling, we categorize them into three types: **(1) Pixel-Level Modeling** (Sec. 5.2.1), which directly models environment dynamics from pixel-level observations such as images or videos; **(2) Word-Level Modeling** (Sec. 5.2.2), which represents environment states in natural language; and **(3) Latent-Level Modeling** (Sec. 5.2.3), which learns structured representations of the environment in a latent space and captures the dynamics through low-dimensional embeddings. An overview of these paradigms is illustrated in Fig. 8.

5.2.1 Pixel-Level Modeling

Pixel-level modeling refers to modeling the environment directly in the raw visual observation space. Rather than relying on semantic abstraction or representation compression, this paradigm learns environment dynamics from high-dimensional visual signals [180], [182], [337], thereby preserving richer physical detail and scene information.

Some approaches focus primarily on modeling the temporal evolution of the visual world with limited emphasis on complex interaction. A representative approach [430] simulates game environments using a world model that compresses spatial representations with a VAE and models temporal dynamics as a probability distribution with an MDN-RNN, where the stochasticity is controlled via a temperature parameter. Subsequent research shifts toward using world models for agent decision-making and interaction, introducing action-conditioned modeling so that the model can predict future visual observations given agent behavior. In embodied tasks, EVA [178] integrates a vision-language model with a video generative model, enabling the framework to generate future visual observations conditioned on language-described actions. In GUI tasks, ViMo [179] uses natural language instructions rather than abstract action codes, and trains an image predictor with both visual features and text instructions as dual conditioning inputs.

Other approaches extend world models to support long-horizon interactive environments. Matrix-Game [431] trains on a large-scale Minecraft dataset to enable interactive video generation conditioned on continuous keyboard and mouse inputs, supporting temporally extended trajectories rather than short clips. Its successor Matrix-Game 2.0 [432] further improves long-horizon consistency by introducing a scalable data production pipeline and a frame-level action injection module, enabling stable interactive generation at the minute scale. NeuralOS [181] targets operating system environments with persistent and evolving states. It models long-range dependencies by maintaining system state with a hierarchical RNN, while combining diffusion-based rendering and multi-stage training to mitigate exposure bias over extended interaction sequences.

5.2.2 Word-Level Modeling

Word-level modeling benefits from the expressive and general-purpose nature of natural language, making it well-suited for modeling environments across diverse domains [191], [194], [433]. Compared to pixel-level modeling, it provides a higher-level representation with lower computational cost, and is more capable of handling complex reasoning and long-horizon planning tasks.

TABLE 5
Representative works for symbolic synthesis. **Base Model** refers to the model used for environment synthesis; "-" indicates that no model was used or that it was not specified. **Data Source** refers to the data on which the environment synthesis was based.

Name	Modality	Architecture	Base Model	Task Domain	Data Source	Evaluation
			Task-Driven Synthesis (Sec. 5.1.1)			
Planetarium [426]	T	LLM	-	Code	IPC	Correctness, Complexity
SWE-Gym [152]	T	LLM	-	Code	GitHub	Correctness
R2E-Gym [159]	T	LLM	Sonnet-3.5-V2	Code	GitHub	Correctness, Complexity
VML PDDL [427]	T	LLM	Qwen2.5-Coder	Code	IPC	Correctness
TheoryCoder [428]	T	LLM	GPT-4o	Game	VGDL, BabyAI, etc.	Correctness
SWE-smith [153]	T	LLM	o3, Claude 3.7	Code	GitHub	Correctness, Complexity
MedAgentGym [141]	T	LLM	GPT-4.1-mini, etc.	Domain-Specific	MIMIC-III, eICU, etc.	Correctness, Complexity, Fidelity
V-GameGym [344]	T I V	VLM	Claude-Sonnet-4	Game	OpenCoder, The Stack v2	Correctness, Diversity, Complexity
Agentic CPT [410]	T	LLM	LLM	Tool	CommonCrawl, Wikipedia,etc.	Correctness
AgentScaler [154]	T	LLM	LLM	Tool	ToolBench, API-Gen	Correctness
PaperArena [385]	T I	VLM	GPT-5, Claude 4, Gemini-2.5	Domain-Specific	OpenReview, Open Access	Diversity, Complexity
Agent2World [155]	T	LLM	GPT-4.1-mini, Llama-3.1-8b	Domain	LIMA	Correctness, Fidelity
SCALER [318]	T	LLM	GLM-4.6	Domain	CodeContests	Correctness, Diversity, Complexity
MEnvAgent [414]	T	LLM	Kimi-K2, Gemini-3, Claude-4.5	Code	GitHub	Correctness, Complexity
EnvScaler [161]	T	LLM	GPT-4.1, Qwen3-235B	Tool	API-Bank, ToolACE	Correctness, Diversity, Complexity
LLM-in-Sandbox [158]	T	LLM	-	Domain	Instruction Pre-Training, etc.	Correctness
DockSmith [415]	T	LLM	Qwen3*	Code	GitHub	Correctness, Complexity
Scale-SWE [160]	T	LLM	DeepSeek, Gemini-3	Code	GitHub, PyPI	Correctness, Diversity
SciAgentGym [417]	T I	VLM	-	Domain-Specific	SciInstruct, GPQA, BMMR, etc.	Correctness, Diversity, Complexity, Fidelity
CLI-Gym [416]	T	LLM	LLM	Code	SWE-smith, GitHub	Correctness
FinMTM [418]	T I	VLM	Gemini-3-Pro	Domain-Specific	-	Correctness, Diversity, Complexity
Nemotron-Terminal [411]	T	LLM	DeepSeek-V3.2	Code	OpenMathReasoning, etc.	Correctness
Hybrid-Gym [429]	T	LLM	GPT-4o-mini, Claude-4.5, etc.	Code	GitHub, SWE-Gym, RepoST	Correctness, Diversity
SWE-Hub [413]	T	LLM	LLM	Code	GitHub	Correctness, Diversity
			Real-World-Driven Synthesis (Sec. 5.1.2)			
EnvGen [163]	T I	VLM	GPT-4	Embodied	Crafter,Heist	Correctness, Fidelity
EmbodiedBench [164]	T I	VLM	GPT-4	Embodied	ALFRED, AI2-THOR, etc.	Correctness, Fidelity
lmgame-Bench [108]	T I	VLM	o3	Game	video games	Correctness, Complexity
AgentSynth [162]	T I	VLM	GPT-4.1	Domain	OSWorld,persona hub	Correctness, Complexity
TaskCraft [165]	T I	VLM	GPT-4.1	GUI, Tool	Web	Correctness, Complexity
OSWorld-MCP [53]	T I	VLM	o3	GUI	OSWorld	Correctness, Complexity
MedMCP-Calc [169]	T	LLM	GPT-OSS, DeepSeek-V3.1, etc.	Domain-Specific	MDCalc, MIMIC-IV, etc.	Correctness
AI GameStore [109]	T I	VLM	Claude-4.5, Gemini-2.5	Game	App Store, Steam	Correctness, Complexity, Fidelity
GameDevBench [419]	T I V	VLM	GPT-5	Game	YouTube, Godot 4	Correctness, Complexity, Fidelity
AutoWebWorld [168]	T I	VLM	GPT-5.1, Gemini-3, etc.	GUI	Facebook, Gmail, etc.	Correctness, Complexity
SWE-Universe [422]	T	LLM	Qwen-Next-80B-A3B*	Code	GitHub	Correctness
VeriEnv [167]	T I	VLM	GPT-5.2	GUI	Web	Correctness, Complexity, Fidelity
DIVE [166]	T	LLM	Claude-4-Sonnet	Tool	Wikipedia, PubMed, etc.	Correctness, Complexity
KAT-Coder-V2 [421]	T	LLM	Claude	Code	GitHub, etc.	Correctness, Complexity
			De Novo Synthesis (Sec. 5.1.3)			
RandomWorld [177]	T	LLM	GPT-4o	Tool	-	Correctness, Complexity
NL2Plan [173]	T	LLM	GPT-4o	Tool	-	Correctness, Complexity, Fidelity
gg-bench [424]	T	LLM	OpenAI o1	Game	-	Correctness, Diversity, Complexity
AutoEnv [170]	T	LLM	Claude-4-Sonnet	Domain	-	Correctness, Diversity, Complexity
AutoForge [175]	T	LLM	Qwen3-Thinking	Code, Tool	-	Correctness, Diversity, Complexity
SWE-Playground [176]	T	LLM	Gemini-2.5, GPT-4.1, etc.	Code	-	Correctness
InfiniteWeb [172]	T I	VLM	GPT-5	GUI	-	Correctness, Diversity
Endless Terminals [423]	T	LLM	o3	Code	-	Correctness, Complexity
Agent World Model [11]	T I	LLM	GPT-5	Tool	-	Correctness, Diversity
ScaleEnv [174]	T	LLM	Deepseek-V3.2, GPT-5.1, etc.	Code	-	Correctness, Diversity, Complexity
LOGIGEN [171]	T	LLM	DeepSeek-V3.2	Tool	-	Correctness, Complexity

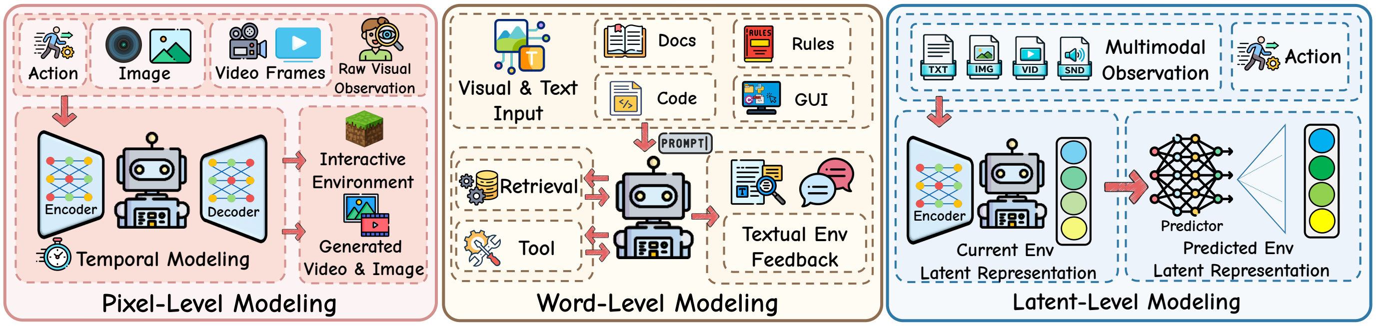

Fig. 8. Three neural environment synthesis paradigms are presented: Pixel-Level Modeling, Word-Level Modeling, and Latent-Level Modeling, trading off fidelity and abstraction.

A line of work focuses on using language models to represent environment knowledge. WKM [189] synthesizes task knowledge (i.e., knowledge required to accomplish a task) and state knowledge (i.e., summaries of environment states based on historical actions), providing prior guidance and constraints for action generation during agent reasoning. Complementary to this, retrieval-augmented approaches extend language world models with external knowledge sources. Specifically, R-WoM [434] enhances environment understanding by retrieving tutorial knowledge and integrating it into reasoning, improving long-horizon simulation through reasoning-aware retrieval and listwise reward estimation.

Other research turns to using language world models to perform dynamic reasoning over the environment. MobileDreamer [435] represents key GUI elements in structured text and introduces a rollout imagination strategy that constructs a tree of predictions to enable multi-step lookahead, thereby improving action selection. Also targeting GUI tasks, Code2World [190] proposes representing GUI states as renderable code, and using a language model to generate code to express GUI state transitions. However, reasoning with language world models inevitably suffers from hallucination, inconsistency, and long-horizon drift, making verifiability a critical concern. RAP [188] incorporates Monte Carlo Tree Search to explore multiple reasoning trajectories and selects the optimal path, ensuring that the output is traceable and more reliable.

5.2.3 Latent-level Modeling

Unlike pixel-level methods that directly model raw visual observations and word-level approaches that describe environments through linguistic symbols, latent-level modeling operates in a learned representation space [201], [436], [437]. **By shifting prediction from the observation space to the latent space, models can focus on the structural information that is most relevant to semantics and dynamics.** I-JEPA [200] predicts representations of image patches in latent space rather than pixels. By masking large target blocks and leveraging spatially distributed context blocks, it encourages the model to learn semantic representations, achieving improvements in both performance and efficiency. seq-JEPA [201] extends this framework from single images to action-observation sequences, introducing inductive biases to separate equivariant and invariant representations within a single model. This enables the model to perform classification with invariant representations and fine-grained discrimination with equivariant representations, while generalizing to sequential tasks such as gaze prediction and path integration.

However, learning latent representations from scratch is costly. Subsequent research therefore turns to reusing representations from pretrained foundation models, enabling cross-domain generalization and diverse applications. A line of work [202], [203], [437] leverages frozen pretrained visual representations from DINOv2 to model environment and temporal dynamics in latent space. These methods avoid learning visual representations from scratch, and train dynamics models on offline video data or trajectories, supporting efficient future forecasting, generalization, and task-adaptive planning. V-JEPA 2 [199], pretrained on one million hours of internet video, requires only 62 hours of robot data to fine-tune and achieves zero-shot robot planning, highlighting the potential of pretrained foundation models to enable scalable and generalizable world modeling.

5.2.4 Summary

Takeaway 5.2

- **Pros and Cons:** Each modeling paradigm involves inherent trade-offs. Pixel-level modeling operates at low abstraction with high fidelity, but suffers from information redundancy. Word-level modeling provides high abstraction with lower fidelity, often leading to information loss due to compression. Latent-level modeling aims to balance representation compactness and predictive capability, while facing challenges in interpretability and reliance on learned representations.
- **Future Directions:** Maintaining long-horizon consistency, preserving essential information, and achieving efficient modeling remain fundamental challenges. Addressing these issues, together with improving controllability and integration across different representation spaces, is critical for building robust and general-purpose neural environment modeling systems.

TABLE 6
Representative methods in neural synthesis. **Modality** refers to the output modality of the model. **Base Model** marked as "-" indicates that no pre-existing foundation model is used, while "base model*" denotes that the base model is fine-tuned.

Name	Modality	Architecture	Base Model	Task Domain	Data Source	Evaluation
			Pixel-Level Modeling (Sec. 5.2.1)			
EVA [178]	V	VLM,Diffusion	ChatUniVi*,Dynamicrafter*	Embodied	COCO,CC3M,etc.	Correctness, Fidelity
ViMo [179]	T I	Diffusion,VLM	Stable Diffusion*,GPT-4o	GUI	Android Control,etc.	Correctness, Complexity, Fidelity
Matrix-Game [431]	V	Diffusion	HunyuanVideo*	Game	MineDojo,etc.	Correctness, Diversity, Fidelity
Matrix-Game 2.0 [432]	V	Diffusion	SkyReelsV2*	Game	Sekai,etc.	Correctness, Diversity, Fidelity
DIAMOND [180]	I	Diffusion	-	Game	Atari 100k	Correctness, Diversity, Complexity, Fidelity
GameNGen [370]	V	Diffusion	Stable Diffusion v1.4*	Game	ViZDoom	Correctness, Complexity, Fidelity
DreamGen [337]	V	Diffusion	WAN2.1*	Embodied	RoboCasa,DROID	Correctness, Diversity, Fidelity
EnerVerse [438]	V	Diffusion	DynamicCrafter*	Embodied	RT1,Taco-Play,etc.	Correctness, Complexity, Fidelity
Genie Envisioner [439]	V	Diffusion	LTX-Video 2B*,COSMOS2 2B*	Embodied	AgiBot-World-Beta	Correctness, Diversity, Fidelity
KeyWorld [440]	V	Diffusion	-	Embodied	LIBERO	Correctness, Complexity, Fidelity
MineWorld [93]	V	LLM,VAE	LLaMA*,VQ-VAE*	Game	Video PreTraining	Correctness, Fidelity
NeuralOS [181]	I	RNN,Diffusion	-	GUI	private	Correctness, Complexity, Fidelity
FlowDreamer [441]	V	Diffusion	Stable Diffusion 2.1*	Embodied	RT-1,Language Table,etc.	Correctness, Fidelity
Cosmos Policy [442]	V	Diffusion	Cosmos-Predict2-2B*	Embodied	LIBERO,RoboCasa,etc.	Correctness, Fidelity
DreamZero [184]	V	Diffusion	Wan2.1-I2V-14B*	Embodied	DROID,etc.	Correctness, Diversity, Fidelity
GAIA-2 [443]	V	Diffusion	-	Embodied	private	Correctness, Diversity, Fidelity
Cosmos-Drive [187]	V	Diffusion	Cosmos-7B-Text2World*	Embodied	Real Driving Scene,etc.	Complexity, Diversity, Fidelity
PEWM [444]	V	Diffusion	DynamiCrafter*	Embodied	RLBench,LIBERO,etc.	Correctness, Fidelity
AdaWorld [185]	V	Diffusion	Stable Video Diffusion*	Embodied,Game	Gym Retro,etc.	Correctness, Diversity, Fidelity
LingBot-VA [445]	V	Diffusion	Wan2.2-5B*	Embodied	Agibot,RoboMind,etc.	Correctness, Complexity, Fidelity
MeWM [183]	T I	VLM,Diffusion	-	Domain-Specific	HCC-TACE-Seg,etc.	Correctness, Fidelity
Cosmos [186]	V	Diffusion	-	Embodied	private	Correctness, Diversity, Fidelity
Pandora [182]	V	VLM,Diffusion	Chat-Univi*,DynamiCrafter*	Cross-Domain	WebVid-10M,etc.	Correctness
LIVE [446]	V	Diffusion	NFD*	Embodied,Game	RealEstate10K,etc.	Correctness, Fidelity
			Word-Level Modeling (Sec. 5.2.2)			
GTM [194]	T	LLM	Qwen2.5-1.5B*	Tool	KernelBench,etc.	Correctness, Fidelity
WKM [189]	T	LLM	Mistral-7B-Instruct-v0.2*	Cross-Domain	ALFWorld,WebShop,etc.	Correctness
Code2World [190]	T	VLM	Qwen3-VL-8B-Instruct*	GUI	Android Control	Correctness, Fidelity
CWM [191]	T	LLM	-	Code	OpenCodeReasoning,etc.	Correctness, Complexity
Dyna-Think [196]	T	LLM	Qwen2.5-32B-Instruct*	GUI	private	Correctness, Fidelity
ITP [433]	T	LLM	Qwen3-8B*	Embodied	ALFWorld,ScienceWorld	Correctness
WebDreamer [192]	T	VLM	GPT-4o,Qwen2-VL-7B-Instruct*	GUI	private	Correctness, Complexity, Fidelity
MobileDreamer [435]	T	LLM	Qwen3-8B*	GUI	Android Control	Correctness, Complexity, Fidelity
VLWM [195]	T	VLM	PerceptionLM-8B*	Cross-Domain	COIN,CrossTask,etc.	Correctness, Complexity
R-WoM [434]	T	VLM	Qwen2.5-VL-72B-Instruct	GUI	-	Correctness, Complexity
RAP [188]	T	LLM	LLaMA-33B	Game	-	Correctness, Complexity
SimuRA [447]	T	LLM	GPT-4o	Cross-Domain	-	Correctness
FPWC [448]	T	VLM	GPT-4V	GUI	-	Correctness, Complexity
WWM [197]	T	LLM	Gemini 2.5 Flash	GUI	-	Correctness, Diversity
DyMo [449]	T	LLM	Cohere's R7B*	Tool	BFCL	Correctness
Simia [193]	T	LLM	o4-mini	Cross-Domain	-	Correctness, Diversity, Fidelity
WebWorld [301]	T	LLM	Qwen3-8/14/32B*	GUI	private	Correctness, Diversity, Fidelity
gWorld [450]	T	VLM	Qwen3-VL-8/32B*	GUI	Android in the Wild,etc.	Correctness, Fidelity
SWE-World [198]	T	LLM	Qwen2.5-32B/72B*	Code	GitHub, SWE-bench, etc.	Correctness
			Latent-Level Modeling (Sec. 5.2.3)			
V-JEPA 2 [199]	E	JEPA	-	Cross-Domain	VideoMix22M	Correctness, Fidelity
I-JEPA [200]	E	JEPA	-	Cross-Domain	ImageNet-1K	Correctness, Diversity
IWM [436]	E	JEPA	-	Cross-Domain	ImageNet	Correctness, Fidelity
DINO-world [202]	E	ViT	DINOv2	Cross-Domain	private	Correctness
EchoJEPA [451]	E	JEPA	-	Domain-Specific	MIMIC-IV-Echo,etc.	Correctness
DINO-WM [437]	E	ViT	DINOv2	Cross-Domain	private	Correctness, Fidelity
DINO-Foresight [203]	E	ViT	DINOv2	Embodied	Cityscapes,nuScenes	Correctness, Complexity, Fidelity
seq-JEPA [201]	E	JEPA	-	Cross-Domain	3DIEBench,CIFAR100,etc.	Correctness, Complexity

5.3 Quality Control and Evaluation of Environments

After synthesizing environments, a central question is how to evaluate and control their quality, namely whether these environments can serve as reliable substrates for training and evaluation. We summarize existing evaluation practices along four complementary dimensions: **correctness**, **diversity**, **complexity**, and **fidelity**. We next discuss how prior work assesses environment quality from these perspectives.

5.3.1 Correctness

Correctness is the most fundamental requirement for synthesized agentic environments. It requires not only that the environment has valid state transitions, but also that the synthesized tasks can be legally executed, have valid solutions, and are paired with reliable verifiers that provide correct reward signals [153], [159], [416].

A common strategy for symbolic environment is program execution and unit testing [413]. In coding and software environments, many works use sandboxes or unit tests to check whether generated environments can run, while filtering runtime errors and timeout cases [126], [413], [422]. For example, SWE-Gym [152] and Scale-SWE [160] validate patches or generated tasks through execution tests. GameDevBench [419] uses deterministic tests in the Godot scripting framework, V-GameGym [344] executes and repairs generated code in a UI sandbox. ScaleEnv [174] verifies generated tools and database code through procedural testing, while Endless Terminals [423] validates container construction, initial-state tests, and final completion tests before retaining a task.

Other works focus more on solvability and trajectory validation. Tool-based environments often execute golden tool sequences or expert trajectories, and use final-state comparison to determine whether the task can be truly completed [155], [303], [417]. For instance, AutoForge [175] obtains the target state by executing golden tool sequences, while AgentSynth [162] uses a verifier to estimate long-horizon task completion.

Some works also evaluate the reliability of the verifier itself. In OS, GUI, and Web environments, task completion often cannot be judged by simple string matching. Execution-based evaluators or expert reviews are therefore used to prevent failed trajectories from being falsely accepted [53], [161], [168]. For example, MCP-Universe [126] uses static and dynamic evaluators instead of relying on unstable LLM-as-a-judge evaluation, while OSWorld-MCP [53] combines execution validation with expert review to ensure tool correctness. InterCode [138] checks custom reward functions with gold commands, and Mobile-Env [54] combines system signals such as screen text and user responses to improve state estimation and outcome evaluation. Some works further measure the agreement between LLM or human verifiers and ground-truth task completion [42], [162], [417].

Unlike the aforementioned evaluation approaches based on explicit execution and verifiers, correctness in neural environment synthesis is difficult to assess through executable tests due to the widespread multi-modal content (e.g., videos and images) in neurally synthesized environments. As a result, correctness is often reformulated as measuring whether the synthesized outputs satisfy predefined constraints. Such methods typically perform evaluation from semantic consistency, interaction consistency, and trajectory deviation. For instance, DreamGen [337] employs Qwen-VL-2.5 to score generated videos and determine whether they faithfully follow the given language instructions, further incorporating human annotations to improve reliability. Matrix-Game [431], on the other hand, utilizes a pretrained inverse dynamics model to infer actions from generated videos and compares them with ground-truth keyboard inputs using average accuracy to evaluate interaction correctness. At a finer granularity, Genie Envisioner [439] combines LLM-based semantic evaluation with rule-based metrics, using symmetric Hausdorff distance to measure spatial deviations between generated and reference trajectories, and normalized dynamic time warping (NDTW) to capture temporal alignment of action sequences. For GUI scenarios, MobileDreamer [435] introduces the mIoU metric to assess spatial accuracy by computing the overlap between predicted and ground-truth element bounding boxes. In addition, some works [189], [196] directly adopt final task completion as an overall criterion to evaluate the validity of synthesized environments.

5.3.2 Diversity

Diversity concerns whether a synthesized environment collection provides broad and non-redundant coverage over task spaces, state spaces, tool spaces, and linguistic expressions. This is crucial for agent training and evaluation: if environments are too homogeneous, agents may overfit to surface patterns rather than acquire generalizable interaction abilities. A high-quality environment set should therefore cover different task types, capabilities, tools, and interaction patterns [198].

Some methods in symbolic environments use similarity filtering to avoid generating many semantically redundant samples [172]. For example, Agent World Model [11] adopts embedding-based deduplication to prevent scenario collapse, EnvScaler [161] uses embedding similarity and t-SNE visualization to analyze topic dispersion, and V-GameGym [344] applies clustering to select diverse game seeds from large code repositories.

Other methods expand diversity through structured coverage. Tool environments often track tool categories and API combinations [38], [40], while OS, GUI, and Web environments emphasize coverage over web tasks, operation types and distractor tools [54], [59], [60]. For example, AutoForge [175] uses random walks over tool dependency graphs to generate diverse tool sequences. MCP-Universe [126] covers multiple real MCP servers and task domains. And other methods improve diversity through the generation process itself [48], [109]. TaskCraft [165] synthesizes multi-hop tool-use tasks from Web, PDF, and image sources.

In neural environment synthesis, beyond improving diversity through expanded task and environment coverage as discussed above, diversity is also commonly characterized by the richness of the model's output distribution under identical input conditions, namely, whether it can produce multiple valid outputs that are semantically consistent yet diverse in details. For example, Genie Envisioner [439] generates multiple videos under the same language instruction and computes pairwise similarities based on CLIP global video embeddings as a diversity score, thereby evaluating the model's ability to produce diverse yet valid trajectories. GAIA-2 [443] perturbs latent representations extracted from real videos and performs conditional denoising to generate diverse environment variants while preserving core semantics and ego-motion trajectories. Furthermore, some works decouple actions from context to verify the model's ability to synthesize diverse environments with consistent core actions. For instance, AdaWorld [185] transfers identical actions to different contexts to evaluate generalization across diverse environments. Similarly, I-JEPA [200] performs multiple stochastic decodings for the same input and target location using the RCDM framework, and assesses diversity by comparing variations in core semantics, object structures, and low-level details across samples.

5.3.3 Complexity

Complexity measures whether synthesized environments have appropriate difficulty. A useful environment should be neither too simple nor systematically impossible. If tasks are too easy, they may fail to improve agent capabilities and can even lead to overfitting; if tasks are too hard or unsolvable, evaluation results provide little useful feedback. Thus, complexity evaluation should not simply increase task length, but construct solvable task distributions that can distinguish agents with different abilities [162], [165], [167].

Some methods quantify complexity through structure parameters. Many symbolic environments report interaction steps, number of tools, API calls, state entities, or lines of code to describe structural difficulty [44], [385]. For example, AutoForge [175] increases the logical complexity of tool-use tasks through DAG structures. OSWorld-MCP [53] characterizes difficulty by the number of tool-calling rounds. LOGIGEN [171] controls complexity by injecting multi-variable conditions, state-dependent constraints, role permissions, and irreversible transitions into the compiled policy environment. NL2Plan [173] stratifies planning tasks by expected optimal plan length, dividing problems into short, medium, and long settings.

Other methods calibrate complexity using strong-model or human performance [109], [153], [424]. gg-bench [424] filters games using win-rate disparity from RL self-play to remove tasks with insufficient discriminative power. In more realistic task settings, AI GameStore [109] uses expert cognitive ratings and human median performance to calibrate game difficulty.

Complexity in neural environment synthesis is often reflected in the reasoning depth and width explored during the generation process. For example, WebDreamer [192] adjusts the planning horizon (1/2/3 steps) to control multi-step prediction depth, thereby influencing task complexity. MobileDreamer [435] varies the number of candidate actions per node to modulate the branching factor of the search space. VLWM [195] regulates complexity by adjusting the number of candidate plans in the System-2 stage, while RAP [188] increases complexity by scaling the number of Monte Carlo Tree Search iterations, thereby expanding search depth.

5.3.4 Fidelity

Fidelity concerns whether synthesized environments faithfully reflect the real systems, physical dynamics or user behaviors they aim to model. It is related to, but distinct from, correctness. A task may be executable inside a synthesized simulator, but if its dynamics, API behavior, or user workflow diverges from the real world, it may still have low fidelity. Fidelity therefore asks a higher-level question: an environment should not only work, but work in a way that resembles the target setting [60], [431], [439].

For symbolic synthesis, fidelity is mainly evaluated through consistency between symbolic rules and reference environments. In PDDL and rule-based game environments, fidelity comes from the alignment between synthesized rules and the original task intent or human intuition [343], [409], [427]. For example, some PDDL-generation works manually compare generated instances with original natural language intents and measure how many edits are needed to make the task semantically correct [427]. OSWorld-MCP [53] uses expert reviews to check whether synthesized tools have real practical value. Therefore, fidelity in symbolic environments mainly emphasizes realistic task logic and real user needs, rather than visual or physical realism.

In neural environment synthesis, fidelity primarily depends on the consistency between synthesized and real environments in terms of perceptual appearance and dynamic evolution, namely, whether generated outputs resemble real-world environments in visual, physical, and temporal aspects. In embodied, gaming, and GUI scenarios, some works [180], [370], [443] employ metrics such as FID, FVD, and LPIPS to measure the similarity between synthesized and real environments in perceptual distributions. DreamGen [337] further evaluates fidelity from a physical perspective by quantifying the realism of rigid-body motion, collisions, gravity, and contact forces, thereby capturing dynamic consistency. Additionally, some approaches assess fidelity from a subjective perspective. For example, WebWorld [301] introduces the Web Turing Score, in which a large language model is tasked with distinguishing between real and synthesized environments, thereby measuring perceptual realism and overall plausibility. From a temporal perspective, dynamic consistency is also considered a key factor. For instance, GAIA-2 [443] adopts Frechet Video Motion Distance (FVMD) to compare motion patterns between generated and real videos based on keypoint trajectories, while EnerVerse [438] incorporates expert evaluations of continuity and stability in robotic behaviors to assess fidelity in long-horizon interactions.

5.3.5 Summary

Takeaway 5.3

- **Dimensions:** Quality covers not only **instance-level** correctness, but also **distribution-level** diversity, complexity, and fidelity.
- **Direction:** Existing work has moved from post generation filtering toward closed-loop generation with continuous validation and refinement for quality-controlled environment synthesis.
- **Limitations:** While correctness has developed well-established evaluation frameworks, the study of diversity, complexity, and fidelity remains preliminary, which affects the credibility.

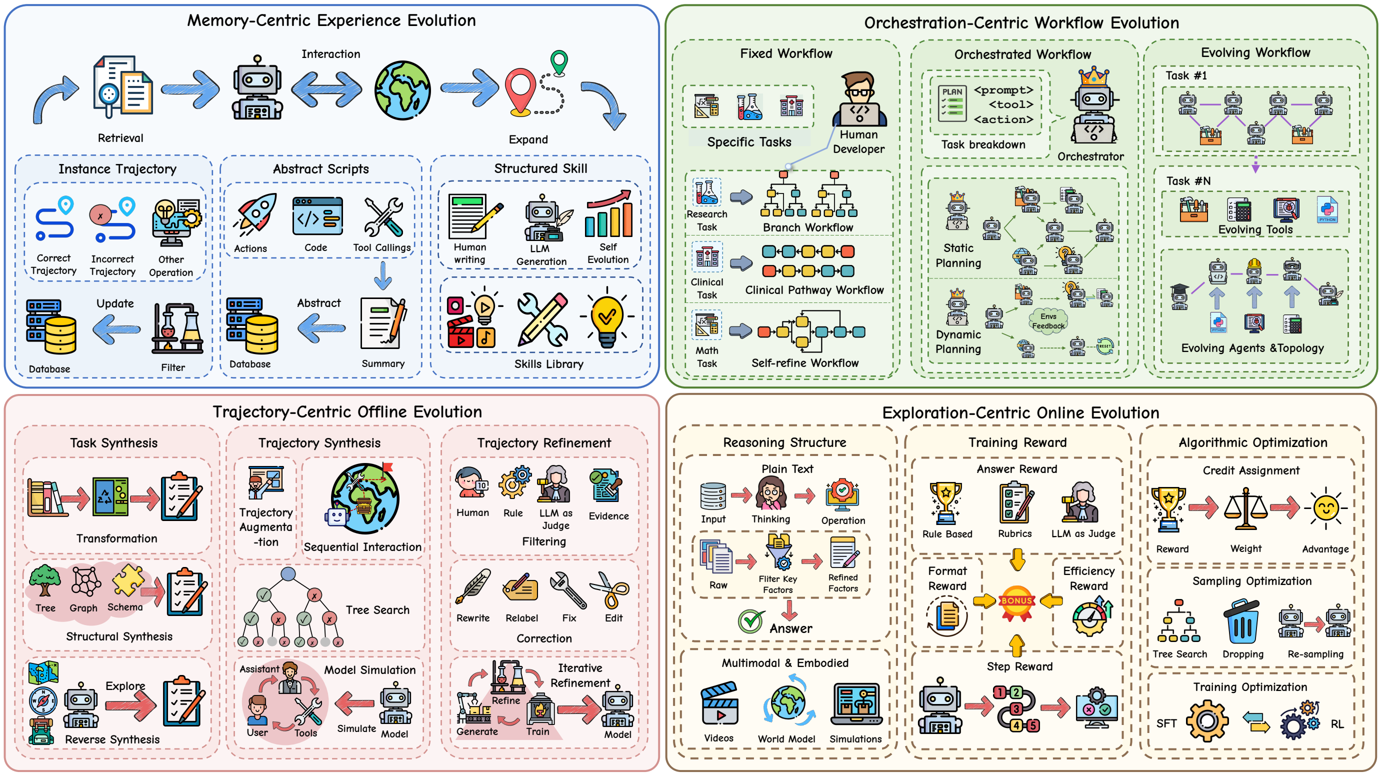

Fig. 9. Overview of agent evolution paradigms. Existing methods are organized into four categories: Memory-Centric Experience Evolution, which enhances agent capabilities through accumulated experiences; Orchestration-Centric Workflow Evolution, which adapts agent workflows to optimize task performance; Trajectory-Centric Offline Evolution, which refines agent behavior through synthetic task interaction data; and Exploration-Centric Online Evolution, which strengthens agent capabilities through real-time learning and adaptation via reinforcement.

6 Agent Evolution

Agent Evolution refers to the process by which an agent's capabilities advance through various mechanisms, typically realized through continuous interaction with the external environment. This process encompasses both external structural adaptations and internal parametric changes. Specifically, external structural adaptations include **Memory-Centric Experience Evolution** (Sec. 6.1), which enhances the model's task-processing capabilities by accumulating and leveraging experience from the external environment, and **Orchestration-Centric Workflow Evolution** (Sec. 6.2), which adapts to diverse environmental tasks through the design of specialized architectures. The internal parametric changes include **Trajectory-Centric Offline Evolution** (Sec. 6.3), which improves the model's task adaptation by synthesizing and designing complex data generation pipelines, and **Exploration-Centric Online Evolution** (Sec. 6.4), which strengthens the model's capabilities through reinforcement learning processes within specific environments. The framework is shown as Fig. 9.

6.1 Memory-Centric Experience Evolution

Experience helps agents better understand and master their current capabilities [455]–[459]. These methods typically involve storing trajectories, experiences, or process knowledge in an external memory base, and then retrieving relevant information to enhance the model's ability to handle problems. Through leveraging experience at different levels of granularity and diverse utilization strategies, the model can maintain long-term memory, strengthen its ability to handle specific problems, and enable the co-evolution of an agent's memory and capabilities across tasks. Based on how experience is summarized and represented, we categorize it into the following types: (1) **Instance Trajectory Experience**: This refers to the complete interaction trajectories and actions of an agent with the environment. (2) **Abstract Scripts Experience**: This type of experience abstracts and summarizes multiple trajectories into reusable script-like knowledge, enabling more efficient reuse across similar tasks. (3) **Structured Skill Experience**: Skills represent a highly structured form of externalized experience that can be modularly stored and invoked on demand.

6.1.1 *Instance Trajectory Experience*

Instance Trajectory Experience represents the most fundamental and specific type of experience, referring to the complete record of actions an agent takes during its interaction with the environment. Instance trajectories directly reflect the agent's decision-making process in a specific task context by recording the correct or incorrect actions, or other agent operations, providing the most detailed information. However, they are highly context-dependent and have limited generalization ability. For example, Synapse [207] provides the LLM with entire trajectories to improve the performance in Web tasks. WorldMM [208] utilizes the multiple categories of memory, where the visual memory provides the detailed scenes of the targets. The maintenance of experience is also crucial. Existing methods typically

TABLE 7
The statistics of Memory-Centric Experience Evolution methods.

Name	Task	Model	Generalization	Experience Update	Train
Instance Trajectory Experience (Sec. 6.1.1)					
OpenAgent [204]	Code	GPT-4, GPT-3.5-Turbo	✓	✓	✗
CoPS [205]	Cross-Domain	Llama 3.1 8B Instruct, Llama 3.1 70B Instruct	✓	✓	✗
ELLMER [206]	Embodied	GPT-4, DALL-E	✓	✓	✗
Memp [452]	Cross-Domain	GPT-4o, Qwen2.5-72B	✗	✓	✗
Synapse [207]	GUI	CodeLlama-7B, GPT-3.5	✓	✓	✗
WorldMM [208]	Image & Video	Qwen3-VL-8B-Instruct	✗	✓	✗
DGM [453]	Code	o3-mini, Claude-3.5-Sonnet	✓	✓	✗
Abstract Scripts Experience (Sec. 6.1.2)					
Reasoning Bank [209]	Cross-Domain	Gemini-2.5-flash, Gemini-2.5-pro, Claude-3.7-sonnet	✓	✓	✗
Agent-Pro [217]	Game	GPT-3.5-Turbo-0613, GPT-4-0613, etc.	✗	✓	✗
Agent-KB [210]	Cross-Domain	GPT-4o, GPT-4.1, etc.	✓	✓	✗
DeepAgent [211]	Tool	QwQ-32B	✓	✓	✓
FLEX [212]	Cross-Domain	Claude-Sonnet-4, DeepSeek-V3.1-Terminus, etc.	✗	✓	✗
O-Mem [213]	Cross-Domain	GPT-4.1, GPT-4o-mini	✗	✓	✗
H-EPM [454]	Tool	GPT-4.1-mini, GPT-4.1, etc.	✓	✓	✓
Reme [214]	Tool	Qwen3-8B, Qwen3-32B	✗	✓	✗
BIFROST [218]	DeepResearch	Llama-3.2-3B-Instruct, etc.	✓	✗	✗
PhysMem [216]	Embodied	Gemini-3.0-Flash, Qwen3-VL	✓	✓	✗
AWM [215]	GUI	GPT-3.5, GPT-4	✓	✓	✗
Structured Skill Experience (Sec. 6.1.3)					
SAGE [219]	Tool	Qwen2.5-32B-Instruct	✗	✓	✓
ASI [220]	Cross-Domain	GPT-4o, Claude-3.5-sonnet	✓	✓	✗
SkillWeaver [221]	DeepResearch	GPT-4o, GPT-4o-mini	✗	✓	✗
SkillRL [222]	Cross-Domain	Qwen2.5-7B-Instruct	✓	✓	✓
SkillOrchestra [223]	Tool	Qwen2.5-3B	✓	✓	✓

organize experiences into databases and perform maintenance operations to ensure their effectiveness [205], [452]. For example, Memp [452] refines historical task trajectories into both fine-grained instructions and high-level abstract scripts, systematically exploring strategies for building, retrieving, and updating procedural memories.

6.1.2 Abstract Scripts Experience

Abstract Scripts Experience is derived by generalizing across multiple instance trajectories to capture reusable task execution patterns. Compared to individual trajectories, script-level experience no longer focuses on specific environment states, but instead extracts the typical sequence of steps or operational logic underlying a class of tasks. By abstracting trajectories into scripts, such experience achieves stronger generalization and enables cross-domain retrieval [210], [215], [218]. Reasoning-Bank [209] extracts generalizable reasoning strategies from the agent's successful and failed experiences and stores them in a structured memory bank. During testing, relevant memories are retrieved to guide decision-making. Besides, some works [213], [460] focus on utilizing the memory to maintain the long-horizon information of previous works, rather than to handle specific problems. For instance, O-Mem [213] further organizes experience into structured categories, including persona memory, working memory, and episodic memory, thereby supporting personalized and consistent long-term interactions. In addition, some approaches focus on how memory is utilized during decision-making [454], [461]. For example, H-EPM [454] integrates tool graphs with procedural memory to guide execution, where each edge in the graph serves as a compact representation of procedural summaries.

Other works emphasize the maintenance and evolution of such experiences to enable continuous self-improvement [212], [217], [462]. For instance, Agent-Pro [217] enhances performance through iteratively reflecting on previous actions and fine-tuning to correct the wrong steps, while FLEX [212] and Reme [214] reflect on both successful and failed experiences to construct and maintain structured experience repositories, facilitating ongoing agent evolution.

6.1.3 Structured Skill Experience

Skills are the experience organization mechanism to expand the LLM's capabilities. Skills have gained widespread attention due to their detail and comprehensiveness. Many methods focus on exploring the automatic synthesis of skills to enhance the capabilities and efficiency of intelligent agents.

Some methods automatically induce and generate procedural skills through interaction with the environment to help the agent complete tasks. ASI [220] induces procedural skills automatically through interaction with the environment, helping the agent execute tasks. SkillWeaver [221] enables the agent to autonomously discover and transform website features into reusable Python API skills.

As research progresses, more works are focused on using skills to complete tasks. By building a skill library, SAGE [219] enhances the self-evolving capabilities of large language model agents. It uses a sequential rollout mechanism to reuse programming skills and integrates skill-based rewards to motivate the generation and utilization of high-quality skills. SkillRL [222] distills redundant raw interaction trajectories into a hierarchical skill bank and co-evolves it with policy learning, thus improving the agent's task success rate and reasoning efficiency.

6.1.4 Summary

Takeaway 6.1

- **Comparative Analysis:** The granularity of experience and its utilization strategies have become central research focuses. The former concerns how experience is acquired and represented, while the latter emphasizes how experience is adapted and evolved for downstream tasks. In this context, both the scale and effectiveness of experience are critical, which has led to increasing attention on **skills** as a structured and reusable form of experience.
- **Future Directions:** An important direction is to develop more systematic and fine-grained frameworks for experience management and utilization. Such systems should be capable of aggregating and integrating large-scale experiences from human contributions and open-source resources. Especially, current experience management mechanisms remain relatively underdeveloped, with limited support for comprehensive operations such as insertion, deletion, updating, and retrieval, and lacking principled design for scalable and efficient management. This is an important direction for experience utilization.

6.2 Orchestration-Centric Workflow Evolution

A **Workflow** is a graph-based topology of multi-step processes powered by LLMs, where nodes encompass LLM calls, tool executions, subagents, or functions, and edges dictate the fixed, conditional, or cyclic routing [10], [17], [504]–[506]. The approach centers on decomposing objectives into detailed steps, enabling the agent to navigate complex environments. Harness frameworks like AutoGen [507] and OpenHands [229] facilitate the deployment of these systems by providing the essential infrastructure and interfaces for inter-agent communication. Based on the degree of structural autonomy, we categorize these methods into three types: **(1) Fixed Workflow**, characterized by explicit data flow and predefined roles; **(2) Automated Workflow**, where a designated orchestrator agent dynamically coordinates the available worker agents; and **(3) Evolving Workflow**, where the entire structure undergoes persistent evolution through interaction with the environment.

6.2.1 Fixed Workflow

Fixed Workflow constitutes a task execution framework with a deterministic logical topology, predefined by developers during the system design phase. The framework incorporates hard-coded sequential logic [463], [477], conditional branching [472], [474], and local loops including prespecified error retry protocols [469], [476]. Within this configuration, the agent functions as a modular unit tasked with performing specific atomic operations. Consequently, the agent has no authority over the global workflow topology and its decision making is bounded by the localized task node. Recent studies demonstrate the utility of this framework in complex engineering scenarios. MetaGPT [224] codifies standard operating procedures for software development into multi-agent pipelines, leveraging role specialization to ensure stability and efficiency. For knowledge-intensive domains, works such as Multimodal DeepResearcher [72] structure report generation as a four-stage pipeline "Researching, Textualization, Planning, and Generation" to replicate the procedural standards of human experts. Moreover, Agentless [225] avoids complex autonomous interactions, showing that a fixed three stage pipeline of localization, repair, and validation is more effective than dynamic agent planning for software bug fixing. Overall, Fixed Workflow aims to codify human expertise into explicit execution logic. By decomposing tasks into multi-stage pipelines, it moves beyond single-shot invocation and enables the model to handle complex tasks in specific scenarios.

6.2.2 Automated Workflow

Automated Workflow typically consists of a central orchestrator agent and a series of worker agents. Based on the input task, the orchestrator autonomously constructs the workflow [150], [481], [508], or intervenes to adjust the existing topology [235], [237], [509]. By decomposing the objective into specific execution steps, the orchestrator manages the coordination logic among execution nodes and may also adjust the subsequent path based on real-time feedback. For tool invocation, works such as ControlLLM, Tool-Planner, and ToolChain* employ task decomposition and structured search for tool workflow planning, while frameworks like OctoTools facilitate extensible tool use through standardized tool cards [487], [510]–[512]. In terms of domain generalization and multi-agent system scalability, Workforce [233] separates strategic planning from specialized execution. It relies on a domain-agnostic "Planner Agent" to decompose tasks, a "Coordinator Agent" to orchestrate subtasks, and specialized "Worker Nodes" to carry out execution. AORCHESTRA [485] further extends this flexibility by creating specific configurations for each task node, allowing the orchestrator to instantiate customized subagents on demand rather than relying on a static pool of predefined workers. Overall, Automated Workflow aims to handle open-ended tasks where execution paths cannot be

TABLE 8
The statistics of Orchestration-Centric Workflow Evolution methods.

Name	Task	Collaboration	Model	Train	Multi-Agent
		Fixed Workflow (Sec. 6.2.1)			
Simulate Before Act [463]	DeepResearch	Tree	GPT-4o	✗	✗
WebVoyager [227]	DeepResearch	Sequential	GPT-4-V, Claude-3-O, GPT-4o	✗	✗
ManuSearch [464]	DeepResearch	Sequential	QwQ-32B, DeepSeek-R1, etc.	✗	✓
Open Deep Search [465]	DeepResearch	Sequential	DeepSeek-R1, Llama-3.1-70B	✗	✗
Search-o1 [466]	DeepResearch	Sequential	QwQ-32B-P	✗	✗
SearchAgent-X [467]	DeepResearch	Sequential	Qwen-7/14B	✗	✗
Thought of Search [468]	DeepResearch	Sequential	GPT-4	✗	✗
WKM [189]	Domain-Specific	Sequential	Mistral-7B, Llama-3-8B, etc.	✓	✗
MedAgent-Zero [228]	Domain-Specific	Iterative	GPT-3.5/4/o, o1-P	✗	✓
QuantAgent [469]	Domain-Specific	Iterative	GPT-4 (0125-P)	✗	✓
Agentless [225]	Code	Sequential	GPT-4o, o1	✗	✗
AutoCodeRover [470]	Code	Sequential	GPT-4 (0125-P)	✗	✓
CodeNav [471]	Code	Iterative	GPT-4, Mixtral-8x22B-IT, etc.	✗	✗
CodeR [472]	Code	Graph	GPT-4 (1106)	✗	✓
MAGIS [473]	Code	Sequential + Iterative	GPT-4, GPT-3.5, Claude-2, etc.	✗	✓
MetaGPT [224]	Code	Sequential	GPT-4, GPT-3.5-T, etc.	✗	✓
SWE-agent [230]	Code	Sequential	GPT-4-T, Claude-3-O	✗	✗
VideoAgent [474]	Video	Sequential	GPT-4, CogAgent, LaViLa	✗	✗
AgileThinker [475]	Game	Sequential	DeepSeek-V3/R1, Gemini-2.5-Flash	✗	✗
LVAgent [476]	Video	Iterative	Qwen2-VL, InternVL-2.5, etc.	✓	✗
Explorer [477]	GUI	Sequential	Phi-3.5V, Qwen2-VL-7B	✓	✓
		Automated Workflow (Sec. 6.2.2)			
AutoFlow [232]	Cross-Domain	Graph	GPT-4-1106-preview, Mixtral-8x7B	✓	✓
ScoreFlow [478]	Cross-Domain	Graph	Llama-3.1-8B-Instruct, GPT-4o-mini, etc.	✓	✓
RobustFlow [479]	Cross-Domain	Graph	Qwen3-32B, GPT-4o-mini, etc.	✓	✓
DyFlow [480]	Cross-Domain	DAG	Phi-4, GPT-4.1, GPT-4o-mini, etc.	✓	✓
AgentFLow [330]	Cross-Domain	Sequential + Iterative	Qwen2.5-7B-IT	✓	✓
MaAS [231]	Cross-Domain	DAG	GPT-4o-mini, Qwen-2.5-72B-Instruct, etc.	✓	✓
FlowReasoner [260]	Code	Graph	DeepSeek-R1-Distill-Qwen-7B, etc.	✓	✓
Mindsearch [481]	DeepResearch	DAG	GPT-4o, InternLM2.5-7B, etc.	✗	✓
WideSeek-R1 [482]	DeepResearch	Tree	Qwen3-4B	✓	✓
Workflow-R1 [483]	DeepResearch	Sequential	Qwen2.5-7B-Instruct, etc.	✓	✓
ResearStudio [484]	DeepResearch	Sequential	GPT-4.1, o4, o3, GPT-4o, etc.	✗	✗
Webpilot [237]	DeepResearch	Iterative	GPT-4o, GPT-3.5	✗	✓
Plan-and-act [235]	Domain-Specific	Sequential	Llama-3.3-70B-I, QWQ-32B	✓	✓
AOrchestra [485]	Domain-Specific	Sequential	Gemini-3-Flash, DeepSeek-V3.2, etc.	✓	✓
Dr. MAS [486]	Domain-Specific	Tree + Iterative	Qwen3-4/8B, Qwen2.5-3/7B-I	✓	✓
Workforce [233]	Domain-Specific	Sequential	Qwen2.5-32B-I, GPT-4o, Claude-3.7-S	✓	✓
ControlLLM [487]	Embodied	DAG	ChatGPT, Llama-7/13B, GPT-4	✓	✗
HuggingGPT [150]	Embodied	DAG	GPT-3.5-T, GPT-4, Alpaca, etc.	✗	✗
HyperAgent [236]	Code	DAG	Claude-3-S/H, Llama-3-70B, etc.	✓	✗
RepairAgent [238]	Code	Graph	GPT-3.5	✗	✗
SWE-Search [234]	Code	Tree	GPT-4o, Qwen2.5-72B-IT, etc.	✓	✗
Alibaba LingmaAgent [488]	Code	Tree	GPT-4-T, GPT-4o, Claude-3-O/S	✗	✗
MindAgent [489]	Embodied	Tree	GPT-4 (0613)	✗	✓
Code Researcher [490]	Code	Graph	GPT-4o, o1	✗	✗
SCLPlan [491]	Embodied	Sequential + Iterative	Llama-3.1-8B, Llama-3-70B	✗	✗
MGA [492]	GUI	Sequential + Iterative	O3	✓	✓
		Evolving Workflow (Sec. 6.2.3)			
AFlow [239]	Cross-Domain	Graph	GPT-4o-mini, DeepSeek-V2.5, etc.	✗	✗
AgentSquare [493]	Cross-Domain	Graph	GPT-4o, GPT-3.5-turbo	✗	✗
Puppeteer [494]	Cross-Domain	Graph	Qwen2.5, Llama-3.1, etc.	✗	✓
ADAS [240]	Cross-Domain	Graph	GPT-4, GPT-3.5, Claude-Haiku, etc.	✗	✗
ReasonRAG [241]	DeepResearch	Iterative	Qwen2.5-7B-IT	✓	✗
MUSE [329]	Domain-Specific	Sequential + Iterative	Gemini-2.5-Flash, DeepSeek-V3	✗	✗
AvaTaR [495]	Domain-Specific	Sequential + Iterative	Claude-3-O, GPT-4/o	✗	✗
ReCreate [496]	Domain-Specific	Iterative	GPT-5, Claude-4.5-O	✓	✗
Yunjue Agent [497]	Domain-Specific	Iterative	Gemini-3-Pro, GPT-5, GPT-5	✗	✓
Chain-of-Agents [245]	Domain-Specific	Sequential	Qwen2.5-3/7/32B-IT	✓	✓
ClinicalReTrial [498]	Domain-Specific	Iterative	DeepSeek-V3, Claude	✗	✓
EvoClinician [499]	Domain-Specific	Iterative	Gemini-3-Pro, GPT-5.1, MedGemma	✗	✓
Grammar-based Search [500]	Domain-Specific	Sequential	GPT-4o, GPT-4.1, GPT-5	✗	✓
CASCADE [501]	Domain-Specific	DAG	GPT-5, O3, Claude-4.5-S, etc.	✗	✓
STEVE [502]	Embodied	Sequential	Llama-2-7/13B	✓	✓
LATM [503]	Cross-Domain	Iterative	GPT-4, GPT-3.5, etc.	✗	✓
Criticize-Reflect [246]	Embodied	Graph	Llama2-70B, GPT-4, GPT-3.5	✗	✓

exhaustively listed. By decoupling high-level planning from low-level execution, it adapts to dynamic environments.

6.2.3 Evolving Workflow

Evolving Workflow refers to a persistently evolving framework where the topological structure adapts to the environment as tasks accumulate, distinguishing it from patterns that merely change states in response to different tasks. This evolution manifests as long-term modifications to task topology [243], [246], [329], [496], the autonomous introduction and persistent storage of new tools or roles during runtime [497], [502], or behavioral shifts in agents simulating the entire workflow after training [241], [245]. In this setting, the workflow is no longer a static program, but a dynamic system that expands functional boundaries or optimizes coordination paths through autonomous iteration. Work such as LATM [503] enables an agent to act as a tool maker, coding Python functions into a persistent cache to broaden the range of tasks the system can handle. Criticize-Reflect [246] and ReCreate [496] further extend this approach by allowing the system to iteratively rewrite its own organizational prompts and execution logic based on feedback, dynamically altering the entire workflow structure. Alternatively, Chain-of-Agents [245] demonstrates that such multi-agent workflows can even be fully internalized into a single model through distillation. Overall, Evolving Workflow aims to break the boundaries between predefined logic and individual tasks. By internalizing empirical insights from execution into the workflow, the agent system develops the ability to improve itself across tasks.

6.2.4 Summary

Takeaway 6.2

- **Comparative Analysis:** Beyond model training, workflows serve as a method to enhance agent capabilities. In **Fixed Workflow**, since role functions are relatively decentralized and static, improving overall coordination through training faces challenges. In contrast, **Automated Workflow** offers a more explicit entry point: by specifically training the central orchestrator, the collective performance of the entire system can be elevated more directly.
- **Future Directions: Evolving Workflow** represents an advanced direction for managing complex and dynamic environments. In highly variable environments, training individual agents in isolation often struggles to meet practical demands. Rather than fragmented training at single nodes, this collective evolution allows the entire system to better adapt to increasingly complex future environments.

6.3 Trajectory-Centric Offline Evolution

Trajectory-Centric Offline Evolution primarily refers to agentic SFT, which leverages interaction trajectories to enhance an agent's capabilities. These trajectories may include environment states, action choices, tool calls, external feedback, intermediate reasoning, and even error recovery. In practice, the process of trajectory synthesis usually consists of three stages, as illustrated in Fig. 10: **(1) Task Synthesis** (Sec. 6.3.1), which determines what tasks the agent should solve; **(2) Trajectory Synthesis** (Sec. 6.3.2), which turns these tasks into interaction trajectories; and **(3) Trajectory Refinement** (Sec. 6.3.3), which improves the quality and reliability of the synthesized trajectories before training.

6.3.1 Task Synthesis

Based on the task construction paradigm, existing approaches can be categorized into three types: **Resource Transformation**, **Reverse Synthesis**, and **Structure-based Synthesis**.

Resource Transformation refers to performing transformations based on prior resources. Typical approaches include adding tool-use annotations via LLMs [38], [265], [521], masking answers [523], replacing entities with target information [252], [519], or utilizing existing resources as seeds and prompting LLMs to synthesize novel tasks [258], [267].

Reverse Synthesis denotes a paradigm where agents first explore the environment and then synthesize task instructions from observations or interaction trajectories. This is particularly common in the GUI [248], [254], [529] and Deep Research [247], [249], [253], [272], [477], [508], [517], [518] domains. To be more specific, WebExplorer [253] leverages LLMs to navigate the web and synthesize corresponding QA pairs. OS-Genesis [248] performs step-wise interactions in GUI environments and retrospectively derives tasks from the resulting transitions. Going beyond naive exploration, HATS [529] generates tasks through hardness-driven exploration, targeting ambiguous interactions during exploration and validating synthesized instructions through iterative alignment.

Structure-based Synthesis synthesizes tasks based on explicit structural representations, including graphs, trees, schemas, abstract syntax trees (ASTs), etc. Some methods leverage trees [256], [321], such as ToolACE [321], which builds a hierarchical API context tree and synthesizes APIs through a speciation-adaptation-evolution process. Many instead rely on graphs [154], [255], [263], [513], [515], [520], [524], [538], including API dependency graph, knowledge graph, database graph, etc. For instance, DeepDive [515] operates over knowledge graphs, constructing QA pairs via random walks and enriching paths with node attributes before using an LLM to obfuscate key cues into complex questions. In contrast, Mobile-Agent-v3.5 [514] relies on mobile DAGs annotated by humans, sampling realistic interaction paths and synthesizing user instructions with LLMs/VLMs. Specifically, WebShaper [251] uses set operations over knowledge projections, where an agentic Expander progressively increases task complexity.

6.3.2 Trajectory Synthesis

In terms of how trajectories are generated, recent studies mainly adopt four kinds of approaches: **Trajectory Augmentation**, **Sequential Interaction**, **Tree Search**, and **Model Simulation**.

Trajectory Augmentation includes rewriting [530], enriching [259], [525], [531] and expanding [516] existing

TABLE 9
The statistics of Trajectory-Centric Offline Evolution Methods. F refers to Filtering, C refers to Correction, I refers to Iterative Refinement.

Name	Domain	Data Size	Teacher Model	Task Source	Trajectory Source	Refinement
Task Synthesis (Sec. 6.3.1)						
BAGEL [247]	Cross-Domain	260	PaLM-2	Reverse	Sequential Interaction	F
ToolACE [321]	Tool	180,000	-	Structural	Model Simulation	F
OS-Genesis [248]	GUI	1,000	GPT-4o	Reverse	Sequential Interaction	F
Insta [249]	GUI	150,000	Qwen3-235B	Reverse	Sequential Interaction	F
APIGen-MT [250]	Tool	5,000	GPT-4o, DeepSeek V3	Structural	Model Simulation	F
WebSailor [513]	Deep Research	2,000	QwQ-32B	Structural	Sequential Interaction	F
WebShaper [251]	Deep Research	5,000	QwQ-32B	Structural	Sequential Interaction	F
WebWatcher [252]	Deep Research	8,000	GPT-4o	Transformation	Sequential Interaction	F
Cognitive Kernel-Pro [508]	Deep Research	47,314	GPT-4.1	Reverse	Sequential Interaction	C
Mobile-Agent-v3.5 [514]	GUI	-	GUI-Owl	Structural	Sequential Interaction	I
WebExplorer [253]	Deep Research	13,000	-	Reverse	Sequential Interaction	F
DeepDive [515]	Deep Research	858	Claude-4-Sonnet-T	Structural	Sequential Interaction	F
AutoPlay [254]	GUI	11,500	GPT-4o, UI-TARS-1.5-7B	Reverse	Sequential Interaction	F
AgentFounder [516]	Cross-Domain	-	-	Transformation	Augmentation	F C
WebAggregator [517]	Deep Research	6,184	GPT-4.1	Reverse	Sequential Interaction	F
CRMWeaver [255]	Domain-Specific	3,000	GPT-4.1	Structural	Sequential Interaction	F
WebLeaper [256]	Deep Research	20,000	-	Structural	Sequential Interaction	F
AgentEvolver [518]	Cross-Domain	-	Qwen-Plus	Reverse	Sequential Interaction	F
VSearcher [519]	Deep Research	1,308	Gemini-3-Pro-T	Transformation	Sequential Interaction	F
OpenSeeker [520]	Deep Research	11,700	-	Reverse	Sequential Interaction	-
Trajectory Synthesis (Sec. 6.3.2)						
ToolCoder [521]	Code	53,000	GPT-3.5-turbo	Transformation	Augmentation	F
ToolBench [38]	Tool	126,486	GPT-3.5-turbo	Transformation	Tree Search	F
ToolAlpaca [257]	Tool	3,938	GPT-3.5	Transformation	Model Simulation	F
AgentTuning [522]	Cross-Domain	1,866	GPT-3.5, GPT-4	Reuse	Sequential Interaction	F
Lingma SWE-GPT [258]	Code	-	GPT-4o	Transformation	Sequential Interaction	F
Aguvis [259]	GUI	35,000	GPT-4o	Transformation	Augmentation	-
MaskSearch [523]	Deep Research	58,000	Qwen-Max	Transformation	Sequential Interaction	F
DreamGen [337]	Embodied	240,000	WAN2.1, IDM	Human&Reuse	Model Simulation	F
Tongyi DeepResearch [524]	Deep Research	-	-	Structural	Sequential Interaction	F
AgentFold [262]	Deep Research	-	GLM-4.5, DeepSeek-V3.1	Reuse	Sequential Interaction	F
SIMA2 [525]	Embodied	-	Human	Transformation	Human&Augmentation	I
O-Researcher [526]	Deep Research	3,500	-	Transformation	Sequential Interaction	F
Pixels2Play [527]	Game	-	Human	Reverse	Human&Augmentation	F
ToolACE-MCP [263]	Tool	15,092	GPT-4o	Structural	Model Simulation	-
ProAct [264]	Game	33,000	-	Reuse	Tree Search	C
OpenResearcher [528]	Deep Research	97,000	GPT-OSS-120B	Reuse	Model Simulation	F
HATS [529]	GUI	1,000	GPT-4o	Reverse	Tree Search	C
Trajectory Refinement (Sec. 6.3.3)						
Toolformer [265]	Tool	68,000	GPT-J-6.7B	Transformation	Augmentation	F
ETO [266]	Cross-Domain	7651	GPT-4	Reuse	Sequential Interaction	I
Agent-FLAN [530]	Cross-Domain	24,703	GPT-3.5-turbo	Transformation	Augmentation	F
Self-Improvement [267]	GUI	58	Qwen-1.5-72B-Chat	Transformation	Sequential Interaction	F
UI-TARS [531]	GUI	-	UI-TARS	Human	Human&Augmentation	I
μCode [532]	Code	-	Llama-3.1-8B-I	Reuse	Sequential Interaction	C
SimpleDeepSearcher [533]	Deep Research	871	QwQ-32B	Reuse	Sequential Interaction	F
EvolveSearch [289]	Deep Research	80,000	Qwen2.5-7B-I	Reuse	Sequential Interaction	I
GUI-Reflection [268]	GUI	32,951	Gemini-2.0-Flash, Gemini-2.5-Pro	Transformation	Augmentation	C
Agent-Reward [534]	Cross-Domain	1,136	GPT-4o	Transformation	Augmentation	F
WebSynthesis [261]	GUI	4,000	GPT-4o	Reuse	Model Simulation	F C
TiG [269]	Game	-	DeepSeek-R1	Human	Human	C
ToolRM [535]	Tool	180,000	Qwen2.5-32B-I, Granite-20b, etc.	Structural	Human&Augmentation	F
Tool-Reflection [536]	Tool	6,000	Human	Transformation	Model Simulation	C
AgentFrontier [270]	Deep Research	12,000	-	Structural	Sequential Interaction	I
WebSTAR [271]	GUI	13,300	GPT-4o	Reuse	Sequential Interaction	F
SynthAgent [272]	GUI	2,500	GPT-4.1	Reverse	Sequential Interaction	C
SERA [537]	Code	200,000	GLM-4.5-Air, GLM-4.6	Reverse	Sequential Interaction	F C
TopoCurate [273]	Tool	2,400	Claude-4.5-Sonnet	Reuse	Sequential Interaction	F

trajectories. Typically, Agent-FLAN [530] rewrites ReAct-style trajectories into multi-turn chat-style conversations. Aguvis [259] first generates trajectories through automated programs and then augments them with reasoning traces produced by GPT-4o. AgentFounder [516] first generates multi-step reasoning and tool-action sequences, and then expands them into diverse decision paths through higher-order action synthesis.

Sequential Interaction means that trajectories are primarily generated through a single forward interaction chain. Among relevant approaches, the ReAct framework is widely adopted [519], [522], [539], while some methods further optimize via complex workflow pipelines [258] or multi-agent systems [260], [523], [526], [540]. As an example, Lingma SWE-GPT [258] synthesizes development trajectories with a three-stage workflow: Repository Understanding, Fault Localization, and Patch Generation. MaskSearch [523] uses a multi-agent system involving a planner, rewriter, and observer. To manage context, Tongyi DeepResearch [524] summarizes context into an evolving report to preserve essential information. Moreover, AgentFold [262] adopts a more fine-grained context-folding strategy, performing either granular condensation of the Latest Interaction or deep consolidation of it with prior summaries into a coarser state summary.

Tree Search denotes a class of work that identifies higher-quality trajectories using tree search algorithms such as depth-first search (DFS) and Monte Carlo Tree Search (MCTS). In particular, ToolBench [38] uses DFS to search for a valid solution path to avoid being trapped in a single ReACT trajectory. ProAct [264] generates trajectories by running MCTS from the current state, sampling both optimal and suboptimal/dead-end paths.

Model Simulation is another type of method that generates trajectories in simulated environments through components including simulated users [154], [250], [263], tools [257], [321], [535], [538], and world models [261], [337]. Specifically, ToolACE-MCP [263] generates multi-turn dialogue trajectories through role-based simulation, with planner, user, assistant, and tool roles simulated by LLMs. ToolRM [535] uses an LLM to simulate the tool's error feedback. DreamGen [337] generates neural trajectories by prompting a robot-adapted video world model.

6.3.3 Trajectory Refinement

Trajectory Refinement determines which trajectories are ultimately incorporated into the training set. We broadly classify it into three methods, namely, **Filtering**, **Correction**, and **Iterative Refinement**.

Filtering removes low-quality samples using rules [256], [258], [265], [533], [541], structure-based scoring [273], LLM-as-judge [248], [249], [254], [271], [523], [534], or model self-scoring [267]. In detail, Lingma SWE-GPT [258] applies rejection sampling based on two criteria: fault localization accuracy and patch similarity. TopoCurate [273] computes three structure-aware metrics on an interaction topology to filter trajectories. WebSTAR [271] performs step-level filtering using a grading model. Self-Improvement [267] filters out trajectories in which the model self-detected a failure, producing a refusal.

Correction chooses to correct [268], [516], rewrite [529], [535], [542], relabel [532] or edit [272] erroneous trajectories instead of simply discarding them. For example, GUI-Reflection [268] identifies the first incorrect action in a trajectory and generates post-error reflection and correction actions for it. WebSynthesis [261] leverages GPT-4 to synthesize reflections for failed branches, tracing back to the common ancestor node and identifying "go_back" as the corrective action for rollback trajectory construction. μCode [532] relabels the collected data with the optimal solutions selected by a local search expert. SynthAgent [272] applies Remove, Reorder, Drop, or Keep edits to trajectories to eliminate task-irrelevant, redundant, or misaligned actions.

Iterative Refinement connects generation, filtering, verification, correction, and retraining into a closed loop, allowing the refinement process to directly shape the data distribution of the next iteration [266], [270], [289], [514], [531], [543]. Among these, ETO [266] updates the agent policy through an iterative cycle of trajectory collection and training. EvolveSearch [289] iteratively alternates between RL exploration and SFT optimization, using high-quality and diverse rollouts collected during RL as the SFT training set. AgentFrontier [270] generates complex-reasoning data within the LLM's Zone of Proximal Development (ZPD), targeting tasks situated at the model's capability frontier and, as the model learns, its ZPD advances, enabling a continuously adaptive curriculum.

6.3.4 Summary

Takeaway 6.3

- **Comparative Analysis:** For task synthesis, current research mainly focuses on making tasks more complex and realistic. Thus, methods have evolved from corpus transformation to reverse and structure-based synthesis. Trajectories are expected to better capture high-level abilities, such as planning, reflection, and recovery, resulting in increasingly sophisticated workflows. At the same time, maintaining verifiable quality boundaries while scaling up requires refinement strategies to shift from simple filtering toward a closed-loop pipeline that includes generation, verification, correction, and retraining.
- **Future Directions:** It is likely that the next competitive edge will no longer lie in simply scaling up data, but in several fundamental directions: how to improve the alignment between synthesized trajectories and training objectives, how to build low-cost but high-trust verification mechanisms for long-horizon trajectories, how to turn failure and recovery processes into stable training assets, and how to prevent models from continually reinforcing their own biases in a self-bootstrapping data flywheel. Whoever can solve these problems more effectively will be more likely to truly advance trajectory synthesis from "sample production" to "capability" engineering."

TABLE 10
The statistics of Exploration-Centric Online Evolution Methods. A refers to outcome reward, P refers to process reward, E refers to efficient reward and F refers to format reward.

Name	Task	RL Algorithm	Model	Training Data	Reward
Reasoning Structure Design (Sec. 6.4.1)					
DeepRetrieval [274]	DeepResearch	PPO	Qwen2.5-3B-I, LLaMA-3.2-3B, etc.	NQ, HotpotQA, etc.	A F
Search-R1 [275]	DeepResearch	PPO	Qwen2.5-3/7B, etc.	NQ, HotpotQA	A
ReSearch [280]	DeepResearch	GRPO	Qwen2.5-7/32B, etc.	MuSiQue	A F
AutoRefine [276]	DeepResearch	GRPO	Qwen2.5-3/7B,	Natural Questions, HotpotQA	A P
SEEA-R1 [277]	Embodied	Tree-GRPO	Qwen2.5-VL-7B-I	ALFWorld, Customize	A
M3-Agent [278]	Image & Video	DAPO	Qwen2.5-Omni-7B, Qwen3-32B, etc.	Customize	A
Video-Thinker [279]	Image & Video	GRPO	Qwen2.5-VL-7B-I	ActivityNet, YouCook2, etc.	A F
Training Reward Design (Sec. 6.4.2)					
MaskSearch [523]	DeepResearch	DAPO	Qwen2.5-1.5/3B, etc.	Wikipedia, HotpotQA, etc.	A F
R1-Searcher [544]	DeepResearch	Reinforce++	Qwen-2.5-7B, Llama-3.1-8B-I	HotpotQA, 2WikiMultiHopQA	A F P
ToolRL [8]	Tool	GRPO	Qwen2.5-1.5/3B-I,	ToolACE, xLAM, etc.	A F P
OTC-PO [545]	Cross-Domain	PPO, GRPO	Qwen2.5-3/7B	NQ, HotpotQA, etc.	A F P E
Tool-N1 [285]	Tool	GRPO	Qwen2.5-7/14B-I	xLAM, ToolACE	A F
ARTIST [546]	Cross-Domain	GRPO	Qwen2.5-7/14B-I	NuminaMath, BFCL v3	A F P
VLA-RL [547]	Embodied	PPO	OpenVLA-7B	LIBERO	A P
VRAG-RL [282]	Image & Video	GRPO	Qwen2.5-VL-3/7B-I	Customize	A F E
R-Search [548]	DeepResearch	PPO, GRPO	Qwen2.5-3/7B-I	2WikiMultiHopQA	A F P
Chain-of-Agents [245]	Cross-Domain	DAPO	Qwen2.5-3/7/32B-I	NQ, HotpotQA, etc.	A F
Video-MTR [549]	Image & Video	PPO	Qwen2.5-VL-7B	NExT-GQA, QVHighlights	A F P
Agent-R1 [281]	Cross-Domain	PPO, GRPO, etc.	Qwen2.5-3B-I	HotpotQA, 2WikiMultiHopQA	A P
ToolOrchestra [286]	Tool	GRPO	Qwen3-8B	ToolScale, GeneralThought-430K	A E P
VideoMem [550]	Image & Video	PRPO	Qwen3-VL-8B	VideoMarathon, LLaVA-Video-178K	A P E
GDPO [283]	Tool	GDPO	Qwen2.5-1.5/3B-I	ToolACE, xLAM, etc.	A F
FlowSteer [284]	Tool	CWRPO	Qwen3-8B	GSM8K, MATH, etc.	A F P
IntentRL [551]	DeepResearch	GRPO	Qwen2.5-7B	DeepResearch Bench	A F P E
SHARP [552]	Tool	GRPO	LLaMA-3.1-8B, Qwen3-8B	MuSiQue	A P
Training Algorithm Optimization (Sec. 6.4.3)					
RAGEN [287]	Cross-Domain	StarPO	Qwen2.5-0.5/3B-I	WebShop	A F
ZeroSearch [288]	DeepResearch	ZeroSearch	Qwen2.5-3/7B, etc.	NQ, HotpotQA	A
GiGPO [290]	Cross-Domain	GiGPO	Qwen2.5-1.5/3/7B-I	ALFWorld, WebShop, etc.	A P
WebAgent-R1 [553]	GUI	M-GRPO	Qwen-2.5-3B,Llama-3.1-8B	WebArena	A
EvolveSearch [289]	DeepResearch	GRPO	Qwen2.5-7B-I	NQ, HotpotQA, etc.	A F
Embodied Planner-R1 [554]	Embodied	IPO	Qwen2.5-7B-I	ALFWorld, ScienceWorld	A
SPIRAL [555]	Game	SPIRAL	Qwen3-4/8B, etc.	Customize	A
MobileGUI-RL [292]	GUI	MobGRPO	Qwen2.5-VL-7/32B	AndroidWorldAvd	A E
ARPO [291]	Cross-Domain	ARPO	Qwen2.5-3B-I, Llama3.1-8B-I, etc.	Tool-Star, STILL, etc.	A F P
Thinking With Videos [556]	Image & Video	DGRPO	Qwen2.5-VL-7B	MTVR-CoT, MTVR-RL, etc.	A F P
ComputerRL [557]	GUI	ComputerRL	GLM-4-9B-0414, GLM-4.1V-9B-T	Customize	A
ExIt [558]	Cross-Domain	ExIt	Llama-3.2-3B-I, Qwen2.5-7B-I, etc.	NuminaMath, BFCL-v3, etc.	A
AgentGym-RL [559]	Cross-Domain	ScalingInter-RL	Qwen2.5-3/7B	WebArena, NQ, etc.	A
UI-S1 [560]	GUI	SOPO	Qwen2.5VL-3/7/32B	AndroidControl-Train, Amex	A F P
SPEAR [293]	Cross-Domain	SPEAR	Qwen2.5-1.5/7B-I, etc.	ALFWorld, WebShop, etc.	A F P
AgentRL [561]	Cross-Domain	AgentRL	Qwen2.5-3/7B-I, etc.	ALFWorld, WebShop, etc.	A F
Stronger-MAS [562]	Cross-Domain	AT-GRPO	Qwen3-1.7/8B	Sudoku, Sokoban, etc.	A P
VAGEN [294]	Cross-Domain	PPO	Qwen2.5-VL-3B	Sokoban, FrozenLake, etc.	A F P
GTR-Turbo [563]	Cross-Domain	GTR	Qwen2.5-VL-7B, Qwen3-VL-8B	Points24, ALFWorld	A P
SeeUPO [295]	Tool	SeeUPO	Qwen3-14B, Qwen2.5-14B-I	AppWorld, BFCL v4	A P
HGPO [564]	Cross-Domain	HGPO	Qwen2.5-1.5/7B-I	ALFWorld, WebShop	A
DeepSWE [565]	Code	GRPO++	Qwen3-32B	R2E-Gym	A E

6.4 Exploration-Centric Online Evolution

Reinforcement learning is a crucial method for agent evolution [20], [315], [322]. Through reinforcement learning, the model can not only enhance its capabilities but also mitigate catastrophic forgetting, thereby achieving better performance across multiple tasks [3], [566]–[573]. By modifying the model's reasoning paradigms, setting up reward mechanisms, or improving training algorithms, reinforcement learning encourages the model to achieve targeted improvements in specific tasks or behaviors [393], [574]–[576]. We summarize the reinforcement learning training process into three parts: **(1) Reasoning Structure Design**: Modifying the model's reasoning structure in downstream tasks and designing corresponding reinforcement learning algorithms. **(2) Training Reward Design**: Designing more effective reward signals to encourage the model to explore and optimize more efficiently. **(3) Training Algorithm Optimization**: Optimizing the training algorithm to enhance the model's stability and effectiveness in complex tasks.

6.4.1 Reasoning Structure Design

The research on inference structure design primarily focuses on how agents organize decision-making and reasoning processes in complex tasks. In recent years, various methods have been proposed to enhance task-solving capabilities by designing specialized inference structures, often integrated with the training process. For example, Search-R1 [275] introduces special tags in the reasoning process to differentiate between the thinking, search, feedback, and answer processes, while ReSearch [280] and DeepRetrieval [274] enhance the model's reasoning capability by incorporating format correctness rewards to ensure the accuracy of generated labels and the correct reasoning format, to promote the effectiveness of structured output.

As research progresses, more works are designing complex inference structures or workflows to further improve model performance. For instance, AutoRefine [276] introduces an explicit refinement step into the inference loop. In this paradigm, after obtaining retrieved documents, the model must first filter, refine, and organize the evidence, extracting key facts. Based on this refined information, the model then decides whether to conduct another search or generate the final answer.

Additionally, in multimodal and embodied systems, Video-Thinker [279] enables the model to think about videos by integrating time localization and content understanding into the reasoning process using special tokens. M3-Agent [278] enhances the handling of audiovisual tasks by incorporating external memory tools to establish connections between roles and memories before the reasoning process. ITP [433] introduces world models to simulate environmental dynamics, supporting more complex reasoning tasks.

6.4.2 Training Reward Design

The core of reward design lies in modeling refined signals to guide agents in exploring and identifying optimal strategies. The reward mechanism has evolved from early, purely result-oriented systems that focused on the correctness of the outcome to a multi-dimensional, task-specific optimization framework.

In terms of efficiency optimization, researchers have introduced a cost penalty mechanism to reduce the agent's over-reliance on tools or redundant searches. For example, OTC-PO [545] penalizes redundant calls by rewarding the product of correctness and efficiency, ReasonRAG [241] drives minimalist reasoning by applying a path-length penalty factor, and VRAG-RL [282] and VideoMem [550] impose sparsity constraints on the RAG process or memory module, respectively, through retrieval time rewards and memory capacity penalties, aiming to solve problems with minimal cost.

In the domain of goal alignment and stability, many methods combine rewards to help models better understand tasks. For example, IntentRL [551] combines answer reward, format compliance, and repetition penalties, while SHARP [552] uses the Shapley Value for marginal contribution attribution, effectively solving the credit assignment problem. GDPO [283] introduces a decoupled normalization mechanism, independently processing accuracy, format, and length rewards to prevent signal collapse. At the same time, FlowSteer [284] employs a conditional release strategy. Final answer rewards are activated only when intermediate metrics of diversity are satisfied. Similarly, Video-MTR [549] uses this strategy, where process answer rewards are triggered only when final answer validity meets the required criteria. For multi-agent frameworks like Chain-of-Agents [245], the correctness reward and format reward are composed to formulate the final reward.

In terms of process supervision, format rewards have become a universal constraint across tasks, while VLA-RL [547] uses a robot process reward model (RPRM), which provides high-density pseudo-reward signals through visual milestones.

6.4.3 Training Algorithm Optimization

The optimization design of reinforcement learning algorithms focuses on improving training stability and sample efficiency in complex tasks through innovations in underlying architectures. In the area of credit allocation, DigiRL [577] proposes an advantage-weighted RL with advanced advantage estimators to account for stochasticity, coupled with an automatic curriculum that derives maximal learning signals. To address the challenge of sparse rewards due to long-range interactions, GiGPO [290] introduces Anchor State Grouping, which aggregates different actions originating from the same state into step-level groups for optimization. The contextual consistency cluster of historical contexts and adaptive weighting in HGPO [564], along with the world modeling reward and Bi-layer Generalized Advantage Estimation mechanism in VAGEN [294], enables precise reward propagation from macro-level episode rewards to micro-level step-level rewards. SPIRAL [555], through Role-Conditioned Advantage Estimation (RAE), effectively mitigates asymmetric variance risk in multi-agent games.

For objective function optimization, AEPO [578] introduces entropy-balanced rollout mechanism and optimization to prevent traditional algorithms from excessively suppressing high-entropy exploration signals, while CEMs [579] impose an information-theoretic metric term to encourage the model to learn efficient communication topologies.

Regarding sampling structures, RAGEN [287] retains trajectories with high reward variance during the sampling process, avoiding overfitting to specific samples. Stronger-MAS [562] employs tree-structured sampling to ensure statistical validity deep within conversations, while WebSailor [513] improves gradient efficiency by dynamically replicating non-zero variance samples, and AgentRL [561] uses cross-policy sampling to enable stabilization.

Finally, in terms of training mechanisms, ComputerRL [557] introduces the novel concept of alternating RL and supervised fine-tuning updates, forcibly injecting successful experiences to tackle the issue of entropy collapse in online RL during later stages. This establishes a systematic optimization paradigm that evolves from local credit alignment to global strategy robustness.

6.4.4 Summary

Takeaway 6.4

- **Comparative Analysis:** The design of the reinforcement learning process remains one of the key challenges in post-training. Current research focuses on improving more efficient reasoning paradigms, designing fine-grained reward signals, enhancing training stability, and enabling more effective exploration. Among these, training algorithm optimization is particularly critical, as it is more tightly coupled with parameter updates and thus offers a more general and scalable solution.
- **Future Directions:** Looking ahead, there are two promising directions. One is to integrate new concepts and capabilities into the training process, such as experience, skills, and personalization, enabling more adaptive and intelligent agents. The other is to develop more general-purpose algorithms that can be reliably deployed in real-world systems, bridging the gap between research and practical applications.

7 Environment Evolution

Environment evolution focuses on how training environments are progressively evolved to support the continuous improvement of agent capability. Existing approaches can be categorized into three paradigms according to how the environment is evolved: **(1) Neural-Driven Evolution** (Sec. 7.1) instantiates the environment as a learned neural model, and evolves the environment by optimizing this model to provide adaptive interactions or simulated transitions for agent learning; **(2) Difficulty-Driven Evolution** (Sec. 7.2) regulates environment complexity to match the agent's capability through curriculum learning; **(3) Scaling-Driven Evolution** (Sec. 7.3) expands the environment distribution itself by increasing scenario diversity or introducing new environment structures. This categorization highlights different ways in which environments can be adapted to support agent learning, and provides a structured view of the design space of environment evolution.

7.1 Neural-Driven Evolution

Neural-Driven environment evolution treats the environment itself as a learned neural model. In this paradigm, the environment is no longer a fixed external simulator or a manually specified interaction process, but a trainable model that directly embodies either self-generated interactive dynamics or learned environment dynamics. Accordingly, environment evolution is realized through the optimization of this model, so that the environment can adapt together with agent learning by providing adaptive interactions or simulated transitions. Existing approaches in this direction can be broadly divided into two categories. The first relies on **self-play**, where the environment is effectively instantiated by the agent itself through self-generated tasks, challenges, or role-based interactions [296]–[299], [580]. The second relies on **world models**, where the environment is instantiated as a separately learned simulator that approximates environment dynamics and provides environment-like feedback [190], [192], [300], [301], [338], [581].

7.1.1 Self-Play

Self-play constructs an effective training environment from the agent itself or its role-specialized variants. Through self-generated tasks, the agent not only learns within the environment but also participates in shaping the environment it learns from, making environment evolution endogenous to self-play. Absolute zero [296] represents the clearest form of this paradigm: a unified model simultaneously plays the roles of proposer and solver, generating tasks that maximize its own learning progress and then improving by solving them. Self-Challenging [297] extends this paradigm to tool-use settings, where the same model first acts as a challenger to synthesize verifiable tasks and then switches to an executor role to learn from them. Beyond direct task generation, some methods realize self-play through structured role interaction. Vision-zero [299] formulates self-improvement as a gamified visual reasoning process, where role-based interactions over visual inputs create increasingly informative training challenges. SSR [580] grounds self-play in software engineering, allowing a single model to alternate between bug injection and bug fixing so that increasingly realistic coding tasks emerge from the self-play loop itself. Active Zero [298] further generalizes this formulation by introducing multiple co-evolving roles, such as Searcher, Questioner, and Solver, enabling the agent to actively retrieve frontier examples and construct adaptive visual reasoning tasks.

7.1.2 World Model

World-model-based neural evolution instantiates the environment as a separately learned simulator that approximates environment dynamics and provides environment-like feedback for agent learning. Existing approaches mainly differ in how the world model is obtained: some methods learn environment dynamics directly from interaction data [190], [301], [338], while others leverage the prior knowledge of foundation models through prompting to simulate environment transitions at inference time [192], [300], [581]. These two directions introduce different trade-offs between training cost, scalability, and simulation fidelity. **Learning-based World Models** obtain the simulator

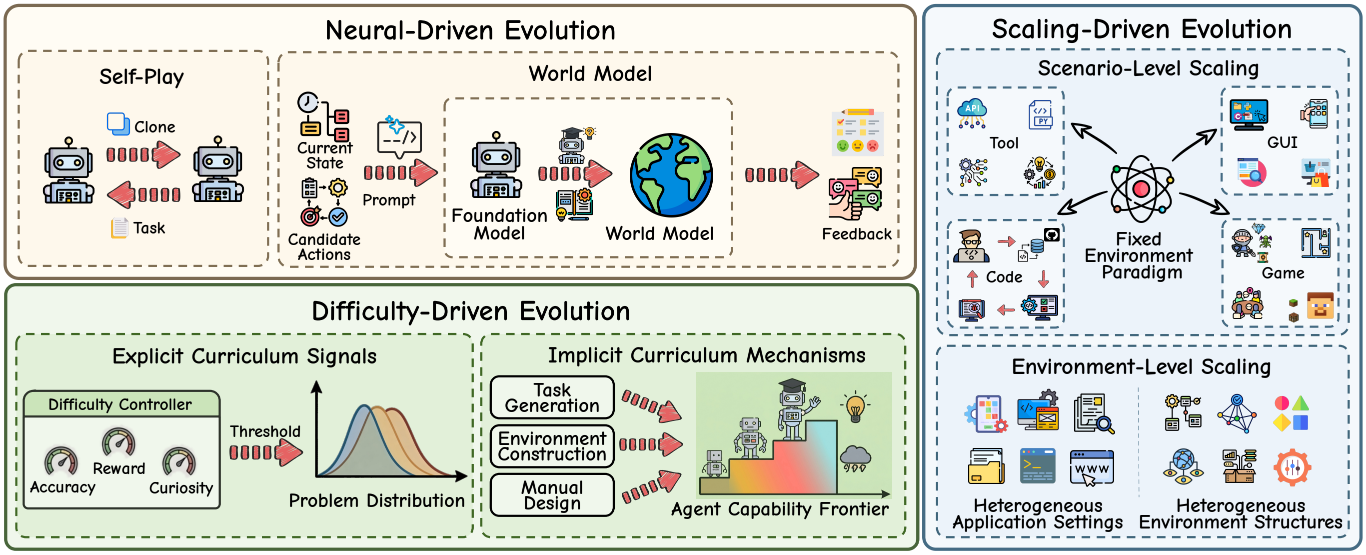

Fig. 10. **Overview of environment evolution paradigms.** Existing methods are organized into three categories: Neural-Driven Evolution, which evolves environments through self-play or world models; Difficulty-Driven Evolution, which adapts task difficulty via explicit curriculum signals or implicit curriculum mechanisms; and Scaling-Driven Evolution, which expands environment diversity at the scenario or environment level.

by explicitly learning environment transitions from interaction data. Code2World [190] models GUI transitions as renderable code, generating the next interface as HTML and aligning predictions with visual outcomes via render-aware reinforcement learning. WebEvolver [338] extends this paradigm by jointly evolving the world model and agent policy, where the learned model is used both to generate training trajectories and to simulate future interactions for planning. WebWorld [301] further scales this line by training a web simulator on over one million real-world interaction trajectories, enabling long-horizon and structured state prediction at scale. **Inference-based World Models** rely on the reasoning and prior knowledge of foundation models to simulate environment transitions without explicitly training a dedicated simulator. WebDreamer [192] demonstrates that a strong LLM can function as a web world model via direct prompting, predicting potential state changes for candidate actions and enabling model-predictive planning before real interaction. UI-Simulator [300] constructs an LLM-based simulator that generates future UI states from prior context and actions, while guided rollouts and trajectory wrappers are introduced to synthesize scalable training data. WAC [581] integrates a world model as an environment expert within a multi-agent pipeline, where simulated outcomes are used to support candidate action generation and pre-execution correction.

7.2 Difficulty-Driven Evolution

Complementary to neural-driven evolution, difficulty-driven environment evolution commonly adopts **curriculum learning**, where environments are progressively adjusted to match the evolving capabilities of agents. Existing approaches can be broadly divided into two categories. The first relies on **explicit curriculum signals**, where environment difficulty is directly regulated by clearly defined signals such as accuracy, regret, reward or curiosity [315]–[318]. The second relies on **implicit curriculum mechanisms**, where curriculum progression is not explicitly controlled by a measurable signal but instead emerges from adaptive task generation, environment construction, or manually designed curricula [303], [304], [306], [307].

7.2.1 Explicit Curriculum Signals

Explicit curriculum signals regulate environment difficulty through clearly defined metrics that determine how tasks should be generated or adapted during training. Most existing approaches adopt performance-related signals as the primary basis for curriculum control, though they differ in their adaptation mechanisms. A common strategy is to use success rate or accuracy as the main curriculum signal. RLVE [315] primarily performs forward difficulty evolution: each adaptive verifiable environment shifts its problem distribution toward harder instances once the empirical success rate on the current upper-bound difficulty exceeds a predefined threshold. SCALER [318] adopts a broader co-adaptive strategy, using an online difficulty controller to adjust instance difficulty based on rollout accuracy so as to maintain training near a target success-rate band, while further reorganizing the training distribution through active environment curation. GenEnv [317] instead formulates this principle as a reward-driven curriculum policy, where the α-Curriculum Reward encourages the generation of tasks whose empirical success rate stays close to a target band, thereby aligning difficulty with the agent's current capability. EnvGen [163] extends this idea to game environments by adapting environment configurations according to task-specific success and failure statistics, so that the generated environments increasingly target the agent's weak skills. Eurekaverse [314] similarly evolves embodied training terrains using policy training statistics, especially success rate and reward-related signals, to iteratively generate more suitable parkour environments. Another important line of work comes from unsupervised environment design, where explicit signals such as regret are used to regulate the

TABLE 11
The statistics of Environment Evolution Methods. **Str** stands for structured text, **Err** for error messages, and **NA** means not applicable or unspecified.

Name	Trait	Domain	Feedback	Evo Criterion	Process Reward
		Neural-Driven Evolution (Sec. 7.1)			
Absolute zero [296]	Self-Play	Code	Str	Success Rate	✗
Self-Challenging [297]	Self-Play	Tool	Str, Tool Response	NA	✗
Vision-zero [299]	Self-Play	Chart, Vision	Image, Str	Accuracy, N/A Rate	✗
SSR [580]	Self-Play	Code	Str, JSON	Success Rate	✗
Active Zero [298]	Self-Play	Vision, Math	Image, Str	Solver Uncertainty, Repetition Penalty	✗
WebDreamer [192]	World Model	GUI	Image, Str	NA	✗
UI-Simulator [300]	World Model	GUI	Str, List	Teacher-forcing Loss	✗
WebEvolver [338]	World Model	GUI	Str, List	NA	✗
Code2World [190]	World Model	GUI	Str, Image	NA	✗
WAC [581]	World Model	GUI	Str	NA	✗
WebWorld [301]	World Model	GUI	Str, List	NA	✗
		Difficulty-Driven Evolution (Sec. 7.2)			
POET [302]	Implicit Signals	Robotics	State	NA	✓
PAIRED [308]	Explicit Signals	Game	State	Regret	Partial
DCD [309]	Explicit Signals	Game, Robotics	State, Image	Regret	Partial
ACCEL [310]	Explicit Signals	Game, Robotics	State	Regret	Partial
MAESTRO [311]	Explicit Signals	Game	State	Regret	Partial
ReMiDi [312]	Explicit Signals	Game	State	Regret	Partial
EnvGen [163]	Explicit Signals	Game	Image	Success Rate	✗
AgentGen [303]	Implicit Signals	Embodied	Str	NA	✗
DataEnvGym [313]	Explicit Signals	Math, Code, Vision, Tool	Str	Student Feedback	✗
Eurekaverse [314]	Explicit Signals	Embodied	NA	Success Rate	✓
Environment Tuning [304]	Implicit Signals	Tool	Hint, Str, Err	Curriculum Schedule	✓
SEC [305]	Implicit Signals	Math, Reasoning	Str	NA	✗
Reasoning Core [306]	Implicit Signals	Reasoning, Planning, Logic	Str, Err	External Curriculum Policy	✗
DreamGym [307]	Implicit Signals	GUI, Embodied	Str	NA	✗
RLVE [315]	Explicit Signals	Math, Algorithm	Str	Success Rate	✗
CuES [316]	Explicit Signals	Tool, GUI	Str, Err	Curiosity	✗
GenEnv [317]	Explicit Signals	Tool, Embodied, Reasoning	Str, Err	α-Curriculum Reward	✗
SCALER [318]	Explicit Signals	Math	Str, Err	Accuracy	✗
		Scaling-Driven Evolution (Sec. 7.3)			
FTRL [319]	Scenario-Level	Tool	Str, Err	NA	✓
ARE [320]	Environment-Level	GUI, Tool	Str, Tool Response	NA	✗
AgentScaler [154]	Scenario-Level	Tool	Str, Err	NA	✗
AutoEnv [170]	Environment-Level	Reasoning, Game	Str	NA	✗
AutoForge [175]	Scenario-Level	Tool	Str, Err	NA	✗
EnvScaler [161]	Scenario-Level	Tool	Str, Err	NA	✓
InfiniteWeb [172]	Scenario-Level	GUI	Image, Str	NA	✓
Agent World Model [11]	Scenario-Level	Tool	Dict, Str, Err	NA	✓
AutoWebWorld [168]	Scenario-Level	GUI	Image, Str	NA	✓

evolution of parameterized environments or replayable levels [308]–[312]. PAIRED [308] uses regret between antagonist and protagonist returns to train an adversarial environment generator, thereby steering the environment toward tasks that are difficult yet still learnable. ACCEL [310] further develops this paradigm in replay-based settings by editing and preserving high-value levels according to regret-style signals, enabling continual environment evolution while avoiding curriculum collapse. Beyond direct performance signals, some methods rely on richer external feedback. DataEnvGym [313] uses student feedback, such as errors and weak skills, to guide teacher-side data and environment generation, turning student performance into an explicit curriculum signal. CuES [316] instead uses intrinsic curiosity as an explicit signal to drive exploration and task synthesis in environments without predefined tasks, enabling the agent to autonomously discover and organize useful training tasks.

7.2.2 Implicit Curriculum Mechanisms

Implicit curriculum mechanisms do not rely on explicit signals, with curriculum progression emerging from task generation and environment construction during training. POET [302] exemplifies this paradigm by jointly evolving environments and agents through mutation, minimal-criterion filtering, and transfer, allowing progressively more complex obstacle courses to emerge without an explicit difficulty controller. DreamGym [307] induces curriculum through adaptive task generation, synthesizing tasks that challenge the current policy and prioritizing those with high reward entropy to construct progressively more informative training experiences. AgentGen [303] similarly relies on task evolution by constructing tasks of increasing difficulty within generated environments and refining the progression through bidirectional evolution over both easier and harder instances. Reasoning Core [306] exemplifies curriculum emerging from environment construction. It emphasizes scalable symbolic reasoning environments with continuous difficulty control and large-scale problem generation, enabling increasingly challenging instances to arise as training progresses. SEC [305] instead realizes curriculum progression through adaptive category selection, dynamically reallocating training focus across problem types during training without relying on a manually specified curriculum schedule. Environment Tuning [304] instead reflects manually designed curricula, organizing training through structured curricula, environment augmentation, and progress-based feedback.

7.3 Scaling-Driven Evolution

Beyond adjusting environment parameters or curriculum difficulty, scaling-driven environment evolution focuses on expanding the breadth of the **environment distribution** itself. The goal is to expose agents to substantially richer interaction spaces by increasing either the diversity of scenarios within a given environment family or the coverage of fundamentally different domains and environment structures. Existing approaches in this direction can be broadly divided into two categories. The first emphasizes **scenario-level scaling**, where evolution is achieved by synthesizing more diverse tasks, trajectories, websites, or tool-use workflows within a shared interaction paradigm [154], [161], [168], [172], [175]. The second emphasizes **environment-level scaling**, where evolution is driven by extending environments across multiple domains and applications, together with their transition, observation, and reward structures, thereby pushing agents toward broader cross-environment generalization [170], [320].

7.3.1 Scenario-Level Scaling

Scenario-level scaling expands environment diversity primarily by increasing the number and variety of tasks, trajectories, and interaction scenarios within a shared environment paradigm. In tool-use settings, several works [154], [161], [175] follow this paradigm by automatically constructing executable tool-interaction scenarios. EnvScaler [161] adopts a two-stage pipeline that first builds environment skeletons and then instantiates them into concrete scenarios with initial states, task descriptions, and rule-based validation. AutoForge [175] further extends this pipeline by composing more challenging workflows from tool dependencies and reasoning constraints, while also introducing RL to stabilize training over synthesized environments. AgentScaler [154] provides a more structured formulation by abstracting function calling as read and write operations over databases, thereby systematically expanding the combinatorial space of tools, user intents, and execution paths. A similar trend appears in GUI environments. InfiniteWeb [172] enables scalable web-agent training by automatically generating functional websites from lightweight specifications, along with corresponding tasks and reward evaluators. AutoWebWorld [168] further improves controllability and verifiability by modeling websites as finite state machines, enabling systematic trajectory enumeration and verification over explicit state spaces.

7.3.2 Environment-Level Scaling

By comparison, environment-level scaling extends the environment distribution itself by introducing diverse application settings and heterogeneous environment structures. Current works focus on expanding the environment space rather than increasing the number of task instances. ARE [320] provides a general platform for constructing and orchestrating agent environments across heterogeneous applications, supporting both synthetic environments and real-world application integration. By enabling flexible configuration of rules, tools, content, and evaluation protocols, it expands the environment ecosystem in which agents are developed and evaluated. AutoEnv [170], in contrast, offers a more fundamental formulation by modeling environments as factorizable distributions over transitions, observations, and rewards. This abstraction allows heterogeneous environments with different dynamics and feedback structures to be generated in a unified manner, effectively scaling the space of environment factors and enabling systematic study of cross-environment learning and generalization.

7.4 Summary

Takeaway 7

- **Unified Perspective:** Taken together, these paradigms reveal three complementary ways of evolving training environments for agent learning: Neural-Driven Evolution represents the environment itself as a learned model, Difficulty-Driven Evolution adapts environment difficulty to match the agent's capability, and Scaling-Driven Evolution expands the environment distribution by broadening scenario diversity or environment structures.
- **Future Directions:** From this unified perspective, advancing agent capability requires not only stronger policies, but also treating environment evolution as a core component of the training process. Looking forward, training environments may need to evolve toward greater realism, with richer feedback, more diverse structures, and closer alignment with real-world interaction settings for agent development.

8 Challenges & Future Directions

8.1 Environment-as-a-Service

Current research faces a fundamental challenge due to the lack of standardization in environment interfaces. Existing environments differ substantially in their definitions of observations, action spaces, reward functions, and interaction paradigms, forcing agents to rely on environment-specific adaptations and limiting their ability to generalize across environments. Such heterogeneity also hinders unified benchmarking and undermines reproducibility. In addition, environment deployment remains a major bottleneck for scaling agent systems. Many environments rely on complex software infrastructures and tightly coupled runtime conditions, such as simulators, web browsers, graphical interfaces, or physical devices, making them difficult to reproduce and deploy across platforms. As a result, deployment overhead often becomes a system-level constraint, hindering large-scale training and evaluation.

To address these challenges, we advocate the concept of **Environment-as-a-Service (EaaS)**. The key idea is to standardize environment interfaces and provide them as cloud-hosted services, decoupling agent development from environment implementation and deployment. Under this paradigm, environments are encapsulated behind unified APIs, allowing researchers to interact with diverse environments through a consistent interface without managing underlying dependencies or runtime configurations.

8.2 The Evolution of Environment Properties

Current environments are still predominantly designed around static, short-horizon, closed-world, and single-modality settings. While such formulations simplify environment construction and evaluation, they fail to reflect the complexity and variability of real-world scenarios. In practice, real-world environments are often dynamic and continuously evolving. Environment states may change while agents are interacting with them, requiring agents to make not only accurate but also timely decisions. At the same time, many real-world tasks involve long-horizon interactions, where actions can influence future states far beyond immediate observations, significantly increasing the difficulty of planning and credit assignment. Moreover, real environments are inherently open-ended, with continuously emerging tasks, expanding state spaces, and evolving interaction patterns, making it difficult for agents to rely solely on fixed task distributions. In addition, the growing emergence of multimodal environments introduces diverse inputs, including text, images, videos, audio, and sensor signals, requiring agents to jointly understand and integrate heterogeneous modalities. These trends fundamentally reshape the capability requirements of agents. Accordingly, developing realistic environments that are dynamic, long-horizon, open-ended, and omni-modal will become a critical direction for advancing practical agent systems.

8.3 From Single-agent to Multi-agent Environments

Existing research has predominantly focused on single-agent environments, where the environment is typically modeled as a static and passive system. In such settings, an agent interacts with the environment independently, receives feedback, and optimizes its policy within a relatively controlled and static framework. While this approach simplifies the problem, it fails to capture the complexity of real-world scenarios. In practice, many tasks inherently involve multiple agents interacting within a shared environment, where decisions are interdependent and outcomes are jointly determined. Capturing such scenarios requires moving beyond single-agent environments to multi-agent environments, which better reflect the dynamics of cooperation, competition, and strategic interaction.

In multi-agent settings, the environment becomes a dynamic system shaped by interactions among agents. Each agent's behavior influences others, leading to increased non-stationarity and complexity. This shift introduces new challenges. Agents must continuously adapt their strategies in response to other agents, resulting in a more uncertain and evolving decision space. At the same time, inter-agent dependencies make credit assignment and stable reinforcement learning significantly more difficult. More importantly, complex multi-agent interactions may also give rise to emergent capabilities and behaviors that are difficult to observe in single-agent settings. Understanding how such emergent behaviors arise from environment dynamics and agent interactions remains an important open problem for future research.

8.4 Neural-Symbolic Environments

Existing environment designs can be broadly categorized into neural and symbolic paradigms, each with distinct advantages and limitations. Neural environments, typically built upon learned models or simulators, offer strong expressiveness and flexibility. They can capture complex, high-dimensional dynamics and support rich multimodal interactions. However, such environments are often opaque and lack interpretability, making it difficult to analyze underlying mechanisms or verify correctness. Moreover, neural environments may suffer from instability, distribution shift, and limited controllability, which can lead to unreliable feedback for agent learning. In contrast, symbolic environments rely on rule-based systems or explicitly defined state transitions, providing clear semantics and high interpretability. Their structured nature enables precise control, reproducibility, and ease of analysis. However, symbolic environments are typically limited in expressiveness and scalability, making it challenging to model complex, real-world scenarios or handle high-dimensional observations. These limitations highlight the need for neural-symbolic integration in environment design. By combining the expressiveness of neural models with the interpretability and controllability of symbolic systems, hybrid environments have the potential to provide both realistic dynamics and reliable structure. Developing such integrated environments remains an important direction for building robust and generalizable agent systems.

8.5 Bridging the Sim-to-Real Gap in Environments

Despite recent advances in generative and simulated environments, a substantial sim-to-real gap still exists between synthetic environments and real-world environments. Current environments are often constructed through hand-crafted rules or LLM-driven generation, aiming to provide scalable and controllable platforms for agent training and evaluation. However, due to the inherent complexity, dynamics, and openness of the real world, existing synthetic environments still struggle to faithfully approximate real-world environments from multiple perspectives.

First, from the perspective of correctness, synthetic environments may contain factual errors and logical inconsistencies. For example, webpages automatically generated by LLMs may exhibit invalid state transitions, incorrect feedback signals, or inconsistent interaction logic. Such issues can lead agents to learn spurious policies that fail to generalize effectively to real-world environments. Second, in terms of difficulty, many synthetic environments are simplified approximations of real-world problems. To reduce construction and training costs, existing environments often simplify state spaces, action spaces, and task procedures, making it difficult to capture the long-horizon dependencies and complex interactions present in realistic scenarios. As a result, agents that achieve strong performance in synthetic environments may still struggle to generalize effectively to more complex real-world tasks. Moreover, regarding diversity, current synthetic environments are often constrained by predefined templates and fixed task distributions. Compared with the continuously evolving and open-ended nature of real-world environments, synthetic environments typically provide limited scenarios and task distributions, which can easily cause agents to overfit to specific patterns. From the perspective of fidelity, substantial discrepancies still exist between synthetic and real-world environments. In digital environments such as GUIs or web systems, real user behaviors, system latency, and dynamic interface changes are often insufficiently modeled. Therefore, an important future direction is to develop more realistic, diverse, and high-fidelity environments that better capture real-world complexity while maintaining scalability and controllability for large-scale agent training and evaluation.

8.6 Co-Evolution of Agents and Environments

Traditional research typically treats environments as fixed and passive systems, primarily focusing on how to optimize agent policies within predefined environments. However, as environments become increasingly complex, the relationship between agents and environments is gradually evolving from one-way adaptation to bidirectional co-evolution. Beyond requiring agents to adapt to environments, environments themselves should also evolve in response to agents by dynamically adjusting task difficulty and environment rules according to the agents' current capabilities, thereby forming a continual co-evolution mechanism. For example, environments may progressively introduce more challenging tasks, longer interaction horizons, and increasingly complex feedback mechanisms as agent capabilities improve. Environment generators may continuously create new tasks, tools, and interaction patterns based on the weaknesses or emerging capabilities of agents.

8.7 Towards Unified Offline–Online Learning

Current agent training methods can generally be categorized into two paradigms: offline training and online training. Offline training relies on high-quality target trajectories generated by external expert models, allowing agents to imitate stronger policies while improving data efficiency and reducing the reliance on costly environment interaction. However, it is limited by fixed data distributions and the lack of continual interaction with real environments, which can lead to distributional mismatch and reduced adaptability to evolving dynamics. In contrast, online training relies on trajectories generated through the model's direct interaction with environments, allowing policies to continuously adapt to evolving environment dynamics. Such a paradigm more naturally reflects real-world agent behavior, but typically requires extensive interaction and often suffers from sparse rewards and high exploration costs.

Consequently, integrating offline and online paradigms is emerging as an increasingly important direction for agent training. On-Policy Distillation (OPD) [582] combines online trajectory generation with high-performing teacher supervision, where the student first explores through on-policy interaction and then receives dense corrective feedback from stronger teacher models. However, the application of OPD to agentic settings remains largely underexplored. Extending OPD to multi-turn agent environments introduces challenges fundamentally different from those in conventional single-turn tasks. Errors made during early interaction steps not only affect the current output, but also alter subsequent environment states and future trajectories, leading to long-horizon error accumulation and increasingly inconsistent teacher supervision. In addition, an important future direction is to investigate how to effectively integrate accurate environment feedback with fine-grained teacher supervision.

8.8 Science of Environment Engineering

Despite the rapid progress in agentic environments, current environment construction remains largely empirical and still lacks a systematic scientific foundation. As environments become increasingly important for agent training, evaluation, and capability development, establishing a more principled understanding of environment engineering is emerging as a critical research direction.

One important future direction is to investigate the scaling laws of environments, namely how factors such as the number of environments, environment diversity, interaction horizon, and environment complexity influence the formation, generalization, and emergence of agent capabilities. Understanding these relationships may help establish more principled guidelines for environment construction and scaling. Beyond environment scale, another fundamental question is whether an environment itself is learnable by agents. Not all environments are equally suitable for agent training, as overly sparse rewards, excessively large state spaces, or long interaction horizons may prevent agents from obtaining effective learning signals, leading to unstable training or even failure to converge. Future research should therefore investigate environment learnability, namely what types of environments are more conducive to stable and efficient agent learning. More importantly, environments and agent capabilities are inherently coupled. Different environments may naturally encourage the development of different capabilities. However, there is still a lack of systematic understanding of what types of environments cultivate what kinds of capabilities. An important future direction is to establish the mapping between environments and agent capabilities, enabling targeted environment construction for developing specific meta-capabilities such as long-term memory, task decomposition, world modeling, and strategic planning.

References

[1] Q. Team, "Qwen3 technical report," *CoRR*, vol. abs/2505.09388, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.09388

[2] DeepSeek-AI, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *CoRR*, vol. abs/2501.12948, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2501.12948

[3] Z. Liu, A. Sims, K. Duan, C. Chen, S. Yu, X. Zhou, H. Xu, S. Xiong, B. Liu, C. Tan, C. Y. Beh, W. Wang, H. Zhu, W. Shi, D. Yang, M. Shieh, Y. W. Teh, W. S. Lee, and M. Lin, "GEM: A gym for agentic llms," *CoRR*, vol. abs/2510.01051, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.01051

[4] X. Li, X. Li, K. Dong, Q. Zhang, R. Ruan, X. Dai, Y. Wang, and R. Tang, "Humanity's last code exam: Can advanced llms conquer human's hardest code competition?" in *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, Eds. Association for Computational Linguistics, 2025, pp. 21 122–21 137. [Online]. Available: https://aclanthology.org/2025.findings-emnlp.1152/

[5] OpenAI. (2025) Introducing gpt-5.4. Accessed: 2026-03-16. [Online]. Available: https://openai.com/index/introducing-gpt-5-4/

[6] Google DeepMind. (2025) Gemini 3.1 pro. Accessed: 2026-03-16. [Online]. Available: https://deepmind.google/models/gemini/pro/

[7] K. Team, "Kimi K2.5: visual agentic intelligence," *CoRR*, vol. abs/2602.02276, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2602.02276

[8] C. Qian, E. C. Acikgoz, Q. He, H. Wang, X. Chen, D. Hakkani-Tür, G. Tur, and H. Ji, "Toolrl: Reward is all tool learning needs," *CoRR*, vol. abs/2504.13958, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2504.13958

[9] J. Xie, K. Zhang, J. Chen, T. Zhu, R. Lou, Y. Tian, Y. Xiao, and Y. Su, "Travelplanner: A benchmark for real-world planning with language agents," in *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, ser. Proceedings of Machine Learning Research, R. Salakhutdinov, Z. Kolter, K. A. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, Eds. PMLR / OpenReview.net, 2024, pp. 54 590–54 613. [Online]. Available: https://proceedings.mlr.press/v235/xie24j.html

[10] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang, S. Gupta, B. P. Majumder, K. Hermann, S. Welleck, A. Yazdanbakhsh, and P. Clark, "Self-refine: Iterative refinement with self-feedback," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/91edff07232fb1b55a505a9e9f6c0ff3-Abstract-Conference.html

[11] Z. Wang, C. Xu, B. Liu, Y. Wang, S. Han, Z. Yao, H. Yao, and Y. He, "Agent world model: Infinity synthetic environments for agentic reinforcement learning," *arXiv preprint arXiv:2602.10090*, 2026.

[12] P. Cao, C. Wang, Z. He, H. Yuan, J. Li, Y. Chen, K. Liu, J. Zhao *et al.*, "Rwku: Benchmarking real-world knowledge unlearning for large language models," vol. 37, 2024, pp. 98 213–98 263.

[13] T. Men, P. Cao, Z. Jin, Y. Chen, K. Liu, and J. Zhao, "A troublemaker with contagious jailbreak makes chaos in honest towns," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025, pp. 17 561–17 587.

[14] F. Lei, J. Meng, Y. Huang, J. Zhao, Y. Zhang, J. Luo, X. Zou, R. Yang, W. Shi, Y. Gao *et al.*, "Dacomp: Benchmarking data agents across the full data intelligence lifecycle," *arXiv preprint arXiv:2512.04324*, 2025.

[15] S. Yao, H. Chen, J. Yang, and K. Narasimhan, "Webshop: Towards scalable real-world web interaction with grounded language agents," in *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., 2022. [Online]. Available: http://papers.nips.cc/paper_files/paper/2022/hash/82ad13ec01f9fe44c01cb91814fd7b8c-Abstract-Conference.html

[16] C. E. Jimenez, J. Yang, A. Wettig, S. Yao, K. Pei, O. Press, and K. R. Narasimhan, "Swe-bench: Can language models resolve real-world github issues?" in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. [Online]. Available: https://openreview.net/forum?id=VTF8yNQM66

[17] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. R. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. [Online]. Available: https://openreview.net/forum?id=WE_vluYUL-X

[18] J. Li, P. Cao, Y. Chen, J. Xu, H. Li, X. Jiang, K. Liu, and J. Zhao, "Fixing the broken compass: Diagnosing and improving inference-time reward modeling," 2025.

[19] Z. Jin, H. Yuan, K. Zhu, J. Li, P. Cao, Y. Chen, K. Liu, and J. Zhao, "Omni-reward: Towards generalist omni-modal reward modeling with free-form preferences," *arXiv preprint arXiv:2510.23451*, 2025.

[20] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, M. Zhang, Y. K. Li, Y. Wu, and D. Guo, "Deepseekmath: Pushing the limits of mathematical reasoning in open language models," *CoRR*, vol. abs/2402.03300, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2402.03300

[21] Q. Yu, Z. Zhang, R. Zhu, Y. Yuan, X. Zuo, Y. Yue, T. Fan, G. Liu, L. Liu, X. Liu, H. Lin, Z. Lin, B. Ma, G. Sheng, Y. Tong, C. Zhang, M. Zhang, W. Zhang, H. Zhu, J. Zhu, J. Chen, J. Chen, C. Wang, H. Yu, W. Dai, Y. Song, X. Wei, H. Zhou, J. Liu, W. Ma, Y. Zhang, L. Yan, M. Qiao, Y. Wu, and M. Wang, "DAPO: an open-source LLM reinforcement learning system at scale," *CoRR*, vol. abs/2503.14476, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2503.14476

[22] Z. Huang, X. Yuan, Y. Ju, J. Zhao, and K. Liu, "Reinforced internal-external knowledge synergistic reasoning for efficient adaptive search agent," *arXiv preprint arXiv:2505.07596*, 2025.

[23] W. Sun, X. Cheng, J. Fan, Y. Xu, X. Yu, S. He, J. Zhao, and K. Liu, "Towards agentic self-learning llms in search environment," *arXiv preprint arXiv:2510.14253*, 2025.

[24] T. Wei, T. Li, Z. Liu, X. Ning, Z. Yang, J. Zou, Z. Zeng, R. Qiu, X. Lin, D. Fu, Z. Li, M. Ai, D. Zhou, W. Bao, Y. Li, G. Li, C. Qian, Y. Wang, X. Tang, Y. Xiao, L. Fang, H. Liu, X. Tang, Y. Zhang, C. Wang, J. You, H. Ji, H. Tong, and J. He, "Agentic reasoning for large language models," *CoRR*, vol. abs/2601.12538, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.12538

[25] P. Cao, T. Men, W. Liu, J. Zhang, X. Li, X. Lin, D. Sui, Y. Cao, K. Liu, and J. Zhao, "Large language models for planning: A comprehensive and systematic survey," *arXiv preprint arXiv:2505.19683*, 2025.

[26] C. Wang, J. Li, Y. Chen, K. Liu, and J. Zhao, "A survey of recent advances in commonsense knowledge acquisition: Methods and resources," *Machine Intelligence Research*, vol. 22, no. 2, pp. 201–218, 2025.

[27] A. Drouin, M. Gasse, M. Caccia, I. H. Laradji, M. D. Verme, T. Marty, D. Vázquez, N. Chapados, and A. Lacoste, "Workarena: How capable are web agents at solving common knowledge work tasks?" in *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, ser. Proceedings of Machine Learning Research, R. Salakhutdinov, Z. Kolter, K. A. Heller, A. Weller, N. Oliver,

J. Scarlett, and F. Berkenkamp, Eds., vol. 235. PMLR / OpenReview.net, 2024, pp. 11 642–11 662. [Online]. Available: https://proceedings.mlr.press/v235/drouin24a.html

[28] T. Xie, D. Zhang, J. Chen, X. Li, S. Zhao, R. Cao, T. J. Hua, Z. Cheng, D. Shin, F. Lei, Y. Liu, Y. Xu, S. Zhou, S. Savarese, C. Xiong, V. Zhong, and T. Yu, "Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments," in *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, Eds., 2024. [Online]. Available: http://papers.nips.cc/paper_files/paper/2024/hash/5d413e48f84dc61244b6be550f1cd8f5-Abstract-Datasets_and_Benchmarks_Track.html

[29] J. Wei, N. Karina, H. W. Chung, Y. J. Jiao, S. Papay, A. Glaese, J. Schulman, and W. Fedus, "Measuring short-form factuality in large language models," 2024. [Online]. Available: https://arxiv.org/abs/2411.04368

[30] G. Mialon, C. Fourrier, T. Wolf, Y. LeCun, and T. Scialom, "GAIA: a benchmark for general AI assistants," in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. [Online]. Available: https://openreview.net/forum?id=fibxvahvs3

[31] J. Wei, Z. Sun, S. Papay, S. McKinney, J. Han, I. Fulford, H. W. Chung, A. T. Passos, W. Fedus, and A. Glaese, "Browsecomp: A simple yet challenging benchmark for browsing agents," *CoRR*, vol. abs/2504.12516, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2504.12516

[32] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox, "ALFRED: A benchmark for interpreting grounded instructions for everyday tasks," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 2020, pp. 10 737–10 746. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/html/Shridhar_ALFRED_A_Benchmark_for_Interpreting_Grounded_Instructions_for_Everyday_Tasks_CVPR_2020_paper.html

[33] M. Shridhar, X. Yuan, M. Côté, Y. Bisk, A. Trischler, and M. J. Hausknecht, "Alfworld: Aligning text and embodied environments for interactive learning," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. [Online]. Available: https://openreview.net/forum?id=0IOX0YcCdTn

[34] R. Wang, P. A. Jansen, M. Côté, and P. Ammanabrolu, "Scienceworld: Is your agent smarter than a 5th grader?" in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Association for Computational Linguistics, 2022, pp. 11 279–11 298. [Online]. Available: https://doi.org/10.18653/v1/2022.emnlp-main.775

[35] L. Hu, Q. Li, A. Xie, N. Jiang, I. Stoica, H. Jin, and H. Zhang, "Gamearena: Evaluating llm reasoning through live computer games," 2025. [Online]. Available: https://arxiv.org/abs/2412.06394

[36] N. Cloos, M. Jens, M. Naim, Y.-L. Kuo, I. Cases, A. Barbu, and C. J. Cueva, "Baba is ai: Break the rules to beat the benchmark," 2025. [Online]. Available: https://arxiv.org/abs/2407.13729

[37] A. Costarelli, M. Allen, R. Hauksson, G. Sodunke, S. Hariharan, C. Cheng, W. Li, J. Clymer, and A. Yadav, "Gamebench: Evaluating strategic reasoning abilities of llm agents," 2024. [Online]. Available: https://arxiv.org/abs/2406.06613

[38] Y. Qin, S. Liang, Y. Ye, K. Zhu, L. Yan, Y. Lu, Y. Lin, X. Cong, X. Tang, B. Qian, S. Zhao, L. Hong, R. Tian, R. Xie, J. Zhou, M. Gerstein, D. Li, Z. Liu, and M. Sun, "Toolllm: Facilitating large language models to master 16000+ real-world apis," in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. [Online]. Available: https://openreview.net/forum?id=dHng2O0Jjr

[39] S. Yao, N. Shinn, P. Razavi, and K. Narasimhan, "τ-bench: A benchmark for tool-agent-user interaction in real-world domains," *CoRR*, vol. abs/2406.12045, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2406.12045

[40] M. Li, Y. Zhao, B. Yu, F. Song, H. Li, H. Yu, Z. Li, F. Huang, and Y. Li, "Api-bank: A comprehensive benchmark for tool-augmented llms," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, H. Bouamor, J. Pino, and K. Bali, Eds. Association for Computational Linguistics, 2023, pp. 3102–3116. [Online]. Available: https://doi.org/10.18653/v1/2023.emnlp-main.187

[41] J. Austin, A. Odena, M. Nye, M. Bosma, H. Michalewski, D. Dohan, E. Jiang, C. Cai, M. Terry, Q. Le *et al.*, "Program synthesis with large language models," *arXiv preprint arXiv:2108.07732*, 2021.

[42] M. A. Merrill, A. G. Shaw, N. Carlini, B. Li, H. Raj, I. Bercovich, L. Shi, J. Y. Shin, T. Walshe, E. K. Buchanan, J. Shen, G. Ye, H. Lin, J. Poulos, M. Wang, M. Nezhurina, J. Jitsev, D. Lu, O. Menis-Mastromichalakis, Z. Xu, Z. Chen, Y. Liu, R. Zhang, L. L. Chen, A. Kashyap, J. Uslu, J. Li, J. Wu, M. Yan, S. Bian, V. Sharma, K. Sun, S. Dillmann, A. Anand, A. Lanpouthakoun, B. Koopah, C. Hu, E. K. Guha, G. H. S. Dreiman, J. Zhu, K. Krauth, L. Zhong, N. Muennighoff, R. Amanfu, S. Tan, S. Pimpalgaonkar, T. Aggarwal, X. Lin, X. Lan, X. Zhao, Y. Liang, Y. Wang, Z. Wang, C. Zhou, D. Heineman, H. Liu, H. Trivedi, J. Yang, J. Lin, M. Shetty, M. Yang, N. Omi, N. Raoof, S. Li, T. Y. Zhuo, W. Lin, Y. Dai, Y. Wang, W. Chai, S. Zhou, D. Wahdany, Z. She, J. Hu, Z. Dong, Y. Zhu, S. Cui, A. Saiyed, A. Kolbeinsson, J. Hu, C. M. Rytting, R. Marten, Y. Wang, A. Dimakis, A. Konwinski, and L. Schmidt, "Terminal-bench: Benchmarking agents on hard, realistic tasks in command line interfaces," *CoRR*, vol. abs/2601.11868, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.11868

[43] A. Ouyang, S. Guo, S. Arora, A. L. Zhang, W. Hu, C. Ré, and A. Mirhoseini, "Kernelbench: Can llms write efficient gpu kernels?" *arXiv preprint arXiv:2502.10517*, 2025.

[44] Y. Jiang, K. C. Black, G. Geng, D. Park, A. Y. Ng, and J. H. Chen, "Medagentbench: Dataset for benchmarking llms as agents in medical applications," *CoRR*, vol. abs/2501.14654, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2501.14654

[45] Z. Chen, S. Chen, Y. Ning, Q. Zhang, B. Wang, B. Yu, Y. Li, Z. Liao, C. Wei, Z. Lu, V. Dey, M. Xue, F. N. Baker, B. Burns, D. Adu-Ampratwum, X. Huang, X. Ning, S. Gao, Y. Su, and H. Sun, "Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery," in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. [Online]. Available: https://openreview.net/forum?id=6z4YKr0GK6

[46] L. Jing, Z. Huang, X. Wang, W. Yao, W. Yu, K. Ma, H. Zhang, X. Du, and D. Yu, "Dsbench: How far are data science agents from becoming data science experts?" in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. [Online]. Available: https://openreview.net/forum?id=DSsSPr0RZJ

[47] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[48] X. Liu, H. Yu, H. Zhang, Y. Xu, X. Lei, H. Lai, Y. Gu, H. Ding, K. Men, K. Yang, S. Zhang, X. Deng, A. Zeng, Z. Du, C. Zhang, S. Shen, T. Zhang, Y. Su, H. Sun, M. Huang, Y. Dong, and J. Tang, "Agentbench: Evaluating llms as agents," in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. [Online]. Available: https://openreview.net/forum?id=zAdUB0aCTQ

[49] Z. He, P. Cao, C. Wang, Z. Jin, Y. Chen, J. Xu, H. Li, K. Liu, and J. Zhao, "Agentscourt: Building judicial decision-making agents with court debate simulation and legal knowledge augmentation," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024, pp. 9399–9416.

[50] K. Zhu, Z. Jin, H. Yuan, J. Li, S. Tu, P. Cao, Y. Chen, K. Liu, and J. Zhao, "Mmr-v: What's left unsaid? a benchmark for multimodal deep reasoning in videos," *arXiv preprint arXiv:2506.04141*, 2025.

[51] J. Li, S. Huang, Z. Jin, C. Zhang, P. Cao, Y. Chen, K. Liu, and J. Zhao, "Mmr-life: Piecing together real-life scenes for multimodal multi-image reasoning," *arXiv preprint arXiv:2603.02024*, 2026.

[52] R. Bonatti, D. Zhao, F. Bonacci, D. Dupont, S. Abdali, Y. Li, Y. Lu, J. Wagle, K. Koishida, A. Bucker *et al.*, "Windows agent arena: Evaluating multi-modal os agents at scale," *arXiv preprint arXiv:2409.08264*, 2024.

[53] H. Jia, J. Liao, X. Zhang, H. Xu, T. Xie, C. Jiang, M. Yan, S. Liu, W. Ye, and F. Huang, "Osworld-mcp: Benchmarking MCP tool invocation in computer-use agents," *CoRR*, vol. abs/2510.24563, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.24563

[54] D. Zhang, Z. Shen, R. Xie, S. Zhang, T. Xie, Z. Zhao, S. Chen, L. Chen, H. Xu, R. Cao *et al.*, "Mobile-env: Building qualified evaluation benchmarks for llm-gui interaction," *arXiv preprint arXiv:2305.08144*, 2023.

[55] C. Rawles, A. Li, D. Rodriguez, O. Riva, and T. P. Lillicrap, "Android in the wild: A large-scale dataset for android device control," *CoRR*, vol. abs/2307.10088, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2307.10088

[56] C. Rawles, S. Clinckemaillie, Y. Chang, J. Waltz, G. Lau, M. Fair, A. Li, W. E. Bishop, W. Li, F. Campbell-Ajala, D. K. Toyama, R. J. Berry, D. Tyamagundlu, T. P. Lillicrap, and O. Riva, "Androidworld: A dynamic benchmarking environment for autonomous agents," in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. [Online]. Available: https://openreview.net/forum?id=il5yUQsrjC

[57] Q. Kong, X. Zhang, Z. Yang, N. Gao, C. Liu, P. Tong, C. Cai, H. Zhou, J. Zhang, L. Chen, Z. Liu, S. Hoi, and Y. Wang, "Mobileworld: Benchmarking autonomous mobile agents in agent-user interactive and mcp-augmented environments," *CoRR*, vol. abs/2512.19432, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2512.19432

[58] S. Deng, W. Xu, H. Sun, W. Liu, T. Tan, L. Liujianfeng, A. Li, J. Luan, B. Wang, R. Yan *et al.*, "Mobile-bench: An evaluation benchmark for llm-based mobile agents," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 8813–8831.

[59] X. Deng, Y. Gu, B. Zheng, S. Chen, S. Stevens, B. Wang, H. Sun, and Y. Su, "Mind2web: Towards a generalist agent for the web," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/5950bf290a1570ea401bf98882128160-Abstract-Datasets_and_Benchmarks.html

[60] S. Zhou, F. F. Xu, H. Zhu, X. Zhou, R. Lo, A. Sridhar, X. Cheng, T. Ou, Y. Bisk, D. Fried, U. Alon, and G. Neubig, "Webarena: A realistic web environment for building autonomous agents," in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. [Online]. Available: https://openreview.net/forum?id=oKn9c6ytLx

[61] J. Y. Koh, R. Lo, L. Jang, V. Duvvur, M. C. Lim, P. Huang, G. Neubig, S. Zhou, R. Salakhutdinov, and D. Fried, "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), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, L. Ku, A. Martins, and V. Srikumar, Eds. Association for Computational Linguistics, 2024, pp. 881–905. [Online]. Available: https://doi.org/10.18653/v1/2024.acl-long.50

[62] R. Wong, J. Wang, J. Zhao, L. Chen, Y. Gao, L. Zhang, X. Zhou, Z. Wang, K. Xiang, G. Zhang, W. Huang, Y. Wang, and K. Wang, "Widesearch: Benchmarking agentic broad info-seeking," *CoRR*, vol. abs/2508.07999, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.07999

[63] Y. Xi, J. Lin, M. Zhu, Y. Xiao, Z. Ou, J. Liu, T. Wan, B. Chen, W. Liu, Y. Wang, R. Tang, W. Zhang, and Y. Yu, "Infodeepseek: Benchmarking agentic information seeking for retrieval-augmented generation," *CoRR*, vol. abs/2505.15872, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.15872

[64] Z. Xia, K. Luo, H. Qian, and Z. Liu, "Open data synthesis for deep research," *CoRR*, vol. abs/2509.00375, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.00375

[65] P. Huang, Z. Zhong, Z. Wan, D. Zhou, S. Alam, X. Wang, Z. Li, Z. Dou, L. Zhu, J. Xiong, C. Tao, Y. Xu, D. Dimitriadis, T. Zhang, and M. Zhang, "Mmdeepresearch-bench: A benchmark for multimodal deep research agents," *CoRR*, vol. abs/2601.12346, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.12346

[66] J. Wu, W. Yin, Y. Jiang, Z. Wang, Z. Xi, R. Fang, L. Zhang, Y. He, D. Zhou, P. Xie, and F. Huang, "Webwalker: Benchmarking llms in web traversal," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds. Association for Computational Linguistics, 2025, pp. 10 290–10 305. [Online]. Available: https://aclanthology.org/2025.acl-long.508/

[67] A. Cattan, A. Jacovi, O. Ram, J. Herzig, R. Aharoni, S. Goldshtein, E. Ofek, I. Szpektor, and A. Caciularu, "Dragged into conflicts: Detecting and addressing conflicting sources in search-augmented llms," *CoRR*, vol. abs/2506.08500, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2506.08500

[68] P. Zhou, B. Leon, X. Ying, C. Zhang, Y. Shao, Q. Ye, D. Chong, Z. Jin, C. Xie, M. Cao, Y. Gu, S. Hong, J. Ren, J. Chen, C. Liu, and Y. Hua, "Browsecomp-zh: Benchmarking web browsing ability of large language models in chinese," *CoRR*, vol. abs/2504.19314, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2504.19314

[69] A. Java, A. Khandelwal, S. P. Midigeshi, A. Halfaker, A. Deshpande, N. Goyal, A. Gupta, N. Natarajan, and A. Sharma, "Characterizing deep research: A benchmark and formal definition," *CoRR*, vol. abs/2508.04183, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.04183

[70] X. Li, W. Jiao, J. Jin, S. Wang, G. Dong, J. Jin, H. Wang, Y. Wang, J. Wen, Y. Lu, and Z. Dou, "Omnigaia: Towards native omni-modal AI agents," *CoRR*, vol. abs/2602.22897, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2602.22897

[71] M. Du, B. Xu, C. Zhu, X. Wang, and Z. Mao, "Deepresearch bench: A comprehensive benchmark for deep research agents," *CoRR*, vol. abs/2506.11763, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2506.11763

[72] Z. Yang, B. Pan, H. Wang, Y. Wang, X. Liu, M. Zhu, B. Zhang, and W. Chen, "Multimodal deepresearcher: Generating text-chart interleaved reports from scratch with agentic framework," *CoRR*, vol. abs/2506.02454, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2506.02454

[73] Y. Yao, Y. Wang, Y. Zhang, Y. Lu, T. Gu, L. Li, D. Zhao, K. Wu, H. Wang, P. Nie, Y. Teng, and Y. Wang, "Dr. bench: A multidimensional evaluation for deep research agents, from answers to reports," 2026. [Online]. Available: https://arxiv.org/abs/2510.02190

[74] T. Bao, M. T. Nayeem, D. Rafiei, and C. Zhang, "Surveygen: Quality-aware scientific survey generation with large language models," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, Eds. Association for Computational Linguistics, 2025, pp. 2712–2736. [Online]. Available: https://doi.org/10.18653/v1/2025.emnlp-main.136

[75] M. Li, Y. Zeng, Z. Cheng, C. Ma, and K. Jia, "Reportbench: Evaluating deep research agents via academic survey tasks," *CoRR*, vol. abs/2508.15804, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.15804

[76] A. Asai, J. He, R. Shao, W. Shi, A. Singh, J. C. Chang, K. Lo, L. Soldaini, S. Feldman, M. D'Arcy, D. Wadden, M. Latzke, M. Tian, P. Ji, S. Liu, H. Tong, B. Wu, Y. Xiong, L. Zettlemoyer, G. Neubig, D. S. Weld, D. Downey, W. Yih, P. W. Koh, and H. Hajishirzi, "Openscholar: Synthesizing scientific literature with retrieval-augmented lms," *CoRR*, vol. abs/2411.14199, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2411.14199

[77] H. Tan, Z. Guo, Z. Shi, L. Xu, Z. Liu, Y. Feng, X. Li, Y. Wang, L. Shang, Q. Liu, and L. Song, "Proxyqa: An alternative framework for evaluating long-form text generation with large language models," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, L. Ku, A. Martins, and V. Srikumar, Eds. Association for Computational Linguistics, 2024, pp. 6806–6827. [Online]. Available: https://doi.org/10.18653/v1/2024.acl-long.368

[78] J. Coelho, J. Ning, J. He, K. Mao, A. Paladugu, P. Setlur, J. Jin, J. Callan, J. Magalhães, B. Martins, and C. Xiong, "Deepresearchgym: A free, transparent, and reproducible evaluation sandbox for deep research," *CoRR*, vol. abs/2505.19253, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.19253

[79] M. Savva, J. Malik, D. Parikh, D. Batra, A. Kadian,

O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, and V. Koltun, "Habitat: A platform for embodied AI research," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019, pp. 9338–9346. [Online]. Available: https://doi.org/10.1109/ICCV.2019.00943

[80] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. D. Reid, S. Gould, and A. van den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, 2018, pp. 3674–3683. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Anderson_Vision-and-Language_Navigation_Interpreting_CVPR_2018_paper.html

[81] Q. Li, Z. Peng, L. Feng, Q. Zhang, Z. Xue, and B. Zhou, "Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3461–3475, 2023. [Online]. Available: https://doi.org/10.1109/TPAMI.2022.3190471

[82] A. Ku, P. Anderson, R. Patel, E. Ie, and J. Baldridge, "Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Association for Computational Linguistics, 2020, pp. 4392–4412. [Online]. Available: https://doi.org/10.18653/v1/2020.emnlp-main.356

[83] J. Krantz, E. Wijmans, A. Majumdar, D. Batra, and S. Lee, "Beyond the nav-graph: Vision-and-language navigation in continuous environments," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXVIII*, ser. Lecture Notes in Computer Science, A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds. Springer, 2020, pp. 104–120. [Online]. Available: https://doi.org/10.1007/978-3-030-58604-1_7

[84] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3019–3026, 2020.

[85] S. Srivastava, C. Li, M. Lingelbach, R. Martín-Martín, F. Xia, K. E. Vainio, Z. Lian, C. Gokmen, S. Buch, C. K. Liu, S. Savarese, H. Gweon, J. Wu, and L. Fei-Fei, "BEHAVIOR: benchmark for everyday household activities in virtual, interactive, and ecological environments," in *Conference on Robot Learning, 8-11 November 2021, London, UK*, ser. Proceedings of Machine Learning Research, A. Faust, D. Hsu, and G. Neumann, Eds., vol. 164. PMLR, 2021, pp. 477–490. [Online]. Available: https://proceedings.mlr.press/v164/srivastava22a.html

[86] Q. Gallouédec, N. Cazin, E. Dellandréa, and L. Chen, "panda-gym: Open-source goal-conditioned environments for robotic learning," *arXiv preprint arXiv:2106.13687*, 2021.

[87] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu, "Robocasa: Large-scale simulation of everyday tasks for generalist robots," *arXiv preprint arXiv:2406.02523*, 2024.

[88] Y. Qin, L. Kang, X. Song, Z. Yin, X. Liu, X. Liu, R. Zhang, and L. Bai, "Robofactory: Exploring embodied agent collaboration with compositional constraints," *CoRR*, vol. abs/2503.16408, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2503.16408

[89] L. Zhang, Y. Wang, H. Gu, A. Hamidizadeh, Z. Zhang, Y. Liu, Y. Wang, D. G. A. Bravo, J. Dong, S. Zhou, T. Cao, X. Quan, Y. Zhuang, Y. Zhang, and J. Hao, "Et-plan-bench: Embodied task-level planning benchmark towards spatial-temporal cognition with foundation models," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2025, Hangzhou, China, October 19-25, 2025*. IEEE, 2025, pp. 21 566–21 573. [Online]. Available: https://doi.org/10.1109/IROS60139.2025.11246658

[90] Z. Cheng, Z. Wang, J. Hu, S. Hu, A. Liu, Y. Tu, P. Li, L. Shi, Z. Liu, and M. Sun, "LEGENT: open platform for embodied agents," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, Y. Cao, Y. Feng, and D. Xiong, Eds. Association for Computational Linguistics, 2024, pp. 335–345. [Online]. Available: https://doi.org/10.18653/v1/2024.acl-demos.32

[91] L. Fan, G. Wang, Y. Jiang, A. Mandlekar, Y. Yang, H. Zhu, A. Tang, D. Huang, Y. Zhu, and A. Anandkumar, "Minedojo: Building open-ended embodied agents with internet-scale knowledge," in *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., 2022. [Online]. Available: http://papers.nips.cc/paper_files/paper/2022/hash/74a67268c5cc5910f64938cac4526a90-Abstract-Datasets_and_Benchmarks.html

[92] Z. Wang, S. Cai, G. Chen, A. Liu, X. Ma, and Y. Liang, "Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents," 2024. [Online]. Available: https://arxiv.org/abs/2302.01560

[93] J. Guo, Y. Ye, T. He, H. Wu, Y. Jiang, T. Pearce, and J. Bian, "Mineworld: a real-time and open-source interactive world model on minecraft," 2025. [Online]. Available: https://arxiv.org/abs/2504.08388

[94] M. U. Nasir, S. James, and J. Togelius, "Gametraversalbenchmark: Evaluating planning abilities of large language models through traversing 2d game maps," 2024. [Online]. Available: https://arxiv.org/abs/2410.07765

[95] Y. Wu, X. Tang, T. M. Mitchell, and Y. Li, "Smartplay: A benchmark for llms as intelligent agents," 2024. [Online]. Available: https://arxiv.org/abs/2310.01557

[96] F. Hudi, G. I. Winata, R. Zhang, and A. F. Aji, "Textgames: Learning to self-play text-based puzzle games via language model reasoning," 2025. [Online]. Available: https://arxiv.org/abs/2502.18431

[97] M. Chevalier-Boisvert, B. Dai, M. Towers, R. de Lazcano, L. Willems, S. Lahlou, S. Pal, P. S. Castro, and J. Terry, "Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks," 2023. [Online]. Available: https://arxiv.org/abs/2306.13831

[98] K. Chalamalasetti, J. Götze, S. Hakimov, B. Madureira, P. Sadler, and D. Schlangen, "Clembench: Using game play to evaluate chat-optimized language models as conversational agents," 2023. [Online]. Available: https://arxiv.org/abs/2305.13455

[99] M. Abdulhai, I. White, C. Snell, C. Sun, J. Hong, Y. Zhai, K. Xu, and S. Levine, "Lmrl gym: Benchmarks for multi-turn reinforcement learning with language models," 2023. [Online]. Available: https://arxiv.org/abs/2311.18232

[100] J. Light, M. Cai, S. Shen, and Z. Hu, "Avalonbench: Evaluating llms playing the game of avalon," 2023. [Online]. Available: https://arxiv.org/abs/2310.05036

[101] S. Wu, L. Zhu, T. Yang, S. Xu, Q. Fu, Y. Wei, and H. Fu, "Enhance reasoning for large language models in the game werewolf," 2024. [Online]. Available: https://arxiv.org/abs/2402.02330

[102] "Whodunitbench: Evaluating llms on murder mystery games," in *NeurIPS 2024 Datasets and Benchmarks Track*, 2024.

[103] L. Guertler, B. Cheng, S. Yu, B. Liu, L. Choshen, and C. Tan, "Textarena," 2025. [Online]. Available: https://arxiv.org/abs/2504.11442

[104] S. Wang, C. Liu, Z. Zheng, S. Qi, S. Chen, Q. Yang, A. Zhao, C. Wang, S. Song, and G. Huang, "Avalon's game of thoughts: Battle against deception through recursive contemplation," 2023. [Online]. Available: https://arxiv.org/abs/2310.01320

[105] J. Ahn, J. Kim, H. Yun, J. Son, D. Park, J. Cho, and G. Kim, "Flashadventure: A benchmark for gui agents solving full story arcs in diverse adventure games," 2025. [Online]. Available: https://arxiv.org/abs/2509.01052

[106] D. Paglieri, B. Cupiał, S. Coward, U. Piterbarg, M. Wolczyk, A. Khan, E. Pignatelli, Łukasz Kuciński, L. Pinto, R. Fergus, J. N. Foerster, J. Parker-Holder, and T. Rocktäschel, "Balrog: Benchmarking agentic llm and vlm reasoning on games," 2025. [Online]. Available: https://arxiv.org/abs/2411.13543

[107] A. L. Zhang, T. L. Griffiths, K. R. Narasimhan, and O. Press, "Videogamebench: Can vision-language models complete popular video games?" 2025. [Online]. Available: https://arxiv.org/abs/2505.18134

[108] L. Hu, M. Huo, Y. Zhang, H. Yu, E. P. Xing, I. Stoica, T. Rosing, H. Jin, and H. Zhang, "lmgame-bench: How good are llms at playing games?" *CoRR*, vol. abs/2505.15146, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.15146

[109] L. Ying, R. Truong, P. Sharma, K. I. Zhao, N. Cloos, K. R. Allen, T. L. Griffiths, K. M. Collins, J. Hernández-

Orallo, P. Isola, S. J. Gershman, and J. B. Tenenbaum, "Ai gamestore: Scalable, open-ended evaluation of machine general intelligence with human games," 2026. [Online]. Available: https://arxiv.org/abs/2602.17594

[110] S. Qi, S. Chen, Y. Li, X. Kong, J. Wang, B. Yang, P. Wong, Y. Zhong, X. Zhang, Z. Zhang, N. Liu, W. Wang, Y. Yang, and S.-C. Zhu, "Civrealm: A learning and reasoning odyssey in civilization for decision-making agents," 2024. [Online]. Available: https://arxiv.org/abs/2401.10568

[111] J. Hopkins, M. Bakler, and A. Khan, "Factorio learning environment," 2025. [Online]. Available: https://arxiv.org/abs/2503.09617

[112] J. Duan, R. Zhang, J. Diffenderfer, B. Kailkhura, L. Sun, E. Stengel-Eskin, M. Bansal, T. Chen, and K. Xu, "Gtbench: Uncovering the strategic reasoning limitations of llms via game-theoretic evaluations," 2024. [Online]. Available: https://arxiv.org/abs/2402.12348

[113] J. Liu, C. Yu, J. Gao, Y. Xie, Q. Liao, Y. Wu, and Y. Wang, "Llm-powered hierarchical language agent for real-time human-ai coordination," 2024. [Online]. Available: https://arxiv.org/abs/2312.15224

[114] C. Ruhdorfer, M. Bortoletto, A. Penzkofer, and A. Bulling, "The overcooked generalisation challenge: Evaluating cooperation with novel partners in unknown environments using unsupervised environment design," 2025. [Online]. Available: https://arxiv.org/abs/2406.17949

[115] H. Sun, S. Zhang, L. Niu, L. Ren, H. Xu, H. Fu, F. Zhao, C. Yuan, and X. Wang, "Collab-overcooked: Benchmarking and evaluating large language models as collaborative agents," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2025, p. 4922–4951. [Online]. Available: http://dx.doi.org/10.18653/v1/2025.emnlp-main.249

[116] J. Chen, X. Hu, S. Liu, S. Huang, W.-W. Tu, Z. He, and L. Wen, "Llmarena: Assessing capabilities of large language models in dynamic multi-agent environments," 2024. [Online]. Available: https://arxiv.org/abs/2402.16499

[117] W. Lin, J. Roberts, Y. Yang, S. Albanie, Z. Lu, and K. Han, "Gamebot: Transparent assessment of llm reasoning in games," 2025. [Online]. Available: https://arxiv.org/abs/2412.13602

[118] H. Trivedi, T. Khot, M. Hartmann, R. Manku, V. Dong, E. Li, S. Gupta, A. Sabharwal, and N. Balasubramanian, "Appworld: A controllable world of apps and people for benchmarking interactive coding agents," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, L. Ku, A. Martins, and V. Srikumar, Eds. Association for Computational Linguistics, 2024, pp. 16 022–16 076. [Online]. Available: https://doi.org/10.18653/v1/2024.acl-long.850

[119] S. G. Patil, H. Mao, F. Yan, C. C. Ji, V. Suresh, I. Stoica, and J. E. Gonzalez, "The berkeley function calling leaderboard (BFCL): from tool use to agentic evaluation of large language models," in *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, ser. Proceedings of Machine Learning Research, A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, Eds., vol. 267. PMLR / OpenReview.net, 2025. [Online]. Available: https://proceedings.mlr.press/v267/patil25a.html

[120] Z. Guo, S. Cheng, H. Wang, S. Liang, Y. Qin, P. Li, Z. Liu, M. Sun, and Y. Liu, "Stabletoolbench: Towards stable large-scale benchmarking on tool learning of large language models," in *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, ser. Findings of ACL, L. Ku, A. Martins, and V. Srikumar, Eds. Association for Computational Linguistics, 2024, pp. 11 143–11 156. [Online]. Available: https://doi.org/10.18653/v1/2024.findings-acl.664

[121] V. Barres, H. Dong, S. Ray, X. Si, and K. Narasimhan, "τ^2-bench: Evaluating conversational agents in a dual-control environment," *CoRR*, vol. abs/2506.07982, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2506.07982

[122] C. Qian, Z. Liu, A. Prabhakar, Z. Liu, J. Zhang, H. Chen, H. Ji, W. Yao, S. Heinecke, S. Savarese, C. Xiong, and H. Wang, "Userbench: An interactive gym environment for user-centric agents," *CoRR*, vol. abs/2507.22034, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2507.22034

[123] N. Farn and R. Shin, "Tooltalk: Evaluating tool-usage in a conversational setting," *CoRR*, vol. abs/2311.10775, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2311.10775

[124] X. Wang, Z. Wang, J. Liu, Y. Chen, L. Yuan, H. Peng, and H. Ji, "MINT: evaluating llms in multi-turn interaction with tools and language feedback," in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. [Online]. Available: https://openreview.net/forum?id=jp3gWrMuIZ

[125] F. Lei, Y. Yang, W. Sun, and D. Lin, "Mcpverse: An expansive, real-world benchmark for agentic tool use," *CoRR*, vol. abs/2508.16260, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.16260

[126] Z. Luo, Z. Shen, W. Yang, Z. Zhao, P. Jwalapuram, A. Saha, D. Sahoo, S. Savarese, C. Xiong, and J. Li, "Mcp-universe: Benchmarking large language models with real-world model context protocol servers," *CoRR*, vol. abs/2508.14704, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.14704

[127] Z. Wang, Q. Chang, H. Patel, S. Biju, C. Wu, Q. Liu, A. Ding, A. Rezazadeh, A. Shah, Y. Bao, and E. Siow, "Mcp-bench: Benchmarking tool-using LLM agents with complex real-world tasks via MCP servers," *CoRR*, vol. abs/2508.20453, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.20453

[128] S. Fan, X. Ding, L. Zhang, and L. Mo, "Mcptoolbench++: A large scale AI agent model context protocol MCP tool use benchmark," *CoRR*, vol. abs/2508.07575, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.07575

[129] K. Zhang, J. Li, G. Li, X. Shi, and Z. Jin, "Codeagent: Enhancing code generation with tool-integrated agent systems for real-world repo-level coding challenges," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 13 643–13 658.

[130] W. Li, X. Zhang, Z. Guo, S. Mao, W. Luo, G. Peng, Y. Huang, H. Wang, and S. Li, "Fea-bench: A benchmark for evaluating repository-level code generation for feature implementation," 2025. [Online]. Available: https://arxiv.org/abs/2503.06680

[131] T. Y. Zhuo, M. C. Vu, J. Chim, H. Hu, W. Yu, R. Widyasari, I. N. B. Yusuf, H. Zhan, J. He, I. Paul, S. Brunner, C. Gong, J. Hoang, A. R. Zebaze, X. Hong, W. Li, J. Kaddour, M. Xu, Z. Zhang, P. Yadav, and et al., "Bigcodebench: Benchmarking code generation with diverse function calls and complex instructions," in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. [Online]. Available: https://openreview.net/forum?id=YrycTjllL0

[132] N. Jain, K. Han, A. Gu, W.-D. Li, F. Yan, T. Zhang, S. Wang, A. Solar-Lezama, K. Sen, and I. Stoica, "Livecodebench: Holistic and contamination free evaluation of large language models for code," *arXiv preprint arXiv:2403.07974*, 2024.

[133] A. Gu, B. Rozière, H. Leather, A. Solar-Lezama, G. Synnaeve, and S. I. Wang, "Cruxeval: A benchmark for code reasoning, understanding and execution," 2024. [Online]. Available: https://arxiv.org/abs/2401.03065

[134] J. Ding, S. Long, C. Pu, H. Zhou, H. Gao, X. Gao, C. He, Y. Hou, F. Hu, Z. Li, W. Shi, Z. Wang, D. Zan, C. Zhang, X. Zhang, Q. Chen, X. Cheng, B. Deng, Q. Gu, K. Hua, J. Lin, P. Liu, M. Li, X. Pan, Z. Peng, Y. Qin, Y. Shan, Z. Tan, W. Xie, Z. Wang, Y. Yuan, J. Zhang, E. Zhao, Y. Zhao, H. Zhu, L. Zhu, C. Zou, M. Ding, J. Jiao, J. Liu, M. Liu, Q. Liu, C. Tao, J. Yang, T. Yang, Z. Zhang, X. Chen, W. Huang, and G. Zhang, "Nl2repo-bench: Towards long-horizon repository generation evaluation of coding agents," 2026. [Online]. Available: https://arxiv.org/abs/2512.12730

[135] J. Yang, C. E. Jimenez, A. L. Zhang, K. Lieret, J. Yang, X. Wu, O. Press, N. Muennighoff, G. Synnaeve, K. R. Narasimhan *et al.*, "Swe-bench multimodal: Do ai systems generalize to visual software domains?" *arXiv preprint arXiv:2410.03859*, 2024.

[136] Y. Xiao, R. Wang, L. Kong, D. Golac, and W. Wang, "Csr-bench: Benchmarking llm agents in deployment of computer science research repositories," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2025, pp. 12 705–12 723.

[137] N. Mündler, M. N. Müller, J. He, and M. Vechev, "Swt-bench: Testing and validating real-world bug-fixes with code agents," 2025. [Online]. Available: https://arxiv.org/abs/2406.12952

[138] J. Yang, A. Prabhakar, K. Narasimhan, and S. Yao, "Intercode: Standardizing and benchmarking interactive coding with execu-

tion feedback," *Advances in Neural Information Processing Systems*, vol. 36, pp. 23 826–23 854, 2023.

[139] X. Deng, J. Da, E. Pan, Y. Y. He, C. Ide, K. Garg, N. Lauffer, A. Park, N. Pasari, C. Rane, K. Sampath, M. Krishnan, S. Kundurthy, S. Hendryx, Z. Wang, C. B. C. Zhang, N. Jacobson, B. Liu, and B. Kenstler, "Swe-bench pro: Can AI agents solve long-horizon software engineering tasks?" *CoRR*, vol. abs/2509.16941, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.16941

[140] R. Tian, Y. Ye, Y. Qin, X. Cong, Y. Lin, Y. Pan, Y. Wu, H. Hui, W. Liu, Z. Liu, and M. Sun, "Debugbench: Evaluating debugging capability of large language models," 2024. [Online]. Available: https://arxiv.org/abs/2401.04621

[141] R. Xu, Y. Zhuang, Y. Zhong, Y. Yu, Z. Wang, X. Tang, H. Wu, M. D. Wang, P. Ruan, D. Yang *et al.*, "Medagentgym: A scalable agentic training environment for code-centric reasoning in biomedical data science," *arXiv preprint arXiv:2506.04405*, 2025.

[142] D. Fa, M. Culjak, B. Pandza, and M. Cupic, "Bioagent bench: An AI agent evaluation suite for bioinformatics," *CoRR*, vol. abs/2601.21800, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.21800

[143] X. Tang, B. Qian, R. Gao, J. Chen, X. Chen, and M. B. Gerstein, "Biocoder: a benchmark for bioinformatics code generation with large language models," *Bioinform.*, vol. 40, no. Supplement_1, pp. i266–i276, 2024. [Online]. Available: https://doi.org/10.1093/bioinformatics/btae230

[144] Y. Zhang, Q. Jiang, X. Han, N. Chen, Y. Yang, and K. Ren, "Benchmarking data science agents," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, L. Ku, A. Martins, and V. Srikumar, Eds. Association for Computational Linguistics, 2024, pp. 5677–5700. [Online]. Available: https://doi.org/10.18653/v1/2024.acl-long.308

[145] J. S. Chan, N. Chowdhury, O. Jaffe, J. Aung, D. Sherburn, E. Mays, G. Starace, K. Liu, L. Maksin, T. Patwardhan, A. Madry, and L. Weng, "Mle-bench: Evaluating machine learning agents on machine learning engineering," in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. [Online]. Available: https://openreview.net/forum?id=6s5uXNWGIh

[146] R. Qiang, Y. Zhuang, Y. Li, R. Zhang, C. Li, I. S.-H. Wong, S. Yang, P. Liang, C. Zhang, B. Dai *et al.*, "Mle-dojo: Interactive environments for empowering llm agents in machine learning engineering," *arXiv preprint arXiv:2505.07782*, 2025.

[147] F. Zhu, X. Y. Ng, Z. Liu, C. Liu, X. Zeng, C. Wang, T. Tan, X. Yao, P. Shao, M. Xu, Z. Wang, J. Wang, X. Li, J. Li, J. Zhu, Y. Zhang, W. Wang, F. Feng, R. Hong, H. Luan, K. Huang, and T. Chua, "Findeepresearch: Evaluating deep research agents in rigorous financial analysis," *CoRR*, vol. abs/2510.13936, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.13936

[148] Y. Chen, Z. Yao, Y. Liu, J. Ye, J. Yu, L. Hou, and J. Li, "Stockbench: Can LLM agents trade stocks profitably in real-world markets?" *CoRR*, vol. abs/2510.02209, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.02209

[149] A. Bigeard, L. Nashold, R. Krishnan, and S. Wu, "Finance agent benchmark: Benchmarking llms on real-world financial research tasks," *CoRR*, vol. abs/2508.00828, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.00828

[150] Y. Shen, K. Song, X. Tan, D. Li, W. Lu, and Y. Zhuang, "Hugginggpt: Solving AI tasks with chatgpt and its friends in hugging face," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/77c33e6a367922d003ff102ffb92b658-Abstract-Conference.html

[151] C. Ma, J. Zhang, Z. Zhu, C. Yang, Y. Yang, Y. Jin, Z. Lan, L. Kong, and J. He, "Agentboard: An analytical evaluation board of multi-turn LLM agents," in *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, Eds., 2024. [Online]. Available: http://papers.nips.cc/paper_files/paper/2024/hash/877b40688e330a0e2a3fc24084208dfa-Abstract-Datasets_and_Benchmarks_Track.html

[152] J. Pan, X. Wang, G. Neubig, N. Jaitly, H. Ji, A. Suhr, and Y. Zhang, "Training software engineering agents and verifiers with swe-gym," in *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, ser. Proceedings of Machine Learning Research, A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, Eds., vol. 267. PMLR / OpenReview.net, 2025. [Online]. Available: https://proceedings.mlr.press/v267/pan25g.html

[153] J. Yang, K. Leret, C. E. Jimenez, A. Wettig, K. Khandpur, Y. Zhang, B. Hui, O. Press, L. Schmidt, and D. Yang, "Swe-smith: Scaling data for software engineering agents," *CoRR*, vol. abs/2504.21798, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2504.21798

[154] R. Fang, S. Cai, B. Li, J. Wu, G. Li, W. Yin, X. Wang, X. Wang, L. Su, Z. Zhang, S. Wu, Z. Tao, Y. Jiang, P. Xie, F. Huang, and J. Zhou, "Towards general agentic intelligence via environment scaling," *CoRR*, vol. abs/2509.13311, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.13311

[155] M. Hu, B. Xia, Y. Wu, A. Yu, Y. Zou, Q. Chen, S. Wang, J. Jin, K. Li, W. Jiao, Y. Lu, and P. Luo, "Agent2world: Learning to generate symbolic world models via adaptive multi-agent feedback," *CoRR*, vol. abs/2512.22336, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2512.22336

[156] M. Hu, T. Chen, Y. Zou, Y. Lei, Q. Chen, M. Li, Y. Mu, H. Zhang, W. Shao, and P. Luo, "Text2world: Benchmarking large language models for symbolic world model generation," in *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, ser. Findings of ACL, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds. Association for Computational Linguistics, 2025, pp. 26 043–26 066. [Online]. Available: https://aclanthology.org/2025.findings-acl.1337/

[157] H. Tang, D. Key, and K. Ellis, "Worldcoder, a model-based LLM agent: Building world models by writing code and interacting with the environment," in *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, Eds., 2024. [Online]. Available: http://papers.nips.cc/paper_files/paper/2024/hash/820c61a0cd419163ccbd2c33b268816e-Abstract-Conference.html

[158] D. Cheng, S. Huang, Y. Gu, H. Song, G. Chen, L. Dong, W. X. Zhao, J. Wen, and F. Wei, "Llm-in-sandbox elicits general agentic intelligence," *CoRR*, vol. abs/2601.16206, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.16206

[159] N. Jain, J. Singh, M. Shetty, L. Zheng, K. Sen, and I. Stoica, "R2e-gym: Procedural environments and hybrid verifiers for scaling open-weights SWE agents," *CoRR*, vol. abs/2504.07164, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2504.07164

[160] J. Zhao, G. Chen, F. Meng, M. Li, J. Chen, H. Xu, Y. Sun, X. Zhao, R. Song, Y. Zhang *et al.*, "Immersion in the github universe: Scaling coding agents to mastery," *arXiv preprint arXiv:2602.09892*, 2026.

[161] X. Song, H. Chang, G. Dong, Y. Zhu, Z. Dou, and J. Wen, "Envscaler: Scaling tool-interactive environments for LLM agent via programmatic synthesis," *CoRR*, vol. abs/2601.05808, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.05808

[162] J. Xie, D. Xu, X. Zhao, and D. Song, "Agentsynth: Scalable task generation for generalist computer-use agents," *CoRR*, vol. abs/2506.14205, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2506.14205

[163] A. Zala, J. Cho, H. Lin, J. Yoon, and M. Bansal, "Envgen: Generating and adapting environments via llms for training embodied agents," *CoRR*, vol. abs/2403.12014, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2403.12014

[164] R. Yang, H. Chen, J. Zhang, M. Zhao, C. Qian, K. Wang, Q. Wang, T. V. Koripella, M. Movahedi, M. Li, H. Ji, H. Zhang, and T. Zhang, "Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents," in *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, ser. Proceedings of Machine Learning Research, A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, Eds., vol. 267. PMLR / OpenReview.net, 2025. [Online]. Available: https://proceedings.mlr.press/v267/yang25f.html

[165] D. Shi, J. Cao, Q. Chen, W. Sun, W. Li, H. Lu, F. Dong, T. Qin, K. Zhu, M. Liu, J. Yang, G. Zhang, J. Liu, C. Zhang, J. Wang, Y. E. Jiang, and W. Zhou, "Taskcraft: Automated generation of agentic tasks," *CoRR*, vol. abs/2506.10055, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2506.10055

[166] A. Chen, C. Zhang, J. Liu, J. Chen, C. Du, Y. Li, M. Zhong, Q. Wang, Z. Zhu, J. Song, K. Ji, J. He, P. Zhao, and Y. Xiao, "Dive: Scaling diversity in agentic task synthesis for generalizable tool use," 2026. [Online]. Available: https://arxiv.org/abs/2603.11076

[167] H. Chae, J. Park, and A. Ritter, "Safe and scalable web agent learning via recreated websites," *arXiv preprint arXiv:2603.10505*, 2026.

[168] Y. Wu, Y. Peng, Y. Chen, J. Ruan, Z. Zhuang, C. Yang, J. Zhang, M. Chen, Y. Tseng, Z. Yu *et al.*, "Autowebworld: Synthesizing infinite verifiable web environments via finite state machines," *arXiv preprint arXiv:2602.14296*, 2026.

[169] Y. Zhu, Y. Huang, S. Qin, Z. Huang, S. Zhang, and X. Zhang, "Medmcp-calc: Benchmarking llms for realistic medical calculator scenarios via MCP integration," *CoRR*, vol. abs/2601.23049, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.23049

[170] J. Zhang, Y. Peng, F. Kong, C. Yang, Y. Wu, Z. Yu, J. Xiang, J. Ruan, J. Wang, M. Song, H. Liu, X. Tang, B. Liu, C. Wu, and Y. Luo, "Autoenv: Automated environments for measuring cross-environment agent learning," *CoRR*, vol. abs/2511.19304, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2511.19304

[171] Y. Zeng, W. Lu, L. Liu, S. Li, Z. Qu, C. Zhu, S. Li, Z. Tan, M. Liu, H. Zhao *et al.*, "Logigen: Logic-driven generation of verifiable agentic tasks," *arXiv preprint arXiv:2603.00540*, 2026.

[172] Z. Zhang, Z. Wang, X. Zhang, Z. Guo, J. Li, B. Li, and Y. Lu, "Infiniteweb: Scalable web environment synthesis for GUI agent training," *CoRR*, vol. abs/2601.04126, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.04126

[173] E. Gestrin, M. Kuhlmann, and J. Seipp, "Nl2plan: Robust llm-driven planning from minimal text descriptions," *CoRR*, vol. abs/2405.04215, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2405.04215

[174] D. Tu, H. Hao, H. Yang, Y. Chen, Y.-K. Zhang, Z. Xia, Y. Yang, Y. Sun, X. Liu, F. Shen *et al.*, "Scaleenv: Scaling environment synthesis from scratch for generalist interactive tool-use agent training," *arXiv preprint arXiv:2602.06820*, 2026.

[175] S. Cai, R. Fang, J. Wu, B. Li, X. Wang, Y. Jiang, L. Su, L. Zhang, W. Yin, Z. Zhang, F. Feng, P. Xie, and X. Wang, "Autoforge: Automated environment synthesis for agentic reinforcement learning," *CoRR*, vol. abs/2512.22857, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2512.22857

[176] Y. Zhu, A. Gandhi, and G. Neubig, "Training versatile coding agents in synthetic environments," *arXiv preprint arXiv:2512.12216*, 2025.

[177] M. Sullivan, M. Hartmann, and A. Koller, "Procedural environment generation for tool-use agents," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, Eds. Association for Computational Linguistics, 2025, pp. 18 544–18 562. [Online]. Available: https://doi.org/10.18653/v1/2025.emnlp-main.936

[178] X. Chi, C. Fan, H. Zhang, X. Qi, R. Zhang, A. Chen, C. Chan, W. Xue, Q. Liu, S. Zhang, and Y. Guo, "Empowering world models with reflection for embodied video prediction," in *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, ser. Proceedings of Machine Learning Research, A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, Eds., vol. 267. PMLR / OpenReview.net, 2025. [Online]. Available: https://proceedings.mlr.press/v267/chi25b.html

[179] D. Luo, B. Tang, K. Li, G. Papoudakis, J. Song, S. Gong, J. Hao, J. Wang, and K. Shao, "Vimo: A generative visual gui world model for app agents," *arXiv preprint arXiv:2504.13936*, 2025.

[180] E. Alonso, A. Jelley, V. Micheli, A. Kanervisto, A. J. Storkey, T. Pearce, and F. Fleuret, "Diffusion for world modeling: Visual details matter in atari," in *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, Eds., 2024. [Online]. Available: http://papers.nips.cc/paper_files/paper/2024/hash/6bdde0373d53d4a501249547084bed43-Abstract-Conference.html

[181] L. Rivard, S. Sun, H. Guo, W. Chen, and Y. Deng, "Neuralos: Towards simulating operating systems via neural generative models," *CoRR*, vol. abs/2507.08800, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2507.08800

[182] J. Xiang, G. Liu, Y. Gu, Q. Gao, Y. Ning, Y. Zha, Z. Feng, T. Tao, S. Hao, Y. Shi, Z. Liu, E. P. Xing, and Z. Hu, "Pandora: Towards general world model with natural language actions and video states," *CoRR*, vol. abs/2406.09455, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2406.09455

[183] Y. Yang, Z. Wang, Q. Liu, S. Sun, K. Wang, R. Chellappa, Z. Zhou, A. L. Yuille, L. Zhu, Y. Zhang, and J. Chen, "Medical world model: Generative simulation of tumor evolution for treatment planning," *CoRR*, vol. abs/2506.02327, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2506.02327

[184] S. Ye, Y. Ge, K. Zheng, S. Gao, S. Yu, G. Kurian, S. Indupuru, Y. L. Tan, C. Zhu, J. Xiang *et al.*, "World action models are zero-shot policies," *arXiv preprint arXiv:2602.15922*, 2026.

[185] S. Gao, S. Zhou, Y. Du, J. Zhang, and C. Gan, "Adaworld: Learning adaptable world models with latent actions," in *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, ser. Proceedings of Machine Learning Research, A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, Eds., vol. 267. PMLR / OpenReview.net, 2025. [Online]. Available: https://proceedings.mlr.press/v267/gao25u.html

[186] N. Agarwal, A. Ali, M. Bala, Y. Balaji, E. Barker, T. Cai, P. Chattopadhyay, Y. Chen, Y. Cui, Y. Ding, D. Dworakowski, J. Fan, M. Fenzi, F. Ferroni, S. Fidler, D. Fox, S. Ge, Y. Ge, J. Gu, S. Gururani, E. He, J. Huang, J. S. Huffman, P. Jannaty, J. Jin, S. W. Kim, G. Klár, G. Lam, S. Lan, L. Leal-Taixé, A. Li, Z. Li, C. Lin, T. Lin, H. Ling, M. Liu, X. Liu, A. Luo, Q. Ma, H. Mao, K. Mo, A. Mousavian, S. Nah, S. Niverty, D. Page, D. Paschalidou, Z. Patel, L. Pavao, M. Ramezanali, F. Reda, X. Ren, V. R. N. Sabavat, E. Schmerling, S. Shi, B. Stefaniak, S. Tang, L. Tchapmi, P. Tredak, W. Tseng, J. Varghese, H. Wang, H. Wang, H. Wang, T. Wang, F. Wei, X. Wei, J. Z. Wu, J. Xu, W. Yang, Y. Lin, X. Zeng, Y. Zeng, J. Zhang, Q. Zhang, Y. Zhang, Q. Zhao, and A. Zólkowski, "Cosmos world foundation model platform for physical AI," *CoRR*, vol. abs/2501.03575, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2501.03575

[187] X. Ren, Y. Lu, T. Cao, R. Gao, S. Huang, A. Sabour, T. Shen, T. Pfaff, J. Z. Wu, R. Chen, S. W. Kim, J. Gao, L. Leal-Taixé, M. Chen, S. Fidler, and H. Ling, "Cosmos-drive-dreams: Scalable synthetic driving data generation with world foundation models," *CoRR*, vol. abs/2506.09042, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2506.09042

[188] S. Hao, Y. Gu, H. Ma, J. J. Hong, Z. Wang, D. Z. Wang, and Z. Hu, "Reasoning with language model is planning with world model," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, H. Bouamor, J. Pino, and K. Bali, Eds. Association for Computational Linguistics, 2023, pp. 8154–8173. [Online]. Available: https://doi.org/10.18653/v1/2023.emnlp-main.507

[189] S. Qiao, R. Fang, N. Zhang, Y. Zhu, X. Chen, S. Deng, Y. Jiang, P. Xie, F. Huang, and H. Chen, "Agent planning with world knowledge model," in *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, Eds., 2024. [Online]. Available: http://papers.nips.cc/paper_files/paper/2024/hash/d032263772946dd5026e7f3cd22bce5b-Abstract-Conference.html

[190] Y. Zheng, L. Zhong, Y. Wang, R. Dai, K. Liu, X. Chu, L. Lv, P. Torr, and K. Q. Lin, "Code2world: A gui world model via renderable code generation," *arXiv preprint arXiv:2602.09856*, 2026.

[191] F. C. team, J. Copet, Q. Carbonneaux, G. Cohen, J. Gehring, J. Kahn, J. Kossen, F. Kreuk, E. McMilin, M. Meyer, Y. Wei, D. Zhang, K. Zheng, J. Armengol-Estapé, P. Bashiri, M. Beck, P. Chambon, A. Charnalia, C. Cummins, J. Decugis, Z. V. Fisches, F. Fleuret, F. Gloeckle, A. Gu, M. Hassid, D. Haziza, B. Y. Idrissi, C. Keller, R. Kindi, H. Leather, G. Maimon, A. H. Markosyan, F. Massa, P. Mazaré, V. Mella, N. Murray, K. Muzumdar, P. W. O'Hearn, M. Pagliardini, D. Pedchenko, T. Remez, V. Seeker, M. Selvi, O. Sultan, S. Wang, L. Wehrstedt,

O. Yoran, L. Zhang, T. Cohen, Y. Adi, and G. Synnaeve, "CWM: an open-weights LLM for research on code generation with world models," *CoRR*, vol. abs/2510.02387, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.02387

[192] Y. Gu, K. Zhang, Y. Ning, B. Zheng, B. Gou, T. Xue, C. Chang, S. Srivastava, Y. Xie, P. Qi, H. Sun, and Y. Su, "Is your LLM secretly a world model of the internet? model-based planning for web agents," *Trans. Mach. Learn. Res.*, vol. 2025, 2025. [Online]. Available: https://openreview.net/forum?id=c6l7yA0HSq

[193] Y. Li, H. A. Inan, X. Yue, W. Chen, L. Wutschitz, J. Kulkarni, R. Poovendran, R. Sim, and S. Rajmohan, "Simulating environments with reasoning models for agent training," *CoRR*, vol. abs/2511.01824, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2511.01824

[194] Z. Ren, X. Zhang, Z. Qian, Y. Gao, Y. Shi, S. Zheng, and J. He, "GTM: simulating the world of tools for AI agents," *CoRR*, vol. abs/2512.04535, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2512.04535

[195] D. Chen, T. Moutakanni, W. Chung, Y. Bang, Z. Ji, A. Bolourchi, and P. Fung, "Planning with reasoning using vision language world model," *CoRR*, vol. abs/2509.02722, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.02722

[196] X. Yu, B. Peng, R. Xu, M. Galley, H. Cheng, S. Nath, J. Gao, and Z. Yu, "Dyna-think: Synergizing reasoning, acting, and world model simulation in AI agents," *CoRR*, vol. abs/2506.00320, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2506.00320

[197] J. Feng, Y. Zhang, C. Zhang, Y. Lu, S. Liu, and M. Wang, "Web world models," *CoRR*, vol. abs/2512.23676, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2512.23676

[198] S. Sun, H. Song, L. Huang, J. Jiang, R. Le, Z. Lv, Z. Chen, Y. Hu, W. Luo, W. X. Zhao, Y. Song, H. Xu, T. Zhang, and J. Wen, "Swe-world: Building software engineering agents in docker-free environments," *CoRR*, vol. abs/2602.03419, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2602.03419

[199] M. Assran, A. Bardes, D. Fan, Q. Garrido, R. Howes, M. Komeili, M. J. Muckley, A. Rizvi, C. Roberts, K. Sinha, A. Zholus, S. Arnaud, A. Gejji, A. Martin, F. R. Hogan, D. Dugas, P. Bojanowski, V. Khalidov, P. Labatut, F. Massa, M. Szafraniec, K. Krishnakumar, Y. Li, X. Ma, S. Chandar, F. Meier, Y. LeCun, M. Rabbat, and N. Ballas, "V-JEPA 2: Self-supervised video models enable understanding, prediction and planning," *CoRR*, vol. abs/2506.09985, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2506.09985

[200] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, and N. Ballas, "Self-supervised learning from images with a joint-embedding predictive architecture," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 15 619–15 629.

[201] H. Ghaemi, E. Muller, and S. Bakhtiari, "Seq-jepa: Autoregressive predictive learning of invariant-equivariant world models," *arXiv preprint arXiv:2505.03176*, 2025.

[202] F. Baldassarre, M. Szafraniec, B. Terver, V. Khalidov, F. Massa, Y. LeCun, P. Labatut, M. Seitzer, and P. Bojanowski, "Back to the features: Dino as a foundation for video world models," *arXiv preprint arXiv:2507.19468*, 2025.

[203] E. Karypidis, I. Kakogeorgiou, S. Gidaris, and N. Komodakis, "Dino-foresight: Looking into the future with dino," *arXiv preprint arXiv:2412.11673*, 2024.

[204] B. Lyu, X. Cong, H. Yu, P. Yang, C. Qian, Z. Wang, Y. Qin, Y. Ye, Y. Lu, C. Qian, Z. Zhang, Y. Yan, Y. Lin, Z. Liu, and M. Sun, "Enhancing open-domain task-solving capability of llms via autonomous tool integration from github," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds. Association for Computational Linguistics, 2025, pp. 17 257–17 277. [Online]. Available: https://aclanthology.org/2025.acl-long.845/

[205] C. Yang, C. Zhao, Q. Gu, and D. Zhou, "Cops: Empowering LLM agents with provable cross-task experience sharing," *CoRR*, vol. abs/2410.16670, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2410.16670

[206] R. Mon-Williams, G. Li, R. Long, W. Du, and C. G. Lucas, "Embodied large language models enable robots to complete complex tasks in unpredictable environments," *Nat. Mac. Intell.*, vol. 7, no. 4, pp. 592–601, 2025. [Online]. Available: https://doi.org/10.1038/s42256-025-01005-x

[207] L. Zheng, R. Wang, X. Wang, and B. An, "Synapse: Trajectory-as-exemplar prompting with memory for computer control," in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. [Online]. Available: https://openreview.net/forum?id=Pc8AU1aF5e

[208] W. Yeo, K. Kim, J. Yoon, and S. J. Hwang, "Worldmm: Dynamic multimodal memory agent for long video reasoning," *CoRR*, vol. abs/2512.02425, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2512.02425

[209] S. Ouyang, J. Yan, I. Hsu, Y. Chen, K. Jiang, Z. Wang, R. Han, L. T. Le, S. Daruki, X. Tang, V. Tirumalashetty, G. Lee, M. Rofouei, H. Lin, J. Han, C. Lee, and T. Pfister, "Reasoningbank: Scaling agent self-evolving with reasoning memory," *CoRR*, vol. abs/2509.25140, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.25140

[210] X. Tang, T. Qin, T. Peng, Z. Zhou, D. Shao, T. Du, X. Wei, P. Xia, F. Wu, H. Zhu, G. Zhang, J. Liu, X. Wang, S. Hong, C. Wu, H. Cheng, C. Wang, and W. Zhou, "Agent KB: leveraging cross-domain experience for agentic problem solving," *CoRR*, vol. abs/2507.06229, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2507.06229

[211] X. Li, W. Jiao, J. Jin, G. Dong, J. Jin, Y. Wang, H. Wang, Y. Zhu, J. Wen, Y. Lu, and Z. Dou, "Deepagent: A general reasoning agent with scalable toolsets," *CoRR*, vol. abs/2510.21618, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.21618

[212] Z. Cai, X. Guo, Y. Pei, J. Feng, J. Chen, Y. Zhang, W. Ma, M. Wang, and H. Zhou, "FLEX: continuous agent evolution via forward learning from experience," *CoRR*, vol. abs/2511.06449, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2511.06449

[213] P. Wang, M. Tian, J. Li, Y. Liang, Y. Wang, Q. Chen, T. Wang, Z. Lu, J. Ma, Y. E. Jiang, and W. Zhou, "O-mem: Omni memory system for personalized, long horizon, self-evolving agents," 2025. [Online]. Available: https://arxiv.org/abs/2511.13593

[214] Z. Cao, J. Deng, L. Yu, W. Zhou, Z. Liu, B. Ding, and H. Zhao, "Remember me, refine me: A dynamic procedural memory framework for experience-driven agent evolution," *CoRR*, vol. abs/2512.10696, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2512.10696

[215] Z. Z. Wang, J. Mao, D. Fried, and G. Neubig, "Agent workflow memory," *CoRR*, vol. abs/2409.07429, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2409.07429

[216] H. Li, Y. You, H. Su, and L. Guibas, "Learning physical principles from interaction: Self-evolving planning via test-time memory," *arXiv preprint arXiv:2602.20323*, 2026.

[217] W. Zhang, K. Tang, H. Wu, M. Wang, Y. Shen, G. Hou, Z. Tan, P. Li, Y. Zhuang, and W. Lu, "Agent-pro: Learning to evolve via policy-level reflection and optimization," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, L. Ku, A. Martins, and V. Srikumar, Eds. Association for Computational Linguistics, 2024, pp. 5348–5375. [Online]. Available: https://doi.org/10.18653/v1/2024.acl-long.292

[218] Q. M. Tran, Z. Huang, W. Zhang, B. Han, K. Yatani, M. Sugiyama, and T. Liu, "Bifrost: Steering strategic trajectories to bridge contextual gaps for self-improving agents," 2026. [Online]. Available: https://arxiv.org/abs/2602.05810

[219] J. Wang, Q. Yan, Y. Wang, Y. Tian, S. S. Mishra, Z. Xu, M. Gandhi, P. Xu, and L. L. Cheong, "Reinforcement learning for self-improving agent with skill library," *CoRR*, vol. abs/2512.17102, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2512.17102

[220] Z. Z. Wang, A. Gandhi, G. Neubig, and D. Fried, "Inducing programmatic skills for agentic tasks," *CoRR*, vol. abs/2504.06821, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2504.06821

[221] B. Zheng, M. Y. Fatemi, X. Jin, Z. Z. Wang, A. Gandhi, Y. Song, Y. Gu, J. Srinivasa, G. Liu, G. Neubig, and Y. Su, "Skillweaver: Web agents can self-improve by discovering and honing skills," *CoRR*, vol. abs/2504.07079, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2504.07079

[222] P. Xia, J. Chen, H. Wang, J. Liu, K. Zeng, Y. Wang, S. Han, Y. Zhou, X. Zhao, H. Chen *et al.*, "Skillrl: Evolving agents via recursive skill-augmented reinforcement learning," *arXiv preprint arXiv:2602.08234*, 2026.

[223] J. Wang, Y. Ming, Z. Ke, S. Joty, A. Albarghouthi, and F. Sala,

"SkillOrchestra: Learning to route agents via skill transfer," *arXiv preprint arXiv:2602.19672*, 2026.

[224] S. Hong, M. Zhuge, J. Chen, X. Zheng, Y. Cheng, J. Wang, C. Zhang, Z. Wang, S. K. S. Yau, Z. Lin, L. Zhou, C. Ran, L. Xiao, C. Wu, and J. Schmidhuber, "Metagpt: Meta programming for A multi-agent collaborative framework," in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. [Online]. Available: https://openreview.net/forum?id=VtmBAGCN7o

[225] C. S. Xia, Y. Deng, S. Dunn, and L. Zhang, "Agentless: Demystifying llm-based software engineering agents," *CoRR*, vol. abs/2407.01489, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2407.01489

[226] G. Li, H. Hammoud, H. Itani, D. Khizbullin, and B. Ghanem, "CAMEL: communicative agents for "mind" exploration of large language model society," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/a3621ee907def47c1b952ade25c67698-Abstract-Conference.html

[227] H. He, W. Yao, K. Ma, W. Yu, Y. Dai, H. Zhang, Z. Lan, and D. Yu, "Webvoyager: Building an end-to-end web agent with large multimodal models," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, L. Ku, A. Martins, and V. Srikumar, Eds. Association for Computational Linguistics, 2024, pp. 6864–6890. [Online]. Available: https://doi.org/10.18653/v1/2024.acl-long.371

[228] J. Li, Y. Lai, W. Li, J. Ren, M. Zhang, X. Kang, S. Wang, P. Li, Y.-Q. Zhang, W. Ma *et al.*, "Agent hospital: A simulacrum of hospital with evolvable medical agents," *arXiv preprint arXiv:2405.02957*, 2024.

[229] X. Wang, S. Rosenberg, J. Michelini, C. Smith, H. Tran, E. Nyst, R. Malhotra, X. Zhou, V. Chen, R. Brennan *et al.*, "The openhands software agent sdk: A composable and extensible foundation for production agents," *arXiv preprint arXiv:2511.03690*, 2025.

[230] J. Yang, C. E. Jimenez, A. Wettig, K. Lieret, S. Yao, K. Narasimhan, and O. Press, "Swe-agent: Agent-computer interfaces enable automated software engineering," *Advances in Neural Information Processing Systems*, vol. 37, pp. 50 528–50 652, 2024.

[231] G. Zhang, L. Niu, J. Fang, K. Wang, L. Bai, and X. Wang, "Multi-agent architecture search via agentic supernet," in *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, ser. Proceedings of Machine Learning Research, A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, Eds. PMLR / OpenReview.net, 2025. [Online]. Available: https://proceedings.mlr.press/v267/zhang25bi.html

[232] Z. Li, S. Xu, K. Mei, W. Hua, B. Rama, O. Raheja, H. Wang, H. Zhu, and Y. Zhang, "Autoflow: Automated workflow generation for large language model agents," *CoRR*, vol. abs/2407.12821, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2407.12821

[233] M. Hu, Y. Zhou, W. Fan, Y. Nie, B. Xia, T. Sun, Z. Ye, Z. Jin, Y. Li, Q. Chen, Z. Zhang, Y. Wang, Q. Ye, B. Ghanem, P. Luo, and G. Li, "OWL: optimized workforce learning for general multi-agent assistance in real-world task automation," *CoRR*, vol. abs/2505.23885, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.23885

[234] A. Antoniades, A. ¨Orwall, K. Zhang, Y. Xie, A. Goyal, and W. Wang, "Swe-search: Enhancing software agents with monte carlo tree search and iterative refinement," *arXiv preprint arXiv:2410.20285*, 2024.

[235] L. E. Erdogan, N. Lee, S. Kim, S. Moon, H. Furuta, G. Anumanchipalli, K. Keutzer, and A. Gholami, "Plan-and-act: Improving planning of agents for long-horizon tasks," in *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, ser. Proceedings of Machine Learning Research, A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, Eds., vol. 267. PMLR / OpenReview.net, 2025. [Online]. Available: https://proceedings.mlr.press/v267/erdogan25a.html

[236] H. N. Phan, T. N. Nguyen, P. X. Nguyen, and N. D. Bui, "Hyperagent: Generalist software engineering agents to solve coding tasks at scale," *arXiv preprint arXiv:2409.16299*, 2024.

[237] Y. Zhang, Z. Ma, Y. Ma, Z. Han, Y. Wu, and V. Tresp, "Webpilot: A versatile and autonomous multi-agent system for web task execution with strategic exploration," in *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, T. Walsh, J. Shah, and Z. Kolter, Eds. AAAI Press, 2025, pp. 23 378–23 386. [Online]. Available: https://doi.org/10.1609/aaai.v39i22.34505

[238] I. Bouzenia, P. Devanbu, and M. Pradel, "Repairagent: An autonomous, llm-based agent for program repair," in *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE, 2025, pp. 2188–2200.

[239] J. Zhang, J. Xiang, Z. Yu, F. Teng, X. Chen, J. Chen, M. Zhuge, X. Cheng, S. Hong, J. Wang, B. Zheng, B. Liu, Y. Luo, and C. Wu, "Aflow: Automating agentic workflow generation," in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. [Online]. Available: https://openreview.net/forum?id=z5uVAKwmjf

[240] S. Hu, C. Lu, and J. Clune, "Automated design of agentic systems," in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. [Online]. Available: https://openreview.net/forum?id=t9U3LW7JVX

[241] W. Zhang, X. Li, K. Dong, Y. Wang, P. Jia, X. Li, Y. Zhang, D. Xu, Z. Du, H. Guo, R. Tang, and X. Zhao, "Process vs. outcome reward: Which is better for agentic RAG reinforcement learning," *CoRR*, vol. abs/2505.14069, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.14069

[242] M. Yüksekgönül, F. Bianchi, J. Boen, S. Liu, Z. Huang, C. Guestrin, and J. Zou, "Textgrad: Automatic "differentiation" via text," *CoRR*, vol. abs/2406.07496, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2406.07496

[243] M. Zhuge, W. Wang, L. Kirsch, F. Faccio, D. Khizbullin, and J. Schmidhuber, "Gptswarm: Language agents as optimizable graphs," in *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, ser. Proceedings of Machine Learning Research, R. Salakhutdinov, Z. Kolter, K. A. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, Eds. PMLR / OpenReview.net, 2024, pp. 62 743–62 767. [Online]. Available: https://proceedings.mlr.press/v235/zhuge24a.html

[244] O. Khattab, A. Singhvi, P. Maheshwari, Z. Zhang, K. Santhanam, S. Vardhamanan, S. Haq, A. Sharma, T. T. Joshi, H. Moazam, H. Miller, M. Zaharia, and C. Potts, "Dspy: Compiling declarative language model calls into state-of-the-art pipelines," in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. [Online]. Available: https://openreview.net/forum?id=sY5N0zY5Od

[245] W. Li, J. Lin, Z. Jiang, J. Cao, X. Liu, J. Zhang, Z. Huang, Q. Chen, W. Sun, Q. Wang, H. Lu, T. Qin, C. Zhu, Y. Yao, S. Fan, X. Li, T. Wang, P. Liu, K. Zhu, H. Zhu, D. Shi, P. Wang, Y. Guan, X. Tang, M. Liu, Y. E. Jiang, J. Yang, J. Liu, G. Zhang, and W. Zhou, "Chain-of-agents: End-to-end agent foundation models via multi-agent distillation and agentic RL," *CoRR*, vol. abs/2508.13167, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.13167

[246] X. Guo, K. Huang, J. Liu, W. Fan, N. Vélez, Q. Wu, H. Wang, T. L. Griffiths, and M. Wang, "Embodied LLM agents learn to cooperate in organized teams," *CoRR*, vol. abs/2403.12482, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2403.12482

[247] S. Murty, C. D. Manning, P. Shaw, M. Joshi, and K. Lee, "BAGEL: bootstrapping agents by guiding exploration with language," in *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, ser. Proceedings of Machine Learning Research, R. Salakhutdinov, Z. Kolter, K. A. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, Eds., vol. 235. PMLR / OpenReview.net, 2024, pp. 36 894–36 910. [Online]. Available: https://proceedings.mlr.press/v235/murty24a.html

[248] Q. Sun, K. Cheng, Z. Ding, C. Jin, Y. Wang, F. Xu, Z. Wu, C. Jia, L. Chen, Z. Liu, B. Kao, G. Li, J. He, Y. Qiao, and Z. Wu, "Os-genesis: Automating GUI agent trajectory construction via reverse task synthesis," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds.

Association for Computational Linguistics, 2025, pp. 5555–5579. [Online]. Available: https://aclanthology.org/2025.acl-long.277/

[249] B. Trabucco, G. Sigurdsson, R. Piramuthu, and R. Salakhutdinov, "Insta: Towards internet-scale training for agents," *arXiv preprint arXiv:2502.06776*, 2025.

[250] A. Prabhakar, Z. Liu, M. Zhu, J. Zhang, T. Awalgaonkar, S. Wang, Z. Liu, H. Chen, T. Hoang, J. C. Niebles, S. Heinecke, W. Yao, H. Wang, S. Savarese, and C. Xiong, "Apigen-mt: Agentic pipeline for multi-turn data generation via simulated agent-human interplay," *CoRR*, vol. abs/2504.03601, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2504.03601

[251] Z. Tao, J. Wu, W. Yin, J. Zhang, B. Li, H. Shen, K. Li, L. Zhang, X. Wang, Y. Jiang, P. Xie, F. Huang, and J. Zhou, "Webshaper: Agentically data synthesizing via information-seeking formalization," *CoRR*, vol. abs/2507.15061, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2507.15061

[252] X. Geng, P. Xia, Z. Zhang, X. Wang, Q. Wang, R. Ding, C. Wang, J. Wu, Y. Zhao, K. Li, Y. Jiang, P. Xie, F. Huang, and J. Zhou, "Webwatcher: Breaking new frontier of vision-language deep research agent," *CoRR*, vol. abs/2508.05748, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.05748

[253] J. Liu, Y. Li, C. Zhang, J. Li, A. Chen, K. Ji, W. Cheng, Z. Wu, C. Du, Q. Xu, J. Song, Z. Zhu, W. Chen, P. Zhao, and J. He, "Webexplorer: Explore and evolve for training long-horizon web agents," *CoRR*, vol. abs/2509.06501, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.06501

[254] R. Ramrakhya, A. Szot, O. Attia, Y. Yang, A. Nguyen, B. Mazoure, Z. Gan, H. Agrawal, and A. Toshev, "Scaling synthetic task generation for agents via exploration," *CoRR*, vol. abs/2509.25047, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.25047

[255] Y. Lai, Y. Yang, J. Wu, F. Mo, Z. Wang, T. Liang, J. Lin, and K. Yang, "Crmweaver: Building powerful business agent via agentic RL and shared memories," *CoRR*, vol. abs/2510.25333, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.25333

[256] Z. Tao, H. Shen, B. Li, W. Yin, J. Wu, K. Li, Z. Zhang, H. Yin, R. Ye, L. Zhang, X. Wang, P. Xie, J. Zhou, and Y. Jiang, "Webleaper: Empowering efficiency and efficacy in webagent via enabling info-rich seeking," *CoRR*, vol. abs/2510.24697, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.24697

[257] Q. Tang, Z. Deng, H. Lin, X. Han, Q. Liang, and L. Sun, "Toolalpaca: Generalized tool learning for language models with 3000 simulated cases," *CoRR*, vol. abs/2306.05301, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2306.05301

[258] Y. Ma, R. Cao, Y. Cao, Y. Zhang, J. Chen, Y. Liu, Y. Liu, B. Li, F. Huang, and Y. Li, "Lingma SWE-GPT: an open development-process-centric language model for automated software improvement," *CoRR*, vol. abs/2411.00622, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2411.00622

[259] Y. Xu, Z. Wang, J. Wang, D. Lu, T. Xie, A. Saha, D. Sahoo, T. Yu, and C. Xiong, "Aguvis: Unified pure vision agents for autonomous GUI interaction," in *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, ser. Proceedings of Machine Learning Research, A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, Eds., vol. 267. PMLR / OpenReview.net, 2025. [Online]. Available: https://proceedings.mlr.press/v267/xu25ae.html

[260] H. Gao, Y. Liu, Y. He, L. Dou, C. Du, Z. Deng, B. Hooi, M. Lin, and T. Pang, "Flowreasoner: Reinforcing query-level meta-agents," 2025. [Online]. Available: https://arxiv.org/abs/2504.15257

[261] Y. Gao, J. Ye, J. Wang, and J. Sang, "Websynthesis: World-model-guided MCTS for efficient webui-trajectory synthesis," *CoRR*, vol. abs/2507.04370, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2507.04370

[262] R. Ye, Z. Zhang, K. Li, H. Yin, Z. Tao, Y. Zhao, L. Su, L. Zhang, Z. Qiao, X. Wang, P. Xie, F. Huang, S. Chen, J. Zhou, and Y. Jiang, "Agentfold: Long-horizon web agents with proactive context management," *CoRR*, vol. abs/2510.24699, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.24699

[263] Z. Yao, Z. Xu, Y. Guo, Z. Han, C. Yang, S. Zhang, W. Zhang, X. Zeng, and W. Liu, "Toolace-mcp: Generalizing history-aware routing from MCP tools to the agent web," *CoRR*, vol. abs/2601.08276, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.08276

[264] Y. Yu, M. Yang, J. Li, Y. Gao, F. Liu, Y. Yang, Z. Lin, J. Lyu, Y. Liu, Z. Lu, D. Ye, and J. Jiang, "Proact: Agentic lookahead in interactive environments," 2026. [Online]. Available: https://arxiv.org/abs/2602.05327

[265] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, E. Hambro, L. Zettlemoyer, N. Cancedda, and T. Scialom, "Toolformer: Language models can teach themselves to use tools," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html

[266] Y. Song, D. Yin, X. Yue, J. Huang, S. Li, and B. Y. Lin, "Trial and error: Exploration-based trajectory optimization for LLM agents," *CoRR*, vol. abs/2403.02502, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2403.02502

[267] A. Patel, M. Hofmarcher, C. Leoveanu-Condrei, M.-C. Dinu, C. Callison-Burch, and S. Hochreiter, "Large language models can self-improve at web agent tasks," *arXiv preprint arXiv:2405.20309*, 2024.

[268] P. Wu, S. Ma, B. Wang, J. Yu, L. Lu, and Z. Liu, "Gui-reflection: Empowering multimodal GUI models with self-reflection behavior," *CoRR*, vol. abs/2506.08012, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2506.08012

[269] Y. Liao, Y. Gu, Y. Sui, Z. Zhu, Y. Lu, G. Tang, Z. Sun, and W. Yang, "Think in games: Learning to reason in games via reinforcement learning with large language models," *CoRR*, vol. abs/2508.21365, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.21365

[270] X. Chen, Z. Qiao, G. Chen, L. Su, Z. Zhang, X. Wang, P. Xie, F. Huang, J. Zhou, and Y. Jiang, "Agentfrontier: Expanding the capability frontier of LLM agents with zpd-guided data synthesis," *CoRR*, vol. abs/2510.24695, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.24695

[271] Y. He, P. Chawla, Y. Souri, S. Som, and X. Song, "Scalable data synthesis for computer use agents with step-level filtering," *arXiv preprint arXiv:2512.10962*, 2025.

[272] Z. Wang, Y. Liang, X. Zhang, Q. Wu, S. Han, A. Bastos, R. Wang, C. Bansal, B. Peng, J. Gao, S. Rajmohan, and H. Yao, "Adapting web agents with synthetic supervision," *CoRR*, vol. abs/2511.06101, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2511.06101

[273] J. Yang, Y. Liu, Z. Chen, C. Han, Y. Sun, Q. Gu, H. Su, X. Cai, F. Wu, and K. Kuang, "Topocurate:modeling interaction topology for tool-use agent training," *CoRR*, vol. abs/2603.01714, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2603.01714

[274] P. Jiang, J. Lin, L. Cao, R. Tian, S. Kang, Z. Wang, J. Sun, and J. Han, "Deepretrieval: Hacking real search engines and retrievers with large language models via reinforcement learning," *CoRR*, vol. abs/2503.00223, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2503.00223

[275] B. Jin, H. Zeng, Z. Yue, D. Wang, H. Zamani, and J. Han, "Search-r1: Training llms to reason and leverage search engines with reinforcement learning," *CoRR*, vol. abs/2503.09516, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2503.09516

[276] Y. Shi, S. Li, C. Wu, Z. Liu, J. Fang, H. Cai, A. Zhang, and X. Wang, "Search and refine during think: Autonomous retrieval-augmented reasoning of llms," *CoRR*, vol. abs/2505.11277, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.11277

[277] W. Tian, S. Zhang, K. Zhang, X. Chi, Y. Luo, J. Lu, C. Fan, Q. Zhou, Y. Zhao, N. L. S. Lin, Z. Qin, X. Ju, S. Zhang, and J. Tang, "SEEA-R1: tree-structured reinforcement fine-tuning for self-evolving embodied agents," *CoRR*, vol. abs/2506.21669, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2506.21669

[278] L. Long, Y. He, W. Ye, Y. Pan, Y. Lin, H. Li, J. Zhao, and W. Li, "Seeing, listening, remembering, and reasoning: A multimodal agent with long-term memory," *CoRR*, vol. abs/2508.09736, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.09736

[279] S. Wang, J. Jin, X. Wang, L. Song, R. Fu, H. Wang, Z. Ge, Y. Lu, and X. Cheng, "Video-thinker: Sparking "thinking with videos" via reinforcement learning," *CoRR*, vol. abs/2510.23473, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.23473

[280] M. Chen, T. Li, H. Sun, Y. Zhou, C. Zhu, H. Wang, J. Z. Pan, W. Zhang, H. Chen, F. Yang, Z. Zhou, and W. Chen, "Research: Learning to reason with search for llms via reinforcement learning," *CoRR*, vol. abs/2503.19470, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2503.19470

[281] M. Cheng, J. Ouyang, S. Yu, R. Yan, Y. Luo, Z. Liu, D. Wang, Q. Liu, and E. Chen, "Agent-r1: Training powerful LLM agents with end-to-end reinforcement learning," *CoRR*, vol. abs/2511.14460, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2511.14460

[282] Q. Wang, R. Ding, Y. Zeng, Z. Chen, L. Chen, S. Wang, P. Xie, F. Huang, and F. Zhao, "VRAG-RL: empower vision-perception-based RAG for visually rich information understanding via iterative reasoning with reinforcement learning," *CoRR*, vol. abs/2505.22019, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.22019

[283] S. Liu, X. Dong, X. Lu, S. Diao, P. Belcak, M. Liu, M. Chen, H. Yin, Y. F. Wang, K. Cheng, Y. Choi, J. Kautz, and P. Molchanov, "GDPO: group reward-decoupled normalization policy optimization for multi-reward RL optimization," *CoRR*, vol. abs/2601.05242, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.05242

[284] M. Zhang, Y. Li, Y. Zhang, Y. Cai, Y. Dong, J. Tang, and N. V. Chawla, "Flowsteer: Interactive agentic workflow orchestration via end-to-end reinforcement learning," *arXiv preprint arXiv:2602.01664*, 2026. [Online]. Available: https://arxiv.org/abs/2602.01664

[285] S. Zhang, Y. Dong, J. Zhang, J. Kautz, B. Catanzaro, A. Tao, Q. Wu, Z. Yu, and G. Liu, "Nemotron-research-tool-n1: Exploring tool-using language models with reinforced reasoning," *CoRR*, vol. abs/2505.00024, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.00024

[286] H. Su, S. Diao, X. Lu, M. Liu, J. Xu, X. Dong, Y. Fu, P. Belcak, H. Ye, H. Yin, Y. Dong, E. Bakhturina, T. Yu, Y. Choi, J. Kautz, and P. Molchanov, "Toolorchestra: Elevating intelligence via efficient model and tool orchestration," *arXiv preprint arXiv:2511.21689*, 2025.

[287] Z. Wang, K. Wang, Q. Wang, P. Zhang, L. Li, Z. Yang, X. Jin, K. Yu, M. N. Nguyen, L. Liu, E. Gottlieb, Y. Lu, K. Cho, J. Wu, L. Fei-Fei, L. Wang, Y. Choi, and M. Li, "RAGEN: understanding self-evolution in LLM agents via multi-turn reinforcement learning," *CoRR*, vol. abs/2504.20073, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2504.20073

[288] H. Sun, Z. Qiao, J. Guo, X. Fan, Y. Hou, Y. Jiang, P. Xie, Y. Zhang, F. Huang, and J. Zhou, "Zerosearch: Incentivize the search capability of llms without searching," *CoRR*, vol. abs/2505.04588, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.04588

[289] D. Zhang, Y. Zhao, J. Wu, L. Zhang, B. Li, W. Yin, Y. Jiang, Y. Li, K. Tu, P. Xie, and F. Huang, "Evolvesearch: An iterative self-evolving search agent," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, Eds. Association for Computational Linguistics, 2025, pp. 13 123–13 136. [Online]. Available: https://doi.org/10.18653/v1/2025.emnlp-main.663

[290] L. Feng, Z. Xue, T. Liu, and B. An, "Group-in-group policy optimization for LLM agent training," *CoRR*, vol. abs/2505.10978, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.10978

[291] G. Dong, H. Mao, K. Ma, L. Bao, Y. Chen, Z. Wang, Z. Chen, J. Du, H. Wang, F. Zhang, G. Zhou, Y. Zhu, J. Wen, and Z. Dou, "Agentic reinforced policy optimization," *CoRR*, vol. abs/2507.19849, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2507.19849

[292] Y. Shi, W. Yu, Z. Li, Y. Wang, H. Zhang, N. Liu, H. Mi, and D. Yu, "Mobilegui-rl: Advancing mobile gui agent through reinforcement learning in online environment," *arXiv preprint arXiv:2507.05720*, 2025.

[293] Y. Qin, X. Tan, Z. He, G. Li, H. Lin, Z. Li, Z. Xu, Y. Shi, S. Cai, R. Rui, S. Cai, Y. Cai, X. Zhang, S. Ye, K. Li, and X. Sun, "Learn the ropes, then trust the wins: Self-imitation with progressive exploration for agentic reinforcement learning," *CoRR*, vol. abs/2509.22601, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.22601

[294] K. Wang, P. Zhang, Z. Wang, Y. Gao, L. Li, Q. Wang, H. Chen, C. Wan, Y. Lu, Z. Yang, L. Wang, R. Krishna, J. Wu, L. Fei-Fei, Y. Choi, and M. Li, "VAGEN: reinforcing world model reasoning for multi-turn VLM agents," *CoRR*, vol. abs/2510.16907, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.16907

[295] T. Hu, Q. Fu, Y. Chen, Z. Liu, and B. Ding, "Seeupo: Sequence-level agentic-rl with convergence guarantees," *arXiv preprint arXiv:2602.06554*, 2026.

[296] A. Zhao, Y. Wu, Y. Yue, T. Wu, Q. Xu, M. Lin, S. Wang, Q. Wu, Z. Zheng, and G. Huang, "Absolute zero: Reinforced self-play reasoning with zero data," *arXiv preprint arXiv:2505.03335*, 2025.

[297] Y. Zhou, S. Levine, J. Weston, X. Li, and S. Sukhbaatar, "Self-challenging language model agents," *CoRR*, vol. abs/2506.01716, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2506.01716

[298] J. He, J. Fang, F. Xiong, Z. Yao, F. Shen, H. Guo, J. Wang, and T. Chua, "Active zero: Self-evolving vision-language models through active environment exploration," *CoRR*, vol. abs/2602.11241, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2602.11241

[299] Q. Wang, B. Liu, T. Zhou, J. Shi, Y. Lin, Y. Chen, H. H. Li, K. Wan, and W. Zhao, "Vision-zero: Scalable vlm self-improvement via strategic gamified self-play," *arXiv preprint arXiv:2509.25541*, 2025.

[300] Y. Wang, D. Yin, Y. Cui, R. Zheng, Z. Li, Z. Lin, D. Wu, X. Wu, C. Ye, Y. Zhou *et al.*, "Llms as scalable, general-purpose simulators for evolving digital agent training," *arXiv preprint arXiv:2510.14969*, 2025.

[301] Z. Xiao, J. Tu, C. Zou, Y. Zuo, Z. Li, P. Wang, B. Yu, F. Huang, J. Lin, and Z. Liu, "Webworld: A large-scale world model for web agent training," *arXiv preprint arXiv:2602.14721*, 2026.

[302] R. Wang, J. Lehman, J. Clune, and K. O. Stanley, "Paired open-ended trailblazer (POET): endlessly generating increasingly complex and diverse learning environments and their solutions," *CoRR*, vol. abs/1901.01753, 2019. [Online]. Available: http://arxiv.org/abs/1901.01753

[303] M. Hu, P. Zhao, C. Xu, Q. Sun, J. Lou, Q. Lin, P. Luo, S. Rajmohan, and D. Zhang, "Agentgen: Enhancing planning abilities for large language model based agent via environment and task generation," *CoRR*, vol. abs/2408.00764, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2408.00764

[304] S. Lu, Z. Wang, H. Zhang, Q. Wu, L. Gan, C. Zhuang, J. Gu, and T. Lin, "Don't just fine-tune the agent, tune the environment," *CoRR*, vol. abs/2510.10197, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.10197

[305] X. Chen, J. Lu, M. Kim, D. Zhang, J. Tang, A. Piché, N. Gontier, Y. Bengio, and E. Kamalloo, "Self-evolving curriculum for LLM reasoning," *CoRR*, vol. abs/2505.14970, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.14970

[306] V. Lacombe, V. Quesnel, and D. Sileo, "Reasoning core: A scalable RL environment for LLM symbolic reasoning," *CoRR*, vol. abs/2509.18083, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.18083

[307] Z. Chen, Z. Zhao, K. Zhang, B. Liu, Q. Qi, Y. Wu, T. Kalluri, S. Cao, Y. Xiong, H. Tong, H. Yao, H. Li, J. Zhu, X. Li, D. Song, B. Li, J. Weston, and D. Huynh, "Scaling agent learning via experience synthesis," *CoRR*, vol. abs/2511.03773, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2511.03773

[308] M. Dennis, N. Jaques, E. Vinitsky, A. M. Bayen, S. Russell, A. Critch, and S. Levine, "Emergent complexity and zero-shot transfer via unsupervised environment design," *CoRR*, vol. abs/2012.02096, 2020. [Online]. Available: https://arxiv.org/abs/2012.02096

[309] M. Jiang, M. Dennis, J. Parker-Holder, J. N. Foerster, E. Grefenstette, and T. Rocktäschel, "Replay-guided adversarial environment design," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021, pp. 1884–1897. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/0e915db6326b6fb6a3c56546980a8c93-Abstract.html

[310] J. Parker-Holder, M. Jiang, M. Dennis, M. Samvelyan, J. N. Foerster, E. Grefenstette, and T. Rocktäschel, "Evolving curricula with regret-based environment design," in *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, Eds. PMLR, 2022, pp. 17 473–17 498. [Online]. Available: https://proceedings.mlr.press/v162/parker-holder22a.html

[311] M. Samvelyan, A. Khan, M. Dennis, M. Jiang, J. Parker-Holder, J. N. Foerster, R. Raileanu, and T. Rocktäschel, "MAESTRO: open-ended environment design for multi-agent reinforcement learning," in *The Eleventh International*

Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net, 2023. [Online]. Available: https://openreview.net/forum?id=sKWlRDzPfd7

[312] M. Beukman, S. Coward, M. T. Matthews, M. Fellows, M. Jiang, M. D. Dennis, and J. N. Foerster, "Refining minimax regret for unsupervised environment design," in *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, ser. Proceedings of Machine Learning Research, R. Salakhutdinov, Z. Kolter, K. A. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, Eds. PMLR / OpenReview.net, 2024, pp. 3637–3657. [Online]. Available: https://proceedings.mlr.press/v235/beukman24a.html

[313] Z. Khan, E. Stengel-Eskin, J. Cho, and M. Bansal, "Dataenvgym: Data generation agents in teacher environments with student feedback," in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. [Online]. Available: https://openreview.net/forum?id=00SnKBGTsz

[314] W. Liang, S. Wang, H.-J. Wang, O. Bastani, D. Jayaraman, and Y. J. Ma, "Eurekaverse: Environment curriculum generation via large language models," *arXiv preprint arXiv:2411.01775*, 2024.

[315] Z. Zeng, H. Ivison, Y. Wang, L. Yuan, S. S. Li, Z. Ye, S. Li, J. He, R. Zhou, T. Chen, C. Zhao, Y. Tsvetkov, S. S. Du, N. Jaques, H. Peng, P. W. Koh, and H. Hajishirzi, "RLVE: scaling up reinforcement learning for language models with adaptive verifiable environments," *CoRR*, vol. abs/2511.07317, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2511.07317

[316] S. Mai, Y. Zhai, Z. Chen, C. Chen, A. Zou, S. Tao, Z. Liu, and B. Ding, "Cues: A curiosity-driven and environment-grounded synthesis framework for agentic RL," *CoRR*, vol. abs/2512.01311, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2512.01311

[317] J. Guo, L. Yang, P. Chen, Q. Xiao, Y. Wang, X. Juan, J. Qiu, K. Shen, and M. Wang, "Genenv: Difficulty-aligned co-evolution between LLM agents and environment simulators," *CoRR*, vol. abs/2512.19682, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2512.19682

[318] C. Xu, C. Xiao, Z. Peng, X. Wang, and Y. Cao, "Scaler:synthetic scalable adaptive learning environment for reasoning," *CoRR*, vol. abs/2601.04809, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.04809

[319] J. Ye, C. Jiang, Z. Du, Y. Xu, X. Yao, Z. Xi, X. Fan, Q. Zhang, X. Huang, and J. Chen, "Feedback-driven tool-use improvements in large language models via automated build environments," *CoRR*, vol. abs/2508.08791, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.08791

[320] P. Andrews, A. Benhalloum, G. M. Bertran, M. Bettini, A. Budhiraja, R. S. Cabral, V. Do, R. Froger, E. Garreau, J. Gaya, H. Laurençon, M. Lecanu, K. Malkan, D. Mekala, P. Ménard, G. Mialon, U. Piterbarg, M. Plekhanov, M. Rita, A. Rusakov, T. Scialom, V. Vorotilov, M. Wang, and I. Yu, "ARE: scaling up agent environments and evaluations," *CoRR*, vol. abs/2509.17158, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.17158

[321] W. Liu, X. Huang, X. Zeng, X. Hao, S. Yu, D. Li, S. Wang, W. Gan, Z. Liu, Y. Yu, Z. Wang, Y. Wang, W. Ning, Y. Hou, B. Wang, C. Wu, X. Wang, Y. Liu, Y. Wang, D. Tang, D. Tu, L. Shang, X. Jiang, R. Tang, D. Lian, Q. Liu, and E. Chen, "Toolace: Winning the points of LLM function calling," in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. [Online]. Available: https://openreview.net/forum?id=8EB8k6DdCU

[322] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017. [Online]. Available: http://arxiv.org/abs/1707.06347

[323] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman, "Training verifiers to solve math word problems," *CoRR*, vol. abs/2110.14168, 2021. [Online]. Available: https://arxiv.org/abs/2110.14168

[324] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, "Measuring massive multitask language understanding," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. [Online]. Available: https://openreview.net/forum?id=d7KBjmI3GmQ

[325] X. Yue, Y. Ni, K. Zhang, T. Zheng, R. Liu, G. Zhang, S. Stevens, D. Jiang, W. Ren, Y. Sun, C. Wei, B. Yu, R. Yuan, R. Sun, M. Yin, B. Zheng, Z. Yang, Y. Liu, W. Huang, H. Sun, Y. Su, and W. Chen, "MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI," *CoRR*, vol. abs/2311.16502, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2311.16502

[326] Z. Jin, H. Yuan, T. Men, P. Cao, Y. Chen, J. Xu, H. Li, X. Jiang, K. Liu, and J. Zhao, "Rag-rewardbench: Benchmarking reward models in retrieval augmented generation for preference alignment," in *Findings of the Association for Computational Linguistics: ACL 2025*, 2025, pp. 17 061–17 090.

[327] S. Mateega, J. Yang, T. Costello, S. Jadhav, N. Tian, and A. Garcinuño, "Ide-bench: Evaluating large language models as ide agents on real-world software engineering tasks," *arXiv preprint arXiv:2601.20886*, 2026.

[328] T. Men, P. Cao, Z. Jin, Y. Chen, K. Liu, and J. Zhao, "Unlocking the future: Exploring look-ahead planning mechanistic interpretability in large language models," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 7713–7724.

[329] C. Yang, X. Yang, L. Wen, D. Fu, J. Mei, R. Wu, P. Cai, Y. Shen, N. Deng, B. Shi, Y. Qiao, and H. Li, "Learning on the job: An experience-driven self-evolving agent for long-horizon tasks," *CoRR*, vol. abs/2510.08002, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.08002

[330] Z. Li, H. Zhang, S. Han, S. Liu, J. Xie, Y. Zhang, Y. Choi, J. Zou, and P. Lu, "In-the-flow agentic system optimization for effective planning and tool use," *CoRR*, vol. abs/2510.05592, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.05592

[331] P. Jiang, J. Lin, Z. Shi, Z. Wang, L. He, Y. Wu, M. Zhong, P. Song, Q. Zhang, H. Wang *et al.*, "Adaptation of agentic ai," *arXiv preprint arXiv:2512.16301*, 2025.

[332] M. Lin, Z. Wu, Z. Xu, H. Liu, X. Tang, Q. He, C. C. Aggarwal, H. Liu, X. Zhang, and S. Wang, "A comprehensive survey on reinforcement learning-based agentic search: Foundations, roles, optimizations, evaluations, and applications," *CoRR*, vol. abs/2510.16724, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.16724

[333] J. Fang, Y. Peng, X. Zhang, Y. Wang, X. Yi, G. Zhang, Y. Xu, B. Wu, S. Liu, Z. Li, Z. Ren, N. Aletras, X. Wang, H. Zhou, and Z. Meng, "A comprehensive survey of self-evolving AI agents: A new paradigm bridging foundation models and lifelong agentic systems," *CoRR*, vol. abs/2508.07407, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.07407

[334] G. Zhang, H. Geng, X. Yu, Z. Yin, Z. Zhang, Z. Tan, H. Zhou, Z. Li, X. Xue, Y. Li, Y. Zhou, Y. Chen, C. Zhang, Y. Fan, Z. Wang, S. Huang, F. P. Velez, Y. Liao, H. Wang, M. Yang, H. Ji, J. Wang, S. Yan, P. Torr, and L. Bai, "The landscape of agentic reinforcement learning for llms: A survey," *Trans. Mach. Learn. Res.*, vol. 2026, 2026. [Online]. Available: https://openreview.net/forum?id=RY19y2RI1O

[335] X. Yang, L. Li, H. Zhou, T. Zhu, X. Qu, Y. Fan, Q. Wei, R. Ye, L. Kang, Y. Qin *et al.*, "Toward efficient agents: Memory, tool learning, and planning," *arXiv preprint arXiv:2601.14192*, 2026.

[336] H. Gao, J. Geng, W. Hua, M. Hu, X. Juan, H. Liu, S. Liu, J. Qiu, X. Qi, Q. Ren, Y. Wu, H. Wang, H. Xiao, Y. Zhou, S. Zhang, J. Zhang, J. Xiang, Y. Fang, Q. Zhao, D. Liu, C. Qian, Z. Wang, M. Hu, H. Wang, Q. Wu, H. Ji, and M. Wang, "A survey of self-evolving agents: What, when, how, and where to evolve on the path to artificial super intelligence," *Trans. Mach. Learn. Res.*, vol. 2026, 2026. [Online]. Available: https://openreview.net/forum?id=CTr3bovS5F

[337] J. Jang, S. Ye, Z. Lin, J. Xiang, J. Bjorck, Y. Fang, F. Hu, S. Huang, K. Kundalia, Y.-C. Lin *et al.*, "Dreamgen: Unlocking generalization in robot learning through video world models," *arXiv preprint arXiv:2505.12705*, 2025.

[338] T. Fang, H. Zhang, Z. Zhang, K. Ma, W. Yu, H. Mi, and D. Yu, "Webevolver: Enhancing web agent self-improvement with co-evolving world model," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, Eds. Association for Computational Linguistics, 2025, pp. 8959–8975. [Online]. Available: https://doi.org/10.18653/v1/2025.emnlp-main.454

[339] Q. Yuan, M. Kazemi, X. Xu, I. Noble, V. Imbrasaite, and D. Ramachandran, "Tasklama: Probing the complex task

understanding of language models," in *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, M. J. Wooldridge, J. G. Dy, and S. Natarajan, Eds. AAAI Press, 2024, pp. 19 468–19 476. [Online]. Available: https://doi.org/10.1609/aaai.v38i17.29918

[340] Y. Shen, K. Song, X. Tan, W. Zhang, K. Ren, S. Yuan, W. Lu, D. Li, and Y. Zhuang, "Taskbench: Benchmarking large language models for task automation," in *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, Eds., 2024. [Online]. Available: http://papers.nips.cc/paper_files/paper/2024/hash/085185ea97db31ae6dcac7497616fd3e-Abstract-Datasets_and_Benchmarks_Track.html

[341] W. Li, W. E. Bishop, A. Li, C. Rawles, F. Campbell-Ajala, D. Tyamagundlu, and O. Riva, "On the effects of data scale on UI control agents," in *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, Eds., 2024. [Online]. Available: http://papers.nips.cc/paper_files/paper/2024/hash/a79f3ef3b445fd4659f44648f7ea8ffd-Abstract-Datasets_and_Benchmarks_Track.html

[342] K. Ma, X. Du, Y. Wang, H. Zhang, Z. Wen, X. Qu, J. Yang, J. Liu, M. Liu, X. Yue, W. Huang, and G. Zhang, "Kor-bench: Benchmarking language models on knowledge-orthogonal reasoning tasks," 2025. [Online]. Available: https://arxiv.org/abs/2410.06526

[343] Y. Li, C. Lin, M. U. Nasir, P. Bontrager, J. Liu, and J. Togelius, "Gvgai-llm: Evaluating large language model agents with infinite games," 2025. [Online]. Available: https://arxiv.org/abs/2508.08501

[344] W. Zhang, J. Yang, R. Tao, L. Chai, S. Guo, J. Wu, X. Chen, G. Cui, N. Ding, X. Xu, H. Wei, and B. Zhou, "V-gamegym: Visual game generation for code large language models," *CoRR*, vol. abs/2509.20136, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.20136

[345] Q. Wu, H. Zhao, M. Saxon, T. Bui, W. Y. Wang, Y. Zhang, and S. Chang, "Vsp: Assessing the dual challenges of perception and reasoning in spatial planning tasks for vlms," *arXiv preprint arXiv:2407.01863*, 2024.

[346] L. Rowe, R. Girgis, A. Gosselin, L. Paull, C. Pal, and F. Heide, "Scenario dreamer: Vectorized latent diffusion for generating driving simulation environments," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*. Computer Vision Foundation / IEEE, 2025, pp. 17 207–17 218. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2025/html/Rowe_Scenario_Dreamer_Vectorized_Latent_Diffusion_for_Generating_Driving_Simulation_Environments_CVPR_2025_paper.html

[347] L. Zheng, Z. Huang, Z. Xue, X. Wang, B. An, and S. Yan, "Agentstudio: A toolkit for building general virtual agents," in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. [Online]. Available: https://openreview.net/forum?id=axUf8BOjnH

[348] J. S. Park, J. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Generative agents: Interactive simulacra of human behavior," in *Proceedings of the 36th annual acm symposium on user interface software and technology*, 2023, pp. 1–22.

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

[350] B. Gou, Z. Huang, Y. Ning, Y. Gu, M. Lin, W. Qi, A. Kopanev, B. Yu, B. J. Gutiérrez, Y. Shu, C. H. Song, J. Wu, S. Chen, H. N. Moussa, T. Zhang, J. Xie, Y. Li, T. Xue, Z. Liao, K. Zhang, B. Zheng, Z. Cai, V. Rozgic, M. Ziyadi, H. Sun, and Y. Su, "Mind2web 2: Evaluating agentic search with agent-as-a-judge," *CoRR*, vol. abs/2506.21506, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2506.21506

[351] Y. Deng, X. Zhang, W. Zhang, Y. Yuan, S. Ng, and T. Chua, "On the multi-turn instruction following for conversational web agents," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, L. Ku, A. Martins, and V. Srikumar, Eds. Association for Computational Linguistics, 2024, pp. 8795–8812. [Online]. Available: https://doi.org/10.18653/v1/2024.acl-long.477

[352] L. K. Jang, Y. Li, D. Zhao, C. Ding, J. Lin, P. P. Liang, R. Bonatti, and K. Koishida, "Videowebarena: Evaluating long context multimodal agents with video understanding web tasks," in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. [Online]. Available: https://openreview.net/forum?id=unDQOUah0F

[353] Y. Pan, D. Kong, S. Zhou, C. Cui, Y. Leng, B. Jiang, H. Liu, Y. Shang, S. Zhou, T. Wu, and Z. Wu, "Webcanvas: Benchmarking web agents in online environments," *CoRR*, vol. abs/2406.12373, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2406.12373

[354] T. Xue, W. Qi, T. Shi, C. H. Song, B. Gou, D. Song, H. Sun, and Y. Su, "An illusion of progress? assessing the current state of web agents," *CoRR*, vol. abs/2504.01382, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2504.01382

[355] L. Wang, Y. Deng, Y. Zha, G. Mao, Q. Wang, T. Min, W. Chen, and S. Chen, "Mobileagentbench: An efficient and user-friendly benchmark for mobile LLM agents," *CoRR*, vol. abs/2406.08184, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2406.08184

[356] T. Xu, P. Lu, L. Ye, X. Hu, and P. Liu, "Researcherbench: Evaluating deep AI research systems on the frontiers of scientific inquiry," *CoRR*, vol. abs/2507.16280, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2507.16280

[357] Y. Qiu, H. Zhang, Z. Xu, M. Li, D. Song, Z. Wang, and K. Zhang, "AI idea bench 2025: AI research idea generation benchmark," *CoRR*, vol. abs/2504.14191, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2504.14191

[358] M. Zhu, Y. Weng, L. Yang, and Y. Zhang, "Deepreview: Improving llm-based paper review with human-like deep thinking process," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds. Association for Computational Linguistics, 2025, pp. 29 330–29 355. [Online]. Available: https://aclanthology.org/2025.acl-long.1420/

[359] G. Starace, O. Jaffe, D. Sherburn, J. Aung, J. S. Chan, L. Maksin, R. Dias, E. Mays, B. Kinsella, W. Thompson, J. Heidecke, A. Glaese, and T. Patwardhan, "Paperbench: Evaluating ai's ability to replicate AI research," in *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, ser. Proceedings of Machine Learning Research, A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, Eds. PMLR / OpenReview.net, 2025. [Online]. Available: https://proceedings.mlr.press/v267/starace25a.html

[360] Y. Zeng, W. Huang, Z. Fang, S. Chen, Y. Shen, Y. Cai, X. Wang, Z. Yin, L. Chen, Z. Chen, S. Huang, Y. Zhao, X. Tang, Y. Hu, P. Torr, W. Ouyang, and S. Cao, "Vision-deepresearch benchmark: Rethinking visual and textual search for multimodal large language models," 2026. [Online]. Available: https://arxiv.org/abs/2602.02185

[361] G. Seo, J. Yang, J. Pyo, N. Kim, J. Lee, and Y. Jo, "Simuhome: A temporal-and environment-aware benchmark for smart home llm agents," *arXiv preprint arXiv:2509.24282*, 2025.

[362] J. D. Gajo, G. P. Merales, J. Escarcha, B. A. Molina, G. Nartea, E. G. Maminta, J. C. Roldán, and R. O. Atienza, "Sari sandbox: A virtual retail store environment for embodied AI agents," *CoRR*, vol. abs/2508.00400, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.00400

[363] Y. Zhou, N. Ye, W. Ljungbergh, T. Li, J. Yang, Z. Yang, H. Zhu, C. Petersson, and H. Li, "Decoupled diffusion sparks adaptive scene generation," *CoRR*, vol. abs/2504.10485, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2504.10485

[364] J. Shi, J. Yang, J. Liu, X. Bu, J. Chen, J. Zhou, K. Ma, Z. Wen, B. Wang, Y. He, L. Song, H. Zhu, S. Li, X. Wang,

W. Zhang, R. Yuan, Y. Yao, W. Yang, Y. Wang, S. Fang, S. Yuan, Q. He, X. Tang, Y. Tan, W. Zhou, Z. Zhang, Z. Li, W. Huang, and G. Zhang, "Korgym: A dynamic game platform for llm reasoning evaluation," 2025. [Online]. Available: https://arxiv.org/abs/2505.14552

[365] H. Zhang, H. Guo, S. Guo, M. Cao, W. Huang, J. Liu, and G. Zhang, "Ing-vp: Mllms cannot play easy vision-based games yet," 2024. [Online]. Available: https://arxiv.org/abs/2410.06555

[366] W. Tang, Y. Zhou, E. Xu, K. Cheng, M. Li, and L. Xiao, "Dsgbench: A diverse strategic game benchmark for evaluating llm-based agents in complex decision-making environments," 2025. [Online]. Available: https://arxiv.org/abs/2503.06047

[367] S. Hu, T. Huang, and L. Liu, "Pokellmon: A human-parity agent for pokemon battles with large language models," 2024. [Online]. Available: https://arxiv.org/abs/2402.01118

[368] J. Yao, K. Wang, R. Hsieh, H. Zhou, T. Zou, Z. Cheng, Z. Wang, and P. Viswanath, "Spin-bench: How well do llms plan strategically and reason socially?" 2025. [Online]. Available: https://arxiv.org/abs/2503.12349

[369] J. Yu, Y. Qin, X. Wang, P. Wan, D. Zhang, and X. Liu, "Gamefactory: Creating new games with generative interactive videos," 2025. [Online]. Available: https://arxiv.org/abs/2501.08325

[370] D. Valevski, Y. Leviathan, M. Arar, and S. Fruchter, "Diffusion models are real-time game engines," 2025. [Online]. Available: https://arxiv.org/abs/2408.14837

[371] A. Padmakumar, J. Thomason, A. Shrivastava, P. Lange, A. Narayan-Chen, S. Gella, R. Piramuthu, G. Tur, and D. Hakkani-Tur, "Teach: Task-driven embodied agents that chat," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 2, 2022, pp. 2017–2025.

[372] T. Kim, C. Min, B. Kim, J. Kim, W. Jeung, and J. Choi, "Realfred: An embodied instruction following benchmark in photo-realistic environments," in *European Conference on Computer Vision*. Springer, 2024, pp. 346–364.

[373] J. Ye, G. Li, S. Gao, C. Huang, Y. Wu, S. Li, X. Fan, S. Dou, T. Ji, Q. Zhang, T. Gui, and X. Huang, "Tooleyes: Fine-grained evaluation for tool learning capabilities of large language models in real-world scenarios," in *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, and S. Schockaert, Eds. Association for Computational Linguistics, 2025, pp. 156–187. [Online]. Available: https://aclanthology.org/2025.coling-main.12/

[374] C. Chen, X. Hao, W. Liu, X. Huang, X. Zeng, S. Yu, D. Li, Y. Huang, X. Liu, X. Wang, and W. Liu, "Acebench: A comprehensive evaluation of LLM tool usage," in *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, Eds. Association for Computational Linguistics, 2025, pp. 12 970–12 998. [Online]. Available: https://aclanthology.org/2025.findings-emnlp.697/

[375] R. Xiao, W. Ma, K. Wang, Y. Wu, J. Zhao, H. Wang, F. Huang, and Y. Li, "Flowbench: Revisiting and benchmarking workflow-guided planning for llm-based agents," in *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, ser. Findings of ACL, Y. Al-Onaizan, M. Bansal, and Y. Chen, Eds. Association for Computational Linguistics, 2024, pp. 10 883–10 900. [Online]. Available: https://doi.org/10.18653/v1/2024.findings-emnlp.638

[376] P. He, Z. Dai, B. He, H. Liu, X. Tang, H. Lu, J. Li, J. Ding, S. Mukherjee, S. Wang, Y. Xing, J. Tang, and B. Dumoulin, "Traject-bench:a trajectory-aware benchmark for evaluating agentic tool use," *CoRR*, vol. abs/2510.04550, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.04550

[377] Y. Hao, P. Cao, Z. Jin, H. Liao, Y. Chen, K. Liu, and J. Zhao, "Evaluating personalized tool-augmented llms from the perspectives of personalization and proactivity," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds. Association for Computational Linguistics, 2025, pp. 21 897–21 935. [Online]. Available: https://aclanthology.org/2025.acl-long.1064/

[378] Z. Wu, X. Liu, X. Zhang, L. Chen, F. Meng, L. Du, Y. Zhao, F. Zhang, Y. Ye, J. Wang *et al.*, "Mcpmark: A benchmark for stress-testing realistic and comprehensive mcp use," *arXiv preprint arXiv:2509.24002*, 2025.

[379] Y. Zhou, M. Zhao, Z. Wang, D. Gu, B. Guo, R. Ye, L. Han, C. Jin, and D. N. Metaxas, "M^3-bench: Multi-modal, multi-hop, multi-threaded tool-using MLLM agent benchmark," *CoRR*, vol. abs/2511.17729, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2511.17729

[380] S. Quan, J. Yang, B. Yu, B. Zheng, D. Liu, A. Yang, X. Ren, B. Gao, Y. Miao, Y. Feng, Z. Wang, J. Yang, Z. Cui, Y. Fan, Y. Zhang, B. Hui, and J. Lin, "Codeelo: Benchmarking competition-level code generation of llms with human-comparable elo ratings," *CoRR*, vol. abs/2501.01257, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2501.01257

[381] Z. Z. Wang, A. Asai, X. V. Yu, F. F. Xu, Y. Xie, G. Neubig, and D. Fried, "Coderag-bench: Can retrieval augment code generation?" 2025. [Online]. Available: https://arxiv.org/abs/2406.14497

[382] D. Zan, Z. Huang, W. Liu, H. Chen, L. Zhang, S. Xin, L. Chen, Q. Liu, X. Zhong, A. Li, S. Liu, Y. Xiao, L. Chen, Y. Zhang, J. Su, T. Liu, R. Long, K. Shen, and L. Xiang, "Multi-swe-bench: A multilingual benchmark for issue resolving," 2025. [Online]. Available: https://arxiv.org/abs/2504.02605

[383] H. S. Zheng, S. Mishra, H. Zhang, X. Chen, M. Chen, A. Nova, L. Hou, H. Cheng, Q. V. Le, E. H. Chi, and D. Zhou, "NATURAL PLAN: benchmarking llms on natural language planning," *CoRR*, vol. abs/2406.04520, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2406.04520

[384] P. A. Jansen, M. Côté, T. Khot, E. Bransom, B. D. Mishra, B. P. Majumder, O. Tafjord, and P. Clark, "Discoveryworld: A virtual environment for developing and evaluating automated scientific discovery agents," in *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, Eds., 2024. [Online]. Available: http://papers.nips.cc/paper_files/paper/2024/hash/13836f251823945316ae067350a5c366-Abstract-Datasets_and_Benchmarks_Track.html

[385] D. Wang, M. Cheng, Q. Liu, S. Yu, Z. Liu, and Z. Guo, "Paperarena: An evaluation benchmark for tool-augmented agentic reasoning on scientific literature," *CoRR*, vol. abs/2510.10909, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.10909

[386] L. Mitchener, J. M. Laurent, B. Tenmann, S. Narayanan, G. P. Wellawatte, A. D. White, L. Sani, and S. G. Rodriques, "Bixbench: a comprehensive benchmark for llm-based agents in computational biology," *CoRR*, vol. abs/2503.00096, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2503.00096

[387] K. Huang, A. Prabhakar, O. Thorat, D. Agarwal, P. K. Choubey, Y. Mao, S. Savarese, C. Xiong, and C. Wu, "Crmarena-pro: Holistic assessment of LLM agents across diverse business scenarios and interactions," *CoRR*, vol. abs/2505.18878, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.18878

[388] R. Min, Z. Qiao, Z. Xu, J. Zhai, W. Gao, X. Chen, H. Sun, Z. Zhang, X. Wang, H. Zhou, W. Yin, B. Zhang, X. Zhou, M. Yan, Y. Jiang, H. Liu, L. Ding, L. Zou, Y. R. Fung, Y. Li, and P. Xie, "Ecombench: Towards holistic evaluation of foundation agents in e-commerce," *CoRR*, vol. abs/2512.08868, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2512.08868

[389] E. Chen, S. Postelnik, K. Black, Y. Jiang, and J. H. Chen, "Medagentbench v2: Improving medical llm agent design," in *Biocomputing 2026: Proceedings of the Pacific Symposium*. World Scientific, 2025, pp. 354–371.

[390] Y. Zhao, W. Zhang, L. Xiao, Y. Zheng, M. Liu, and W. Y. B. Lim, "Advancing esg intelligence: An expert-level agent and comprehensive benchmark for sustainable finance," *arXiv preprint arXiv:2601.08676*, 2026.

[391] L. Gupta, L. Li, Y. Liu, S. G. Subramanian, K. Suleman, Z. Zhang, H. Lu, and S. Pasupalak, "World of workflows: a benchmark for bringing world models to enterprise systems," *CoRR*, vol. abs/2601.22130, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.22130

[392] S. K. R. Malay, S. Nayak, A. Tiwari, J. S. Nair, S. T. Madhusudhan, S. Davasam, S. Sunkara, and S. Rajeswar, "Enterpriseops-gym: Environments and evaluations for stateful agentic planning and tool use in enterprise settings."

[393] P. Xia, J. Chen, X. Yang, H. Tu, J. Liu, K. Xiong, S. Han, S. Qiu, H. Ji, Y. Zhou *et al.*, "Metaclaw: Just talk–an agent that meta-learns and evolves in the wild," *arXiv preprint arXiv:2603.17187*, 2026.

[394] B. Ye, R. Li, Q. Yang, Y. Liu, L. Yao, H. Lv, Z. Xie, C. An, L. Li, L. Kong, Q. Liu, Z. Sui, and T. Yang, "Claw-eval: Toward trustworthy evaluation of autonomous agents," 2026. [Online]. Available: https://arxiv.org/abs/2604.06132

[395] H. Ji, K. Xiong, S. Han, P. Xia, S. Qiu, Y. Zhou, J. Liu, J. Li, B. Li, Z. Zheng *et al.*, "Clawarena: Benchmarking ai agents in evolving information environments," *arXiv preprint arXiv:2604.04202*, 2026.

[396] Z. Xi, Y. Ding, W. Chen, B. Hong, H. Guo, J. Wang, D. Yang, C. Liao, X. Guo, W. He, S. Gao, L. Chen, R. Zheng, Y. Zou, T. Gui, Q. Zhang, X. Qiu, X. Huang, Z. Wu, and Y. Jiang, "Agentgym: Evolving large language model-based agents across diverse environments," *CoRR*, vol. abs/2406.04151, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2406.04151

[397] S. Qiao, R. Fang, Z. Qiu, X. Wang, N. Zhang, Y. Jiang, P. Xie, F. Huang, and H. Chen, "Benchmarking agentic workflow generation," in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. [Online]. Available: https://openreview.net/forum?id=vunPXOFmoi

[398] D. Nathani, L. Madaan, N. Roberts, N. Bashlykov, A. Menon, V. Moens, A. Budhiraja, D. Magka, V. Vorotilov, G. Chaurasia *et al.*, "Mlgym: A new framework and benchmark for advancing ai research agents," *arXiv preprint arXiv:2502.14499*, 2025.

[399] S. Chen, P. Moreira, Y. Xiao, S. Schmidgall, J. L. Warner, H. J. W. L. Aerts, T. Hartvigsen, J. Gallifant, and D. S. Bitterman, "Medbrowsecomp: Benchmarking medical deep research and computer use," *CoRR*, vol. abs/2505.14963, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.14963

[400] R. Awal, M. Massoud, A. Feizi, Z. Li, S. Wang, C. Pal, A. Agrawal, D. Vázquez, S. Reddy, J. A. Rodríguez, P. Taslakian, S. Gella, and S. Rajeswar, "Webmmu: A benchmark for multimodal multilingual website understanding and code generation," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, Eds. Association for Computational Linguistics, 2025, pp. 25 118–25 145. [Online]. Available: https://doi.org/10.18653/v1/2025.emnlp-main.1276

[401] H. Xu, X. Huang, Y. Liu, K. Yu, and Z. Deng, "Tps-bench: Evaluating AI agents' tool planning \& scheduling abilities in compounding tasks," *CoRR*, vol. abs/2511.01527, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2511.01527

[402] K. Li, J. Shi, Y. Xiao, M. Jiang, J. Sun, Y. Wu, S. Xia, X. Cai, T. Xu, W. Si, W. Li, D. Wang, and P. Liu, "Agencybench: Benchmarking the frontiers of autonomous agents in 1m-token real-world contexts," *CoRR*, vol. abs/2601.11044, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.11044

[403] Z. Su, J. Gao, H. Guo, Z. Liu, L. Zhang, X. Geng, S. Huang, P. Xia, G. Jiang, C. Wang, Y. Zhang, Y. R. M. Fung, and J. He, "Agentvista: Evaluating multimodal agents in ultra-challenging realistic visual scenarios," *CoRR*, vol. abs/2602.23166, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2602.23166

[404] DeepSeek-AI, "Deepseek-v3.2: Pushing the frontier of open large language models," *CoRR*, vol. abs/2512.02556, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2512.02556

[405] M. L. Team, "Longcat-flash-thinking-2601 technical report," *CoRR*, vol. abs/2601.16725, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.16725

[406] L. Xiaomi, "Mimo-v2-flash technical report," *CoRR*, vol. abs/2601.02780, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.02780

[407] T. Silver, S. Dan, K. Srinivas, J. B. Tenenbaum, L. P. Kaelbling, and M. Katz, "Generalized planning in PDDL domains with pretrained large language models," in *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, M. J. Wooldridge, J. G. Dy, and S. Natarajan, Eds. AAAI Press, 2024, pp. 20 256–20 264. [Online]. Available: https://doi.org/10.1609/aaai.v38i18.30006

[408] R. Wang, G. Todd, Z. Xiao, X. Yuan, M. Côté, P. Clark, and P. A. Jansen, "Can language models serve as text-based world simulators?" in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, L. Ku, A. Martins, and V. Srikumar, Eds. Association for Computational Linguistics, 2024, pp. 1–17. [Online]. Available: https://doi.org/10.18653/v1/2024.acl-short.1

[409] S. Mahdavi, R. Aoki, K. Tang, and Y. Cao, "Leveraging environment interaction for automated PDDL translation and planning with large language models," in *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, Eds., 2024. [Online]. Available: http://papers.nips.cc/paper_files/paper/2024/hash/44af065477781e7f8a8589b14a62c489-Abstract-Conference.html

[410] L. Su, Z. Zhang, G. Li, Z. Chen, C. Wang, M. Song, X. Wang, K. Li, J. Wu, X. Chen, Z. Qiao, Z. Zhang, H. Yin, S. Cai, R. Fang, Z. Tao, W. Yin, C. Qian, Y. Jiang, P. Xie, F. Huang, and J. Zhou, "Scaling agents via continual pre-training," *CoRR*, vol. abs/2509.13310, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.13310

[411] R. Pi, G. Lam, M. Shoeybi, P. Jannaty, B. Catanzaro, and W. Ping, "On data engineering for scaling llm terminal capabilities," *arXiv preprint arXiv:2602.21193*, 2026.

[412] W. Du, H. Gong, Z. Ling, K. Liu, L. Shen, X. Yao, Y. Xu, D. Shi, Y. Yang, and J. Chen, "Generalizable end-to-end tool-use RL with synthetic codegym," *CoRR*, vol. abs/2509.17325, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.17325

[413] Y. Zeng, S. Li, D. Dong, R. Xu, Z. Chen, L. Zheng, Y. Li, Z. Zhou, H. Zhao, L. Tian *et al.*, "Swe-hub: A unified production system for scalable, executable software engineering tasks," *arXiv preprint arXiv:2603.00575*, 2026.

[414] C. Guo, J. Wu, S. He, Y. Chen, Z. Kuang, S. Fan, B. Chen, S. Bao, J. Liu, H. Wu, Q. Zhu, W. Che, and H. Wang, "Menvagent: Scalable polyglot environment construction for verifiable software engineering," *CoRR*, vol. abs/2601.22859, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.22859

[415] J. Zhang, L. Ma, Y. Li, F. Wan, D. Qi, X. Zhao, J. Hou, Z. Xie, M. Ren, X. Wu, Z. Huang, L. Chen, Y. Ma, Q. Han, and X. Zhang, "Docksmith: Scaling reliable coding environments via an agentic docker builder," *CoRR*, vol. abs/2602.00592, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2602.00592

[416] Y. Lin, H. Wang, S. Wu, L. Fan, F. Pan, S. Zhao, and D. Tu, "Cli-gym: Scalable cli task generation via agentic environment inversion," *arXiv preprint arXiv:2602.10999*, 2026.

[417] Y. Shen, Y. Yang, Z. Xi, B. Hu, H. Sha, J. Zhang, Q. Peng, J. Shang, J. Huang, Y. Fan *et al.*, "Sciagentgym: Benchmarking multi-step scientific tool-use in llm agents," *arXiv preprint arXiv:2602.12984*, 2026.

[418] C. Zhang, Z. Gan, L. Zhu, Y. Pang, Q. Zhang, and R. Zhang, "Finmtm: A multi-turn multimodal benchmark for financial reasoning and agent evaluation," *CoRR*, vol. abs/2602.03130, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2602.03130

[419] W. Chi, Y. Fang, A. Yayavaram, S. Yayavaram, S. Karten, Q. A. Wei, R. Chen, A. Wang, V. Chen, A. Talwalkar *et al.*, "Gamedevbench: Evaluating agentic capabilities through game development," *arXiv preprint arXiv:2602.11103*, 2026.

[420] GLM, "GLM-5: from vibe coding to agentic engineering," *CoRR*, vol. abs/2602.15763, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2602.15763

[421] F. Li, H. Zhang, H. Huang, J. Wang, J. Hao, K. Yuan, M. Li, M. Zhang, P. Xu, W. Zhuang, Y. Shao, Z. Feng, C. Tang, C. Wang, C. Tong, F. Yang, G. Xiong, H. Gao, H. Gao, H. Wang, H. Liu, H. Sun, J. Li, J. Chang, J. Du, J. Peng, L. Cui, M. Jing, M. Wu, S. Yan, S. Qi, S. Xu, W. Zhao, X. Sun, X. Xie, Y. Wang, Y. Xia, Y. Cui, Y. Chen, Y. Wang, Y. Shi, Z. Shen, Z. Wang, M. Sun, L. Ye, and B. Chen, "Kat-coder-v2 technical report," 2026. [Online]. Available: https://arxiv.org/abs/2603.27703

[422] M. Chen, L. Zhang, Y. Feng, X. Wang, W. Zhao, R. Cao, J. Yang, J. Chen, M. Li, Z. Ma, H. Ge, Z. Zhang, Z. Cui, D. Liu, J. Zhou, J. Sun, J. Lin, and B. Hui, "Swe-universe: Scale real-world verifiable environments to millions," *CoRR*, vol. abs/2602.02361, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2602.02361

[423] K. Gandhi, S. Garg, N. D. Goodman, and D. Papailiopoulos, "Endless terminals: Scaling RL environments for terminal agents," *CoRR*, vol. abs/2601.16443, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.16443

[424] V. Verma, D. Huang, W. Chen, D. Klein, and N. Tomlin, "Measuring general intelligence with generated games," *CoRR*, vol. abs/2505.07215, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.07215

[425] D. Silver and R. S. Sutton, "Welcome to the era of experience," *Google AI*, vol. 1, p. 11, 2025.

[426] M. Zuo, F. P. Velez, X. Li, M. Littman, and S. H. Bach, "Planetarium: A rigorous benchmark for translating text to structured planning languages," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, L. Chiruzzo, A. Ritter, and L. Wang, Eds. Association for Computational Linguistics, 2025, pp. 11 223–11 240. [Online]. Available: https://doi.org/10.18653/v1/2025.naacl-long.560

[427] Z. Yu, Y. Yuan, T. Z. Xiao, F. F. Xia, J. Fu, G. Zhang, G. Lin, and W. Liu, "Generating symbolic world models via test-time scaling of large language models," *Trans. Mach. Learn. Res.*, vol. 2025, 2025. [Online]. Available: https://openreview.net/forum?id=zVo6PfBa0K

[428] Z. Ahmed, J. B. Tenenbaum, C. Bates, and S. J. Gershman, "Synthesizing world models for bilevel planning," *Trans. Mach. Learn. Res.*, vol. 2025, 2025. [Online]. Available: https://openreview.net/forum?id=m9V4JHLJrD

[429] Y. Xie, E. Liu, G. Zhang, N. Kotalwar, S. Gandhi, S. Acharya, X. Wang, C. Rose, G. Neubig, and D. Fried, "Hybrid-gym: Training coding agents to generalize across tasks," *arXiv preprint arXiv:2602.16819*, 2026.

[430] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., 2018, pp. 2455–2467. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/2de5d16682c3c35007e4e92982f1a2ba-Abstract.html

[431] Y. Zhang, C. Peng, B. Wang, P. Wang, Q. Zhu, F. Kang, B. Jiang, Z. Gao, E. Li, Y. Liu, and Y. Zhou, "Matrix-game: Interactive world foundation model," *CoRR*, vol. abs/2506.18701, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2506.18701

[432] X. He, C. Peng, Z. Liu, B. Wang, Y. Zhang, Q. Cui, F. Kang, B. Jiang, M. An, Y. Ren, B. Xu, H. Guo, K. Gong, C. Wu, W. Li, X. Song, Y. Liu, E. Li, and Y. Zhou, "Matrix-game 2.0: An open-source, real-time, and streaming interactive world model," *CoRR*, vol. abs/2508.13009, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.13009

[433] Y. Liu, J. Wang, H. Wang, B. Guo, and W. Li, "Imagine-then-plan: Agent learning from adaptive lookahead with world models," *CoRR*, vol. abs/2601.08955, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.08955

[434] K. Mei, J. Guo, S. Chang, M. Dong, D. Lee, X. Niu, and J. Jiang, "R-wom: Retrieval-augmented world model for computer-use agents," *CoRR*, vol. abs/2510.11892, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.11892

[435] Y. Cao, Y. Zhong, Z. Zeng, L. Zheng, J. Huang, H. Qiu, P. Shi, W. Mao, and W. Guanglu, "Mobiledreamer: Generative sketch world model for GUI agent," *CoRR*, vol. abs/2601.04035, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.04035

[436] Q. Garrido, M. Assran, N. Ballas, A. Bardes, L. Najman, and Y. LeCun, "Learning and leveraging world models in visual representation learning, 2024," *URL https://arxiv. org/abs/2403.00504*.

[437] G. Zhou, H. Pan, Y. LeCun, and L. Pinto, "Dino-wm: World models on pre-trained visual features enable zero-shot planning," *arXiv preprint arXiv:2411.04983*, 2024.

[438] S. Huang, L. Chen, P. Zhou, S. Chen, Z. Jiang, Y. Hu, P. Gao, H. Li, M. Yao, and G. Ren, "Enerverse: Envisioning embodied future space for robotics manipulation," *CoRR*, vol. abs/2501.01895, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2501.01895

[439] Y. Liao, P. Zhou, S. Huang, D. Yang, S. Chen, Y. Jiang, H. Yue, J. Cai, S. Liu, J. Luo, L. Chen, S. Yan, M. Yao, and G. Ren, "Genie envisioner: A unified world foundation platform for robotic manipulation," *CoRR*, vol. abs/2508.05635, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.05635

[440] S. Li, Q. Hao, Y. Shang, and Y. Li, "Keyworld: Key frame reasoning enables effective and efficient world models," *CoRR*, vol. abs/2509.21027, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.21027

[441] J. Guo, X. Ma, Y. Wang, M. Yang, H. Liu, and Q. Li, "Flowdreamer: A RGB-D world model with flow-based motion representations for robot manipulation," *IEEE Robotics Autom. Lett.*, vol. 11, no. 3, pp. 2466–2473, 2026. [Online]. Available: https://doi.org/10.1109/LRA.2026.3653273

[442] M. J. Kim, Y. Gao, T. Lin, Y. Lin, Y. Ge, G. Lam, P. Liang, S. Song, M. Liu, C. Finn, and J. Gu, "Cosmos policy: Fine-tuning video models for visuomotor control and planning," *CoRR*, vol. abs/2601.16163, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.16163

[443] L. Russell, A. Hu, L. Bertoni, G. Fedoseev, J. Shotton, E. Arani, and G. Corrado, "GAIA-2: A controllable multi-view generative world model for autonomous driving," *CoRR*, vol. abs/2503.20523, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2503.20523

[444] Q. Sun, L. Yang, W. Tang, W. Huang, K. Xu, Y. Chen, M. Liu, J. Yang, H. Zhu, Y. Wang, T. He, Y. Chen, X. Dai, N. Ye, and Q. Gu, "Learning primitive embodied world models: Towards scalable robotic learning," *CoRR*, vol. abs/2508.20840, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.20840

[445] L. Li, Q. Zhang, Y. Luo, S. Yang, R. Wang, F. Han, M. Yu, Z. Gao, N. Xue, X. Zhu, Y. Shen, and Y. Xu, "Causal world modeling for robot control," *CoRR*, vol. abs/2601.21998, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.21998

[446] J. Huang, Z. Ye, X. Hu, T. He, G. Zhang, S. Shi, J. Bian, and L. Jiang, "LIVE: long-horizon interactive video world modeling," *CoRR*, vol. abs/2602.03747, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2602.03747

[447] M. Deng, J. Hou, Z. Hu, and E. Xing, "Simura: A world-model-driven simulative reasoning architecture for general goal-oriented agents," *arXiv preprint arXiv:2507.23773*, 2025.

[448] X. Yin, X. Luo, H. Wu, L. Gao, and J. Song, "Unlocking smarter device control: Foresighted planning with a world model-driven code execution approach," in *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, Eds. Association for Computational Linguistics, 2025, pp. 3982–4005. [Online]. Available: https://aclanthology.org/2025.findings-emnlp.213/

[449] S. Guo, O. D. Domingues, R. Avalos, A. Courville, and F. Strub, "World modelling improves language model agents," *arXiv preprint arXiv:2506.02918*, 2025.

[450] W. Koh, S. Han, S. Lee, S. Yun, and J. Shin, "Generative visual code mobile world models," *CoRR*, vol. abs/2602.01576, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2602.01576

[451] A. Munim, A. Fallahpour, T. Szasz, A. Attarpour, R. Jiang, B. Sooriyakanthan, M. Sooriyakanthan, H. Whitney, J. Slivnick, B. Rubin *et al.*, "Echojepa: A latent predictive foundation model for echocardiography," *arXiv preprint arXiv:2602.02603*, 2026.

[452] R. Fang, Y. Liang, X. Wang, J. Wu, S. Qiao, P. Xie, F. Huang, H. Chen, and N. Zhang, "Memp: Exploring agent procedural memory," *CoRR*, vol. abs/2508.06433, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.06433

[453] J. Zhang, S. Hu, C. Lu, R. T. Lange, and J. Clune, "Darwin godel machine: Open-ended evolution of self-improving agents," *CoRR*, vol. abs/2505.22954, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.22954

[454] S. Li, Y. Huang, Z. Liu, Z. Li, J. fu, L. Song, J. Bian, J. Zhang, and R. Wang, "Experience-evolving multi-turn tool-use agent with hybrid episodic-procedural memory," 2026. [Online]. Available: https://arxiv.org/abs/2512.07287

[455] Y. Hu, S. Liu, Y. Yue, G. Zhang, B. Liu, F. Zhu, J. Lin, H. Guo, S. Dou, Z. Xi, S. Jin, J. Tan, Y. Yin, J. Liu, Z. Zhang, Z. Sun, Y. Zhu, H. Sun, B. Peng, Z. Cheng, X. Fan, J. Guo, X. Yu, Z. Zhou, Z. Hu, J. Huo, J. Wang, Y. Niu, Y. Wang, Z. Yin, X. Hu, Y. Liao, Q. Li, K. Wang, W. Zhou, Y. Liu, D. Cheng, Q. Zhang, T. Gui, S. Pan, Y. Zhang, P. Torr, Z. Dou, J. Wen, X. Huang, Y. Jiang, and S. Yan, "Memory in the age of AI agents," *CoRR*, vol. abs/2512.13564, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2512.13564

[456] Y. Du, W. Huang, D. Zheng, Z. Wang, S. Montella, M. Lapata, K. Wong, and J. Z. Pan, "Rethinking memory in AI: taxonomy, operations, topics, and future directions," *CoRR*, vol. abs/2505.00675, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.00675

[457] W. Huang, W. Zhang, Y. Liang, Y. Bei, Y. Chen, T. Feng, X. Pan, Z. Tan, Y. Wang, T. Wei, S. Wu, R. Xu, L. Yang, R. Yang, W. Yang, C. Yeh, H. Zhang, H. Zhang, S. Zhu, H. P. Zou, W. Zhao, S. Wang, W. Xu, Z. Ke, Z. Hui, D. Li, Y. Wu, L. He, C. Wang, X. Xu, B. Huang, J. Tan, S. Heinecke, H. Wang, C. Xiong, A. A. Metwally, J. Yan, C. Lee, H. Zeng, Y. Xia, X. Wei, A. Payani, Y. Wang, H. Ma, W. Wang, C. Wang, Y. Zhang, X. Wang, Y. Zhang, J. You, H. Tong, X. Luo, X. Liu, Y. Sun, W. Wang, J. J. McAuley, J. Zou, J. Han, P. S. Yu, and K. Shu, "Rethinking memory mechanisms of foundation agents in the second half: A survey," *CoRR*, vol. abs/2602.06052, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2602.06052

[458] L. Wang, Z. Jin, Y. Hao, Y. Chen, K. Liu, Y. Ao, and J. Zhao, "Think while watching: Online streaming segment-level memory for multi-turn video reasoning in multimodal large language models," *arXiv preprint arXiv:2603.11896*, 2026.

[459] S. Liang, P. Cao, J. Zhao, W. Teng, X. Liao, J. Zhao, and K. Liu, "Learning how to remember: A meta-cognitive management method for structured and transferable agent memory," *arXiv preprint arXiv:2601.07470*, 2026.

[460] C. Packer, V. Fang, S. G. Patil, K. Lin, S. Wooders, and J. E. Gonzalez, "Memgpt: Towards llms as operating systems," *CoRR*, vol. abs/2310.08560, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2310.08560

[461] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar, "Voyager: An open-ended embodied agent with large language models," *Trans. Mach. Learn. Res.*, vol. 2024, 2024. [Online]. Available: https://openreview.net/forum?id=ehfRiF0R3a

[462] Y. Cai, S. Cai, Y. Shi, Z. Xu, L. Chen, Y. Qin, X. Tan, G. Li, Z. Li, H. Lin, Y. Mao, K. Li, and X. Sun, "Training-free group relative policy optimization," *CoRR*, vol. abs/2510.08191, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.08191

[463] Y. Gu, B. Zheng, B. Gou, K. Zhang, C. Chang, S. Srivastava, Y. Xie, P. Qi, H. Sun, and Y. Su, "Simulate before act: Model-based planning for web agents."

[464] L. Huang, Y. Liu, J. Jiang, R. Zhang, J. Yan, J. Li, and X. Zhao, "Manusearch: Democratizing deep search in large language models with a transparent and open multi-agent framework," in *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, Eds. Association for Computational Linguistics, 2025, pp. 2403–2417. [Online]. Available: https://aclanthology.org/2025.findings-emnlp.130/

[465] S. Alzubi, C. Brooks, P. Chiniya, E. Contente, C. von Gerlach, L. Irwin, Y. Jiang, A. Kaz, W. Nguyen, S. Oh, H. Tyagi, and P. Viswanath, "Open deep search: Democratizing search with open-source reasoning agents," *CoRR*, vol. abs/2503.20201, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2503.20201

[466] X. Li, G. Dong, J. Jin, Y. Zhang, Y. Zhou, Y. Zhu, P. Zhang, and Z. Dou, "Search-o1: Agentic search-enhanced large reasoning models," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, Eds. Association for Computational Linguistics, 2025, pp. 5420–5438. [Online]. Available: https://doi.org/10.18653/v1/2025.emnlp-main.276

[467] T. Yang, Z. Yao, B. Jin, L. Cui, Y. Li, G. Wang, and X. Liu, "Demystifying and enhancing the efficiency of large language model based search agents," *CoRR*, vol. abs/2505.12065, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.12065

[468] M. Katz, H. Kokel, K. Srinivas, and S. Sohrabi, "Thought of search: Planning with language models through the lens of efficiency," in *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, Eds., 2024. [Online]. Available: http://papers.nips.cc/paper_files/paper/2024/hash/fa080fe0f218871faec1d8ba20e491d5-Abstract-Conference.html

[469] S. Wang, H. Yuan, L. M. Ni, and J. Guo, "Quantagent: Seeking holy grail in trading by self-improving large language model," *CoRR*, vol. abs/2402.03755, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2402.03755

[470] Y. Zhang, H. Ruan, Z. Fan, and A. Roychoudhury, "Autocoderover: Autonomous program improvement," in *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis, ISSTA 2024, Vienna, Austria, September 16-20, 2024*, M. Christakis and M. Pradel, Eds. ACM, 2024, pp. 1592–1604. [Online]. Available: https://doi.org/10.1145/3650212.3680384

[471] T. Gupta, L. Weihs, and A. Kembhavi, "Codenav: Beyond tool-use to using real-world codebases with llm agents," *arXiv preprint arXiv:2406.12276*, 2024.

[472] D. Chen, S. Lin, M. Zeng, D. Zan, J.-G. Wang, A. Cheshkov, J. Sun, H. H. Yu, G. Dong, A. Aliev *et al.*, "Coder: Issue resolving with multi-agent and task graphs," *arXiv preprint arXiv:2406.01304*, 2024.

[473] W. Tao, Y. Zhou, Y. Wang, W. Zhang, H. Zhang, and Y. Cheng, "Magis: Llm-based multi-agent framework for github issue resolution," *Advances in Neural Information Processing Systems*, vol. 37, pp. 51 963–51 993, 2024.

[474] X. Wang, Y. Zhang, O. Zohar, and S. Yeung-Levy, "Videoagent: Long-form video understanding with large language model as agent," in *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXX*, ser. Lecture Notes in Computer Science, A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, Eds., vol. 15138. Springer, 2024, pp. 58–76. [Online]. Available: https://doi.org/10.1007/978-3-031-72989-8_4

[475] Y. Wen, Y. Ye, Y. Zhang, D. Yang, and H. Zhu, "Real-time reasoning agents in evolving environments," *CoRR*, vol. abs/2511.04898, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2511.04898

[476] B. Chen, Z. Yue, S. Chen, Z. Wang, Y. Liu, P. Li, and Y. Wang, "Lvagent: Long video understanding by multi-round dynamical collaboration of MLLM agents," *CoRR*, vol. abs/2503.10200, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2503.10200

[477] V. Pahuja, Y. Lu, C. Rosset, B. Gou, A. Mitra, S. Whitehead, Y. Su, and A. H. Awadallah, "Explorer: Scaling exploration-driven web trajectory synthesis for multimodal web agents," in *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, ser. Findings of ACL, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds., vol. ACL 2025. Association for Computational Linguistics, 2025, pp. 6300–6323. [Online]. Available: https://aclanthology.org/2025.findings-acl.326/

[478] Y. Wang, L. Yang, G. Li, M. Wang, and B. Aragam, "Scoreflow: Mastering LLM agent workflows via score-based preference optimization," *CoRR*, vol. abs/2502.04306, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2502.04306

[479] S. Xu, J. Zhang, S. Di, Y. Luo, L. Yao, H. Liu, J. Zhu, F. Liu, and M. Zhang, "Robustflow: Towards robust agentic workflow generation," *CoRR*, vol. abs/2509.21834, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.21834

[480] Y. Wang, Z. Xu, Y. Huang, X. Wang, Z. Song, L. Gao, C. Wang, X. Tang, Y. Zhao, A. Cohan, X. Zhang, and X. Chen, "Dyflow: Dynamic workflow framework for agentic reasoning," *CoRR*, vol. abs/2509.26062, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.26062

[481] Z. Chen, K. Liu, Q. Wang, J. Liu, W. Zhang, K. Chen, and F. Zhao, "Mindsearch: Mimicking human minds elicits deep AI searcher," in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. [Online]. Available: https://openreview.net/forum?id=xgtXkyqw1f

[482] Z. Xu, Z. Xu, R. Zhang, C. Zhu, S. Yu, W. Liu, Q. Zhang, W. Ding, C. Yu, and Y. Wang, "Wideseek-r1: Exploring width scaling for broad information seeking via multi-agent reinforcement learning," *CoRR*, vol. abs/2602.04634, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2602.04634

[483] M. Kong, Z. Qu, Z. Zhou, P. Liang, X. Li, Z. Shang, Z. Hong, K. Huang, Z. Wang, and Z. Dai, "Workflow-r1: Group sub-sequence policy optimization for multi-turn workflow construction," *CoRR*, vol. abs/2602.01202, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2602.01202

[484] L. Yang and Y. Weng, "Researstudio: A human-intervenable framework for building controllable deep research agents," in *Proceedings of the 2025 Conference on Empirical Methods in Natural*

Language Processing, EMNLP 2025 - System Demonstrations, Suzhou, China, November 4-9, 2025, I. Habernal, P. Schulam, and J. Tiedemann, Eds. Association for Computational Linguistics, 2025, pp. 896–905. [Online]. Available: https://doi.org/10.18653/v1/2025.emnlp-demos.69

[485] J. Ruan, Z. Xu, Y. Peng, F. Ren, Z. Yu, X. Liang, J. Xiang, Y. Chen, B. Liu, C. Wu, Y. Luo, and J. Zhang, "Aorchestra: Automating sub-agent creation for agentic orchestration," 2026. [Online]. Available: https://arxiv.org/abs/2602.03786

[486] F. Lang, T. Long, and B. An, "Dr. mas: Stable reinforcement learning for multi-agent llm systems," *arXiv preprint arXiv:2602.08847*, 2026.

[487] Z. Liu, Z. Lai, Z. Gao, E. Cui, Z. Li, X. Zhu, L. Lu, Q. Chen, Y. Qiao, J. Dai, and W. Wang, "Controlllm: Augment language models with tools by searching on graphs," in *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XII*, ser. Lecture Notes in Computer Science, A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, Eds., vol. 15070. Springer, 2024, pp. 89–105. [Online]. Available: https://doi.org/10.1007/978-3-031-73254-6_6

[488] Y. Ma, Q. Yang, R. Cao, B. Li, F. Huang, and Y. Li, "Alibaba lingmaagent: Improving automated issue resolution via comprehensive repository exploration," in *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*, 2025, pp. 238–249.

[489] R. Gong, Q. Huang, X. Ma, Y. Noda, Z. Durante, Z. Zheng, D. Terzopoulos, L. Fei-Fei, J. Gao, and H. Vo, "Mindagent: Emergent gaming interaction," in *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, ser. Findings of ACL, K. Duh, H. Gómez-Adorno, and S. Bethard, Eds., vol. NAACL 2024. Association for Computational Linguistics, 2024, pp. 3154–3183. [Online]. Available: https://doi.org/10.18653/v1/2024.findings-naacl.200

[490] R. Singh, S. Joel, A. Mehrotra, N. Wadhwa, R. B. Bairi, A. Kanade, and N. Natarajan, "Code researcher: Deep research agent for large systems code and commit history," *CoRR*, vol. abs/2506.11060, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2506.11060

[491] G. Byrd, C. Rivera, B. Kemp, M. Booker, A. Schmidt, C. M. de Melo, L. Seenivasan, and M. Unberath, "Constrained natural language action planning for resilient embodied systems," *CoRR*, vol. abs/2510.06357, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.06357

[492] W. Cheng, E. Ni, W. Wang, Y. Sun, J. Liu, W. Shen, Y. Chen, B. Shi, and D. Wang, "MGA: memory-driven GUI agent for observation-centric interaction," *CoRR*, vol. abs/2510.24168, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.24168

[493] Y. Shang, Y. Li, K. Zhao, L. Ma, J. Liu, F. Xu, and Y. Li, "Agentsquare: Automatic LLM agent search in modular design space," in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. [Online]. Available: https://openreview.net/forum?id=mPdmDYIQ7f

[494] Y. Dang, C. Qian, X. Luo, J. Fan, Z. Xie, R. Shi, W. Chen, C. Yang, X. Che, Y. Tian, X. Xiong, L. Han, Z. Liu, and M. Sun, "Multi-agent collaboration via evolving orchestration," *CoRR*, vol. abs/2505.19591, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.19591

[495] S. Wu, S. Zhao, Q. Huang, K. Huang, M. Yasunaga, K. Cao, V. N. Ioannidis, K. Subbian, J. Leskovec, and J. Y. Zou, "Avatar: Optimizing LLM agents for tool usage via contrastive reasoning," in *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, Eds., 2024. [Online]. Available: http://papers.nips.cc/paper_files/paper/2024/hash/2db8ce969b000fe0b3fb172490c33ce8-Abstract-Conference.html

[496] Z. Hao, H. Wang, J. Luo, J. Zhang, Y. Zhou, Q. Lin, C. Wang, H. Dong, and J. Chen, "Recreate: Reasoning and creating domain agents driven by experience," *CoRR*, vol. abs/2601.11100, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.11100

[497] H. Li, S. Yang, W. Qi, S. Zhao, R. Hua, M. Song, X. Yang, and C. Peng, "Yunjue agent tech report: A fully reproducible, zero-start in-situ self-evolving agent system for open-ended tasks," *CoRR*, vol. abs/2601.18226, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.18226

[498] S. Xing, X. Xia, K. Wu, M. Jiang, J. Chen, and T. Fu, "Clinicalretrial: A self-evolving AI agent for clinical trial protocol optimization," *CoRR*, vol. abs/2601.00290, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.00290

[499] Y. He, J. Liu, Z. Hu, Y. Chen, Y. Liu, Y. Sui, Y. Li, N. Chen, J. Hu, B. Hooi, X. Xu, and J. Bian, "Evoclinician: A self-evolving agent for multi-turn medical diagnosis via test-time evolutionary learning," *CoRR*, vol. abs/2601.22964, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.22964

[500] M. Singh, V. Yadav, S. K. R. Malay, S. Nayak, S. Rajeswar, S. T. Madhusudhan, and E. Blanco, "Grammar search for multi-agent systems," *CoRR*, vol. abs/2512.14079, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2512.14079

[501] X. Huang, J. Chen, Y. Fei, Z. Li, P. Schwaller, and G. Ceder, "CASCADE: cumulative agentic skill creation through autonomous development and evolution," *CoRR*, vol. abs/2512.23880, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2512.23880

[502] Z. Zhao, W. Chai, X. Wang, B. Li, S. Hao, S. Cao, T. Ye, and G. Wang, "See and think: Embodied agent in virtual environment," in *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part VIII*, ser. Lecture Notes in Computer Science, A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, Eds., vol. 15066. Springer, 2024, pp. 187–204. [Online]. Available: https://doi.org/10.1007/978-3-031-73242-3_11

[503] T. Cai, X. Wang, T. Ma, X. Chen, and D. Zhou, "Large language models as tool makers," in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. [Online]. Available: https://openreview.net/forum?id=qV83K9d5WB

[504] S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, and K. Narasimhan, "Tree of thoughts: Deliberate problem solving with large language models," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/271db9922b8d1f4dd7aaef84ed5ac703-Abstract-Conference.html

[505] B. Xu, Z. Peng, B. Lei, S. Mukherjee, Y. Liu, and D. Xu, "Rewoo: Decoupling reasoning from observations for efficient augmented language models," *CoRR*, vol. abs/2305.18323, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2305.18323

[506] B. Liu, Y. Jiang, X. Zhang, Q. Liu, S. Zhang, J. Biswas, and P. Stone, "LLM+P: empowering large language models with optimal planning proficiency," *CoRR*, vol. abs/2304.11477, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2304.11477

[507] Q. Wu, G. Bansal, J. Zhang, Y. Wu, S. Zhang, E. Zhu, B. Li, L. Jiang, X. Zhang, and C. Wang, "Autogen: Enabling next-gen LLM applications via multi-agent conversation framework," *CoRR*, vol. abs/2308.08155, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2308.08155

[508] T. Fang, Z. Zhang, X. Wang, R. Wang, C. Qin, Y. Wan, J. Ma, C. Zhang, J. Chen, X. Li, H. Zhang, H. Mi, and D. Yu, "Cognitive kernel-pro: A framework for deep research agents and agent foundation models training," *CoRR*, vol. abs/2508.00414, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.00414

[509] T. Abuelsaad, D. Akkil, P. Dey, A. Jagmohan, A. Vempaty, and R. Kokku, "Agent-e: From autonomous web navigation to foundational design principles in agentic systems," *CoRR*, vol. abs/2407.13032, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2407.13032

[510] Y. Liu, X. Peng, J. Cao, S. Bo, Y. Zhang, X. Zhang, S. Cheng, X. Wang, J. Yin, and T. Du, "Tool-planner: Task planning with clusters across multiple tools," in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. [Online]. Available: https://openreview.net/forum?id=dRz3cizftU

[511] Y. Zhuang, X. Chen, T. Yu, S. Mitra, V. S. Bursztyn, R. A. Rossi, S. Sarkhel, and C. Zhang, "Toolchain*: Efficient action space navigation in large language models with a* search," in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net,

2024. [Online]. Available: https://openreview.net/forum?id=B6pQxqUcT8
[512] P. Lu, B. Chen, S. Liu, R. Thapa, J. Boen, and J. Zou, "Octotools: An agentic framework with extensible tools for complex reasoning," *CoRR*, vol. abs/2502.11271, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2502.11271
[513] K. Li, Z. Zhang, H. Yin, L. Zhang, L. Ou, J. Wu, W. Yin, B. Li, Z. Tao, X. Wang, W. Shen, J. Zhang, D. Zhang, X. Wu, Y. Jiang, M. Yan, P. Xie, F. Huang, and J. Zhou, "Websailor: Navigating super-human reasoning for web agent," *CoRR*, vol. abs/2507.02592, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2507.02592
[514] H. Xu, X. Zhang, H. Liu, J. Wang, Z. Zhu, S. Zhou, X. Hu, F. Gao, J. Cao, Z. Wang *et al.*, "Mobile-agent-v3.5: Multi-platform fundamental gui agents," *arXiv preprint arXiv:2602.16855*, 2026.
[515] R. Lu, Z. Hou, Z. Wang, H. Zhang, X. Liu, Y. Li, S. Feng, J. Tang, and Y. Dong, "Deepdive: Advancing deep search agents with knowledge graphs and multi-turn RL," *CoRR*, vol. abs/2509.10446, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.10446
[516] L. Su, Z. Zhang, G. Li, Z. Chen, C. Wang, M. Song, X. Wang, K. Li, J. Wu, X. Chen, Z. Qiao, Z. Zhang, H. Yin, S. Cai, R. Fang, Z. Tao, W. Yin, C. Qian, Y. Jiang, P. Xie, F. Huang, and J. Zhou, "Scaling agents via continual pre-training," *CoRR*, vol. abs/2509.13310, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.13310
[517] R. Wang, C. Zhang, J. Ma, J. Zhang, H. Wang, Y. Chen, B. Xue, T. Fang, Z. Zhang, H. Zhang, H. Mi, D. Yu, and K. Wong, "Explore to evolve: Scaling evolved aggregation logic via proactive online exploration for deep research agents," *CoRR*, vol. abs/2510.14438, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.14438
[518] Y. Zhai, S. Tao, C. Chen, A. Zou, Z. Chen, Q. Fu, S. Mai, L. Yu, J. Deng, Z. Cao, Z. Liu, B. Ding, and J. Zhou, "Agentevolver: Towards efficient self-evolving agent system," *CoRR*, vol. abs/2511.10395, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2511.10395
[519] R. Zhang, Q. Sun, C. Song, Y. Qi, and Z. Zheng, "Vsearcher: Long-horizon multimodal search agent via reinforcement learning," *arXiv preprint arXiv:2603.02795*, 2026.
[520] Y. Du, R. Ye, S. Tang, X. Zhu, Y. Lu, Y. Cai, and S. Chen, "Openseeker: Democratizing frontier search agents by fully open-sourcing training data," *arXiv preprint arXiv:2603.15594*, 2026.
[521] K. Zhang, G. Li, J. Li, Z. Li, and Z. Jin, "Toolcoder: Teach code generation models to use API search tools," *CoRR*, vol. abs/2305.04032, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2305.04032
[522] A. Zeng, M. Liu, R. Lu, B. Wang, X. Liu, Y. Dong, and J. Tang, "Agenttuning: Enabling generalized agent abilities for llms," in *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, ser. Findings of ACL, L. Ku, A. Martins, and V. Srikumar, Eds., vol. ACL 2024. Association for Computational Linguistics, 2024, pp. 3053–3077. [Online]. Available: https://doi.org/10.18653/v1/2024.findings-acl.181
[523] W. Wu, X. Guan, S. Huang, Y. Jiang, P. Xie, F. Huang, J. Cao, H. Zhao, and J. Zhou, "Masksearch: A universal pre-training framework to enhance agentic search capability," *CoRR*, vol. abs/2505.20285, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.20285
[524] B. Li, B. Zhang, D. Zhang, F. Huang, G. Li, G. Chen, H. Yin, J. Wu, J. Zhou, K. Li, L. Su, L. Ou, L. Zhang, P. Xie, R. Ye, W. Yin, X. Yu, X. Wang, X. Wu, X. Chen, Y. Zhao, Z. Zhang, Z. Tao, Z. Zhang, Z. Qiao, C. Wang, D. Yu, G. Fu, H. Shen, J. Yang, J. Lin, J. Zhang, K. Zeng, L. Yang, H. Yin, M. Song, M. Yan, P. Xia, Q. Xiao, R. Min, R. Ding, R. Fang, S. Chen, S. Huang, S. Wang, S. Cai, W. Shen, X. Wang, X. Guan, X. Geng, Y. Shi, Y. Wu, Z. Chen, Z. Li, and Y. Jiang, "Tongyi deepresearch technical report," *CoRR*, vol. abs/2510.24701, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.24701
[525] S. Team and G. DeepMind, "SIMA 2: A generalist embodied agent for virtual worlds," *CoRR*, vol. abs/2512.04797, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2512.04797
[526] Y. Yao, H. Zhu, P. Wang, J. Ren, X. Yang, Q. Chen, X. Li, D. Shi, J. Li, Q. Wang, S. Wang, X. Liu, J. Wu, M. Liu, and W. Zhou, "O-researcher: An open ended deep research model via multi-agent distillation and agentic RL," *CoRR*, vol. abs/2601.03743, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.03743
[527] Y. Yue, I. Salia, S. Hunt, C. Green, W. Shi, and J. J. Hunt, "Scaling behavior cloning improves causal reasoning: An open model for real-time video game playing," *CoRR*, vol. abs/2601.04575, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.04575
[528] Z. Li, D. Jiang, X. Ma, H. Zhang, P. Nie, Y. Zhang, K. Zou, J. Xie, Y. Zhang, and W. Chen, "Openresearcher: A fully open pipeline for long-horizon deep research trajectory synthesis," 2026. [Online]. Available: https://arxiv.org/abs/2603.20278
[529] R. Shao, R. Gao, B. Xie, Y. Li, K. Zhou, S. Wang, W. Guan, and G. Chen, "Hats: Hardness-aware trajectory synthesis for gui agents," *arXiv preprint arXiv:2603.12138*, 2026.
[530] Z. Chen, K. Liu, Q. Wang, W. Zhang, J. Liu, D. Lin, K. Chen, and F. Zhao, "Agent-flan: Designing data and methods of effective agent tuning for large language models," in *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, ser. Findings of ACL, L. Ku, A. Martins, and V. Srikumar, Eds., vol. ACL 2024. Association for Computational Linguistics, 2024, pp. 9354–9366. [Online]. Available: https://doi.org/10.18653/v1/2024.findings-acl.557
[531] Y. Qin, Y. Ye, J. Fang, H. Wang, S. Liang, S. Tian, J. Zhang, J. Li, Y. Li, S. Huang, W. Zhong, K. Li, J. Yang, Y. Miao, W. Lin, L. Liu, X. Jiang, Q. Ma, J. Li, X. Xiao, K. Cai, C. Li, Y. Zheng, C. Jin, C. Li, X. Zhou, M. Wang, H. Chen, Z. Li, H. Yang, H. Liu, F. Lin, T. Peng, X. Liu, and G. Shi, "UI-TARS: pioneering automated GUI interaction with native agents," *CoRR*, vol. abs/2501.12326, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2501.12326
[532] A. K. Jain, G. Gonzalez-Pumariega, W. Chen, A. M. Rush, W. Zhao, and S. Choudhury, "Multi-turn code generation through single-step rewards," in *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, ser. Proceedings of Machine Learning Research, A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, Eds., vol. 267. PMLR / OpenReview.net, 2025. [Online]. Available: https://proceedings.mlr.press/v267/jain25a.html
[533] S. Sun, H. Song, Y. Wang, R. Ren, J. Jiang, J. Zhang, F. Bai, J. Deng, X. Zhao, Z. Liu, L. Fang, Z. Wang, and J. Wen, "Simpledeepsearcher: Deep information seeking via web-powered reasoning trajectory synthesis," in *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, Eds. Association for Computational Linguistics, 2025, pp. 13 705–13 720. [Online]. Available: https://aclanthology.org/2025.findings-emnlp.739/
[534] T. Men, Z. Jin, P. Cao, Y. Chen, K. Liu, and J. Zhao, "Agent-rewardbench: Towards a unified benchmark for reward modeling across perception, planning, and safety in real-world multimodal agents," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025, pp. 17 521–17 541.
[535] M. Agarwal, I. Abdelaziz, K. Basu, M. Unuvar, L. A. Lastras, Y. Rizk, and P. Kapanipathi, "Toolrm: Outcome reward models for tool-calling large language models," *CoRR*, vol. abs/2509.11963, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.11963
[536] Z. Ma, J. Liu, X. Luo, Z. Huang, Q. Zhu, and W. Che, "Advancing tool-augmented large language models via meta-verification and reflection learning," in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V.2, KDD 2025, Toronto ON, Canada, August 3-7, 2025*, L. Antonie, J. Pei, X. Yu, F. Chierichetti, H. W. Lauw, Y. Sun, and S. Parthasarathy, Eds. ACM, 2025, pp. 2078–2089. [Online]. Available: https://doi.org/10.1145/3711896.3736835
[537] E. Shen, D. Tormoen, S. Shah, A. Farhadi, and T. Dettmers, "SERA: soft-verified efficient repository agents," *CoRR*, vol. abs/2601.20789, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.20789
[538] X. Ren, S. Dong, C. Yang, Q. Gao, Y. Zhao, Y. Liu, X. Geng, X. Li, D. Yan, Y. Li, C. Huang, D. Zhu, J. Ye, B. Yue, Y. Fu, M. Lv, Z. Feng, B. Zhou, B. Wang, X. Huang, Y.-G. Jiang, T. Gui, Q. Zhang, and Y. Zhang, "Magicagent: Towards generalized agent planning," *arXiv preprint arXiv:2602.19000*, 2026.
[539] J. Wu, B. Li, R. Fang, W. Yin, L. Zhang, Z. Tao,

D. Zhang, Z. Xi, Y. Jiang, P. Xie, F. Huang, and J. Zhou, "Webdancer: Towards autonomous information seeking agency," *CoRR*, vol. abs/2505.22648, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.22648

[540] Z. Li, X. Guan, B. Zhang, S. Huang, H. Zhou, S. Lai, M. Yan, Y. Jiang, P. Xie, F. Huang, J. Zhang, and J. Zhou, "Webweaver: Structuring web-scale evidence with dynamic outlines for open-ended deep research," *CoRR*, vol. abs/2509.13312, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.13312

[541] Z. Qiao, G. Chen, X. Chen, D. Yu, W. Yin, X. Wang, Z. Zhang, B. Li, H. Yin, K. Li, R. Min, M. Liao, Y. Jiang, P. Xie, F. Huang, and J. Zhou, "Webresearcher: Unleashing unbounded reasoning capability in long-horizon agents," *CoRR*, vol. abs/2509.13309, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.13309

[542] C. Gao, Z. Gu, Y. Liu, X. Qiu, S. Shen, Y. Wen, T. Xia, Z. Xu, Z. Zeng, B. Zhou *et al.*, "Ui-venus-1.5 technical report," *arXiv preprint arXiv:2602.09082*, 2026.

[543] B. Seed, "UI-TARS-2 technical report: Advancing GUI agent with multi-turn reinforcement learning," *CoRR*, vol. abs/2509.02544, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.02544

[544] H. Song, J. Jiang, Y. Min, J. Chen, Z. Chen, W. X. Zhao, L. Fang, and J. Wen, "R1-searcher: Incentivizing the search capability in llms via reinforcement learning," *CoRR*, vol. abs/2503.05592, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2503.05592

[545] H. Wang, C. Qian, W. Zhong, X. Chen, J. Qiu, S. Huang, B. Jin, M. Wang, K. Wong, and H. Ji, "OTC: optimal tool calls via reinforcement learning," *CoRR*, vol. abs/2504.14870, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2504.14870

[546] J. Singh, R. Magazine, Y. Pandya, and A. Nambi, "Agentic reasoning and tool integration for llms via reinforcement learning," *CoRR*, vol. abs/2505.01441, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.01441

[547] G. Lu, W. Guo, C. Zhang, Y. Zhou, H. Jiang, Z. Gao, Y. Tang, and Z. Wang, "VLA-RL: towards masterful and general robotic manipulation with scalable reinforcement learning," *CoRR*, vol. abs/2505.18719, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.18719

[548] Q. Zhao, R. Wang, D. Xu, D. Zha, and L. Liu, "R-search: Empowering LLM reasoning with search via multi-reward reinforcement learning," *CoRR*, vol. abs/2506.04185, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2506.04185

[549] Y. Xie, T. Chen, Z. Ge, and L. Ni, "Video-mtr: Reinforced multi-turn reasoning for long video understanding," *CoRR*, vol. abs/2508.20478, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.20478

[550] H. Jin, Q. Wang, W. Zhang, Y. Liu, and S. Cheng, "Videomem: Enhancing ultra-long video understanding via adaptive memory management," *CoRR*, vol. abs/2512.04540, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2512.04540

[551] H. Luo, Z. Li, Y. Xie, W. Zhang, Y. Li, and Y. Shen, "Intentrl: Training proactive user-intent agents for open-ended deep research via reinforcement learning," *CoRR*, vol. abs/2602.03468, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2602.03468

[552] Y. Li, X. Zhang, W. Lu, Z. Tang, M. Wu, H. Luo, T. Wu, Z. Peng, H. Mi, Y. Feng, N. Tan, C. Huang, H. Chen, and L. Shen, "Who deserves the reward? sharp: Shapley credit-based optimization for multi-agent system," *arXiv preprint arXiv:2602.08335*, 2026.

[553] Z. Wei, W. Yao, Y. Liu, W. Zhang, Q. Lu, L. Qiu, C. Yu, P. Xu, C. Zhang, B. Yin, H. Yun, and L. Li, "Webagent-r1: Training web agents via end-to-end multi-turn reinforcement learning," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, Eds. Association for Computational Linguistics, 2025, pp. 7909–7928. [Online]. Available: https://doi.org/10.18653/v1/2025.emnlp-main.401

[554] Z. Fei, L. Ji, S. Wang, J. Shi, J. Gong, and X. Qiu, "Unleashing embodied task planning ability in llms via reinforcement learning," *CoRR*, vol. abs/2506.23127, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2506.23127

[555] B. Liu, L. Guertler, S. Yu, Z. Liu, P. Qi, D. Balcells, M. Liu, C. Tan, W. Shi, M. Lin, W. S. Lee, and N. Jaques, "Spiral: Self-play on zero-sum games incentivizes reasoning via multi-agent multi-turn reinforcement learning," *arXiv:2506.24119*, 2025. [Online]. Available: https://arxiv.org/abs/2506.24119

[556] H. Zhang, X. Gu, J. Li, C. Ma, S. Bai, C. Zhang, B. Zhang, Z. Zhou, D. He, and Y. Tang, "Thinking with videos: Multimodal tool-augmented reinforcement learning for long video reasoning," *CoRR*, vol. abs/2508.04416, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.04416

[557] H. Lai, X. Liu, Y. Zhao, H. Xu, H. Zhang, B. Jing, Y. Ren, S. Yao, Y. Dong, and J. Tang, "Computerrl: Scaling end-to-end online reinforcement learning for computer use agents," *CoRR*, vol. abs/2508.14040, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.14040

[558] M. Jiang, A. Lupu, and Y. Bachrach, "Bootstrapping task spaces for self-improvement," *CoRR*, vol. abs/2509.04575, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.04575

[559] Z. Xi, J. Huang, C. Liao, B. Huang, H. Guo, J. Liu, R. Zheng, J. Ye, J. Zhang, W. Chen, W. He, Y. Ding, G. Li, Z. Chen, Z. Du, X. Yao, Y. Xu, J. Chen, T. Gui, Z. Wu, Q. Zhang, X. Huang, and Y. Jiang, "Agentgym-rl: Training LLM agents for long-horizon decision making through multi-turn reinforcement learning," *CoRR*, vol. abs/2509.08755, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.08755

[560] Z. Lu, J. Ye, F. Tang, Y. Shen, H. Xu, Z. Zheng, W. Lu, M. Yang, F. Huang, J. Xiao, and Y. Zhuang, "UI-S1: advancing GUI automation via semi-online reinforcement learning," *CoRR*, vol. abs/2509.11543, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.11543

[561] H. Zhang, X. Liu, B. Lv, X. Sun, B. Jing, I. L. Iong, Z. Hou, Z. Qi, H. Lai, Y. Xu, R. Lu, H. Wang, J. Tang, and Y. Dong, "Agentrl: Scaling agentic reinforcement learning with a multi-turn, multi-task framework," *CoRR*, vol. abs/2510.04206, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.04206

[562] Y. Zhao, L. Hu, J. Zhao *et al.*, "Stronger-mas: Multi-agent reinforcement learning for collaborative llms," *arXiv preprint arXiv:2510.11062*, 2025.

[563] T. Wei, Y. Yang, C. Zhang, J. Xing, Y. Shi, Z. Lu, and D. Ye, "Gtr-turbo: Merged checkpoint is secretly a free teacher for agentic VLM training," *CoRR*, vol. abs/2512.13043, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2512.13043

[564] S. He, L. Feng, Q. Wei, X. Cheng, L. Feng, and B. An, "Hierarchy-of-groups policy optimization for long-horizon agentic tasks," *arXiv preprint arXiv:2602.22817*, 2026.

[565] M. Luo, "Deepswe: Training a fully open-sourced, state-of-the-art coding agent by scaling rl, jul 2025," *URL https://www. together. ai/blog/deepswe*.

[566] B. Ichter, A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian, D. Kalashnikov, S. Levine, Y. Lu, C. Parada, K. Rao, P. Sermanet, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, M. Yan, N. Brown, M. Ahn, O. Cortes, N. Sievers, C. Tan, S. Xu, D. Reyes, J. Rettinghouse, J. Quiambao, P. Pastor, L. Luu, K. Lee, Y. Kuang, S. Jesmonth, N. J. Joshi, K. Jeffrey, R. J. Ruano, J. Hsu, K. Gopalakrishnan, B. David, A. Zeng, and C. K. Fu, "Do as I can, not as I say: Grounding language in robotic affordances," in *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, ser. Proceedings of Machine Learning Research, K. Liu, D. Kulic, and J. Ichnowski, Eds. PMLR, 2022, pp. 287–318. [Online]. Available: https://proceedings.mlr.press/v205/ichter23a.html

[567] J. Feng, S. Huang, X. Qu, G. Zhang, Y. Qin, B. Zhong, C. Jiang, J. Chi, and W. Zhong, "Retool: Reinforcement learning for strategic tool use in llms," *CoRR*, vol. abs/2504.11536, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2504.11536

[568] R. Luo, L. Wang, W. He, and X. Xia, "GUI-R1 : A generalist r1-style vision-language action model for GUI agents," *CoRR*, vol. abs/2504.10458, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2504.10458

[569] X. Wang, H. Zhang, H. Wang, Y. Shi, R. Li, K. Han, C. Tong, H. Deng, R. Sun, A. Taylor, Y. Zhu, J. Cong, Y. Sun, and W. Wang, "Arlarena: A unified framework for stable agentic reinforcement learning," *CoRR*, vol. abs/2602.21534, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2602.21534

[570] Z. Zhang, K. Song, X. Wang, Y. Hu, W. Yan, C. Zhao, H. P. Zou, H. Deng, S. R. Indurthi, S. Liu, S. Ma, X. Wang, X. E. Wang, and S. Wang, "CM2: reinforcement learning with checklist rewards for multi-turn and multi-step agentic tool

use," *CoRR*, vol. abs/2602.12268, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2602.12268

[571] Z. Xue, L. Zheng, Q. Liu, Y. Li, X. Zheng, Z. Ma, and B. An, "Simpletir: End-to-end reinforcement learning for multi-turn tool-integrated reasoning," *CoRR*, vol. abs/2509.02479, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2509.02479

[572] Z. Qi, X. Liu, I. L. Iong, H. Lai, X. Sun, J. Sun, X. Yang, Y. Yang, S. Yao, W. Xu, J. Tang, and Y. Dong, "Webrl: Training LLM web agents via self-evolving online curriculum reinforcement learning," in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. [Online]. Available: https://openreview.net/forum?id=oVKEAFjEqv

[573] Y. Zhang, S. Tang, Z. Li, Z. Han, and V. Tresp, "Webarbiter: A principle-guided reasoning process reward model for web agents," *CoRR*, vol. abs/2601.21872, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.21872

[574] K. Fan, K. Feng, M. Zhang, T. Peng, Z. Li, Y. Jiang, S. Chen, P. Pei, X. Cai, and X. Yue, "Exploring reasoning reward model for agents," *CoRR*, vol. abs/2601.22154, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2601.22154

[575] C. Yang, S. Su, S. Liu, X. Dong, Y. Yu, W. Su, X. Wang, Z. Liu, J. Zhu, H. Li, W. Wang, Y. Qiao, X. Zhu, and J. Dai, "Zerogui: Automating online GUI learning at zero human cost," *CoRR*, vol. abs/2505.23762, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2505.23762

[576] T. Wang, Z. Wu, J. Liu, J. Hao, J. Wang, and K. Shao, "Distrl: An asynchronous distributed reinforcement learning framework for on-device control agent," in *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. [Online]. Available: https://openreview.net/forum?id=LPG8pPSfQD

[577] H. Bai, Y. Zhou, J. Pan, M. Cemri, A. Suhr, S. Levine, and A. Kumar, "Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning," in *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, and C. Zhang, Eds., 2024. [Online]. Available: http://papers.nips.cc/paper_files/paper/2024/hash/1704ddd0bb89f159dfe609b32c889995-Abstract-Conference.html

[578] G. Dong, L. Bao, Z. Wang, K. Zhao, X. Li, J. Jin, J. Yang, H. Mao, F. Zhang, K. Gai, G. Zhou, Y. Zhu, J. Wen, and Z. Dou, "Agentic entropy-balanced policy optimization," *CoRR*, vol. abs/2510.14545, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2510.14545

[579] X. Zhang, J. Yu, and Z. Zhong, "Learning efficient communication protocols for multi-agent reinforcement learning," *arXiv:2511.09171*, 2025.

[580] Y. Wei, Z. Sun, E. McMilin, J. Gehring, D. Zhang, G. Synnaeve, D. Fried, L. Zhang, and S. I. Wang, "Toward training superintelligent software agents through self-play SWE-RL," *CoRR*, vol. abs/2512.18552, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2512.18552

[581] Z. Shen, X. Hu, X. Li, T. Fang, J. Li, and S. Zhang, "World-model-augmented web agents with action correction," *arXiv preprint arXiv:2602.15384*, 2026.

[582] K. Lu and T. M. Lab, "On-policy distillation," *Thinking Machines Lab: Connectionism*, 2025, https://thinkingmachines.ai/blog/on-policy-distillation.