ENVIRONMENTS AS SCAFFOLD: ENRICHING FEEDBACK TO BOOTSTRAP SELF-EVOLVING AGENTS IN LONG-HORIZON TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models demonstrate remarkable proficiency in static reasoning, yet training them as autonomous agents through Reinforcement Learning (RL) for long-horizon tasks is often hindered by severe reward sparsity. While conventional *agent-side warming* up via supervised fine-tuning (SFT) can alleviate this, it is frequently limited by data scarcity and constrained exploration. To address this, we propose a paradigm shift to *environment-side adaptation* by constructing **F**eedback-**E**nriched **E**nvironments (**FEEs**). Through a pilot study, we establish a feedback design strategy that reformulates environments by transitioning from action guidance to observation enrichment during the later stages of both intra-episode exploration and inter-episode evolution. Large-scale experiments on SciWorld and BFCL benchmarks using various Qwen3 model scales and RL algorithms such as GRPO, GSPO, and DAPO demonstrate that FEEs consistently yield performance improvements over standard settings. Furthermore, our analysis reveals that training with FEEs **(1)** stabilizes training dynamics by reducing entropy volatility, **(2)** facilitates proactive state-space exploration in difficult tasks, **(3)** ensures the internalization of environmental guidance into policy weights rather than acting as a mere inference-time prior, and **(4)** identifies intra-group feedback consistency as a critical boundary for stable optimization.

1 INTRODUCTION

Large Language Models (LLMs) (Yang et al., 2025a; OpenAI, 2023; Dubey et al., 2024; DeepSeek-AI et al., 2025) have demonstrated remarkable proficiency in domains such as mathematical reasoning (Glazer et al., 2024; Du et al., 2025) and code generation (Yang et al., 2024; Jimenez et al., 2024). As the field advances, the research focus is shifting from solving these *static, single-turn tasks* to building autonomous agents capable of exploring *dynamic, complex real-world environments*, such as web navigation (Zhou et al., 2024a; Bai et al., 2026) and terminal-based computer usage (Merrill et al., 2026a; Gandhi et al., 2026). These tasks are inherently **long-horizon** (Zhang et al., 2025e;b), often necessitating sequences of dozens of interaction steps to achieve a solution. To solve them, reinforcement learning (RL) based approaches (Chen et al., 2025b; Feng et al., 2025a; Wang et al., 2025a) have emerged as a central direction, where agents learn to adapt their policies through iterative feedback from the environments.

However, RL on long-horizon tasks suffers greatly from **reward sparsity**. Specifically, the agent lacks the capability for effective exploration and is prone to getting trapped in zero-reward trajectories, leading to vanishing gradients that leave the agent with no signal to learn from. While warming up the agent via supervised fine-tuning (SFT) on expert trajectories effectively increases the probability of obtaining initial positive rewards (Wen et al., 2025; Liu et al., 2025b), it faces significant bottlenecks. Not only is the collection of ground-truth trajectories expensive and hard to scale (Chen et al., 2025a), but over-optimizing the SFT objective also risks overly constraining the agent's behavior, thereby restricting the exploration potential necessary for effective RL (Kang et al., 2025).

To address this, we propose a paradigm shift from **agent-side** warming up to **environment-side** adaptation. Specifically, instead of seeking a better-initialized agent, we propose to construct more suitable environments during RL by systematically diversifying and enriching their feedback signals.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

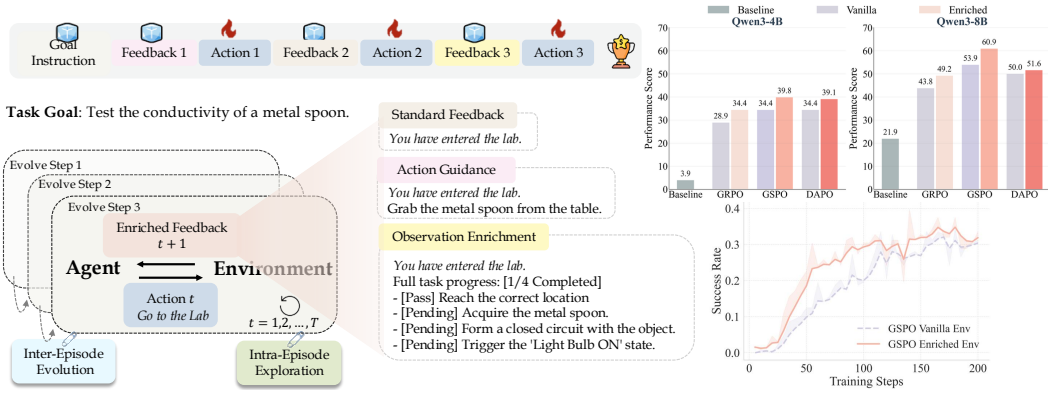

Figure 1: Illustration of the Feedback-Enriched Environments (FEEs).

As shown in Figure 1, while standard feedback leaves the agent searching blindly, enriched feedback guides it to enter the lab and retrieve the metal spoon, which boosts the success rate and helps the agent better understand the environment.

In particular, we study two key dimensions of environment feedback design: **what** information to provide and **when** to deliver it to the agent. For the former, we consider *action guidance* for suggesting immediate next steps and *observation enrichment* for providing supplementary state information. For the latter, we analyze when these signals are presented across two temporal scales: *intra-episode exploration* within a single trajectory and *inter-episode evolution* throughout the training lifecycle. Our initial study suggests that the most effective enriching strategy is to deliver action guidance during early interaction steps and training phases, and then transition to observation enrichment in the later stages of both processes.

To systematically validate this strategy, we build **F**eedback-**E**nriched **E**nvironments (**FEEs**) based on standard environments from two widely adopted benchmarks: SciWorld (Wang et al., 2022) and BFCL (Patil et al., 2025). Within these FEEs, we train Qwen3-4B and Qwen3-8B models with various agent-side RL algorithms, including GRPO (Shao et al., 2024), DAPO (Yu et al., 2025), and GSPO (Zheng et al., 2025). Extensive empirical results demonstrate that training in FEEs consistently outperforms training in standard environments, yielding an average improvement of 2.82% across various model scales and RL algorithms.

Additionally, to gain insights beyond performance scores, we further analyze the impacts of training agents with our proposed feedback-enriched environments. **(1)** *How do FEEs contribute to agent-side RL training stability?* Through an analysis of policy entropy dynamics, we find that FEEs act as a stabilizer that reduces gradient volatility and ensures smoother convergence, a benefit that persists even under explicit entropy regularization. **(2)** *Do FEEs promote more effective state-space exploration?* By analyzing average success rates during training and accuracy across environments of varying difficulty levels, we find that FEEs enable agents to access previously unreachable states while retaining robust exploration capabilities in standard environments. **(3)** *Is the environmental feedback internalized into the agent's policy weights?* Our analysis shows that feedback is internalized into the policy weights rather than acting as a mere inference-time prior. **(4)** *Should environmental feedback remain consistent or diverse within sampling groups?* We investigate the boundary of feedback diversification and find that intra-group consistency is crucial for stable optimization, as excessive stochasticity within a single rollout group introduces harmful noise.

Our contributions can be summarized as follows:

- We propose a paradigm shift from agent-side training to environment-side adaptation. We develop a systematic feedback design strategy that optimizes what information to provide (action guidance vs. observation enrichment) and when to provide it (intra-episode vs. inter-episode) to facilitate more effective RL training in long-horizon tasks.
- Based on our strategy, we construct FEEs using the SciWorld and BFCL benchmarks. Extensive experiments across various model scales (Qwen3-4B/8B) and RL algorithms

(GRPO, DAPO, GSPO) demonstrate that FEEs consistently improve performance compared with the standard environments.

- We provide an in-depth analysis of the impacts of training with FEEs. Our findings reveal that these environments promote proactive state-space exploration in difficult tasks, stabilize training dynamics through entropy reduction, ensure the internalization of external guidance into policy weights, and identify intra-group feedback consistency as a critical factor for stable optimization.

2 PRELIMINARY

Environment Definition The environment can be defined as a goal-conditioned Partially Observable Markov Decision Process (POMDP):

$$e_\phi = (\mathcal{G}, \mathcal{S}, \mathcal{A}, O, \mathcal{T}, r) \tag{1}$$

where $\mathcal{G}$ is a set of potential goals to be accomplished by the agent, S is a set of internal states, A is the set of actions available for the agent, O is a set of observations accessible to the agent, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the state transition probability function, and $r : \mathcal{S} \times \mathcal{A} \times \mathcal{G} \to \mathbb{R}$ is the reward function.

Agent Exploration Formally, the agent π_θ interacts with the environment e_ϕ to achieve a goal g within a finite horizon of T discrete time steps. Given the current partial observation $o_t \in O$ and the interaction history h_t, the agent generates a textual action $a_t \in \mathcal{A}$. Accordingly, the agent's behavior is modeled as a conditional distribution over the output tokens, formally defined as:

$$\begin{aligned} a_t &\sim \pi_\theta\Big(a_t \mid g, o_t, h_t\Big), \\ h_t &= (o_1, a_1), (o_2, a_2), \ldots, (o_{t-1}, a_{t-1}) \end{aligned} \tag{2}$$

Upon executing a_t, the environment transitions to the next state s_{t+1} governed by the transition function $\mathcal{T}(s_{t+1} \mid s_t, a_t)$ and generates the subsequent observation o_{t+1}. The episode trajectory is represented as $\tau = (o_1, a_1, \ldots, o_T, a_T)$, where the final performance is evaluated by a trajectory-level reward r_τ, typically serving as a binary indicator of task success or failure. We define **enriched feedback** as the result of an intervention on the observation space, $o_t^+ = \mathcal{E}(o_t, h_t)$, forming a new environment $e_\phi^+ = (\mathcal{G}, \mathcal{S}, \mathcal{A}, \Omega^+, \mathcal{T}, \mathcal{R})$. Consequently, the agent's behavior $a_t \sim \pi_\theta(a_t \mid g, o_t^+, h_t)$ is shifted, implicitly reshaping the action distribution to vary task difficulty and encourage trajectory diversity. For clarity, we henceforth refer to the original environment e_ϕ as the **standard environment** and e_ϕ^+ as the **enriched environment**.

Agentic RL Given a goal g, the agent samples a batch of N candidate trajectories $\{\tau_1, \tau_2, \ldots, \tau_N\}$ according to its current policy π_θ. Upon completion, each trajectory τ_k receives the final trajectory-level reward $r(\tau_k)$. The advantage $A(\tau_k)$ for each trajectory is computed based on the group statistics:

$$A(\tau_k) = GroupComputation[(r_{\tau_k})_{k=1}^N] \tag{3}$$

Subsequently, the computed trajectory-level advantage $A(\tau_k)$ is assigned uniformly to all **agent-generated tokens** $[a_i]_{i=1}^T$ within τ_k. Notably, tokens corresponding to **environment-generated tokens** $[o_i]_{i=1}^T$ are masked out, ensuring that the policy gradient is estimated solely based on the agent's actions. We term the continuous refinement of the agent's policy π_θ via these gradients as **agent evolving**. More details can be found in Appendix A.

3 FEEDBACK DESIGN STRATEGY FOR FEES

In this section, we empirically investigate how to design effective FEEs through the lens of **what** information to provide and **when** to provide it. Specifically, we categorize feedback into action guidance and observation enrichment, examining their roles during intra-episode exploration and inter-episode evolution. The results demonstrate that action guidance is better suited for the initial stages, while state enrichment is more beneficial for the later stages of both exploration and training.

3.1 Design Choices

Action Guidance (AG). Action guidance integrates procedural hints into environmental feedback to effectively narrow the search space of the policy π_θ. This acts as a soft intervention, preventing the agent from becoming trapped in redundant exploration loops. For instance, as shown in Figure 1, in a conductivity experiment, an unguided agent might erroneously waste steps exploring a classroom for materials. Action guidance, however, explicitly prompts the agent to "*enter the laboratory*", thereby pruning this irrelevant branch and steering the trajectory efficiently toward the goal.

Observation Enrichment (OE). Complementarily, observation enrichment augments raw observations with supplementary semantic information to address the inherent partial observability of the POMDP environment. For instance, in a circuit assembly task, while a generic observation merely states "*wire connected*", enriched feedback elucidates hidden states, such as "*the cathode remains unpowered*". This transparency empowers the agent to identify the missing link and execute remedial actions, rather than guessing blindly.

Intra-episode Exploration. This dimension corresponds to the step-wise interaction process within a single episode's finite horizon T. Specifically, it investigates whether enriched feedback should be introduced during the initial stages of exploration or delayed until the later phases of the episode.

Inter-episode Evolution. This dimension tracks the continuous refinement of the agent's policy π_θ across the entire training lifecycle. Specifically, it investigates whether enriched feedback should be introduced in the early stages of evolution when the agent's capabilities are limited, or in the later stages when it has become more proficient.

3.2 Empirical Validation

Experimental Setup. To evaluate different feedback strategies, we construct enriched variants of the standard SciWorld environment (Wang et al., 2022), which evaluates the capability of agents to design and execute elementary science experiments within interactive text-based environments. Specifically, we limit each episode to 15 steps, where **AG-Early** and **AG-Late** provide action guidance at steps 1–3 and 6–10, respectively, while **OE-Early** and **OE-Late** introduce observation enrichment during the same intervals. Throughout the entire agent evolution process, each enrichment operation is applied stochastically with a 0.5 probability. More details can be found in Appendix B.

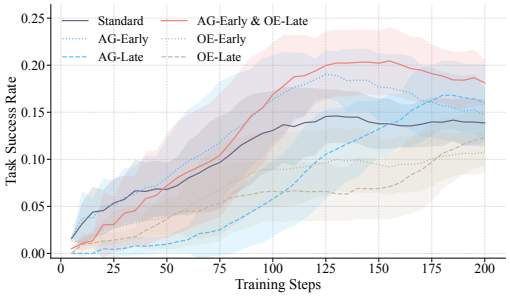

Figure 2: RL training dynamics under various feedback enrichment strategies. Task success rate is evaluated on the standard environment. Curves are smoothed for visual clarity, and shaded regions denote variance.

Implementation. Training is conducted with `Qwen3-4B-Thinking-2507` using GRPO (Shao et al., 2024) for 200 steps across the four enriched environments. Each training step utilizes 16 parallel environments with a 15-interaction rollout length and a 1×10^{-6} learning rate. The task success rate is evaluated on the standard environment every 5 training steps, using a strictly disjoint set of tasks to prevent training contamination.

3.3 Results and Analysis

Action guidance generally outperforms observation enrichment. As shown in Figure 2, the blue curves consistently maintain a superior performance margin over the brown curves throughout the training process. A plausible rationale is that action guidance explicitly prunes the massive search space by prescribing valid next steps, effectively bypassing the exploration bottleneck. In contrast, observation enrichment provides supplementary semantic information which imposes a higher cognitive load. The agent must implicitly learn to map these new features to optimal actions via complex causal reasoning, a process inherently slower than following procedural instructions.

Table 1: Main evaluation results on two selected benchmarks. Color blocks group identical RL algorithms to show the impact of training environments. Performance scores are reported as percentages (%). Values in parentheses indicate the absolute percentage point improvement (↑) or degradation (↓) of FEE training over the standard baseline.

Model	BFCL V3 Multi-Turn				SciWorld	Average
	Base	Long Context	Miss Func	Miss Param		
GPT-5.4	47.00	56.00	56.00	49.00	52.34	52.07
Kimi-K2-Thinking	69.00	54.00	68.00	57.00	19.53	53.51
Qwen3-235B-Thinking	36.00	29.00	32.00	23.00	55.47	35.09
Qwen3-4B	35.00	28.00	27.00	24.00	3.90	23.58
GRPO + Standard	58.00	52.00	34.00	32.00	28.91	40.98
GRPO + Enriched	68.00 (↑10.00%)	46.00 (↓6.00%)	43.00 (↑9.00%)	30.00 (↓2.00%)	34.38 (↑5.47%)	44.27 (↑3.29%)
DAPO + Standard	61.00	44.00	39.00	41.00	34.38	43.88
DAPO + Enriched	71.00 (↑10.00%)	50.00 (↑6.00%)	45.00 (↑6.00%)	34.00 (↓7.00%)	39.06 (↑4.68%)	47.81 (↑3.93%)
GSPO + Standard	54.00	40.00	29.00	25.00	34.38	36.48
GSPO + Enriched	54.00	40.00	37.00 (↑8.00%)	22.00 (↓3.00%)	39.84 (↑5.46%)	38.57 (↑2.09%)
Qwen3-8B	35.00	24.00	30.00	25.00	21.88	27.18
GRPO + Standard	66.00	35.00	52.00	37.00	43.75	46.75
GRPO + Enriched	66.00	36.00 (↑1.00%)	48.00 (↓4.00%)	38.00 (↑1.00%)	49.22 (↑5.47%)	47.44 (↑0.69%)
DAPO + Standard	65.00	33.00	47.00	34.00	50.00	45.80
DAPO + Enriched	71.00 (↑6.00%)	35.00 (↑2.00%)	49.00 (↑2.00%)	34.00	51.56 (↑1.56%)	48.11 (↑2.31%)
GSPO + Standard	58.00	29.00	41.00	27.00	53.91	41.78
GSPO + Enriched	65.00 (↑7.00%)	32.00 (↑3.00%)	40.00 (↓1.00%)	34.00 (↑7.00%)	60.94 (↑7.03%)	46.39 (↑4.61%)

What we enrich defines when we enrich. As shown in Figure 2, AG-Early outperforms AG-Late in earlier stages, while OE-Late exhibits a sharp upward trend in the later stages compared to OE-Early. This indicates that from both intra- and inter-episode perspectives, AG should be applied during the initial stage, while OE should be introduced in the later stage. A plausible explanation is that action guidance effectively prunes the initial combinatorial search space to bootstrap early learning, but becomes redundant once basic navigation is mastered. Conversely, observation enrichment imposes a higher cognitive load and requires a foundational policy to interpret the additional semantics, making it less effective initially but highly potent later on.

Combining *AG-Early* and *OE-Late* yields superior performance. To further validate that distinct strategies suit different training stages, we introduce a hybrid approach applying AG-Early for the first 100 steps and switching to OE-Late for the remaining 100 steps. As shown by the red curve in Figure 2, this combination surpasses all other settings, achieving the highest success rate. This success highlights a general strategy for environment enrichment: *across both the agent-environment interaction and agent evolution processes, action guidance should be leveraged during the early stage to bootstrap exploration, while observation enrichment should be introduced in the later stage for advanced policy refinement*.

4 Experiments

4.1 Experiment Setting

Enriched Environments. To systematically validate our *AG-Early and OE-Late* strategy, we conduct enriched environments for agent training from two primary benchmarks: SciWorld and BFCL-v3 Multi-Turn (Patil et al., 2025), which is introduced to evaluate the agents' capability to handle dynamic and realistic user interactions across multiple dialogue turns via predefined APIs. In SciWorld, action guidance represents concrete valid actions, and observation enrichment represents real-time task progress tracking. In BFCL, action guidance represents suggested actions appended to the user query, and observation enrichment represents expanded raw return contents of invoked tools. Consistent with Section 3.2, we train the agent for 200 steps with a 0.5 enrichment probability, transitioning from AG-Early to OE-Late at step 100, while ensuring that the standard evaluation tasks remain strictly disjoint. Detailed implementations and examples are provided in Appendix B.

RL Algorithms. We evaluate our approach with three representative RL algorithms: GRPO (Shao et al., 2024), DAPO (Yu et al., 2025), and GSPO (Zheng et al., 2025). GRPO estimates the advantage directly by computing relative scores within a group, thereby bypassing the need for a separate critic model. DAPO incorporates specialized strategies such as clip-higher and dynamic sampling to stabilize optimization. GSPO calculates the importance ratio based on sequence-level likelihoods. Details of the mathematical expressions of the algorithms can be found in Appendix A.

Training. We adopt the 4B and 8B variants of the Qwen3 series as the foundational backbones for RL training. Training utilizes a 1×10^{-6} learning rate with evaluations conducted every 5 steps, where the peak performance is recorded in Table 1. To establish a performance ceiling, we incorporate leading proprietary models as baselines, including GPT-5.4, Kimi-K2-Thinking (Team, 2025), Qwen3-235B-Thinking (Yang et al., 2025a).

4.2 Main Results

RL training effectively bridges the gap between open-source models and leading proprietary models. RL training significantly enhances the model's capabilities in multi-turn interactions. As shown in Table 1, Qwen3-4B and Qwen3-8B models initially exhibit limited proficiency, yielding average scores of only 23.58% and 27.18%, respectively. After RL training, Qwen3-4B and 8B reach 47.81% and 48.11%, comfortably outperforming Qwen3-235B-Thinking and closely approaching top-tier closed-source models like GPT-5.4. This confirms that RL training effectively transforms foundational knowledge into specialized execution skills for complex agentic tasks.

Training agents in FEEs consistently yields superior performance compared to standard environments. This advantage remains robust across all evaluated model scales, optimization algorithms, and benchmarks. For instance, as shown in Table 1, the Qwen3-8B model optimized via GSPO on SciWorld improves from 53.91% to 60.94% when trained with enriched feedback. Similarly, the Qwen3-4B model using GRPO on BFCL-Base tasks achieves a 10.00% absolute performance increase. These consistent gains validate the effectiveness of our feedback enrichment strategy.

FEEs may lead agents to rely too much on environmental feedback, weakening their ability to question the input context. Despite widespread gains, training with FEEs may cause slight performance regressions in scenarios where agents must question input sufficiency rather than blindly execute commands. As shown in Table 1, with FEEs, the Qwen3-4B model trained with DAPO drops from 41.00% to 34.00% on Miss Param tasks where essential information is missing from user requests. Similarly, the Qwen3-8B model with GRPO declines from 52.00% to 48.00% on Miss Func tasks where no available tools are provided to fulfill the user request. We hypothesize that the proactive guidance in FEEs makes agents overly inclined to follow interaction cues, leading them to prioritize execution over necessary clarification.

5 Discussion

5.1 Training Stability

Exp 1. To investigate how FEEs affect training stability, we track the evolution of policy entropy across four experimental configurations. Specifically, we compare standard environments and FEEs under two distinct settings: with and without an entropy regularization loss. Since entropy loss explicitly forces the agent to maintain high policy diversity for exploration, we examine whether FEEs provide independent stability even under such volatile conditions. The resulting entropy curves of Qwen3-4B over 300 training steps on SciWorld related environments are presented in Figure 3 (Left).

Res 1. As shown in Figure 3 (Left), training within FEEs significantly delays or prevents premature policy collapse compared to standard environments. Specifically, in configurations without explicit entropy regularization represented by the solid lines, the agent trained in FEEs maintains steady entropy throughout the entire 300 steps without collapsing, whereas the standard environment suffers from a sharp drop to zero at approximately 250 steps. When entropy regularization is introduced to force exploration represented by the dashed lines, FEEs successfully sustain stable training for nearly 200 steps, while the standard environment collapses much earlier at around 130 steps.

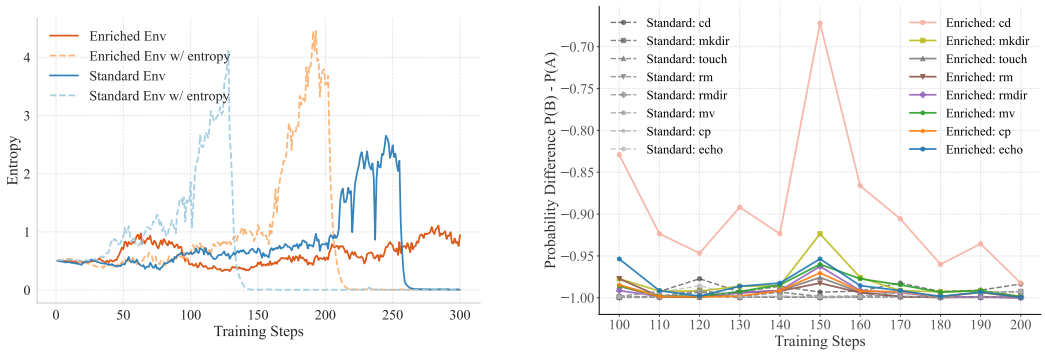

Figure 3: **Left:** Policy entropy curves of Qwen3-4B over 300 training steps in standard SciWorld environments and FEEs, with and without entropy regularization. **Right:** Probability difference between the original and enriched feedback options across training steps.

Takeaway 1. FEEs stabilize RL training with more diverse environment feedback, regardless of explicit entropy regularization.

5.2 State-Space Exploration

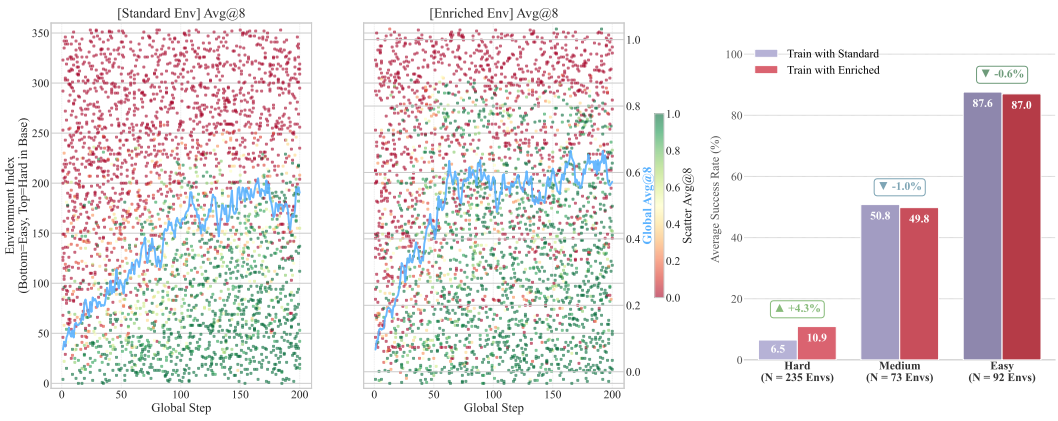

Figure 4: **Left & Middle:** Exploration dynamics of Qwen3-4B on BFCL under standard environments and FEEs. The x-axis denotes training steps and the y-axis denotes environment indices ordered by difficulty (bottom: easy, top: hard). Each point represents the avg@8 success rate of a sampled environment, while the blue curve shows the global average avg@8. **Right:** Validation success rates of models trained in standard environments and FEEs, evaluated on standard environments partitioned into hard, medium, and easy difficulty tiers.

Exp 2. We examine whether FEEs facilitate more effective and persistent state-space exploration from both the training and evaluation perspectives. Specifically, during training, we monitor Qwen3-4B on the BFCL benchmark by calculating the average success rate across 8 rollouts for each sampled environment. To assess the persistence of these capabilities during evaluation, we partition 400 environments into easy ($[0.66, 1]$), medium ($[0.33, 0.66]$), and hard ($[0, 0.33]$) difficulty tiers based on the success rates achieved by models trained in standard environments, and then evaluate models trained in FEEs across these tiers.

Res 2. The experimental results are illustrated in Figure 4, yielding the following analysis. (1) From the training perspective, FEEs improve the agent's exploration efficiency during RL training. For instance, the FEE-trained agent exhibits a significantly higher density of green squares in the later stages of training, whereas the standard baseline remains dominated by red squares across many environments. This suggests that enriched feedback enables the agent to explore and master a broader range of environment states. (2) From the evaluation perspective, the exploration capability

learned with FEEs transfers effectively to standard environments. For instance, the FEE-trained model consistently outperforms the baseline across all difficulty levels, achieving a notable 4.3% improvement on "hard" environments. This indicates that the agent develops a more proactive exploration strategy that allows it to navigate low-probability states even without enriched feedback.

Takeaway 2. FEEs encourage persistent exploration that is internalized by the policy and remains effective in standard, high-difficulty environments.

5.3 Feedback Internalization

Exp 3. We investigate whether the enriched information in FEEs is internalized into the agent's policy weights or merely functions as a temporary inference-time hint. To study this, we design a controlled prediction experiment based on the GorillaFileSystem environment in BFCL. Specifically, we select a subset of enriched file-system operations, including `cd`, `mkdir`, `touch`, `rm`, `rmdir`, `mv`, `cp`, and `echo`. While standard feedback only provides standard tool outputs, the enriched feedback additionally includes the current absolute path. We then formulate multiple-choice probe questions that ask the model to predict the expected tool output after executing a command, and compare the output probabilities assigned to the original and enriched feedback options. Details of these probe questions can be found in Table 2 and Appendix C. We further apply these probe questions to Qwen3-4B throughout the later stages of BFCL training, evaluating them every 10 training steps to monitor the evolution of knowledge internalization.

Table 2: Probing Questions in Research Question 3

`name`	`Q1_cd_bfcl_history`
`question`	You have just been trained in the BFCL environment. Based on your recent training history, when you successfully execute the tool `cd(folder='../src')`, does the environment's return value include the `_new_directory_contents` information?
`options`	**A:** No, it does not include `_new_directory_contents`.
	B: Yes, it automatically includes `_new_directory_contents`.

Res 3. Figure 3 (Right) illustrates the probability margin between the original and enriched feedback options across training steps. When tested in standard environments, models trained with FEEs assign progressively higher probabilities to the enriched feedback option, whereas standard-trained models remain consistently biased toward the original tool output with negligible variation.

Takeaway 3. FEE-trained agents internalize the enriched information rather than treating it as a temporary inference-time hint.

5.4 Intra-group Feedback Consistency

Exp 4. We investigate whether the enriched environmental feedback should remain consistent or stochastic within a single rollout group, as feedback enrichment is typically injected with a probability

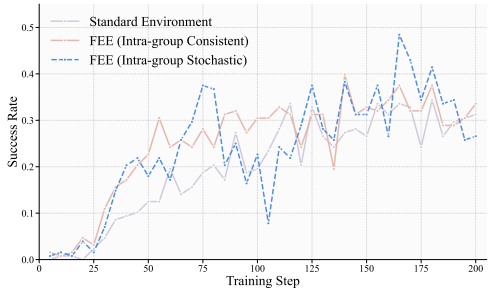

Figure 5: Validation success rates of Qwen3-4B in standard SciWorld environments during GRPO training under different feedback consistency settings.

of 0.5 during training. To study this, we compare two training configurations: a stochastic setting where different enriched feedback variants are randomly assigned within the same sampling group, and a consistent setting where all agents in a group receive identical feedback for a given state. Specifically, we conduct this experiment by training Qwen3-4B on SciWorld with GRPO for 200 training steps and monitor the validation performance in standard environments.

Res 4. The results show that intra-group feedback consistency is crucial for stable optimization. As illustrated in Figure 5, the configuration with consistent intra-group feedback exhibits a steady increase in success rate throughout training. In contrast, providing diverse feedback within a single group leads to severe instability, characterized by erratic performance fluctuations with sharp plunges and sudden spikes. This may stem from the nature of group-based agentic RL, where advantages are estimated within each sampling group. Excessive stochasticity in intra-group feedback can distort advantage estimation, leading to noisy optimization signals and unstable convergence.

Takeaway 4. Intra-group feedback consistency is a prerequisite for stable optimization, as excessive diversity within a single rollout group triggers severe performance volatility.

6 Related Work

6.1 RL for Long-horizon Tasks

Agentic language models tackle long-horizon tasks by integrating natural language reasoning with grounded actions such as tool manipulation (Yao et al., 2024; Wang et al., 2025b; Trivedi et al., 2024; Lei et al., 2025), web navigation (Yao et al., 2022; Zhou et al., 2024b; Koh et al., 2024; Rawles et al., 2025; Xie et al., 2024), and code execution (Ouyang et al., 2025; Merrill et al., 2026b; Yang et al., 2025b), requiring an extended sequence of steps to achieve an ultimate objective. To enhance the capabilities of LLMs serving as the backbone of agentic systems, reinforcement-learning-based approaches have been instrumental (Zhang et al., 2025e; Wang et al., 2025c; Feng et al., 2025b; Jin et al., 2025a; Hu et al., 2026). Typically, a supervised fine-tuning warm-up phase is conducted prior to reinforcement learning to boost the agent's initial capabilities (Li et al., 2025b; Wei et al., 2025; Lu et al., 2025; Jin et al., 2025b). However, in this paper, we introduce a complementary environment-side enrichment framework that is orthogonal to existing agent-centric optimization algorithms and refinement techniques.

6.2 Environments for Evolving Agents

As modern LLMs evolve into autonomous agents capable of sequential decision making in complex environments, training these agents within interactive, gym-style environments is becoming a standard practice (Liu et al., 2025a; Stojanovski et al., 2025; Meng et al., 2026; Aggarwal et al., 2026). Consequently, the focus of research has evolved from scaling static datasets (Wang et al., 2023; Xu et al., 2024) to scaling the complexity and diversity of interactive environments (Fang et al., 2025; Song et al., 2026; Gandhi et al., 2026; Tu et al., 2026). In the era of scaling data, to solve the reward sparsity problem, many studies focus on modulating task difficulty by incorporating linguistic hints as a form of scaffolding to facilitate model training (Li et al., 2025a; Zhang et al., 2025d;a;c; Huang et al., 2025). However, in the era of scaling environments, how to systematically reconstruct multi-turn, dynamic environments to facilitate agent evolution remains relatively under-explored.

7 Conclusion

Training long-horizon LLM agents with RL remains challenging due to sparse rewards and ineffective exploration. In this work, we shift the focus from agent-side warming to environment-side adaptation, and propose a general strategy for constructing effective feedback-enriched environments. By systematically designing what feedback to provide and when to deliver it, the resulting FEEs consistently improve RL training across multiple benchmarks, model scales, and optimization algorithms. Beyond performance gains, our findings show that FEEs stabilize training dynamics, encourage proactive state-space exploration, internalize environmental guidance into policy weights, and highlight intra-group feedback consistency as an important condition for stable optimization.

AI USE STATEMENT

We used LLM-based assistants only for grammar checking, language polishing, and assistance with LaTeX formatting. No experimental data or reported numerical results were changed during this process. The authors are responsible for the paper's technical content, claims, and conclusions.

ETHICS STATEMENT

Our experiments use publicly available tasks from SciWorld and BFCL and do not involve human participants or the collection of personal data. All agent interactions take place within benchmark environments.

REPRODUCIBILITY STATEMENT

The feedback design and enrichment schedule are described in Section 3. Section 4.1 reports the benchmarks, model sizes, training algorithms, learning rate, and evaluation schedule used in the main experiments. Appendix A gives the RL objectives, and Appendix B explains how feedback is constructed for SciWorld and BFCL, including the BFCL domain split and examples of enriched interactions. It also identifies the RLVMR and verl-agent codebases used in our implementation. For the internalization experiment, Section 5.3 describes the procedure, while Table 2 and Appendix C provide the probe questions.

REFERENCES

Pranjal Aggarwal, Graham Neubig, and Sean Welleck. Gym-anything: Turn any software into an agent environment, 2026. URL https://arxiv.org/abs/2604.06126.

Hao Bai, Alexey Taymanov, Tong Zhang, Aviral Kumar, and Spencer Whitehead. Webgym: Scaling training environments for visual web agents with realistic tasks. *CoRR*, abs/2601.02439, 2026. doi: 10.48550/ARXIV.2601.02439. URL https://doi.org/10.48550/arXiv.2601.02439.

Kevin Chen, Marco F. Cusumano-Towner, Brody Huval, Aleksei Petrenko, Jackson Hamburger, Vladlen Koltun, and Philipp Krähenbühl. Reinforcement learning for long-horizon interactive LLM agents. *CoRR*, abs/2502.01600, 2025a. doi: 10.48550/ARXIV.2502.01600. URL https://doi.org/10.48550/arXiv.2502.01600.

Zhaorun Chen, Zhuokai Zhao, Kai Zhang, Bo Liu, Qi Qi, Yifan Wu, Tarun Kalluri, Sara Cao, Yuanhao Xiong, Haibo Tong, Huaxiu Yao, Hengduo Li, Jiacheng Zhu, Xian Li, Dawn Song, Bo Li, Jason Weston, and Dat Huynh. Scaling agent learning via experience synthesis. *CoRR*, abs/2511.03773, 2025b. doi: 10.48550/ARXIV.2511.03773. URL https://doi.org/10.48550/arXiv.2511.03773.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, and S. S. Li. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025. doi: 10.48550/ARXIV.2501.12948. URL https://doi.org/10.48550/arXiv.2501.12948.

Wei Du, Shubham Toshniwal, Branislav Kisacanin, Sadegh Mahdavi, Ivan Moshkov, George Armstrong, Stephen Ge, Edgar Minasyan, Feng Chen, and Igor Gitman. Nemotron-math: Efficient long-context distillation of mathematical reasoning from multi-mode supervision. *CoRR*, abs/2512.15489, 2025. doi: 10.48550/ARXIV.2512.15489. URL https://doi.org/10.48550/arXiv.2512.15489.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. The llama 3 herd

of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Runnan Fang, Shihao Cai, Baixuan Li, Jialong Wu, Guangyu Li, Wenbiao Yin, Xinyu Wang, Xiaobin Wang, Liangcai Su, Zhen Zhang, Shibin Wu, Zhengwei Tao, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Towards general agentic intelligence via environment scaling. *CoRR*, abs/2509.13311, 2025. doi: 10.48550/ARXIV.2509.13311. URL https://doi.org/10.48550/arXiv.2509.13311.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for LLM agent training. *CoRR*, abs/2505.10978, 2025a. doi: 10.48550/ARXIV.2505.10978. URL https://doi.org/10.48550/arXiv.2505.10978.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for LLM agent training. *CoRR*, abs/2505.10978, 2025b. doi: 10.48550/ARXIV.2505.10978. URL https://doi.org/10.48550/arXiv.2505.10978.

Kanishk Gandhi, Shivam Garg, Noah D. Goodman, and Dimitris Papailiopoulos. Endless terminals: Scaling RL environments for terminal agents. *CoRR*, abs/2601.16443, 2026. doi: 10.48550/ARXIV.2601.16443. URL https://doi.org/10.48550/arXiv.2601.16443.

Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, Olli Järviniemi, Matthew Barnett, Robert Sandler, Matej Vrzala, Jaime Sevilla, Qiuyu Ren, Elizabeth Pratt, Lionel Levine, Grant Barkley, Natalie Stewart, Bogdan Grechuk, Tetiana Grechuk, Shreepranav Varma Enugandla, and Mark Wildon. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in AI. *CoRR*, abs/2411.04872, 2024. doi: 10.48550/ARXIV.2411.04872. URL https://doi.org/10.48550/arXiv.2411.04872.

Tianyi Hu, Qingxu Fu, Yanxi Chen, Zhaoyang Liu, and Bolin Ding. Seeupo: Sequence-level agentic-rl with convergence guarantees. *CoRR*, abs/2602.06554, 2026. doi: 10.48550/ARXIV.2602.06554. URL https://doi.org/10.48550/arXiv.2602.06554.

Qihan Huang, Long Chan, Jinlong Liu, Wanggui He, Hao Jiang, Mingli Song, Jingyuan Chen, Chang Yao, and Jie Song. Boosting MLLM reasoning with text-debiased hint-grpo. *CoRR*, abs/2503.23905, 2025. doi: 10.48550/ARXIV.2503.23905. URL https://doi.org/10.48550/arXiv.2503.23905.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik 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. URL https://openreview.net/forum?id=VTF8yNQM66.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *CoRR*, abs/2503.09516, 2025a. doi: 10.48550/ARXIV.2503.09516. URL https://doi.org/10.48550/arXiv.2503.09516.

Hongbo Jin, Qingyuan Wang, Wenhao Zhang, Yang Liu, and Sijie Cheng. Videomem: Enhancing ultra-long video understanding via adaptive memory management. *CoRR*, abs/2512.04540, 2025b. doi: 10.48550/ARXIV.2512.04540. URL https://doi.org/10.48550/arXiv.2512.04540.

Feiyang Kang, Michael Kuchnik, Karthik Padthe, Marin Vlastelica, Ruoxi Jia, Carole-Jean Wu, and Newsha Ardalani. Quagmires in SFT-RL post-training: When high SFT scores mislead and what to use instead. *CoRR*, abs/2510.01624, 2025. doi: 10.48550/ARXIV.2510.01624. URL https://doi.org/10.48550/arXiv.2510.01624.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. In Lun-Wei Ku, Andre Martins, and Vivek

Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 881–905. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.50. URL https://doi.org/10.18653/v1/2024.acl-long.50.

Fei Lei, Yibo Yang, Wenxiu Sun, and Dahua Lin. Mcpverse: An expansive, real-world benchmark for agentic tool use. *CoRR*, abs/2508.16260, 2025. doi: 10.48550/ARXIV.2508.16260. URL https://doi.org/10.48550/arXiv.2508.16260.

Jiazheng Li, Hong Lu, Kaiyue Wen, Zaiwen Yang, Jiaxuan Gao, Hongzhou Lin, Yi Wu, and Jingzhao Zhang. Questa: Expanding reasoning capacity in llms via question augmentation. *CoRR*, abs/2507.13266, 2025a. doi: 10.48550/ARXIV.2507.13266. URL https://doi.org/10.48550/arXiv.2507.13266.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, Weizhou Shen, Junkai Zhang, Dingchu Zhang, Xixi Wu, Yong Jiang, Ming Yan, Pengjun Xie, Fei Huang, and Jingren Zhou. Websailor: Navigating super-human reasoning for web agent. *CoRR*, abs/2507.02592, 2025b. doi: 10.48550/ARXIV.2507.02592. URL https://doi.org/10.48550/arXiv.2507.02592.

Zichen Liu, Anya Sims, Keyu Duan, Changyu Chen, Simon Yu, Xiangxin Zhou, Haotian Xu, Shaopan Xiong, Bo Liu, Chenmien Tan, Chuen Yang Beh, Weixun Wang, Hao Zhu, Weiyan Shi, Diyi Yang, Michael Shieh, Yee Whye Teh, Wee Sun Lee, and Min Lin. GEM: A gym for agentic llms. *CoRR*, abs/2510.01051, 2025a. doi: 10.48550/ARXIV.2510.01051. URL https://doi.org/10.48550/arXiv.2510.01051.

Zihan Liu, Zhuolin Yang, Yang Chen, Chankyu Lee, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Acereason-nemotron 1.1: Advancing math and code reasoning through SFT and RL synergy. *CoRR*, abs/2506.13284, 2025b. doi: 10.48550/ARXIV.2506.13284. URL https://doi.org/10.48550/arXiv.2506.13284.

Zhengxi Lu, Jiabo Ye, Fei Tang, Yongliang Shen, Haiyang Xu, Ziwei Zheng, Weiming Lu, Ming-Hsuan Yang, Fei Huang, Jun Xiao, and Yueting Zhuang. UI-S1: advancing GUI automation via semi-online reinforcement learning. *CoRR*, abs/2509.11543, 2025. doi: 10.48550/ARXIV.2509.11543. URL https://doi.org/10.48550/arXiv.2509.11543.

Fanqing Meng, Lingxiao Du, Jiawei Gu, Jiaqi Liao, Linjie Li, Zijian Wu, Xiangyan Liu, Ziqi Zhao, Mengkang Hu, Zichen Liu, Jiaheng Zhang, and Michael Qizhe Shieh. Gym-v: A unified vision environment system for agentic vision research, 2026. URL https://arxiv.org/abs/2603.15432.

Mike A. Merrill, Alexander Glenn Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Yeon Shin, Thomas Walshe, Estefany Kelly Buchanan, Junhong Shen, Guanghao Ye, Haowei Lin, Jason Poulos, Maoyu Wang, Marianna Nezhurina, Jenia Jitsev, Di Lu, Orfeas Menis-Mastromichalakis, Zhiwei Xu, Zizhao Chen, Yue Liu, Robert Zhang, Leon Liangyu Chen, Anurag Kashyap, Jan-Lucas Uslu, Jeffrey Li, Jianbo Wu, Minghao Yan, Song Bian, Vedang Sharma, Ke Sun, Steven Dillmann, Akshay Anand, Andrew Lanpouthakoun, Bardia Koopah, Changran Hu, Etash Kumar Guha, Gabriel H. S. Dreiman, Jiacheng Zhu, Karl Krauth, Li Zhong, Niklas Muennighoff, Robert Amanfu, Shangyin Tan, Shreyas Pimpalgaonkar, Tushar Aggarwal, Xiangning Lin, Xin Lan, Xuandong Zhao, Yiqing Liang, Yuanli Wang, Zilong Wang, Changzhi Zhou, David Heineman, Hange Liu, Harsh Trivedi, John Yang, Junhong Lin, Manish Shetty, Michael Yang, Nabil Omi, Negin Raoof, Shanda Li, Terry Yue Zhuo, Wuwei Lin, Yiwei Dai, Yuxin Wang, Wenhao Chai, Shang Zhou, Dariush Wahdany, Ziyu She, Jiaming Hu, Zhikang Dong, Yuxuan Zhu, Sasha Cui, Ahson Saiyed, Arinbjörn Kolbeinsson, Jesse Hu, Christopher Michael Rytting, Ryan Marten, Yixin Wang, Alex Dimakis, Andy Konwinski, and Ludwig Schmidt. Terminal-bench: Benchmarking agents on hard, realistic tasks in command line interfaces. *CoRR*, abs/2601.11868, 2026a. doi: 10.48550/ARXIV.2601.11868. URL https://doi.org/10.48550/arXiv.2601.11868.

Mike A. Merrill, Alexander Glenn Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Yeon Shin, Thomas Walshe, Estefany Kelly Buchanan, Junhong Shen, Guanghao Ye, Haowei Lin, Jason Poulos, Maoyu Wang, Marianna Nezhurina, Jenia Jitsev, Di Lu, Orfeas

Menis-Mastromichalakis, Zhiwei Xu, Zizhao Chen, Yue Liu, Robert Zhang, Leon Liangyu Chen, Anurag Kashyap, Jan-Lucas Uslu, Jeffrey Li, Jianbo Wu, Minghao Yan, Song Bian, Vedang Sharma, Ke Sun, Steven Dillmann, Akshay Anand, Andrew Lanpouthakoun, Bardia Koopah, Changran Hu, Etash Kumar Guha, Gabriel H. S. Dreiman, Jiacheng Zhu, Karl Krauth, Li Zhong, Niklas Muennighoff, Robert Amanfu, Shangyin Tan, Shreyas Pimpalgaonkar, Tushar Aggarwal, Xiangning Lin, Xin Lan, Xuandong Zhao, Yiqing Liang, Yuanli Wang, Zilong Wang, Changzhi Zhou, David Heineman, Hange Liu, Harsh Trivedi, John Yang, Junhong Lin, Manish Shetty, Michael Yang, Nabil Omi, Negin Raoof, Shanda Li, Terry Yue Zhuo, Wuwei Lin, Yiwei Dai, Yuxin Wang, Wenhao Chai, Shang Zhou, Dariush Wahdany, Ziyu She, Jiaming Hu, Zhikang Dong, Yuxuan Zhu, Sasha Cui, Ahson Saiyed, Arinbjörn Kolbeinsson, Jesse Hu, Christopher Michael Rytting, Ryan Marten, Yixin Wang, Alex Dimakis, Andy Konwinski, and Ludwig Schmidt. Terminal-bench: Benchmarking agents on hard, realistic tasks in command line interfaces. *CoRR*, abs/2601.11868, 2026b. doi: 10.48550/ARXIV.2601.11868. URL https://doi.org/10.48550/arXiv.2601.11868.

OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/ARXIV.2303.08774. URL https://doi.org/10.48550/arXiv.2303.08774.

Anne Ouyang, Simon Guo, Simran Arora, Alex L. Zhang, William Hu, Christopher Ré, and Azalia Mirhoseini. Kernelbench: Can llms write efficient GPU kernels? In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/ouyang25a.html.

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (BFCL): from tool use to agentic evaluation of large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/patil25a.html.

Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William E. Bishop, Wei Li, Folawiyo Campbell-Ajala, Daniel Kenji Toyama, Robert James Berry, Divya Tyamagundlu, Timothy P. Lillicrap, and Oriana 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. URL https://openreview.net/forum?id=il5yUQsrjC.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. doi: 10.48550/ARXIV.2402.03300. URL https://doi.org/10.48550/arXiv.2402.03300.

Xiaoshuai Song, Haofei Chang, Guanting Dong, Yutao Zhu, Zhicheng Dou, and Ji-Rong Wen. Envscaler: Scaling tool-interactive environments for LLM agent via programmatic synthesis. *CoRR*, abs/2601.05808, 2026. doi: 10.48550/ARXIV.2601.05808. URL https://doi.org/10.48550/arXiv.2601.05808.

Zafir Stojanovski, Oliver Stanley, Joe Sharratt, Richard Jones, Abdulhakeem Adefioye, Jean Kaddour, and Andreas Köpf. REASONING GYM: reasoning environments for reinforcement learning with verifiable rewards. *CoRR*, abs/2505.24760, 2025. doi: 10.48550/ARXIV.2505.24760. URL https://doi.org/10.48550/arXiv.2505.24760.

Kimi Team. Kimi K2: open agentic intelligence. *CoRR*, abs/2507.20534, 2025. doi: 10.48550/ARXIV.2507.20534. URL https://doi.org/10.48550/arXiv.2507.20534.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. Appworld: A controllable world of apps and people for benchmarking interactive coding agents. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024, pp. 16022–16076. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.850. URL https://doi.org/10.18653/v1/2024.acl-long.850.

Dunwei Tu, Hongyan Hao, Hansi Yang, Yihao Chen, Yi-Kai Zhang, Zhikang Xia, Yu Yang, Yueqing Sun, Xingchen Liu, Furao Shen, Qi Gu, Hui Su, and Xunliang Cai. Scaleenv: Scaling environment synthesis from scratch for generalist interactive tool-use agent training. *CoRR*, abs/2602.06820, 2026. doi: 10.48550/ARXIV.2602.06820. URL https://doi.org/10.48550/arXiv.2602.06820.

Jiawei Wang, Jiacai Liu, Yuqian Fu, Yingru Li, Xintao Wang, Yuan Lin, Yu Yue, Lin Zhang, Yang Wang, and Ke Wang. Harnessing uncertainty: Entropy-modulated policy gradients for long-horizon LLM agents. *CoRR*, abs/2509.09265, 2025a. doi: 10.48550/ARXIV.2509.09265. URL https://doi.org/10.48550/arXiv.2509.09265.

Ruoyao Wang, Peter A. Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Scienceworld: Is your agent smarter than a 5th grader? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pp. 11279–11298. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.775. URL https://doi.org/10.18653/v1/2022.emnlp-main.775.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 13484–13508. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.754. URL https://doi.org/10.18653/v1/2023.acl-long.754.

Zhenting Wang, Qi Chang, Hemani Patel, Shashank Biju, Cheng-En Wu, Quan Liu, Aolin Ding, Alireza Rezazadeh, Ankit Shah, Yujia Bao, and Eugene Siow. Mcp-bench: Benchmarking tool-using LLM agents with complex real-world tasks via MCP servers. *CoRR*, abs/2508.20453, 2025b. doi: 10.48550/ARXIV.2508.20453. URL https://doi.org/10.48550/arXiv.2508.20453.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. RAGEN: understanding self-evolution in LLM agents via multi-turn reinforcement learning. *CoRR*, abs/2504.20073, 2025c. doi: 10.48550/ARXIV.2504.20073. URL https://doi.org/10.48550/arXiv.2504.20073.

Zhepei Wei, Wenlin Yao, Yao Liu, Weizhi Zhang, Qin Lu, Liang Qiu, Changlong Yu, Puyang Xu, Chao Zhang, Bing Yin, Hyokun Yun, and Lihong Li. Webagent-r1: Training web agents via end-to-end multi-turn reinforcement learning. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 7909–7928. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.EMNLP-MAIN.401. URL https://doi.org/10.18653/v1/2025.emnlp-main.401.

Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, Haosheng Zou, Yongchao Deng, Shousheng Jia, and Xiangzheng Zhang. Light-r1: Curriculum sft, DPO and RL for long COT from scratch and beyond. In Georg Rehm and Yunyao Li (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 318–327. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.ACL-INDUSTRY.24. URL https://doi.org/10.18653/v1/2025.acl-industry.24.

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

Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/5d413e48f84dc61244b6be550f1cd8f5-Abstract-Datasets_and_Benchmarks_Track.html.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=CfXh93NDgH.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jian Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025a. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/arXiv.2505.09388.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/5a7c947568c1b1328ccc5230172e1e7c-Abstract-Conference.html.

John Yang, Carlos E. Jimenez, Alex L. Zhang, Kilian Lieret, Joyce Yang, Xindi Wu, Ori Press, Niklas Muennighoff, Gabriel Synnaeve, Karthik R. Narasimhan, Diyi Yang, Sida Wang, and Ofir Press. Swe-bench multimodal: Do AI systems generalize to visual software domains? In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025b. URL https://openreview.net/forum?id=riTiq3i21b.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *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*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/82ad13ec01f9fe44c01cb91814fd7b8c-Abstract-Conference.html.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *CoRR*, abs/2406.12045, 2024. doi: 10.48550/ARXIV.2406.12045. URL https://doi.org/10.48550/arXiv.2406.12045.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Weinan Dai, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: an open-source LLM reinforcement learning system at scale. *CoRR*, abs/2503.14476, 2025. doi: 10.48550/ARXIV.2503.14476. URL https://doi.org/10.48550/arXiv.2503.14476.

Feng Zhang, Zezhong Tan, Xinhong Ma, Ziqiang Dong, Xi Leng, Jianfei Zhao, Xin Sun, and Yang Yang. Adhint: Adaptive hints with difficulty priors for reinforcement learning. *CoRR*, abs/2512.13095, 2025a. doi: 10.48550/ARXIV.2512.13095. URL https://doi.org/10.48550/arXiv.2512.13095.

Kai Zhang, Xiangchao Chen, Bo Liu, Tianci Xue, Zeyi Liao, Zhihan Liu, Xiyao Wang, Yuting Ning, Zhaorun Chen, Xiaohan Fu, Jian Xie, Yuxuan Sun, Boyu Gou, Qi Qi, Zihang Meng, Jianwei Yang, Ning Zhang, Xian Li, Ashish Shah, Dat Huynh, Hengduo Li, Zi Yang, Sara Cao, Lawrence Jang, Shuyan Zhou, Jiacheng Zhu, Huan Sun, Jason Weston, Yu Su, and Yifan Wu. Agent learning via early experience. *CoRR*, abs/2510.08558, 2025b. doi: 10.48550/ARXIV.2510.08558. URL https://doi.org/10.48550/arXiv.2510.08558.

Kaiyi Zhang, Ang Lv, Jinpeng Li, Yongbo Wang, Feng Wang, Haoyuan Hu, and Rui Yan. Stephint: Multi-level stepwise hints enhance reinforcement learning to reason. *CoRR*, abs/2507.02841, 2025c. doi: 10.48550/ARXIV.2507.02841. URL https://doi.org/10.48550/arXiv.2507.02841.

Xichen Zhang, Sitong Wu, Yinghao Zhu, Haoru Tan, Shaozuo Yu, Ziyi He, and Jiaya Jia. Scaf-grpo: Scaffolded group relative policy optimization for enhancing LLM reasoning. *CoRR*, abs/2510.19807, 2025d. doi: 10.48550/ARXIV.2510.19807. URL https://doi.org/10.48550/arXiv.2510.19807.

Zijing Zhang, Ziyang Chen, Mingxiao Li, Zhaopeng Tu, and Xiaolong Li. RLVMR: reinforcement learning with verifiable meta-reasoning rewards for robust long-horizon agents. *CoRR*, abs/2507.22844, 2025e. doi: 10.48550/ARXIV.2507.22844. URL https://doi.org/10.48550/arXiv.2507.22844.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization. *CoRR*, abs/2507.18071, 2025. doi: 10.48550/ARXIV.2507.18071. URL https://doi.org/10.48550/arXiv.2507.18071.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham 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, 2024a. URL https://openreview.net/forum?id=oKn9c6ytLx.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham 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, 2024b. URL https://openreview.net/forum?id=oKn9c6ytLx.

A RL ALGORITHMS

In this section, we elaborate on the specific formulations and optimization objectives of the RL algorithms. We follow the notations in Section 2: for each goal g, the old policy $\pi_{\theta_{\text{old}}}$ samples a group of N trajectories $\{\tau_k\}_{k=1}^{N}$, and each trajectory receives a trajectory-level reward $r(\tau_k)$. Let $\mathbf{a}_k = (a_{k,1}, \ldots, a_{k,L_k})$ denote the concatenation of all agent-generated tokens in τ_k, where L_k is the number of such tokens. The conditioning context for token $a_{k,t}$, including the goal, the observations, and the previous interaction history, is denoted by $c_{k,t}$. Environment-generated tokens are excluded from all policy-gradient sums below. Unless otherwise specified, the expectations are taken over goals and trajectory groups sampled from $\pi_{\theta_{\text{old}}}$.

The group-normalized trajectory advantage is computed as

$$A_k = A(\tau_k) = \frac{r(\tau_k) - \text{mean}\left(\{r(\tau_j)\}_{j=1}^{N}\right)}{\text{std}\left(\{r(\tau_j)\}_{j=1}^{N}\right)}, \tag{4}$$

and is assigned to every agent-generated token in the same trajectory, i.e., $A_{k,t} = A_k$.

GRPO. Group Relative Policy Optimization (GRPO) uses token-level importance ratios while estimating advantages from the relative rewards within the sampled group. Its clipped objective can be written as

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}\left[\frac{1}{N}\sum_{k=1}^{N}\frac{1}{L_k}\sum_{t=1}^{L_k}\min\left(\rho_{k,t}(\theta)A_k, \text{clip}\big(\rho_{k,t}(\theta), 1-\varepsilon, 1+\varepsilon\big)A_k\right)\right], \tag{5}$$

where

$$\rho_{k,t}(\theta) = \frac{\pi_\theta(a_{k,t} \mid c_{k,t})}{\pi_{\theta_{\text{old}}}(a_{k,t} \mid c_{k,t})}. \tag{6}$$

Thus, GRPO performs clipping and policy-gradient weighting independently for each generated token, while the reward signal remains trajectory-level.

DAPO. Decoupled Clip and Dynamic Sampling Policy Optimization (DAPO) preserves the group-relative advantage estimation but modifies the GRPO reduction and clipping rule. In particular, it uses a token-level loss reduction over all agent-generated tokens in the group and decouples the lower and upper clipping ranges:

$$\mathcal{J}_{\text{DAPO}}(\theta) = \mathbb{E}\left[\frac{1}{\sum_{k=1}^{N} L_k}\sum_{k=1}^{N}\sum_{t=1}^{L_k}\min\left(\rho_{k,t}(\theta)A_k, \text{clip}\big(\rho_{k,t}(\theta), 1-\varepsilon_{\text{low}}, 1+\varepsilon_{\text{high}}\big)A_k\right)\right], \tag{7}$$

with the same token-level ratio $\rho_{k,t}(\theta)$ and group-normalized advantage A_k as above. Dynamic sampling further keeps only non-degenerate trajectory groups. For binary success rewards, this condition is:

$$0 < |\{\tau_k \mid r(\tau_k) = 1\}| < N, \tag{8}$$

or, more generally, groups whose rewards have non-zero variance. This filtering avoids batches in which all trajectories receive identical rewards and therefore produce zero normalized advantages.

GSPO. Group Sequence Policy Optimization (GSPO) instead aligns the optimization unit with the trajectory-level reward by defining the importance ratio at the sequence level. For each trajectory, the length-normalized sequence ratio over agent-generated tokens is defined as the following form:

$$s_k(\theta) = \left(\prod_{t=1}^{L_k}\frac{\pi_\theta(a_{k,t} \mid c_{k,t})}{\pi_{\theta_{\text{old}}}(a_{k,t} \mid c_{k,t})}\right)^{\frac{1}{L_k}} = \exp\left(\frac{1}{L_k}\sum_{t=1}^{L_k}\log\frac{\pi_\theta(a_{k,t} \mid c_{k,t})}{\pi_{\theta_{\text{old}}}(a_{k,t} \mid c_{k,t})}\right), \tag{9}$$

GSPO optimizes the following objective:

$$\mathcal{J}_{\text{GSPO}}(\theta) = \mathbb{E}\left[\frac{1}{N}\sum_{k=1}^{N}\min\left(s_k(\theta)A_k, \text{clip}\big(s_k(\theta), 1-\varepsilon, 1+\varepsilon\big)A_k\right)\right]. \tag{10}$$

The clipping decision is therefore made once per trajectory rather than once per token, while all agent-generated tokens in the trajectory share the same sequence-level weight and advantage.

B Enrichment Strategies

In this section, we detail the construction of enriched environments based on two standard environments, SciWorld and BFCL, following the given enrichment strategy. For each environment, we first introduce the basic rule-based implementation and then provide concrete reference examples.

B.1 Enriched Environments for SciWorld

For SciWorld, the implementation of action guidance leverages the ground-truth expert trajectories provided for each task. For observation enrichment we utilize the task progress tracking logic available in the SciWorld source code. SciWorld internally maintains rule-based checks for evaluating task completion conditions and intermediate progress. Based on these signals, we enrich the environment observations with supplementary state information reflecting the current task status, enabling the agent to better perceive hidden environment dynamics and long-horizon progress. Representative examples of the enriched environments from SciWorld are shown in Table 3 and Table 4. The implementation is based on RLVMR [1].

B.2 Enriched Environments for BFCL-V3

For BFCL-V3, we construct enriched environments across four domains: *Vehicle Control*, *Trading Bots*, *Travel Booking*, and *Gorilla File System*. We manually designate Gorilla File System and Vehicle Control for training, while reserving Trading Bots and Travel Booking for held-out evaluation. The action guidance in BFCL is implemented by appending lightweight hints after each user query. For observation enrichment, we augment the outputs of selected tools with additional execution information. Concretely, rather than returning sparse or empty responses, enriched feedback may indicate whether a tool invocation has successfully completed, provide intermediate execution status, or expose supplementary contextual information relevant to the current interaction state. For example, in Gorilla File System, enriched feedback can include additional path-related information, while in Vehicle Control, tool outputs may be extended with status indicators or auxiliary environment details. Representative examples of BFCL enrichments are shown in Table 5, Table 6 and Table 7. The implementation is based on verl-agent [2].

C Internalization Experiments

In this section, we provide the probe questions constructed for the Gorilla File System operations used in Exp 3. The complete examples corresponding to `cd`, `mkdir`, `touch`, `rm`, `rmdir`, `mv`, `cp`, and `echo` are presented in Tables 2 and 8. For each probe question, the model predicts between two candidate outputs: the original feedback option $P(A)$ and the enriched feedback option $P(B)$. A higher $P(A)$ indicates that the model remains biased toward standard tool outputs and shows limited memory of the enriched feedback, whereas a higher $P(B)$ suggests that the model recalls and internalizes the information introduced by the enriched feedback.

Table 3: Action Guidance in SciWorld

You are an expert agent operating in the ScienceWorld environment, which is a text-based virtual environment centered around accomplishing tasks from the elementary science curriculum.

Your current task is : Your task is to determine whether round seed shape is a dominant or recessive trait in the unknown E plant. If the trait is dominant, focus on the green box. If the trait is recessive, focus on the blue box.

Your current observation is : This room is called the greenhouse. In it, you see: the agent a substance called air a bee hive. The bee hive door is closed. a blue box (containing nothing) a flower pot 3 (contain-

[1] https://github.com/Tencent/DigitalHuman/tree/main/RLVMR

[2] https://github.com/langfengQ/verl-agent

ing soil) a flower pot 4 (containing soil) a flower pot 5 (containing soil) a flower pot 6 (containing soil) a flower pot 8 (containing soil) a flower pot 9 (containing soil) a green box (containing nothing) a jug (containing nothing) a seed jar (containing a square blue unknown E seed, a round blue unknown E seed) a sink, which is turned off. In the sink is: nothing. You also see: A door to the hallway (that is open) A door to the outside (that is open) Here are the actions you may take:

[

"action": "open OBJ", "description": "open a container",

"action": "close OBJ", "description": "close a container",

"action": "activate OBJ", "description": "activate a device",

"action": "deactivate OBJ", "description": "deactivate a device",

"action": "connect OBJ to OBJ", "description": "connect electrical components",

"action": "disconnect OBJ", "description": "disconnect electrical components",

"action": "use OBJ [on OBJ]", "description": "use a device/item",

"action": "look around", "description": "describe the current room",

"action": "look at OBJ", "description": "describe an object in detail",

"action": "look in OBJ", "description": "describe a container's contents",

"action": "read OBJ", "description": "read a note or book",

"action": "move OBJ to OBJ", "description": "move an object to a container",

"action": "pick up OBJ", "description": "move an object to the inventory",

"action": "put down OBJ", "description": "drop an inventory item",

"action": "pour OBJ into OBJ", "description": "pour a liquid into a container",

"action": "dunk OBJ into OBJ", "description": "dunk a container into a liquid",

"action": "mix OBJ", "description": "chemically mix a container",

"action": "go to LOC", "description": "move to a new location",

"action": "eat OBJ", "description": "eat a food",

"action": "flush OBJ", "description": "flush a toilet",

"action": "focus on OBJ", "description": "signal intent on a task object",

"action": "wait", "description": "take no action for 10 iterations",

"action": "wait1", "description": "take no action for 1 iteration",

"action": "task", "description": "describe current task",

"action": "inventory", "description": "list your inventory"

]

Current available actions :

Valid_actions: ['activate OBJ', 'close OBJ', 'deactivate OBJ', 'dunk OBJ in OBJ', 'eat OBJ', 'flush OBJ', 'focus on OBJ', 'go OBJ', 'inventory', 'look around', 'look at OBJ', 'look in OBJ', 'mix OBJ', 'move OBJ to OBJ', 'open OBJ', 'pick up OBJ', 'pour OBJ in OBJ', 'put down OBJ', 'read OBJ', 'reset task', 'task', 'teleport OBJ', 'use OBJ on OBJ', 'wait', 'wait1'], OBJ needs to be replaced with one of the following objects: ['agent', 'air', 'bee hive', 'blue box', 'ceramic cup', 'door to hallway', 'door to outside', 'flower pot 3', 'flower pot 4', 'flower pot 5', 'flower pot 6', 'flower pot 8', 'flower pot 9', 'green box', 'greenhouse', 'hallway', 'jug', 'outside', 'round blue unknown e seed', 'seed square blue unknown e seed', 'sink', 'soil in flower pot 3', 'soil in flower pot 4', 'soil in flower pot 5', 'soil in flower pot 6', 'soil in flower pot 8', 'soil in flower pot 9']

example: <action>focus on door</action>

Now it's your turn to take an action.

You should first reason step-by-step about the current situation. This reasoning process MUST be enclosed within <think> </think> tags.

Once you've finished your reasoning, you should choose an appropriate action for the current step and present it within <action> </action> tags.

[Hint] Start with the following action to start your exploration!

['look around', 'pick up seed jar', 'look around']

Table 4: Observation Enrichment in SciWorld

You are an expert agent operating in the ScienceWorld environment, which is a text-based virtual environment centered around accomplishing tasks from the elementary science curriculum.

Your current task is : Your task is to find a(n) non-living thing. First, focus on the thing. Then, move it to the yellow box in the living room.

Prior to this step, you have already taken 6 step(s).

Below are the most recent 2 observations and the corresponding actions you took:

[Step 1, Action 1: 'go to living room']

[Step 2, Action 2: 'focus on yellow box']

[Step 3, Action 3: 'pick up non-living thing']

[Step 4, Action 4: 'pick up object']

[Step 5, Observation 5: 'You move the pillow to the inventory.', Action 5: 'move pillow to yellow box']

[Step 6, Observation 6: 'You move the pillow to the yellow box.', Action 6: 'reset task']

You are now at step 7 and your current observation is:

You reset the goal progress and focus.

Here are the actions you may take:

[

"action": "open OBJ", "description": "open a container",

"action": "close OBJ", "description": "close a container",

"action": "activate OBJ", "description": "activate a device",

"action": "deactivate OBJ", "description": "deactivate a device",

"action": "connect OBJ to OBJ", "description": "connect electrical components",

"action": "disconnect OBJ", "description": "disconnect electrical components",

"action": "use OBJ [on OBJ]", "description": "use a device/item",

"action": "look around", "description": "describe the current room",

"action": "look at OBJ", "description": "describe an object in detail",

"action": "look in OBJ", "description": "describe a container's contents",

"action": "read OBJ", "description": "read a note or book",

"action": "move OBJ to OBJ", "description": "move an object to a container",

"action": "pick up OBJ", "description": "move an object to the inventory",

"action": "put down OBJ", "description": "drop an inventory item",

"action": "pour OBJ into OBJ", "description": "pour a liquid into a container",

"action": "dunk OBJ into OBJ", "description": "dunk a container into a liquid",

"action": "mix OBJ", "description": "chemically mix a container",

"action": "go to LOC", "description": "move to a new location",

"action": "eat OBJ", "description": "eat a food",

"action": "flush OBJ", "description": "flush a toilet",

"action": "focus on OBJ", "description": "signal intent on a task object",

"action": "wait", "description": "take no action for 10 iterations",

"action": "wait1", "description": "take no action for 1 iteration",

"action": "task", "description": "describe current task",

"action": "inventory", "description": "list your inventory"

]

Current available actions:

Valid_actions: ['activate OBJ', 'close OBJ', 'deactivate OBJ', 'dunk OBJ in OBJ', 'eat OBJ', 'flush OBJ', 'focus on OBJ', 'go OBJ', 'inventory', 'look around', 'look at OBJ', 'look in OBJ', 'mix OBJ', 'move OBJ to OBJ', 'open OBJ', 'pick up OBJ', 'pour OBJ in OBJ', 'put down OBJ', 'read OBJ', 'reset

task', 'task', 'teleport OBJ', 'use OBJ on OBJ', 'wait', 'wait1'], OBJ needs to be replaced with one of the following objects: ['agent', 'air', 'book shelf', 'chair', 'cloth sittable', 'door', 'hallway', 'living room', 'object', 'painting', 'steel table', 'yellow box'] example: <action>focus on door</action>

Now it's your turn to take an action.

You should first reason step-by-step about the current situation. This reasoning process MUST be enclosed within <think> </think> tags.

Once you've finished your reasoning, you should choose an appropriate action for the current step and present it within <action> </action> tags.

[Hint] Considering The current goal progress is Completed keys:

```
Sequential Subgoals:
0    false                  GoalFocusOnNonlivingThing    focus on a
non-living thing
1    false                     GoalObjectInContainer    move living
thing to answer box
Unordered and Optional Subgoals:
0    false                     GoalObjectInContainer    Pick up
object
1    true                    GoalInRoomWithOpenDoor    Be in a room
with an open door
2    true                     GoalMoveToNewLocation    Move to a new
location
3    true                        GoalMoveToLocation    Move to the
location of the answer box
```

Table 5: Action Guidance in BFCL-V3

[System] You are an expert in composing functions.You are given a question and a set of possible functions. Based on the question, you will need to make one or more function/tool calls to achieve the purpose. If none of the functions can be used, point it out. If the given question lacks the parameters required by the function, also point it out.

You should only return the function calls in your response.

If you decide to invoke any of the function(s), you MUST put it in the format of [func_name1(params_name1=params_value1, params_name2=params_value2...), func_name2(params)]. You SHOULD NOT include any other text in the response.

At each turn, you should try your best to complete the tasks requested by the user within the current turn. Continue to output functions to call until you have fulfilled the user's request to the best of your ability. Once you have no more functions to call, the system will consider the current turn complete and proceed to the next turn or task.

Here is a list of functions in json format that you can invoke.

.....

[User] Find a file named 'config.py' somewhere deep in the file system and once you have located it, display the last line of the first occurring file.

Hint: considering start with the following operation: cd(folder='projects')

[Assistant] [cd(folder='projects'), find(path='.', name='config.py'), tail(file_name='config.py', lines=1)]

[Tool] {"current_working_directory": "projects"}

[Tool] {"matches": ["./deep_folder/config.py", "./deep_folder/real_config.py"]}

[Tool] {"error": "tail: config.py: No such file or directory"}

Table 6: Action Guidance in BFCL-V3

[System] You are an expert in composing functions.You are given a question and a set of possible functions. Based on the question, you will need to make one or more function/tool calls to achieve the purpose. If none of the functions can be used, point it out. If the given question lacks the parameters required by the function, also point it out.

You should only return the function calls in your response.

If you decide to invoke any of the function(s), you MUST put it in the format of [func_name1(params_name1=params_value1, params_name2=params_value2...), func_name2(params)]. You SHOULD NOT include any other text in the response.

At each turn, you should try your best to complete the tasks requested by the user within the current turn. Continue to output functions to call until you have fulfilled the user's request to the best of your ability. Once you have no more functions to call, the system will consider the current turn complete and proceed to the next turn or task.

Here is a list of functions in json format that you can invoke.

.....

[User]I need you to set up a fresh folder wherever you're currently working.

Hint: Essential information is missing from the user's request. Do not make any assumptions; explicitly ask the user to clarify the missing parameters.

[Assistant] The user's request is missing the name of the folder to create. Please provide the name of the folder.

[User] The fresh folder should name 'WebDevProjects'.

[Assistant] [mkdir(dir_name='WebDevProjects')]

[Tool] None

[Assistant] The directory 'WebDevProjects' has been successfully created. Is there anything else you need assistance with?

Table 7: Observation Enrichment in BFCL-V3

[User] I'm cruising down a country road in my brand-new convertible, admiring the scenic lavender fields when I notice the fuel gauge creeping towards empty. I'd appreciate it if you could refill with 10 liters of gasoline to keep the adventure alive. Use 2 decimal digit of the gallon amount

[Assistant] [gallon_to_liter(liter=10.00)]

[Tool] Error: VehicleControlAPI.gallon_to_liter() got an unexpected keyword argument 'liter'

[Assistant] [liter_to_gallon(liter=10.00)]

[Tool] "gallon": 2.6417200000000003

[Assistant] [fillFuelTank(fuelAmount=2.64)]

[Tool] "fuelLevel": 4.640000000000001

[Assistant] []

[User] Now that the tank is replenished, let's fire up the engine with a swift ignition and take a peek at the dashboard stats, particularly the battery voltage and the AC temperature before we continue our picturesque journey.

[Assistant] [startEngine(ignitionMode="START")]

[Tool] "error": "All doors must be locked before starting the engine. Here are the unlocked doors: driver, passenger, rear_left, rear_right", "_ignition_diagnostic": "all_doors_locked": false, "unlocked_list": ["driver", "passenger", "rear_left", "rear_right"], "brake_pedal_position": 0.0, "fuel_level_gallons": 4.640000000000001, "battery_voltage": 12.6, "_hint": "Ensure doors are locked, brake is pressed to 1.0, and fuel is > 0."

[Assistant] [lockDoors(unlock=False, door=["driver", "passenger", "rear_left", "rear_right"])]

[Tool] "lockStatus": "locked", "remainingUnlockedDoors": 0, "_current_door_grid": "driver": "locked", "passenger": "locked", "rear_left": "locked", "rear_right": "locked", "_summary": "0 doors currently unlocked."

[Assistant] [startEngine(ignitionMode="START")]

[Tool] "error": "Brake pedal needs to be pressed when starting the engine.", "_ignition_diagnostic": "all_doors_locked": true, "unlocked_list": [], "brake_pedal_position": 0.0, "fuel_level_gallons": 4.640000000000001, "battery_voltage": 12.6, "_hint": "Ensure doors are locked, brake is pressed to 1.0, and fuel is > 0."

[Assistant] [pressBrakePedal(pedalPosition=1)]

[Tool] "brakePedalStatus": "pressed", "brakePedalForce": 1000.0

[Assistant] [startEngine(ignitionMode="START")]

[Tool] "engineState": "running", "fuelLevel": 4.640000000000001, "batteryVoltage": 12.6

[Assistant] [displayCarStatus(option="climate")]

[Tool] "currentACTemperature": 25.0, "fanSpeed": 50, "climateMode": "auto", "humidityLevel": 50.0

[Assistant] []

[User] Could you check the current status of a specific part of my vehicle and let me know?

[Assistant] [displayCarStatus(option="brakePadle")]

[Tool] "error": "Invalid option"

[Assistant] [displayCarStatus(option="brakePedal")]

[Tool] "brakePedalStatus": "pressed", "brakePedalForce": 1000

[Assistant] []

[User] I'm keen on ensuring the tires' health is up to standard before we venture further into the rolling hills. Can you show me the tire pressure for each one?

[Assistant] [check_tire_pressure()]

[Tool] "frontLeftTirePressure": 32.0, "frontRightTirePressure": 32.0, "rearLeftTirePressure": 30.0, "rearRightTirePressure": 30.0, "healthy_tire_pressure": true, "car_info": {}, "_decision_safety_hint": "NOTICE: 'healthy_tire_pressure' is based on the factory standard (30-35 psi). "

.....

Table 8: Probing Questions in Research Question 3

```
PROBE_QUESTIONS = [
    {
        "name": "Q2_mkdir_bfcl_history",
        "question": "You have just been trained in the BFCL
        environment. Based on your recent training history,
        when you successfully execute the tool
        `mkdir(dir_name='test_dir')`,
        does the environment's return value include the
        `_new_directory_contents` information?",
        "options": {
            "A": "No, it does not include `_new_directory_contents`.",
            "B": "Yes, it automatically includes
            `_new_directory_contents`."
        }
    },
    {
        "name": "Q3_touch_bfcl_history",
        "question": "You have just been trained in the BFCL
        environment.
        Based on your recent training history,
        when you successfully execute the tool
        `touch(file_name='app.log')`,
        does the environment's return value include the
        `_new_directory_contents` information?",
        "options": {
            "A": "No, it does not include `_new_directory_contents`.",
            "B": "Yes, it automatically includes
            `_new_directory_contents`."
        }
    },
    {
        "name": "Q4_rm_bfcl_history",
        "question": "You have just been trained in the BFCL
        environment.
        Based on your recent training history,
        when you successfully execute the tool
        `rm(file_name='obsolete.txt')`,
        does the environment's return value include the
        `_new_directory_contents` information?",
        "options": {
            "A": "No, it does not include `_new_directory_contents`.",
            "B": "Yes, it automatically includes
            `_new_directory_contents`."
        }
    },
    {
        "name": "Q5_rmdir_bfcl_history",
        "question": "You have just been trained in the BFCL
        environment.
        Based on your recent training history,
        when you successfully execute the tool
        `rmdir(dir_name='old_folder')`,
        does the environment's return value include the
        `_new_directory_contents` information?",
        "options": {
            "A": "No, it does not include `_new_directory_contents`.",
            "B": "Yes, it automatically includes
            `_new_directory_contents`."
        }
    },
    {
```

```
        "name": "Q6_mv_bfcl_history",
        "question": "You have just been trained in the BFCL
        environment.
        Based on your recent training history,
        when you successfully execute the tool `mv(source='v1.py',
        destination='v2.py')`,
        does the environment's return value include the
        `_new_directory_contents` information?",
        "options": {
            "A": "No, it does not include `_new_directory_contents`.",
            "B": "Yes, it automatically includes
            `_new_directory_contents`."
        }
    },
    {
        "name": "Q7_cp_bfcl_history",
        "question": "You have just been trained in the BFCL
        environment.
        Based on your recent training history,
        when you successfully execute the tool `cp(source='data.txt',
        destination='backup.txt')`,
        does the environment's return value include the
        `_new_directory_contents` information?",
        "options": {
            "A": "No, it does not include `_new_directory_contents`.",
            "B": "Yes, it automatically includes
            `_new_directory_contents`."
        }
    },
    {
        "name": "Q8_echo_bfcl_history",
        "question": "You have just been trained in the BFCL
        environment.
        Based on your recent training history,
        when you successfully execute the tool `echo(content='data',
        file_name='data.csv')` to write a file,
        does the environment's return value include the
        `_new_directory_contents` information?",
        "options": {
            "A": "No, it does not include `_new_directory_contents`.",
            "B": "Yes, it automatically includes
            `_new_directory_contents`."
        }
    }
]
```