SELF-GENERATED SUPERVISION FOR SELF-EVOLVING AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon interactive agents often rely on task rewards, expert trajectories, or stronger teacher models to improve their performance. However, many complex, open-ended tasks lack verifiable outcomes and reliable external supervision, making it difficult for agents to improve through these approaches. Yet agents can actively interact with their environments, and the feedback from these interactions provides a natural source of learning signals. Building on this observation, we propose a simple framework for online agent self-improvement through on-policy self-distillation. At selected student-visited states, the current student explores alternative actions under a larger interaction budget, while an EMA teacher uses the resulting environment feedback to make a better-informed decision that is distilled into the zero-lookahead student. Our method generates supervision at individual decision steps, without relying on task rewards, explicit value estimates, or the outcomes of complete trajectories. As the student policy is updated, it generates new higher-budget teacher supervision, allowing both the student and teacher to improve over time. Experiments on ALFWorld show that, without using task rewards or expert supervision during training, the student's task success rate increases from 28.4% to 90.3%, while that of the higher-budget teacher increases from 76.9% to 96.3%. The trained student also surpasses the initial higher-budget teacher, demonstrating that agents can achieve sustained self-improvement through their own interactions with the environment. Our code and models will be publicly released.

1 INTRODUCTION

Long-horizon interactive agents must complete complex tasks through continuous interactions with their environments (Yao et al., 2022b). Their decisions affect not only immediate outcomes but also subsequent task progress. In recent years, LLM-based agents have shown considerable potential in applications such as web navigation (Yao et al., 2022a), embodied tasks (Shridhar et al., 2020), and scientific experimentation. Improving their decision-making capabilities in long-horizon interactive tasks has therefore attracted increasing research attention.

Existing agent learning methods typically rely on task rewards or externally provided decision supervision to obtain learning signals. Reinforcement learning methods optimize agent policies using environment rewards or evaluations of task outcomes (Feng et al., 2026b), while imitation learning and policy distillation methods (Li et al., 2026) learn from expert trajectories or teacher models. However, in many complex, open-ended tasks, task outcomes can be difficult to verify (Su et al., 2026). For example, evaluating the overall correctness of a long-horizon coding task (Jimenez et al., 2024) may require completing the entire task. Some open-ended tasks may even lack well-defined criteria for correctness (Gan et al., 2026). Moreover, intermediate steps may also lack reliable quality assessments, and expert demonstrations or high-quality teacher supervision can be difficult to obtain. Therefore, enabling agents to continuously improve without reliable external supervision remains an important challenge.

Notably, agents can obtain additional information by trying alternative actions and observing their consequences through further interactions with the environment (Anthony et al., 2017). This feedback can help agents make better-informed decisions at the current state. These decisions can then serve as learning targets for a policy that acts without access to the additional information. Since

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.

Table 1: External supervision requirements (✓: free; ✗: required). Reward/verifier requires terminal task-completion scores or success checks; demonstration requires expert or example trajectories; teacher requires an additional stronger or supervisory model.

Requirement	WebRL	Learn-by-Interact	Dyna-Mind	IWM	Reasoning-Bank	RL-based	OPD-based	Ours
Reward/Verifier-Free	✗	✗	✗	✓	✗	✗	✓	✓
Demonstration-Free	✗	✗	✓	✗	✓	✓	✓	✓
Teacher-Free	✗	✗	✗	✓	✓	✓	✗	✓

these learning targets can be generated at individual decision steps, agents can learn from their own interactions without waiting for the final outcomes of complete tasks.

Building on this observation, we propose SAGE, a simple framework for continual agent self-improvement through online on-policy self-distillation. At selected states visited by the student, the student uses a larger interaction budget to try alternative actions and observe their consequences in the real environment. An EMA teacher then uses this feedback to make a better-informed decision for the original state, whose token-level predictions are distilled into the student through forward KL divergence. SAGE directly uses interaction feedback for policy updates without first organizing the experience into predefined forms such as reflections or skills. Unlike distillation from a fixed teacher, its supervision improves together with the student: as the policy becomes stronger, it produces more effective lookahead interactions and better supervision for subsequent rounds. SAGE thus obtains learning signals from its own interactions without relying on task rewards, an additional stronger teacher model, or the final outcomes of complete trajectories. Table 1 compares the external supervision requirements of SAGE and existing methods.

We evaluate SAGE on multiple long-horizon interactive tasks. Results show that the student improves its performance through distillation without increasing its interaction budget at inference time, eventually surpassing the initial higher-budget teacher. Meanwhile, the higher-budget teacher also improves as the student policy is updated. On ALFWorld and WebShop, the student's task success rate increases from 28.4% to 90.3% and from 14.1% to 61.6%, respectively, while the higher-budget teacher's success rate increases from 76.9% to 96.3% and from 50.6% to 68.8%, respectively.

Our main contributions are: (i) We propose SAGE, which constructs training supervision from the agent's own real environment interactions without task rewards, expert demonstrations, or an additional stronger teacher model. (ii) We develop foresight distillation, where improved policies generate more effective lookahead interactions and stronger supervision for subsequent training rounds. (iii) Experiments on ALFWorld and WebShop demonstrate strong reward-free performance and continual self-improvement.

2 METHOD

An overview of SAGE is illustrated in Figure 1. Section 2.1 introduces the task formulation and interaction budget used in SAGE. Section 2.2 presents foresight distillation, which uses additional environment feedback to construct teacher supervision and distill it into the student policy. Section 2.3 describes the iterative training procedure for continual agent self-improvement.

2.1 PRELIMINARY

Task Formulation. We consider a sequential decision-making process in which a large language model (LLM) policy π_θ interacts with an environment $\mathcal{E}$ over multiple steps. Given a task instruction q, the interaction history at step t is denoted by $h_t = (q, o_0, a_0, o_1, a_1, \ldots, a_{t-1}, o_t)$, where o_t is the environment observation and a_t is the action executed by the agent. At each step, the agent generates a decision response y_t based on the interaction history h_t, which is parsed into an executable action a_t. The environment then executes the action and returns a new observation:

$$y_t \sim \pi_\theta(\cdot \mid h_t), \qquad a_t = \mathrm{Parse}(y_t), \qquad o_{t+1} \sim \mathcal{E}(\cdot \mid h_t, a_t). \tag{1}$$

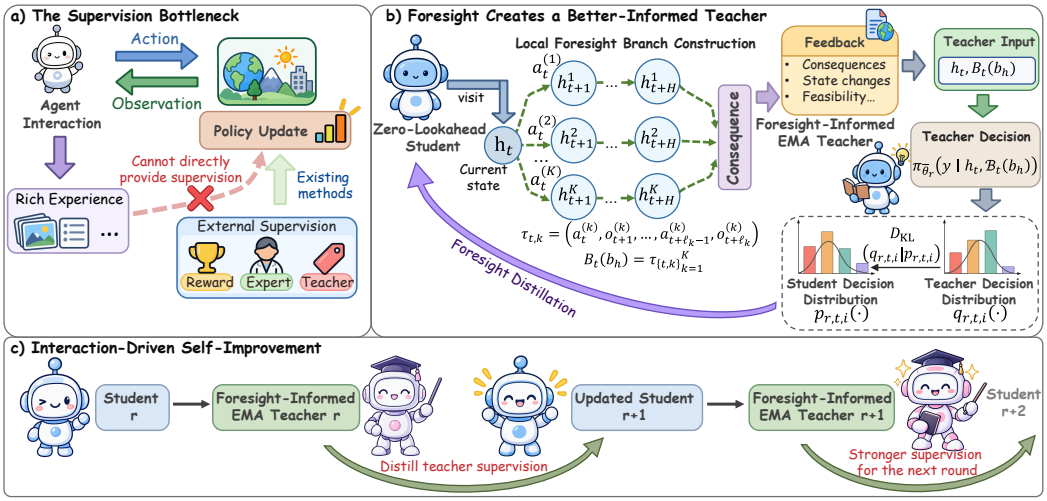

Figure 1: Overview of SAGE. (a) Existing agent learning methods rely on external supervision, such as rewards, verifiers, expert trajectories, or stronger teacher models. (b) SAGE uses the current student to explore alternative actions under a larger interaction budget, while an EMA teacher uses the resulting feedback to supervise the zero-lookahead student. (c) As the student improves, it generates more effective lookahead interactions and stronger supervision for subsequent rounds.

The interaction history is then updated as $h_{t+1} = (h_t, a_t, o_{t+1})$, and the agent proceeds to the next decision step.

Interaction Budget. At each decision step, the agent can try alternative actions and observe their consequences through interactions with the real environment. We define the interaction budget as $b = (K, H)$, where K is the number of candidate root actions explored from the current decision point, and H is the maximum number of interaction steps along each lookahead branch. All lookahead interactions are performed in the real environment rather than through model-based simulation.

During training, given an interaction history h_t, the policy π_θ first generates K candidate root actions $\{a_t^{(k)}\}_{k=1}^{K}$. Before executing each candidate action, the environment is restored to the original state corresponding to h_t. The policy continues interacting with the environment based on the original history h_t and the interactions collected within that branch, until the maximum depth H is reached or the environment terminates. These lookahead branches are used only to collect environment feedback. After lookahead, the agent resumes execution from the original environment state.

The local lookahead trajectory for the k-th candidate action is denoted by

$$\tau_{t,k} = (a_t^{(k)}, o_{t+1}^{(k)}, \ldots, a_{t+\ell_k-1}^{(k)}, o_{t+\ell_k}^{(k)}), \qquad 1 \le \ell_k \le H, \tag{2}$$

where ℓ_k is the actual number of actions executed in the branch. The lookahead information obtained at decision point h_t under budget b is defined as

$$\mathcal{B}_t(b) = \{\tau_{t,k}\}_{k=1}^{K}. \tag{3}$$

The lookahead information consists only of actions and observations from the real environment, without additional task rewards, value estimates, or other scalar evaluation signals. In particular, we define $b_0 = (0, 0)$ as the zero-lookahead budget, for which $\mathcal{B}_t(b_0) = \varnothing$.

2.2 Foresight Distillation

Additional lookahead interactions allow the agent to observe the actual consequences of alternative actions and reconsider its current decision. We use this difference in available information to construct distillation targets at interaction histories h_t visited by the student. Specifically, in training round r, the agent policy π_{θ_r} collects lookahead information $\mathcal{B}_t(b_h)$ using a higher interaction budget $b_h = (K, H)$. The higher-budget teacher $\pi_{\bar{\theta}_r}$ uses this information to generate a decision

Algorithm 1 SAGE: Iterative Self-Improvement

Require: Initial policy θ_0; environment $\mathcal{E}$; budgets $b_0 = (0, 0), b_h = (K, H)$; EMA factor α; reference weight λ_{ref}; rounds R
Ensure: Student policy π_{θ_R}

Initialize $\bar{\theta}_0 \leftarrow \theta_0$ and freeze reference policy $\pi_{\text{ref}} \leftarrow \pi_{\theta_0}$
for $r = 0, \ldots, R-1$ **do**
  $\mathcal{D}_r \leftarrow \text{Rollout}(\pi_{\theta_r}, \mathcal{E}, b_0)$; $\widetilde{\mathcal{D}}_r \leftarrow \text{Select}(\mathcal{D}_r)$; $\mathcal{T}_r \leftarrow \varnothing$
  for $h_t \in \widetilde{\mathcal{D}}_r$ **do**
    Generate candidate root actions $\{a_t^{(k)}\}_{k=1}^{K}$ using π_{θ_r} at h_t
    for $k = 1, \ldots, K$ **do**
      Restore $\mathcal{E}$ to the original state corresponding to h_t; execute $a_t^{(k)}$, and continue the branch with π_{θ_r}
      Record $\tau_{t,k}$ from at most H real environment steps
    end for
    $\mathcal{B}_t(b_h) \leftarrow \{\tau_{t,k}\}_{k=1}^{K}$ ▷ Used only for supervision construction
    $q_{r,t}(y) \leftarrow \pi_{\bar{\theta}_r}(y \mid h_t, \mathcal{B}_t(b_h))$
    Sample $y = (y^{(1)}, \ldots, y^{(L)}) \sim q_{r,t}$ and compute $\{q_{r,t,i}\}_{i=1}^{L}$
    $\mathcal{T}_r \leftarrow \mathcal{T}_r \cup \{(h_t, y, \{q_{r,t,i}\}_{i=1}^{L})\}$
  end for
  Compute $\mathcal{L}^{(r)} = \mathcal{L}_{\text{FD}}^{(r)} + \lambda_{\text{ref}} \mathcal{L}_{\text{ref}}^{(r)}$
  $\theta_{r+1} \leftarrow \text{MultiStepOptimize}(\theta_r, \mathcal{T}_r, \mathcal{L}^{(r)})$ ▷ Update student only
  $\bar{\theta}_{r+1} \leftarrow \alpha \bar{\theta}_r + (1-\alpha)\theta_{r+1}$ ▷ EMA teacher update
end for
return π_{θ_R}

distribution for the original interaction history h_t, while the student makes its decision based only on h_t. Their conditional distributions are defined as

$$q_{r,t}(y) = \pi_{\bar{\theta}_r}(y \mid h_t, \mathcal{B}_t(b_h)), \qquad p_{r,t}(y) = \pi_{\theta_r}(y \mid h_t). \tag{4}$$

Here, y denotes a complete decision response, including the action format and content, and $\bar{\theta}_r$ and θ_r are the teacher and student parameters in round r, respectively. We use $q_{r,t}$ as the distillation target, enabling the student to learn from decisions informed by additional environment feedback.

Momentum Teacher. As the student policy is continuously updated during training, directly copying the student parameters to the teacher would cause the distillation target to change rapidly. To provide more stable supervision, we maintain the teacher using an exponential moving average (EMA) of the student parameters:

$$\bar{\theta}_{r+1} = \alpha \bar{\theta}_r + (1-\alpha)\theta_{r+1}, \tag{5}$$

where $\alpha \in [0, 1)$ is the momentum coefficient. The teacher parameters remain fixed during each distillation round and are not updated by gradients.

Foresight Distillation Objective. For a decision response $y = (y^{(1)}, \ldots, y^{(L)}) \sim q_{r,t}$ sampled from the teacher distribution, we define the teacher and student token distributions at position i, conditioned on the same teacher-generated prefix $y^{(<i)}$, as

$$\begin{aligned} q_{r,t,i}(\cdot) &= \pi_{\bar{\theta}_r}\left(\cdot \mid h_t, \mathcal{B}_t(b_h), y^{(<i)}\right), \\ p_{r,t,i}(\cdot) &= \pi_{\theta_r}\left(\cdot \mid h_t, y^{(<i)}\right). \end{aligned} \tag{6}$$

We perform token-level distillation by minimizing the forward KL divergence:

$$\mathcal{L}_{\text{FD}}^{(r)} = \mathbb{E}_{\substack{h_t \sim \rho_r \\ y \sim q_{r,t}}}\left[\frac{1}{L}\sum_{i=1}^{L} D_{\text{KL}}\left(q_{r,t,i} \| p_{r,t,i}\right)\right], \tag{7}$$

where ρ_r denotes the sampling distribution over student-visited interaction histories selected for distillation in training round r. Minimizing $\mathcal{L}_{\text{FD}}^{(r)}$ trains the student policy to approximate the teacher

distribution without requiring access to additional lookahead information $\mathcal{B}_t(b_h)$, transferring decisions informed by lookahead interactions to the student policy. To stabilize training, we additionally regularize the student toward the frozen initial policy π_{θ_0}. The overall training objective is

$$\mathcal{L}^{(r)} = \mathcal{L}^{(r)}_{\mathrm{FD}} + \lambda_{\mathrm{ref}} \mathcal{L}^{(r)}_{\mathrm{ref}}, \tag{8}$$

where $\mathcal{L}^{(r)}_{\mathrm{ref}}$ denotes the reference-policy KL regularization term and λ_{ref} controls its strength.

2.3 Iterative Self-Improvement

The overall training procedure is summarized in Algorithm 1. In training round r, the zero-lookahead student policy π_{θ_r} interacts with the real environment and collects a set of interaction histories $\mathcal{D}_r = \{h_t\}$. We then select a subset of these histories for distillation, denoted by $\widetilde{\mathcal{D}}_r \subseteq \mathcal{D}_r$. On the selected histories, we construct higher-budget teacher supervision following Section 2.2 and optimize the student policy for multiple gradient steps using the foresight distillation objective, yielding the updated parameters θ_{r+1}. After student optimization, we update the teacher parameters using the EMA rule and proceed to the next training round. In subsequent rounds, the updated student policy collects new interaction histories and performs lookahead interactions, producing improved teacher supervision based on the new interaction experience. Through this iterative process, improvements in the student policy lead to better-informed lookahead interactions and improved teacher supervision in subsequent rounds, enabling continual self-improvement.

3 Experiments

We evaluate SAGE on ALFWorld and WebShop from three aspects. Section 3.2 compares SAGE with existing agent learning methods and examines its continual self-improvement. Section 3.3 compares SAGE with reward-based reinforcement learning in task performance and training dynamics. Section 3.4 presents ablation studies on the key designs of SAGE.

3.1 Implementation Details

We implement SAGE with VERL, using FSDP for training and vLLM for inference on four NVIDIA A100 80GB GPUs. The student and teacher are initialized from the same Qwen3-4B (Yang et al., 2025). We train LoRA adapters (Hu et al., 2021) with rank 128 and scaling factor $\alpha = 256$, while keeping the backbone frozen. At each online iteration, we sample 128 training tasks and generate one student trajectory per task, with at most 30 environment steps. We select up to 512 student-visited states for distillation, with roughly balanced coverage across action types. For each selected state, we use a maximum interaction budget of $b_h = (8, 4)$, corresponding to up to eight lookahead branches with depth at most four. The student generates the root and branch actions, while the EMA teacher provides supervision for the zero-lookahead student.

We optimize the student with token-level forward KL distillation at temperature 2.0. At each token position, we retain the teacher's top-256 tokens and merge the remaining teacher and student probability mass into a single tail bucket; the distillation loss is scaled by the squared temperature. We further apply a KL regularizer to the frozen initial policy with coefficient 0.1, using Schulman's low-variance k_3 estimator. Each online iteration performs 16 optimizer steps with a peak learning rate of 2×10^{-6}. We use linear warmup for the first five iterations followed by cosine decay, and train for 110 iterations in total. After each iteration, the teacher LoRA parameters are updated once by EMA with momentum 0.98. No task reward or success label is used as a training target or explicitly included in the lookahead prompts. Complete prompts and additional implementation details are provided in Appendices A and B.

3.2 Self-Improvement without External Supervision

We first investigate whether SAGE can effectively improve long-horizon agent performance and sustain self-improvement without relying on task rewards, expert demonstrations, or external teacher models. We compare SAGE with three representative categories of methods on ALFWorld and WebShop: the prompting method ReAct (Yao et al., 2022b), world experience methods including IWM (Zhang et al., 2025d), WKM (Qiao et al., 2024), ITP-I (Liu et al., 2026a),

Table 2: Supervision requirements and task success rates (%) on ALFWorld and WebShop. ✓ indicates that the method is free from the corresponding requirement; ✗ indicates that it requires the supervision. Avg. is the unweighted mean of the three reported success rates. For OPD-based methods, the external teacher is Qwen3-30B-A3B.

Category	Method	Policy	Demo-Free	Reward-Free	Teacher-Free	ALFWorld		WebShop	Avg.
						Seen SR ↑	Unseen SR ↑	SR ↑	SR ↑
Prompting	ReAct	Qwen3-4B	✓	✓	✓	32.1	26.1	14.06	24.09
	ReAct	Qwen3-30B-A3B	✓	✓	✓	43.57	40.80	19.25	34.54
World Experience	IWM	Qwen3-4B	✗	✓	✓	64.29	47.20	44.70	52.06
	WKM		✗	✗	✗	62.14	44.90	45.80	50.95
	ITP-I		✗	✓	✓	31.43	28.70	19.80	26.64
	ITP-R		✗	✗	✓	67.15	48.54	52.35	56.01
	CoMAP		✗	✗	✓	71.53	70.68	59.60	67.27
OPD-based	Vanilla OPD	Qwen3-4B	✓	✓	✗	31.43	32.09	13.50	25.67
	AdaSwitch		✓	✓	✗	29.29	29.10	12.25	23.55
	TCOD		✓	✓	✗	32.86	31.34	19.00	27.73
	Guided-OPD		✓	✓	✗	37.14	35.08	17.75	29.99
Self-Evolution	**Ours**	**Qwen3-4B**	✓	✓	✓	**89.50**	**90.29**	**61.56**	**80.45**

ITP-R, and CoMAP (Liu et al., 2026b), and on-policy distillation methods including Vanilla OPD, AdaSwitch (Peng et al., 2025), TCOD (Wang et al., 2026b), and Guided-OPD (Li et al., 2026). For the world experience and on-policy distillation methods, we report the corresponding baseline results provided by CoMAP and Guided-OPD, respectively. Except for ReAct with Qwen3-30B-A3B, all methods use Qwen3-4B as the policy backbone. We also indicate whether each method relies on expert demonstrations, task rewards, or an external teacher model to compare performance under different supervision requirements.

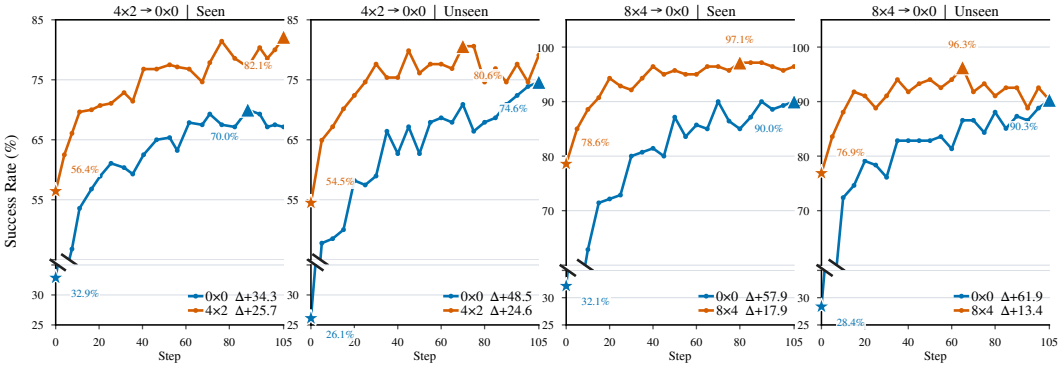

Figure 2: Held-out performance of the student policy during training on ALFWorld Seen and Unseen tasks. The left two panels use a 4×2 interaction budget and the right two use 8×4, comparing zero-lookahead and higher-budget evaluation of the same student policy.

As shown in Table 2, SAGE achieves the highest task success rate on all three evaluation settings, reaching 89.50% on ALFWorld Seen, 90.29% on ALFWorld Unseen, and 61.56% on WebShop. Compared with the strongest existing baseline, CoMAP, SAGE improves these results by 17.97, 19.61, and 1.96 percentage points, respectively. Notably, world experience methods typically rely on demonstrations or reward signals, while OPD methods require an external teacher model. In contrast, SAGE improves its policy solely through its own environment interactions, without task rewards, expert demonstrations, or external teachers. These results show that self-generated interaction supervision can support strong task performance without these sources of external supervision.

Figure 2 further illustrates the continual self-improvement of SAGE. We evaluate the same student policy with zero lookahead and higher interaction budgets throughout training. Taking ALFWorld Unseen under the 8×4 setting as an example, the zero-lookahead student improves from 28.4% to 90.3%, eventually surpassing the initial student's 76.9% success rate under the 8×4 budget.

Table 3: Comparison with RL-based methods on ALFWorld and WebShop. All methods use Qwen3-4B as the policy backbone. ✓ indicates reward-free training, while ✗ indicates that task rewards are required. ALFWorld reports success rates for each task type and overall, while WebShop reports task score and success rate.

Method	Reward-Free	ALFWorld							WebShop	
		Pick	Look	Clean	Heat	Cool	Pick2	Overall	Score	Success
GRPO	✗	86.6±0.6	59.5±5.1	77.7±4.7	67.1±4.1	55.5±4.1	55.0±6.2	70.6±1.3	81.9±3.6	69.5±1.5
GRPO+OPSD	✗	88.0±0.4	52.8±3.9	75.0±4.1	62.5±3.4	68.3±8.1	36.9±2.5	68.8±2.6	82.5±1.0	71.3±1.2
RLSD	✗	87.9±4.0	83.4±1.8	80.5±0.9	70.3±9.4	79.6±4.0	60.6±4.3	79.2±1.0	83.7±1.5	68.5±1.2
SDAR	✗	85.2±4.8	63.9±4.0	76.8±5.8	80.0±3.6	72.9±4.4	42.6±6.9	73.2±1.3	83.6±1.8	71.9±0.7
OCSD	✗	91.9±1.4	77.2±4.4	**90.2±4.6**	**84.7±4.4**	80.9±4.3	63.8±0.9	82.8±0.7	**86.9±1.7**	**73.7±1.6**
SAGE (Ours)	✓	**99.1±1.4**	**95.4±4.0**	85.6±3.7	81.9±3.5	**84.8±2.5**	**86.7±5.8**	**89.5±1.5**	75.1±3.0	61.6±2.3

Meanwhile, the same policy under the 8×4 interaction budget further improves to 96.3%. These results show that the student is not simply approaching a fixed source of supervision: the zero-lookahead student surpasses the initial high-budget policy, while the policy's higher-budget capability continues to improve. This shows that the policy's higher-budget decision capability improves together with the student, providing progressively stronger supervision in subsequent rounds. Appendix C provides qualitative examples of how teacher supervision informed by environment feedback changes the student's action decisions.

3.3 Comparison with RL-based Methods

We further compare SAGE with reinforcement learning methods to examine whether it can achieve competitive performance without task rewards and how its training dynamics differ from reward-based optimization. All methods use Qwen3-4B as the policy backbone. Following OCSD (Yang et al., 2026d), we compare against GRPO (Shao et al., 2024), GRPO+OPSD (Zhao et al., 2026), RLSD (Yang et al., 2026a), SDAR (Lu et al., 2026), and OCSD using the reported baseline results under the same evaluation setting. We report success rates for individual ALFWorld task types and overall performance, as well as task score and success rate on WebShop. We further compare SAGE and GRPO in terms of KL divergence from the initial policy, per-step training time, and success rate throughout training. We also evaluate the effect of applying GRPO after SAGE training.



Figure 3: KL divergence from the frozen initial policy for GRPO and SAGE during training.

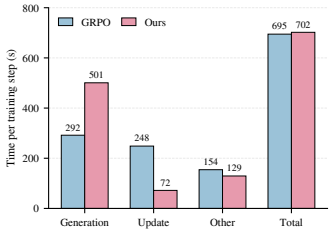

Figure 4: Per-step training time for GRPO and SAGE, including generation, update, and other.

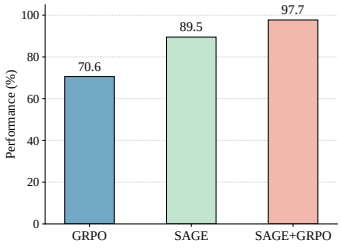

Figure 5: Performance of GRPO, SAGE, and SAGE followed by GRPO training.

As shown in Table 3, SAGE achieves an overall success rate of 89.5% on ALFWorld without using task rewards, outperforming all reward-based baselines. It also obtains the best results on Pick, Look, Cool, and Pick2. These results show that supervision constructed from the agent's own interactions can support strong long-horizon decision making without reward-based optimization. On WebShop, SAGE achieves a 61.6% success rate despite using no task rewards during training.

Figure 3 compares the KL divergence of SAGE and GRPO from the frozen initial policy during training. The KL divergence of GRPO increases substantially and shows larger fluctuations as training proceeds, whereas SAGE remains lower and more concentrated. This indicates that SAGE improves task performance with a more controlled policy shift and smoother updates.

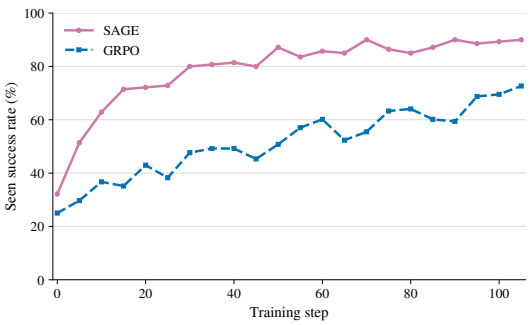

Figure 6: Comparison of task success rates throughout training on the ALFWorld Seen subset for GRPO and SAGE with $8 \times 4 \rightarrow 0 \times 0$ distillation.

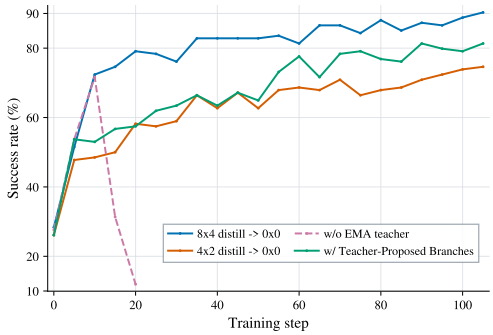

Figure 7: Ablation study on ALFWorld Unseen, comparing two lookahead budgets, direct teacher updates without EMA, and lookahead branches generated by the EMA teacher.

Figures 4 and 6 compare the training dynamics and computational cost of SAGE and GRPO. On ALFWorld Seen, SAGE reaches a higher success rate early in training and remains above GRPO throughout the training process, reaching 89.5% compared with 70.6% for GRPO. Their total per-step training times are similar, at approximately 702 s for SAGE and 695 s for GRPO. SAGE spends more time on generation because of the additional lookahead interactions, but its distillation updates are substantially cheaper than the policy updates in GRPO. As a result, SAGE achieves faster performance gains and a higher final success rate at a similar per-step training cost.

Finally, Figure 5 examines whether SAGE can be combined with subsequent reward-based optimization. GRPO and SAGE alone achieve 70.6% and 89.5% success rates on ALFWorld Seen, respectively. Applying GRPO after SAGE further improves the success rate to 97.7%. This result shows that SAGE is complementary to reward-based optimization and can provide a stronger starting point for subsequent reinforcement learning.

3.4 Ablation Study

Figure 7 presents the ablation study on ALFWorld Unseen. First, $8 \times 4 \rightarrow 0 \times 0$ consistently outperforms $4 \times 2 \rightarrow 0 \times 0$ during training and achieves a higher final success rate, showing that a larger lookahead budget provides more effective supervision for distillation.

We next remove the EMA Teacher and directly copy the student parameters to the teacher after each training round. This variant improves briefly at the beginning but then degrades rapidly, suggesting that a teacher that fully follows the student produces a rapidly changing distillation target. EMA smooths the teacher updates and provides more stable supervision across rounds.

Finally, we replace the default branches generated by the student with branches generated by the EMA teacher. This variant still improves throughout training, but learns more slowly than the default setting. Since the EMA teacher changes more slowly than the student, its lookahead branches reflect a less up-to-date policy. Using the current student for branch interaction therefore makes better use of the latest policy capability and produces effective supervision earlier in training.

4 Related Work

4.1 Self-Evolving Agents

Self-evolving agents continuously improve their policies through environment interactions and supervision signals such as task feedback. Early studies filter self-generated data and reuse it for training, enabling models to improve iteratively from their previous outputs (Zelikman et al., 2022; Wang et al., 2023b). Interactive agents use feedback from their interactions to refine their decisions and plans, and apply these refinements in subsequent attempts (Shinn et al., 2023; Madaan et al., 2023; Gou et al., 2024; Sun et al., 2023). To extend improvement beyond individual attempts, interaction histories are accumulated as reusable experiences and memories (Zhao et al., 2024; Park

et al., 2023; Zhong et al., 2024), skills (Wang et al., 2023a), or workflows (Wang et al., 2024). Improved behaviors can also be converted into training data for policy updates (Zeng et al., 2024). Further studies use model self-evaluation (Yuan et al., 2024), verifiers (Hosseini et al., 2024), reinforcement learning from self-correction (Kumar et al., 2025), environment rewards (Bai et al., 2024; Qi et al., 2025), and search and preference feedback (Putta et al., 2024) to guide iterative generation and optimization. Task generation (Hu et al., 2025a), prompt optimization driven by textual feedback (Yuksekgonul et al., 2024), workflow search (Zhang et al., 2025c), and agent architecture optimization (Hu et al., 2025b; Shang et al., 2025) further extend iterative improvement from agent behavior to task and system design.

As iterative improvement extends to long-horizon interactions, assigning task feedback to individual decisions becomes an important research problem. Reinforcement learning for long-horizon agents uses trajectory-level rewards (Wang et al., 2025d; Jin et al., 2025; Feng et al., 2026a; Dong et al., 2025) and introduces step-level advantage estimation (Feng et al., 2026b; Dong et al., 2026) and progress-based credit assignment (Wang et al., 2025a). Other studies develop frameworks for end-to-end agent reinforcement learning and optimize query-level multi-agent systems using execution feedback (Gao et al., 2025). Experience reuse has also advanced from memory organization and augmentation (Xu et al., 2026; Chhikara et al., 2025; Salama et al., 2025) to distilling reasoning experiences (Ouyang et al., 2026), reward-based memory management and retrieval (Yan et al., 2026; Zhou et al., 2025), and reflection and experience updates in mobile (Wang et al., 2025c) and long-horizon tasks (Wu et al., 2025; Zhai et al., 2025). Meanwhile, execution feedback supports iterative code improvement (Robeyns et al., 2025; Zhang et al., 2026; Novikov et al., 2025), while task performance guides workflow (Zhang et al., 2025a; Wang et al., 2025b) and multi-agent architecture optimization (Zhang et al., 2025b; Dang et al., 2026). Population-based search and co-evolution also provide signals for iterative improvement (Ke et al., 2025). Real environment interactions have further been used to learn environment simulators (Yu et al., 2026) and transferable skills across tasks (Yao et al., 2026).

4.2 On-Policy Distillation for Agents

On-policy distillation (OPD) matches the teacher distribution on student-generated sequences (Agarwal et al., 2024), while different teacher and student generation strategies can be used to adjust the training data (Peng et al., 2025). Self-distillation further constructs dense supervision using verifiable reasoning (Zhao et al., 2026), environment rewards (Hübotter et al., 2026), or self-correction based on binary rewards (He et al., 2026). When extended to multi-turn interactions, temporal curriculum learning (Wang et al., 2026b), step-level sampling and multi-agent debate (Wang et al., 2026a), curriculum-based teacher guidance (Li et al., 2026), and prefix reuse (Liao et al., 2026) have been explored to address trajectory distribution shifts. Other methods distill retrospective skills from completed trajectories (Yang et al., 2026c; Wu et al., 2026) or validate teacher guidance through subsequent student trajectories (Chen et al., 2026). To improve supervision reliability, recent studies examine the limitations of additional information and noisy teachers (Kaur et al., 2026; Ding & Zhang, 2026). They also explore filtering distillation signals using verifiable rewards (Lin et al., 2026), aligning intermediate representations (Yang et al., 2026b), and deriving supervision from self-generated outputs without external annotations (Li et al., 2026).

5 Conclusion

We study how long-horizon interactive agents can continually improve without task rewards, expert demonstrations, or additional stronger teacher models. We propose SAGE, which constructs self-generated supervision from additional real environment interactions and uses foresight distillation to transfer higher-budget decision capability to a zero-lookahead student. As the student improves, it produces more effective lookahead information, which provides better supervision for subsequent training rounds and enables continual self-improvement. Experiments on ALFWorld and WebShop show that SAGE achieves strong task performance, with the zero-lookahead student surpassing the initial higher-budget policy. SAGE can also be combined with reinforcement learning for further improvement. Currently, SAGE uses a fixed lookahead budget across states. Adapting the interaction budget to state difficulty or policy uncertainty may further improve training efficiency.

AI USE STATEMENT

In this work, we used generative AI tools for partial code implementation and debugging. We have not used generative AI tools for the development of the research content and research ideas, experimental design, or the generation and analysis of experimental results, and the remaining tasks requiring mandatory disclosure are not applicable to this work. Additionally, we used generative AI tools for literature search and preliminary organization, as well as language polishing and improvement of the manuscript's clarity and expression. We have reviewed all AI-assisted work. Code implemented or debugged with AI assistance was manually inspected and verified by the authors; literature retrieved or organized with AI assistance was checked against the original sources by the authors; and the research content, research ideas, experimental design, and experimental results presented in this paper were independently produced and verified by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Thomas Anthony, Zheng Tian, and David Barber. Thinking fast and slow with deep learning and tree search. *Advances in neural information processing systems*, 30, 2017.

Hao Bai, Yifei Zhou, Mert Cemri, Jiayi Pan, Alane Suhr, Sergey Levine, and Aviral Kumar. Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning. *Advances in Neural Information Processing Systems*, 37:12461–12495, 2024.

Chishui Chen, Yaoyou Fan, Te Sun, Yi Yang, Chenghao Sun, Delin Mao, Hongbo Qiao, Zuowei Zhang, Junxi Wang, Chenxing Sun, et al. Look ahead before you distill: Future trajectory validation of teacher guidance for agentic on-policy distillation. *arXiv preprint arXiv:2608.01953*, 2026.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

Yufan Dang, Chen Qian, Xueheng Luo, Jingru Fan, Zihao Xie, Ruijie Shi, Weize Chen, Cheng Yang, Xiaoyin Che, Ye Tian, et al. Multi-agent collaboration via evolving orchestration. *Advances in neural information processing systems*, 38:165025–165059, 2026.

Yi Ding and Ruqi Zhang. Does on-policy distillation really distill? from noisy teacher to self-improvement. *arXiv preprint arXiv:2608.31046*, 2026.

Guanting Dong, Yifei Chen, Xiaoxi Li, Jiajie Jin, Hongjin Qian, Yutao Zhu, Hangyu Mao, Guorui Zhou, Zhicheng Dou, and Ji-Rong Wen. Tool-star: Empowering llm-brained multi-tool reasoner via reinforcement learning. *arXiv preprint arXiv:2505.16410*, 2025.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, et al. Agentic reinforced policy optimization. In *International Conference on Learning Representations*, volume 2026, pp. 16981–17017, 2026.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in llms. In *International Conference on Learning Representations*, volume 2026, pp. 37909–37926, 2026a.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *Advances in Neural Information Processing Systems*, 38:46375–46408, 2026b.

Shuyu Gan, James Mooney, Pan Hao, Renxiang Wang, Mingyi Hong, Qianwen Wang, and Dongyeop Kang. Scaling unverifiable rewards: A case study on visual insights. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 34537–34569, 2026.

Hongcheng Gao, Yue Liu, Yufei He, Longxu Dou, Chao Du, Zhijie Deng, Bryan Hooi, Min Lin, and Tianyu Pang. Flowreasoner: Reinforcing query-level meta-agents. *arXiv preprint arXiv:2504.15257*, 2025.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Nan Duan, Weizhu Chen, et al. Critic: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, volume 2024, pp. 57734–57811, 2024.

Yinghui He, Simran Kaur, Adithya Bhaskar, Yongjin Yang, Jiarui Liu, Narutatsu Ri, Liam Fowl, Abhishek Panigrahi, Danqi Chen, and Sanjeev Arora. Self-distillation zero: Self-revision turns binary rewards into dense supervision. *arXiv preprint arXiv:2604.12002*, 2026.

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. V-star: Training verifiers for self-taught reasoners. *arXiv preprint arXiv:2402.06457*, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

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

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. In *International Conference on Learning Representations*, volume 2025, pp. 21344–21377, 2025b.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? In *International Conference on Learning Representations*, volume 2024, pp. 54107–54157, 2024.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Simran Kaur, Narutatsu Ri, Yinghui He, Liam Fowl, and Sanjeev Arora. Rethinking on-policy self-distillation for thinking models. *arXiv preprint arXiv:2607.05184*, 2026.

Zixuan Ke, Austin Xu, Yifei Ming, Xuan-Phi Nguyen, Ryan Chin, Caiming Xiong, and Shafiq Joty. Mas-zero: Designing multi-agent systems with zero supervision. *arXiv preprint arXiv:2505.14996*, 2025.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. Training language models to self-correct via reinforcement learning. In *International Conference on Learning Representations*, volume 2025, pp. 54523–54549, 2025.

Gengsheng Li, Mao Zheng, Mingyang Song, Ruiqi Liu, Tianyu Yang, Jie Sun, Qiyong Zhong, Haiyun Guo, Junfeng Fang, Dan Zhang, et al. On-policy distillation with curriculum turn-level guidance for multi-turn agents. *arXiv preprint arXiv:2606.15912*, 2026.

Baohao Liao, Hanze Dong, Christof Monz, Xinxing Xu, Li Dong, and Furu Wei. Multi-turn on-policy distillation with prefix replay. *arXiv preprint arXiv:2607.04763*, 2026.

Wenze Lin, Jiale Zhao, Xitai Jiang, Songde Rao, Yining Li, Shenzhi Wang, Bingxiang He, and Gao Huang. On-policy distillation with verifiable reward. *arXiv preprint arXiv:2608.24696*, 2026.

Youwei Liu, Jian Wang, Hanlin Wang, Beichen Guo, and Wenjie Li. Imagine-then-plan: Agent learning from adaptive lookahead with world models. *arXiv preprint arXiv:2601.08955*, 2026a.

Youwei Liu, Jian Wang, Hanlin Wang, and Wenjie Li. Comap: Co-evolving world models and agent policies for llm agents. *arXiv preprint arXiv:2606.02372*, 2026b.

Zhengxi Lu, Zhiyuan Yao, Zhuowen Han, Zi-Han Wang, Jinyang Wu, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, et al. Self-distilled agentic reinforcement learning. *arXiv preprint arXiv:2605.15155*, 2026.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, et al. Alphaevolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

Siru Ouyang, Jun Yan, I Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long Le, Samira Daruki, Xiangru Tang, et al. Reasoningbank: Scaling agent self-evolving with reasoning memory. In *International Conference on Learning Representations*, volume 2026, pp. 94327–94354, 2026.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

Jingyu Peng, Maolin Wang, Hengyi Cai, Yuchen Li, Kai Zhang, Shuaiqiang Wang, Dawei Yin, and Xiangyu Zhao. Adaswitch: Adaptive switching generation for knowledge distillation. *arXiv e-prints*, pp. arXiv–2510, 2025.

Pranav Putta, Edmund Mills, Naman Garg, Sumeet Motwani, Chelsea Finn, Divyansh Garg, and Rafael Rafailov. Agent q: Advanced reasoning and learning for autonomous ai agents. *arXiv preprint arXiv:2408.07199*, 2024.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Jiadai Sun, Xinyue Yang, Yu Yang, Shuntian Yao, Wei Xu, et al. Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning. In *International Conference on Learning Representations*, volume 2025, pp. 79791–79821, 2025.

Shuofei Qiao, Runnan Fang, Ningyu Zhang, Yuqi Zhu, Xiang Chen, Shumin Deng, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. Agent planning with world knowledge model. *Advances in Neural Information Processing Systems*, 37:114843–114871, 2024.

Maxime Robeyns, Martin Szummer, and Laurence Aitchison. A self-improving coding agent. *arXiv preprint arXiv:2504.15228*, 2025.

Rana Salama, Jason Cai, Michelle Yuan, Anna Currey, Monica Sunkara, Yi Zhang, and Yassine Benajiba. Meminsight: Autonomous memory augmentation for llm agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 33124–33140, 2025.

Yu Shang, Yu Li, Keyu Zhao, Likai Ma, Jiahe Liu, Fengli Xu, and Yong Li. Agentsquare: Automatic llm agent search in modular design space. In *International Conference on Learning Representations*, volume 2025, pp. 3841–3865, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

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

Yi Su, Dian Yu, Linfeng Song, Juntao Li, Haitao Mi, Zhaopeng Tu, Min Zhang, and Dong Yu. Crossing the reward bridge: Expanding reinforcement learning with verifiable rewards across diverse domains. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3872–3892, 2026.

Haotian Sun, Yuchen Zhuang, Lingkai Kong, Bo Dai, and Chao Zhang. Adaplanner: Adaptive planning from feedback with language models. *Advances in neural information processing systems*, 36:58202–58245, 2023.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023a.

Hanlin Wang, Chak Tou Leong, Jiashuo Wang, Jian Wang, and Wenjie Li. Spa-rl: Reinforcing llm agents via stepwise progress attribution. *arXiv preprint arXiv:2505.20732*, 2025a.

Jianze Wang, Ying Liu, Jinlong Chen, Xuchun Hu, Qilong Zhang, Yu Cao, Jun Wang, Hua Yang, Yong Xie, and Qianglong Chen. Mad-opd: Breaking the ceiling in on-policy distillation via multi-agent debate. *arXiv preprint arXiv:2605.01347*, 2026a.

Jiaqi Wang, Wenhao Zhang, Weijie Shi, Yaliang Li, and James Cheng. Tcod: Exploring temporal curriculum in on-policy distillation for multi-turn autonomous agents. *arXiv preprint arXiv:2604.24005*, 2026b.

Yinjie Wang, Ling Yang, Guohao Li, Mengdi Wang, and Bryon Aragam. Scoreflow: Mastering llm agent workflows via score-based preference optimization. *arXiv preprint arXiv:2502.04306*, 2025b.

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 *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 13484–13508, 2023b.

Zhenhailong Wang, Haiyang Xu, Junyang Wang, Xi Zhang, Ming Yan, Ji Zhang, Fei Huang, and Heng Ji. Mobile-agent-e: Self-evolving mobile assistant for complex tasks. *arXiv preprint arXiv:2501.11733*, 2025c.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, et al. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025d.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. *arXiv preprint arXiv:2409.07429*, 2024.

Jinyang Wu, Shuo Yang, Zhengxi Lu, Fan Zhang, Yuhao Shen, Lang Feng, Haoran Luo, Zheng Lian, Shuai Zhang, Zhengqi Wen, et al. Seed: Self-evolving on-policy distillation for agentic reinforcement learning. *arXiv preprint arXiv:2607.14777*, 2026.

Rong Wu, Xiaoman Wang, Jianbiao Mei, Pinlong Cai, Daocheng Fu, Cheng Yang, Licheng Wen, Xuemeng Yang, Yufan Shen, Yuxin Wang, et al. Evolver: Self-evolving llm agents through an experience-driven lifecycle. *arXiv preprint arXiv:2510.16079*, 2025.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 38:17577–17604, 2026.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z Pan, et al. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12805–12825, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled rlvr. *arXiv preprint arXiv:2604.03128*, 2026a.

Shenzhi Yang, Guangcheng Zhu, Bowen Song, Haobo Wang, Mingxuan Xia, Xing Zheng, Yingfan Ma, Zhongqi Chen, Weiqiang Wang, Junbo Zhao, et al. Oprd: On-policy representation distillation. *arXiv preprint arXiv:2606.06021*, 2026b.

Shuo Yang, Jinyang Wu, Zhengxi Lu, Yuhao Shen, Fan Zhang, Lang Feng, Shuai Zhang, Haoran Luo, Zheng Lian, Zhengqi Wen, et al. Opid: On-policy skill distillation for agentic reinforcement learning. *arXiv preprint arXiv:2606.26790*, 2026c.

Yi Yang, Cong Qin, Xiaodan Liu, Chishui Chen, Qing Dong, Yan Zhang, Cao Liu, Zhao Yang, Lu Pan, Jiaye Lin, et al. Agentic reinforcement learning with observation-calibrated self-distillation. *arXiv preprint arXiv:2608.04788*, 2026d.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022b.

Zhiyuan Yao, Yuxin Chen, Zhengxi Lu, Zishan Xu, Yueqing Sun, Yifu Guo, Yuquan Lu, Zhengzhou Cai, Kangning Zhang, Zhuowen Han, et al. Skillrise: Agentic reinforcement learning for cross-task skill evolution. *arXiv preprint arXiv:2607.26784*, 2026.

Xiao Yu, Baolin Peng, Michel Galley, Hao Cheng, Qianhui Wu, Janardhan Kulkarni, Suman Nath, Zhou Yu, and Jianfeng Gao. Dyna-mind: Learning to simulate from experience for better ai agents. In *International Conference on Learning Representations*, volume 2026, pp. 32435–32459, 2026.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. Textgrad: Automatic" differentiation" via text. *arXiv preprint arXiv:2406.07496*, 2024.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. Agenttuning: Enabling generalized agent abilities for llms. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3053–3077, 2024.

Yunpeng Zhai, Shuchang Tao, Cheng Chen, Anni Zou, Ziqian Chen, Qingxu Fu, Shinji Mai, Li Yu, Jiaji Deng, Zouying Cao, et al. Agentevolver: Towards efficient self-evolving agent system. *arXiv preprint arXiv:2511.10395*, 2025.

Guibin Zhang, Kaijie Chen, Guancheng Wan, Heng Chang, Hong Cheng, Kun Wang, Shuyue Hu, and Lei Bai. Evoflow: Evolving diverse agentic workflows on the fly. *arXiv preprint arXiv:2502.07373*, 2025a.

Guibin Zhang, Luyang Niu, Junfeng Fang, Kun Wang, Lei Bai, and Xiang Wang. Multi-agent architecture search via agentic supernet. *arXiv preprint arXiv:2502.04180*, 2025b.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin gödel machine: open-ended evolution of self-improving agents. In *International Conference on Learning Representations*, volume 2026, pp. 104223–104294, 2026.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, et al. Aflow: Automating agentic workflow generation. In *International Conference on Learning Representations*, volume 2025, pp. 34040–34077, 2025c.

Kai Zhang, Xiangchao Chen, Bo Liu, Tianci Xue, Zeyi Liao, Zhihan Liu, Xiyao Wang, Yuting Ning, Zhaorun Chen, Xiaohan Fu, et al. Agent learning via early experience. *arXiv preprint arXiv:2510.08558*, 2025d.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 19724–19731, 2024.

Huichi Zhou, Yihang Chen, Siyuan Guo, Xue Yan, Kin Hei Lee, Zihan Wang, Ka Yiu Lee, Guchun Zhang, Kun Shao, Linyi Yang, et al. Memento: Fine-tuning llm agents without fine-tuning llms. *arXiv preprint arXiv:2508.16153*, 2025.

A Baselines and Benchmarks

A.1 Prompting Methods

ReAct. ReAct (Yao et al., 2022b) interleaves reasoning, actions, and environment observations to guide multi-step interactions. The agent selects actions based on the task and interaction history, incorporating environment feedback into subsequent decisions. We evaluate ReAct with 4B and Qwen3-30B-A3B models to compare their interactive performance without task-specific parameter updates.

A.2 World Experience Methods

IWM. IWM (Zhang et al., 2025d) collects real environment transitions by executing alternative actions from expert-visited states and observing their consequences. It learns environment dynamics through state-transition prediction and subsequently improves the policy through imitation learning on expert trajectories. Its training does not directly optimize task rewards but relies on expert demonstrations.

WKM. WKM (Qiao et al., 2024) extracts task-level and state-level knowledge from interaction trajectories and trains a world knowledge model to organize and retrieve this experience. During task execution, the agent uses task-level knowledge for planning and retrieves state-relevant knowledge to guide action selection. Knowledge construction and model training rely on pre-collected trajectories, whose sources and selection criteria vary across environments.

ITP (ITP-I / ITP-R). ITP (Liu et al., 2026a) uses a learned world model to predict the consequences of candidate actions and adaptively allocate lookahead budgets. ITP-I uses world-model predictions to guide a frozen policy at inference time, whereas ITP-R further uses lookahead-derived supervision for policy warm-up and online reinforcement learning to optimize the policy and budget allocation. Both variants rely on a pretrained world model, while ITP-R additionally uses expert data and environment rewards for policy learning.

CoMAP. CoMAP (Liu et al., 2026b) jointly trains an agent policy and a textual world model. The policy uses predicted action consequences to refine its decisions, while real environment transitions provide supervision for updating the world model. Training begins with expert demonstrations to initialize both models, followed by online updates using interaction experience and action-refinement signals derived from subsequent task outcomes.

A.3 Agent RL-based Methods

GRPO. GRPO (Shao et al., 2024) samples a group of trajectories for each task and computes relative advantages from their rewards. It optimizes the policy using group-based advantages without a separate value network, while still relying on environment rewards or task-outcome scores.

GRPO+OPSD. GRPO+OPSD (Zhao et al., 2026) combines reward-based GRPO with On-Policy Self-Distillation (OPSD). In addition to the trajectory-level RL objective, it uses a privileged teacher context to provide token-level supervision over student-generated trajectories. Its policy optimization therefore combines reward-based reinforcement learning with self-distillation signals.

RLSD. RLSD (Yang et al., 2026a) incorporates self-distillation signals into reward-based reinforcement learning to refine token-level credit assignment. It uses teacher–student policy differences to modulate the update magnitude of individual tokens while preserving the trajectory-level advantage. The overall update direction is still determined by task rewards.

SDAR. SDAR (Lu et al., 2026) introduces self-distillation as a gated auxiliary objective on top of reward-based reinforcement learning. It adaptively weights token-level distillation signals according to the discrepancy between the teacher and student policies. The primary policy optimization still relies on task rewards, while self-distillation provides additional fine-grained supervision.

OCSD. OCSD (Yang et al., 2026d) calibrates self-distillation using future observations obtained from real environment interactions. It contrasts full and observation-ablated replay contexts to isolate the information contributed by future observations and uses this signal to modulate token-level policy updates. Its trajectory-level optimization remains driven by task rewards.

A.4 OPD-BASED METHODS

Vanilla OPD. Vanilla OPD collects online trajectories through student-environment interactions, while an external teacher provides action or token-level probability distributions at student-visited states. The student updates its policy by matching the teacher's distributions, without requiring terminal task rewards.

AdaSwitch. AdaSwitch (Peng et al., 2025) adaptively switches between student and teacher generation based on their predictive discrepancies, balancing student exploration and teacher guidance. The student is subsequently distilled using teacher-provided distributions. When applied to interactive agents, the switching mechanism operates during action generation.

TCOD. TCOD (Wang et al., 2026b) progressively adjusts the interaction horizon in online policy distillation. Its F2B variant gradually extends student-controlled trajectory prefixes, while B2F starts from prefixes of pre-collected successful teacher trajectories and lets the student complete the remaining interactions. Both variants use teacher distributions for distillation.

Guided-OPD. Guided-OPD (Li et al., 2026) alternates between student and teacher actions within online trajectories and gradually reduces teacher intervention during training. The student updates its policy using teacher supervision, progressively transitioning from teacher-guided interactions to independent decision-making without requiring separately collected expert trajectories.

A.5 BENCHMARKS

ALFWorld. ALFWorld (Shridhar et al., 2020) is a text-based embodied agent benchmark for multi-step household tasks. Given a natural language instruction, the agent must explore the environment, locate relevant objects, and perform actions such as picking up, placing, cleaning, heating, and cooling. Completing a task requires sequential decision making based on environment feedback.

WebShop. WebShop (Yao et al., 2022a) is an interactive shopping benchmark in which agents follow natural language requests to find and purchase products. The agent searches for products, navigates pages, inspects product information, and selects attributes to satisfy constraints such as category, price, color, and size.

B PROMPT TEMPLATES

The templates below are the runtime forms used in our experiments. Braced fields are filled with the task, current state, interaction history, admissible actions, and additional interaction evidence.

B.1 ALFWORLD

Direct Action Selection Template

```
[system]
Interact with a household environment to complete the task.
[user]
Task: {task_description}
Observation: {initial_observation}
{task_and_history}
Available actions:
{available_actions}
Rules:
- Choose exactly one action from Available actions.
```

- Copy the action exactly.
- If the previous action had no effect, choose another useful action.
- Output no reasoning or extra text.

Output exactly:
Action: <one exact current admissible action>

Candidate Action Generation Template

[system]
{alfworld_system_prompt}

[user]
{task_and_history}

Available actions:
{available_actions}

Rules:
- Select exactly {branch_count} useful and diverse actions.
- Choose only from the current Available actions.
- Copy each action exactly and output no explanations.

Output exactly {branch_count} numbered lines:
Action 1: <one exact current admissible action>
...

Feedback-Guided Action Selection Template

[system]
{alfworld_system_prompt}

[user]
{task_and_history}

Available actions:
{available_actions}

Additional interaction evidence:
{preview_evidence}

Rules:
- Use the additional evidence only to inform the current decision.
- Choose exactly one action from the current Available actions.
- Copy the action exactly and output no reasoning or extra text.

Output exactly:
Action: <one exact current admissible action>

B.2 WEBSHOP

Direct Action Selection Template

<system>
Interact with a WebShop environment to complete the shopping task.
</system>
<user>
Task: {task_description}
{real_history}

Current state:
Page: {page}
Current product: {current_product}
Selected options: {selected_options}

Available actions:
[
{available_actions}
]

Rules:
- Choose exactly one action from Available actions.
- Copy the action exactly.
- Output no reasoning or extra text.

Output exactly:
Action: <one exact current admissible action>
</user>

Candidate Action Generation Template

```
<system>
Interact with a WebShop environment to complete the shopping task.
</system>
<user>
Task: {task_description}
{real_history}
Current state:
Page: {page}
Current product: {current_product}
Selected options: {selected_options}
Available actions:
[
{available_actions}
]
Rules:
- Select exactly {branch_count} distinct and useful actions.
- Choose only from the current Available actions.
- Prefer actions that reveal task-relevant product, option, or page information.
- Copy each action exactly and output no reasoning or extra text.
Output exactly {branch_count} numbered lines:
Action 1: <one exact current admissible action>
Action 2: <one exact current admissible action>
...
Action {branch_count}: <one exact current admissible action>
</user>
```

Feedback-Guided Action Selection Template

```
<system>
Interact with a WebShop environment to complete the shopping task.
</system>
<user>
Task: {task_description}
{real_history}
Additional interaction evidence:
{candidate_evidence}
Current real state:
Page: {page}
Current product: {current_product}
Selected options: {selected_options}
Requested options: {requested_options}
Available actions:
[
{available_actions}
]
Rules:
- Treat the current real state as authoritative.
- Use the additional interactions only as evidence; their actions and selections have not occurred in the current state.
- Choose exactly one action from the current Available actions.
- Before buying, ensure all required option values are selected in the current real state.
- Match requested option values exactly and do not substitute similar values.
- Copy the action exactly and output no reasoning or extra text.
Output exactly:
Action: <one exact current admissible action>
</user>
```

C Case Studies

C.1 ALFWorld: Lookahead Avoids Inefficient Exploration

Task. put a clean plate in countertop. At the decision point the agent has just reached countertop 2.

ALFWorld Candidate Action Generation Template

[system]
Interact with a household environment to complete the task.

[user]
Now it's your turn.
You are in the middle of a room. Looking quickly around you, you see a cabinet 6, a cabinet 5, a cabinet 4, a cabinet 3, a cabinet 2, a cabinet 1, a coffeemachine 1, a countertop 3, a countertop 2, a countertop 1, a drawer 3, a drawer 2, a drawer 1, a fridge 1, a garbagecan 1, a microwave 1, a shelf 3, a shelf 2, a shelf 1, a sinkbasin 1, a stoveburner 4, a stoveburner 3, a stoveburner 2, a stoveburner 1, and a toaster 1.
Your task is to: put a clean plate in countertop.

[assistant]
Action: go to countertop 1

[user]
Observation: You arrive at countertop 1. On the countertop 1, you see a dishsponge 1, a egg 2, and a glassbottle 2.

[assistant]
Action: go to sinkbasin 1

[user]
Observation: You arrive at sinkbasin 1. On the sinkbasin 1, you see nothing.

[assistant]
Action: go to countertop 2

[user]
Observation: You arrive at countertop 2. On the countertop 2, you see a bread 1, a cellphone 1, a creditcard 2, a creditcard 1, a lettuce 1, a mug 1, a plate 3, a spatula 1, and a tomato 1.

Available actions:
- examine countertop 2
- go to cabinet 1
- go to cabinet 2
- go to cabinet 3
- go to cabinet 4
- go to cabinet 5
- go to cabinet 6
- go to coffeemachine 1
- go to countertop 1
- go to countertop 3
- go to drawer 1
- go to drawer 2
- go to drawer 3
- go to fridge 1
- go to garbagecan 1
- go to microwave 1
- go to shelf 1
- go to shelf 2
- go to shelf 3
- go to sinkbasin 1
- go to stoveburner 1
- go to stoveburner 2
- go to stoveburner 3
- go to stoveburner 4
- go to toaster 1
- help
- inventory
- look
- take bread 1 from countertop 2
- take cellphone 1 from countertop 2
- take creditcard 1 from countertop 2
- take creditcard 2 from countertop 2
- take lettuce 1 from countertop 2
- take mug 1 from countertop 2
- take plate 3 from countertop 2
- take spatula 1 from countertop 2
- take tomato 1 from countertop 2

Rules:
- Select exactly 4 useful and diverse actions.
- Choose only from the current Available actions.
- Copy each action exactly and output no explanations.

Output exactly 4 numbered lines:
Action 1: <one exact current admissible action>
...

Candidate Actions

Action 1: examine countertop 2
Action 2: take plate 3 from countertop 2
Action 3: go to sinkbasin 1
Action 4: go to fridge 1

ALFWorld Feedback-Guided Action Selection Template

[system]
Interact with a household environment to complete the task.

[user]
Now it's your turn.
You are in the middle of a room. Looking quickly around you, you see a cabinet 6, a cabinet 5, a cabinet 4, a cabinet 3, a cabinet 2, a cabinet 1, a coffeemachine 1, a countertop 3, a countertop 2, a countertop 1, a drawer 3, a drawer 2, a drawer 1, a fridge 1, a garbagecan 1, a microwave 1, a shelf 3, a shelf 2, a shelf 1, a sinkbasin 1, a stoveburner 4, a stoveburner 3, a stoveburner 2, a stoveburner 1, and a toaster 1.
Your task is to: put a clean plate in countertop.

[assistant]
Action: go to countertop 1

[user]
Observation: You arrive at countertop 1. On the countertop 1, you see a dishsponge 1, a egg 2, and a glassbottle 2.

[assistant]
Action: go to sinkbasin 1

[user]
Observation: You arrive at sinkbasin 1. On the sinkbasin 1, you see nothing.

[assistant]
Action: go to countertop 2

[user]
Observation: You arrive at countertop 2. On the countertop 2, you see a bread 1, a cellphone 1, a creditcard 2, a creditcard 1, a lettuce 1, a mug 1, a plate 3, a spatula 1, and a tomato 1.

Available actions:
- examine countertop 2
- go to cabinet 1
- go to cabinet 2
- go to cabinet 3
- go to cabinet 4
- go to cabinet 5
- go to cabinet 6
- go to coffeemachine 1
- go to countertop 1
- go to countertop 3
- go to drawer 1
- go to drawer 2
- go to drawer 3
- go to fridge 1
- go to garbagecan 1
- go to microwave 1
- go to shelf 1
- go to shelf 2
- go to shelf 3
- go to sinkbasin 1
- go to stoveburner 1
- go to stoveburner 2
- go to stoveburner 3
- go to stoveburner 4
- go to toaster 1
- help
- inventory
- look
- take bread 1 from countertop 2
- take cellphone 1 from countertop 2
- take creditcard 1 from countertop 2
- take creditcard 2 from countertop 2
- take lettuce 1 from countertop 2
- take mug 1 from countertop 2
- take plate 3 from countertop 2
- take spatula 1 from countertop 2
- take tomato 1 from countertop 2

Additional interaction evidence:
- Sampled future 1:

step 1 executed action: examine countertop 2
step 1 next observation: On the countertop 2, you see a bread 1, a cellphone 1, a creditcard 2, a creditcard 1, a lettuce 1, a mug 1, a plate 3, a spatula 1, and a tomato 1.
step 2 executed action: take plate 3 from countertop 2
step 2 next observation: You pick up the plate 3 from the countertop 2.
- Sampled future 2:
step 1 executed action: take plate 3 from countertop 2
step 1 next observation: You pick up the plate 3 from the countertop 2.
step 2 executed action: move plate 3 to countertop 2
step 2 next observation: You move the plate 3 to the countertop 2.
- Sampled future 3:
step 1 executed action: go to sinkbasin 1
step 1 next observation: You arrive at sinkbasin 1. On the sinkbasin 1, you see nothing.
step 2 executed action: go to countertop 3
step 2 next observation: You arrive at countertop 3. On the countertop 3, you see a bowl 3, a butterknife 2, a butterknife 1, a fork 2, a houseplant 1, a soapbottle 2, a spatula 2, a spoon 1, and a statue 1.
- Sampled future 4:
step 1 executed action: go to fridge 1
step 1 next observation: You arrive at fridge 1. The fridge 1 is closed.
step 2 executed action: open fridge 1
step 2 next observation: You open the fridge 1. The fridge 1 is open. In it, you see a apple 2, a bowl 2, a cup 1, a potato 3, a potato 2, and a potato 1.

Rules:
- Use the additional evidence only to inform the current decision.
- Choose exactly one action from the current Available actions.
- Copy the action exactly and output no reasoning or extra text.

Output exactly:
Action: <one exact current admissible action>

Final action

Action: take plate 3 from countertop 2

What the evidence changes

Without candidate lookahead. The direct run ends with `Action: move plate 3 to countertop 1`; its executed route is:

1. `go to countertop 1`
2. `examine countertop 1`
3. `go to sinkbasin 1`
4. `go to dishsponge 1`
5. `examine sinkbasin 1`
6. `go to countertop 2`
7. `take plate 3 from countertop 2`
8. `go to countertop 1`
9. `move plate 3 to countertop 1`
10. `go to sinkbasin 1`
11. `go to sinkbasin 1`
12. `examine sinkbasin 1`
13. `go to countertop 1`
14. `take dishsponge 1 from countertop 1`
15. `move dishsponge 1 to countertop 1`
16. `go to sinkbasin 1`
17. `examine sinkbasin 1`
18. `go to countertop 1`

19. `take plate 3 from countertop 1`
20. `move plate 3 to countertop 1`
21. `go to sinkbasin 1`
22. `go to sinkbasin 1`
23. `examine sinkbasin 1`
24. `go to countertop 1`
25. `take plate 3 from countertop 1`
26. `examine plate 3`
27. `examine plate 3`
28. `go to sinkbasin 1`
29. `clean plate 3 with sinkbasin 1`
30. `move plate 3 to countertop 1`

It cleans and places the plate only at the 30-step interaction limit, and the episode is therefore counted as unsuccessful.

With candidate lookahead. The executed route is:

1. `go to countertop 1`
2. `examine countertop 1`
3. `go to sinkbasin 1`
4. `go to countertop 1`
5. `go to countertop 2`
6. `take plate 3 from countertop 2`
7. `go to sinkbasin 1`
8. `clean plate 3 with sinkbasin 1`
9. `go to countertop 3`
10. `move plate 3 to countertop 3`

It cleans the plate and places it at countertop 3, reaching the successful terminal state.

C.2 WEBSHOP: EIGHT REAL BRANCHES FIND THE EXACT PRODUCT

Task. Find me slip resistant, non slip men's loafers & slip-ons with rubber outsole, rubber sole with color: 228r.brown, and size: 7, and price lower than 70.00 dollars.

WebShop Candidate Action Generation Template

<system>
Interact with a WebShop environment to complete the shopping task.
</system>
<user>
Task: Find me slip resistant, non slip men's loafers & slip-ons with rubber outsole, rubber sole with color: 228r.brown, and size: 7, and price lower than 70.00 dollars
<assistant>
Action: search[slip resistant, non slip men's loafers & slip-ons with rubber outsole, rubber sole color: 228r.brown, size: 7, price lower than 70.00 dollars]
</assistant>
<user>
Observation: Instruction:
Find me slip resistant, non slip men's loafers & slip-ons with rubber outsole, rubber sole with color: 228r.brown, and size: 7, and price lower than 70.00 dollars
Back to Search
Page 1 (Total results: 50)
Next >

B07HP6LVRS
MCICI Mens Loafers Moccasin Driving Shoes Premium Genuine Leather Casual Slip On Flats Fashion Slipper Breathable Big Size
$15.99 to $32.99
B082MT9162
GLOBALWIN Men's Casual Slip On Penny Loafers Lightweight Driving Shoes
$29.99 to $49.99
B07S7HDC88
VanciLin Mens Casual Leather Fashion Slip-on Loafers
$33.99 to $35.99
B09S6VN97V
Skechers Women's, Relaxed Fit: Arch Fit - Commute Clog Taupe 11 M
$74.95
B07RPMCNKD
ULTRAIDEAS Women's Comfy Lightweight Slippers Non-Slip House Shoes for Indoor & Outdoor
$19.99 to $21.99
B008QYOGZ2
OluKai MEA Ola - Men's Supportive Sandal Charcoal/Dkjava - 14
$100.0
B07CN9RZ1C
Ecco Men's Melbourne Plain Toe Slip on Loafer
$114.99
B09F2MFG1Q
Womens Mens house slippers Memory Foam Garden Plush Lining Slip On Rubber sole Indoor Outdoor House Casual Shoes (Black adult-10.5-11)
$100.0
B06W51MMKY
Alegria Women's Kourtney
$62.99 to $129.99
B09QXF3V3X
DEUVOUM Summer Trend Mesh Shoes Men's Sports Shoes Solid Color Lace-Up Sneakers Fashion All-Match Walking Shoes Outdoor Hiking Shoes Non-Slip Shock-Absorbing Casual Sports Shoes
$100.0

Current state:
Page: search_results
Current product: none
Selected options: unknown

Available actions:
[
'click[back to search]',
'click[next >]',
'click[b07hp6lvrs]',
'click[b082mt9162]',
'click[b07s7hdc88]',
'click[b09s6vn97v]',
'click[b07rpmcnkd]',
'click[b008qyogz2]',
'click[b07cn9rz1c]',
'click[b09f2mfg1q]',
'click[b06w51mmky]',
'click[b09qxf3v3x]'
]

Rules:
- Select exactly 8 distinct and useful actions.
- Choose only from the current Available actions.
- Prefer actions that reveal task-relevant product, option, or page information.
- Copy each action exactly and output no reasoning or extra text.

Output exactly 8 numbered lines:
Action 1: <one exact current admissible action>
Action 2: <one exact current admissible action>
...
Action 8: <one exact current admissible action>
</user>

Candidate actions

Action 1: click[b07hp6lvrs]
Action 2: click[b082mt9162]
Action 3: click[b07s7hdc88]
Action 4: click[next >]
Action 5: click[b07rpmcnkd]
Action 6: click[b008qyogz2]

Action 7: click[b07cn9rz1c]
Action 8: click[b09f2mfg1q]

WebShop Feedback-Guided Action Selection Template

<system>
Interact with a WebShop environment to complete the shopping task.
</system>
<user>
Task: Find me slip resistant, non slip men's loafers & slip-ons with rubber outsole, rubber sole with color: 228r.brown, and size: 7, and price lower than 70.00 dollars
<assistant>
Action: search[slip resistant, non slip men's loafers & slip-ons with rubber outsole, rubber sole color: 228r.brown, size: 7, price lower than 70.00 dollars]
</assistant>
<user>
Observation: Instruction:
Find me slip resistant, non slip men's loafers & slip-ons with rubber outsole, rubber sole with color: 228r.brown, and size: 7, and price lower than 70.00 dollars
Back to Search
Page 1 (Total results: 50)
Next >
B07HP6LVRS
MCICI Mens Loafers Moccasin Driving Shoes Premium Genuine Leather Casual Slip On Flats Fashion Slipper Breathable Big Size
$15.99 to $32.99
B082MT9162
GLOBALWIN Men's Casual Slip On Penny Loafers Lightweight Driving Shoes
$29.99 to $49.99
B07S7HDC88
VanciLin Mens Casual Leather Fashion Slip-on Loafers
$33.99 to $35.99
B09S6VN97V
Skechers Women's, Relaxed Fit: Arch Fit - Commute Clog Taupe 11 M
$74.95
B07RPMCNKD
ULTRAIDEAS Women's Comfy Lightweight Slippers Non-Slip House Shoes for Indoor & Outdoor
$19.99 to $21.99
B008QYOGZ2
OluKai MEA Ola - Men's Supportive Sandal Charcoal/Dkjava - 14
$100.0
B07CN9RZ1C
Ecco Men's Melbourne Plain Toe Slip on Loafer
$114.99
B09F2MFG1Q
Womens Mens house slippers Memory Foam Garden Plush Lining Slip On Rubber sole Indoor Outdoor House Casual Shoes (Black adult-10.5-11)
$100.0
B06W51MMKY
Alegria Women's Kourtney
$62.99 to $129.99
B09QXF3V3X
DEUVOUM Summer Trend Mesh Shoes Men's Sports Shoes Solid Color Lace-Up Sneakers Fashion All-Match Walking Shoes Outdoor Hiking Shoes Non-Slip Shock-Absorbing Casual Sports Shoes
$100.0
Additional interaction evidence:
- Sampled future 1:
step 1 executed action: click[b07hp6lvrs]
- Page observed: item_page
- Observable selected options: unknown
- Product title: MCICI Mens Loafers Moccasin Driving Shoes Premium Genuine Leather Casual Slip On Flats Fashion Slipper Breathable Big Size
- Product price: $15.99 to $32.99
- Visible size options: 6.5; 7; 8; 8.5; 9.5; 10; 10.5; 11; 11.5
- Visible color options: black; blue; brown; white; yellow; black-b; blue-a; yellow-a
- Page changed from search_results to item_page.
- Current product became MCICI Mens Loafers Moccasin Driving Shoes Premium Genuine Leather Casual Slip On Flats Fashion Slipper Breathable Big Size.
- Sampled future 2:
step 1 executed action: click[b082mt9162]
- Page observed: item_page
- Observable selected options: unknown
- Product title: GLOBALWIN Men's Casual Slip On Penny Loafers Lightweight Driving Shoes

- Product price: $29.99 to $49.99
- Visible size options: 7; 7.5 (9 other visible values omitted from this compact evidence)
- Visible color options: 2005brown; 2005grey; 2005black
- Page changed from search_results to item_page.
- Current product became GLOBALWIN Men's Casual Slip On Penny Loafers Lightweight Driving Shoes.
- Sampled future 3:
step 1 executed action: click[b07s7hdc88]
- Page observed: item_page
- Observable selected options: unknown
- Product title: VanciLin Mens Casual Leather Fashion Slip-on Loafers
- Product price: $33.99 to $35.99
- Visible size options: 7; 8 (11 other visible values omitted from this compact evidence)
- Visible color options: 228r.brown (23 other visible values omitted from this compact evidence)
- Page changed from search_results to item_page.
- Current product became VanciLin Mens Casual Leather Fashion Slip-on Loafers.
- Sampled future 4:
step 1 executed action: click[next >]
- Page observed: search_results
- Observable selected options: unknown
- Search result page: Page 2 (Total results: 50)
- Observed product: B09HMCKZQW Copercn Dress Ankle Booties For Women Ladies Fashion Casual Chunky Block High Heels Dress Pump Thermal Short Boots Winter Fall Dressy Short Boots For Business Work Wedding Party Dresses $17.99 to $23.99 | $17.99 to $23.99
- Observed product: B094RBNS2K Mens Running Tennis Blade Shoes Lightweight Casual Walking Sneakers Mesh Breathable Sport Sneakers Sheet Shoes $27.0 | $27.0
- Observed product: B09QRV4YLR SWEETSHOPZ Irish Dance Shape Background Active Running Walking Sneakers Shoes for Women Men Unisex Kids Adults Fashion Sports Black $100.0 | $100.0
- Observed product: B09RPRRNVP ZSWWang Slip On Sneakers Flats Shoes for Women Walking Sock Shoes Lightweight Breathable Mesh Knit Yoga Sneakers Fashion Casual Comfortable Platform Wedge Work Outdoor Walking Sports Shoes $15.99 | $15.99
- Observed product: B07MNW91JH Easy Works by Easy Street womens Leeza Clog, Black, 7.5 Wide US $69.97 | $69.97
- Observed product: B09JC3K6R6 5 Pairs Skin Color No Show Socks Women Lace Low Cut Liner Socks Non Slip Flat Boat Crew Socks (2.39/pair) $11.99 | $11.99
- Observed product: B001J6NXFS TOMS Womens Classic Canvas Slip-On,White,5.5 M US $76.24 | $76.24
- Observed product: B09B9QLDKD Professional Pedicure Tools Kit, 20/23 in 1 Stainless Steel Foot Spa Care Set Foot Rasp Dead Skin Callus Remover Foot File Foot Scrubber for Women Men Salon or Home Use $100.0 | $100.0
- Observed product: B08XH9JMTG meyarn Interdental Brush for Braces Toothpick 50Count Tooth Floss Oral Hygiene Interdental Brush Toothpick Teeth Healthy Care Tight 0.8mm $11.89 | $11.89
- Observed product: B09HH6P68W MAPOLO Hand Drawn Christmas Elements Collection Men's Boys Casual Walking Shoes Sneaker Lightweight Stylish Athletic Tennis Sports Running Shoes for Outdoor Hiking Travel Driving $100.0 | $100.0
- Page changed from search_results to search_results.
...

Current real state:
Page: search_results
Current product: none
Selected options: unknown
Requested options: color=228r.brown; size=7

Available actions:
[
'click[back to search]',
'click[next >]',
'click[b07hp6lvrs]',
'click[b082mt9162]',
'click[b07s7hdc88]',
'click[b09s6vn97v]',
'click[b07rpmcnkd]',
'click[b008qyogz2]',
'click[b07cn9rz1c]',
'click[b09f2mfg1q]',
'click[b06w51mmky]',
'click[b09qxf3v3x]'
]

Rules:
- Treat the current real state as authoritative.
- Use the additional interactions only as evidence; their actions and selections have not occurred in the current state.
- Choose exactly one action from the current Available actions.
- Before buying, ensure all required option values are selected in the current real state.
- Match requested option values exactly and do not substitute similar values.
- Copy the action exactly and output no reasoning or extra text.

Output exactly:
Action: <one exact current admissible action>
</user>

Final action

Action: click[b07s7hdc88]

What the preview evidence changes

Direct failure. The no-preview route is:

1. `search[slip resistant, non slip men's loafers & slip-ons with rubber outsole, rubber sole color: 228r.brown, size: 7, price lower than 70.00 dollars]`
2. `click[b07hp6lvrs]`
3. `click[7]`
4. `click[brown]`
5. `click[description]`
6. `click[back to search]`
7. `search[slip resistant, non slip men's loafers & slip-ons rubber outsole rubber sole color: 228r.brown size: 7 price lower than 70.00 dollars]`
8. `click[b07hp6lvrs]`
9. `click[7]`
10. `click[brown]`
11. `click[buy now]`

It purchases ASIN B07HP6LVRS with `color=brown` and `size=7`, scoring 0.7143; the requested color is `228r.brown`.

Branch comparison. Branches 1–2 have size 7 but no exact `228r.brown`; branch 3 exposes both required labels at 33.99–35.99; branch 4 changes to page 2.

Lookahead success. The selector outputs `Action: click[b07s7hdc88]`; the executed route is:

1. `search[slip resistant, non slip men's loafers & slip-ons with rubber outsole, rubber sole color: 228r.brown, size: 7, price lower than 70.00 dollars]`
2. `click[b07s7hdc88]`
3. `click[7]`
4. `click[228r.brown]`
5. `click[buy now]`

The terminal purchase records `color=228r.brown` and `size=7` with score 1.0.