SEED: SELF-EVOLVING ON-POLICY DISTILLATION FOR AGENTIC REINFORCEMENT LEARNING

Jinyang Wu[1*†], **Shuo Yang**[1*], **Zhengxi Lu**[2], **Fan Zhang**[3], **Yuhao Shen**[2], **Lang Feng**[4], **Haoran Luo**[4], **Zheng Lian**[5], **Shuai Zhang**[1], **Zhengqi Wen**[1], **Jianhua Tao**[1]

[1]Tsinghua University [2]Zhejiang University [3]The Chinese University of Hong Kong
[4]Nanyang Technological University [5]Tongji University
Corresponding to: wu-jy23@mails.tsinghua.edu.cn

Project Page | Code | Model

ABSTRACT

Large language models are increasingly trained as interactive agents for long-horizon tasks involving multi-turn interaction, tool use, and environment feedback. Outcome-based reinforcement learning (RL) provides a practical optimization paradigm, but its sparse trajectory-level rewards offer limited guidance on intermediate decisions, leaving a supervision gap between episode-level outcomes and token-level policy learning. We propose **SEED** (**SE**lf-**E**volving On-Policy **D**istillation), a self-evolving framework that converts completed on-policy trajectories into training-time hindsight skills and distills their behavioral effect back into the policy model. SEED first fine-tunes the policy to analyze completed trajectories and generate natural-language skills that capture reusable workflows, decisive observations, or failure-avoidance rules. During RL, the current policy both collects trajectories and serves as the analyzer that extracts hindsight skills from them. Policy updates therefore improve subsequent decision making and skill analysis together, allowing hindsight supervision to evolve with the policy. SEED then re-scores the sampled actions under ordinary and skill-augmented contexts, converting the skill-induced probability shift into a dense token-level on-policy distillation signal. This signal is jointly optimized with outcome-based RL, keeping the auxiliary supervision aligned with the current trajectory distribution. Extensive experiments on text-based and vision-based agentic tasks show that SEED consistently improves performance and sample efficiency, exhibiting robust generalization to unseen scenarios. Our code is available at jinyangwu/SEED.

1 INTRODUCTION

Recent large language model (LLM) systems are moving beyond single-turn response generation toward multi-turn agentic interaction, where a model repeatedly reasons, acts, uses tools, and incorporates feedback from its environment (Liu et al., 2023; Schick et al., 2023; Patil et al., 2024; Luo et al., 2025a; Xi et al., 2025; Wu et al., 2026b; Xu et al., 2026). Such settings require an agent to make sequential decisions whose consequences may only become visible after many interaction steps. The model must learn when to gather information, when to call a tool, how to interpret feedback, and how to revise a plan after partial progress or failure. Reinforcement learning (RL) has therefore become an important post-training paradigm for LLM-based agents, since it directly optimizes policies against task-level feedback from environments, simulators, or verifiers (Shao et al., 2024; Wang et al., 2025; Luo et al., 2025b; Wu et al., 2026a).

Despite this progress, outcome-based agentic RL provides only coarse supervision. In long-horizon environments, rewards are often sparse, delayed, and assigned at the trajectory level: they indicate whether an episode succeeds but not which intermediate observations, actions, or tool calls should

*Equal Contribution
†Project Leader

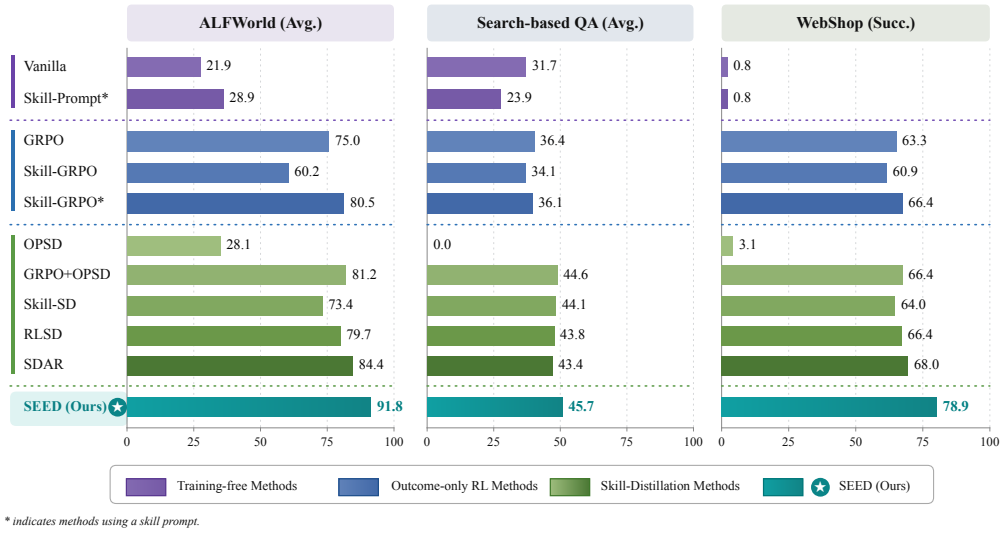

Figure 1: **Overall performance overview.** Compared with powerful baseline methods, SEED achieves the strongest average performance across three representative agentic benchmarks.

be reinforced or corrected (Andrychowicz et al., 2017; Arjona-Medina et al., 2019; Uesato et al., 2022; Lightman et al., 2024). This leaves a supervision gap between episode-level outcomes and token-level policy learning. A failed trajectory may contain useful partial behaviors but fail because of a few local mistakes, whereas a successful trajectory may contain reusable strategies that the scalar reward never identifies. As a result, outcome-only optimization provides limited guidance for fine-grained, decision-level credit assignment over long interaction histories.

A key observation is that completed trajectories reveal hindsight unavailable during online decision making. Once an episode terminates, the full interaction history reveals which subgoals were achieved, where the agent deviated from an effective strategy, which observations were decisive, and which behavioral patterns may transfer to future attempts. This view is related to hindsight learning in RL, where completed experience can be reinterpreted to improve learning under sparse feedback (Andrychowicz et al., 2017), and to language-agent methods based on verbal reflection, episodic memory, or experience summaries (Shinn et al., 2023; Zhao et al., 2024; Wang et al., 2023; Madaan et al., 2023). However, many such methods treat hindsight as static experience, inference-time context, or retrieved memory. For practical agentic RL, hindsight supervision should not remain fixed: as the policy improves and encounters new states, strategies, and failure modes, the hindsight extracted from its trajectories must adapt to its current behavior (Andrychowicz et al., 2017; Zhang et al., 2026a). Our goal is therefore to convert policy-generated hindsight into parametric supervision, enabling the policy to internalize reusable behavioral guidance without external memory or additional deployment-time prompts.

On-policy distillation (OPD) offers a natural mechanism for converting hindsight information into decision-level learning signals. Classical knowledge distillation transfers teacher behavior into a student through token- or sequence-level supervision (Hinton et al., 2015; Kim & Rush, 2016), while on-policy variants reduce distribution mismatch by supervising outputs sampled from the student policy itself (Ross et al., 2011; Agarwal et al., 2024). Recent on-policy self-distillation methods further avoid a separate external teacher by comparing the same model under different contexts, such as privileged reasoning traces or feedback-conditioned prompts (Zhao et al., 2026; Hübotter et al., 2026). Related agentic methods have explored skill- or feedback-conditioned distillation for multi-turn interaction and tool use (Wang et al., 2026; Lu et al., 2026a; Zhong et al., 2026; Ko et al., 2026; Yang et al., 2026b). Together, these advances point to three requirements for effective hindsight supervision in agentic RL.

First, the supervision should be *on-policy*, because useful corrections depend on the states, actions, and failure modes induced by the current policy. Second, it should be *dense*, allowing trajectory-level hindsight to guide individual decision tokens rather than only the final outcome. Third, it

should be *self-evolving*: as the policy improves, its decision-making and trajectory-analysis capabilities should advance together. Fixed teachers, static skill datasets, and one-time distillation cannot continually adapt to the policy's evolving capabilities and trajectory distribution.

We propose **SEED** (**SE**lf-**E**volving On-Policy **D**istillation), a framework that turns completed on-policy trajectories into hindsight skills and distills their behavioral effect back into the policy model through a self-evolving training loop. During RL, the current policy collects trajectories and also serves as the analyzer that extracts natural-language skills describing reusable workflows, decisive observations, or failure-avoidance rules. Because both roles share the same model, policy updates improve subsequent decision making and skill analysis together, allowing hindsight supervision to evolve with the policy. Given an extracted skill, SEED keeps the sampled actions fixed and rescores them under ordinary and skill-augmented contexts. The resulting probability shift provides a dense token-level OPD signal, which is jointly optimized with the outcome-based RL objective. Specifically, SEED consists of two stages. First, **hindsight-skill supervised fine-tuning** equips the model to analyze completed interaction histories and generate reusable trajectory-level skills. Second, **self-evolving OPD** repeatedly uses the latest policy checkpoint for both trajectory collection and skill analysis, then updates the policy with the joint RL and OPD objective. The generated skills act only as privileged supervision during training and require neither external memory nor additional prompts at inference time.

We evaluate SEED across embodied interaction, web navigation, search-based QA, and visual perception and planning. SEED achieves superior task performance, sample efficiency, and robustness. Taken together, our work makes the following contributions:

- We propose **SEED**, a self-evolving OPD framework that continually transforms the policy's completed trajectories into hindsight skills and internalizes their behavioral guidance during agentic RL, allowing decision-making and skill analysis to improve together.
- We introduce a policy-synchronized hindsight OPD mechanism that converts skill-induced log-probability shifts on sampled actions into dense token-level supervision and jointly optimizes this signal with outcome-based RL.
- Extensive experiments across diverse long-horizon agentic benchmarks show that SEED improves task performance, sample efficiency, and robustness over representative baselines.

2 Related Work

Reinforcement learning for agentic LLMs. LLMs are increasingly trained as interactive agents that reason, use tools, and act over long horizons (Liu et al., 2023; Schick et al., 2023; Patil et al., 2024; Luo et al., 2025a; Xi et al., 2025; Wu et al., 2026b; Xu et al., 2026). Reinforcement learning provides a natural post-training paradigm for such agents by directly optimizing task-level outcomes (Shao et al., 2024; Wang et al., 2025; Luo et al., 2025b; Wu et al., 2026a; Lu et al., 2026b). However, outcome-based RL typically assigns sparse and delayed rewards at the trajectory level, offering limited guidance on which intermediate observations, actions, or tool calls should be reinforced. SEED retains outcome-based RL as the optimization backbone while supplementing it with dense supervision derived from completed trajectories.

Hindsight learning for language agents. Completed trajectories expose reusable strategies, decisive observations, and failure causes that are unavailable during online decision making. Prior work exploits such information through hindsight relabeling, return decomposition, process supervision, verbal reflection, and experience memory (Andrychowicz et al., 2017; Arjona-Medina et al., 2019; Uesato et al., 2022; Lightman et al., 2024; Shinn et al., 2023; Zhao et al., 2024; Wang et al., 2023; Madaan et al., 2023). Nevertheless, hindsight knowledge is often stored as static experience or introduced as additional inference-time context. SEED instead converts completed on-policy trajectories into natural-language skills and internalizes their behavioral guidance into the policy, requiring neither external memory nor skill prompts at inference time.

On-policy self-distillation for agentic RL. Knowledge distillation transfers teacher behavior through token- or sequence-level supervision, while on-policy variants reduce distribution mismatch by training on outputs sampled from the learner itself (Hinton et al., 2015; Kim & Rush, 2016; Ross et al., 2011; Agarwal et al., 2024). Recent methods construct privileged self-teachers from reasoning

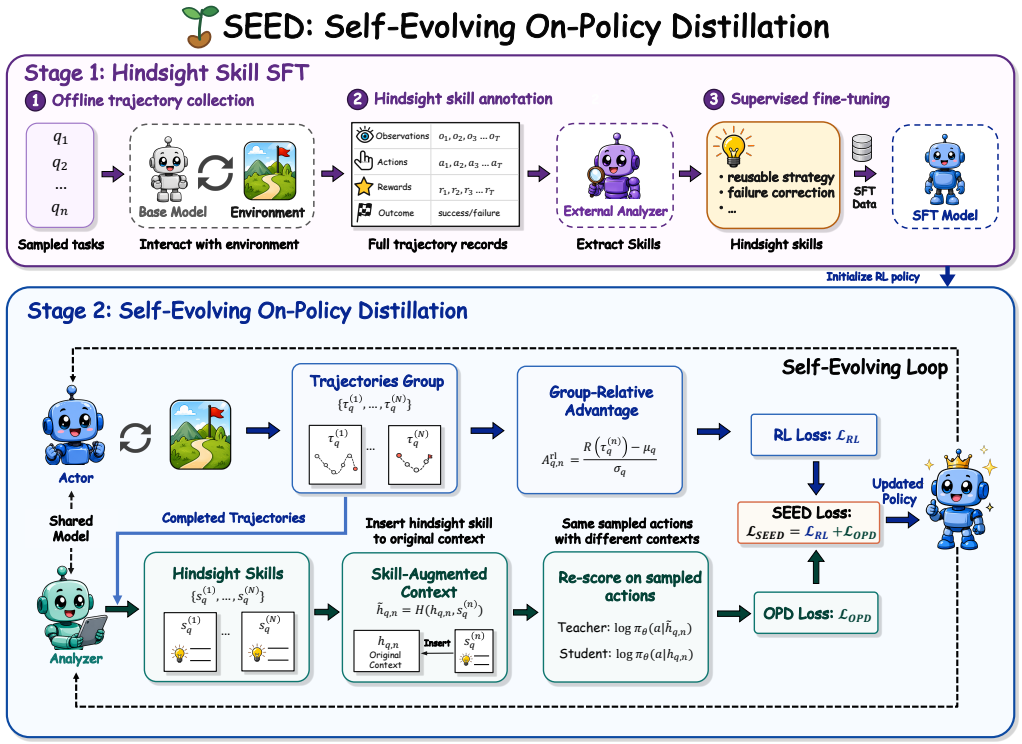

Figure 2: **Overview of SEED. Stage 1** (Hindsight Skill SFT) equips the policy to extract hindsight skills from completed trajectories. **Stage 2** (Self-Evolving On-Policy Distillation) jointly optimizes outcome-based RL and skill-conditioned OPD in a self-evolving agentic loop.

traces, feedback, or skills and use them to provide token-level guidance for agentic RL (Zhao et al., 2026; Wang et al., 2026; Lu et al., 2026a; Zhong et al., 2026; Yang et al., 2026b). However, the source of privileged supervision is often static, externally generated, or updated independently of the policy. As the policy improves and encounters new states and failure modes, such supervision can become stale or mismatched with its current behavior. SEED addresses this limitation through a self-evolving loop in which the latest policy checkpoint serves simultaneously as the rollout actor and the trajectory analyzer. After each policy update, the shared checkpoint is refreshed for both roles, allowing decision making and hindsight supervision to co-evolve.

3 METHOD

We present **SEED**, a self-evolving OPD framework for agentic RL. SEED is motivated by the observation that completed agent trajectories contain rich hindsight information: even when reward is sparse, a full trajectory often reveals useful behavioral patterns, failure causes, and reusable strategies that are not directly available at intermediate decision steps. SEED converts such hindsight information into **hindsight skills** and distills their behavioral effect back into the ordinary policy.

As shown in Figure 2, SEED has two training stages. First, hindsight-skill supervised fine-tuning (SFT) equips a single policy model to analyze completed trajectories. Second, the current policy snapshot both collects on-policy trajectories and analyzes them into hindsight skills. The same sampled actions are then re-scored under ordinary and skill-augmented contexts to construct a token-level OPD signal, which is optimized jointly with RL. At inference time, SEED removes the analyzer, so deployment requires only the learned policy.

3.1 PROBLEM FORMULATION

We formulate long-horizon agentic tasks as partially observable Markov decision processes:

$$(\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \Omega, \mathcal{R}, \gamma)\,,$$

where $\mathcal{S}$ is the latent state space, $\mathcal{A}$ is the action space, $\mathcal{O}$ is the observation space, $\mathcal{T}$ is the transition kernel, Ω is the observation kernel, $\mathcal{R}$ is the reward function, and γ is the discount factor. At timestep t, the agent receives an observation $o_t \in \mathcal{O}$ and maintains an interaction history

$$h_t = (o_0, a_0, o_1, a_1, \ldots, o_t),$$

where a_i denotes the textual response or executable action produced by the agent. The policy generates the next action according to

$$a_t \sim \pi_\theta(\cdot \mid h_t).$$

A completed trajectory is denoted by

$$\tau = \{(o_t, a_t, r_t)\}_{t=0}^{T-1},$$

with an episode-level outcome $R(\tau) \in \mathbb{R}$. In many agentic environments, $R(\tau)$ is sparse and becomes available only after task completion, such as a binary success indicator or a final task score. The standard reinforcement learning objective is

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[R(\tau)].$$

This formulation exposes a key challenge in long-horizon agentic RL: the environment provides only trajectory-level feedback, while the policy must learn from many token-level decisions distributed across the interaction history. SEED bridges this granularity gap by deriving an auxiliary training-time token-level signal from hindsight skills extracted after trajectory completion.

3.2 HINDSIGHT SKILL SUPERVISED FINE-TUNING

The first stage initializes the policy with the ability to analyze full interaction histories and express reusable behavioral guidance as natural-language hindsight skills.

Offline trajectory collection. We first collect an offline pool of trajectories using a base policy without skill augmentation. Let $\mathcal{Q}_{\text{sft}} = \{q_j\}_{j=1}^{M}$ be a set of training tasks. For each task q_j, we run K_0 rollouts with the base policy $\pi_{\theta_{\text{base}}}$:

$$\mathcal{B}_j = \{\tau_{j,k}\}_{k=1}^{K_0}, \qquad \tau_{j,k} \sim \pi_{\theta_{\text{base}}}(\cdot \mid q_j).$$

The complete offline trajectory pool is

$$\mathcal{B} = \bigcup_{j=1}^{M} \mathcal{B}_j.$$

Each trajectory contains the task description, observations, actions, rewards, and final outcome. These trajectories are collected without any hindsight skill in the decision context, ensuring that the SFT data are derived from ordinary agent-environment interaction.

Hindsight skill annotation. Given a completed trajectory τ, an external analyzer A_{ext} produces a hindsight-skill annotation:

$$s_\tau = A_{\text{ext}}(\tau),$$

where s_τ denotes the skill annotation produced from trajectory τ. For a successful trajectory, s_τ typically captures reusable strategies or workflows that contributed to task completion. For a failed trajectory, s_τ can encode corrective or avoidance guidance inferred from the observed failure.

We retain the annotation only if the generated skill is correctly formatted. Let $v_\tau \in \{0, 1\}$ denote the validity indicator. The accepted SFT set is

$$\mathcal{D}_{\text{sft}} = \{(x_\tau, s_\tau) : \tau \in \mathcal{B}, v_\tau = 1\},$$

where x_τ is the serialized trajectory-analysis input and s_τ is the corresponding hindsight skill target.

Supervised fine-tuning. We then fine-tune the policy model to predict the hindsight skill from the completed trajectory. The same autoregressive model later initializes both the RL actor and the synchronized trajectory analyzer. For an accepted SFT example $(x_\tau, s_\tau) \in \mathcal{D}_{\text{sft}}$, the model is optimized with the standard negative log-likelihood objective:

$$\mathcal{L}_{\text{sft}}(\theta) = -\mathbb{E}_{(x_\tau, s_\tau)\sim\mathcal{D}_{\text{sft}}} \left[\sum_{\ell=1}^{|s_\tau|} \log \pi_\theta(s_{\tau,\ell} \mid x_\tau, s_{\tau,<\ell}) \right],$$

where $s_{\tau,\ell}$ is the ℓ-th token of s_τ. The resulting checkpoint θ_{sft} initializes the later RL policy.

During RL, the trajectory analyzer is instantiated directly from the current policy checkpoint. Thus, the same model can act in the environment under ordinary interaction histories and generate hindsight skills from completed trajectories, without a separately trained analyzer.

3.3 SELF-EVOLVING ON-POLICY DISTILLATION

The second stage performs agentic RL with an additional token-level OPD signal. At the start of each update, SEED freezes the current policy as $\pi_{\theta_{\text{old}}}$. This snapshot collects trajectories and also parameterizes the analyzer that extracts their hindsight skills. A trainable policy π_θ, initialized from $\pi_{\theta_{\text{old}}}$, is then optimized with both the environment-driven GRPO and OPD objectives. After the update, the optimized policy π_θ becomes the policy snapshot for the next iteration.

This design keeps the distillation signal on-policy: the policy collects its own trajectories, the analyzer agent converts these completed trajectories into hindsight skills, and the behavioral effect of such hindsight guidance is distilled back into the ordinary policy.

On-policy hindsight skill generation. For each task prompt q, SEED samples a group of N trajectories using the frozen policy:

$$\mathcal{G}_q = \left\{ \tau_q^{(1)}, \tau_q^{(2)}, \ldots, \tau_q^{(N)} \right\}, \qquad \tau_q^{(n)} \sim \pi_{\theta_{\text{old}}}(\cdot \mid q).$$

For each completed trajectory $\tau_q^{(n)}$, SEED constructs its trajectory-analysis input $x_{\tau_q^{(n)}}$. The analyzer agent instantiated from the same policy snapshot $\pi_{\theta_{\text{old}}}$ then analyzes the completed trajectory and generates a hindsight skill:

$$s_q^{(n)} = A_{\theta_{\text{old}}}\left(x_{\tau_q^{(n)}}\right),$$

where $A_{\theta_{\text{old}}}$ denotes the analyzer role of the shared model. Although the actor and analyzer share the same parameters at each update, they play different roles: the actor interacts with the environment, while the analyzer summarizes completed trajectories into trajectory-level hindsight skills. The generated skill $s_q^{(n)}$ thus provides reusable behavioral guidance.

This shared parameterization creates SEED's self-evolving loop. Refreshing θ_{old} changes both the trajectories encountered by the actor and the model capability used for skill analysis. Consequently, the experience distribution and its hindsight supervision evolve together.

On-policy distillation objective. SEED keeps the original on-policy actions fixed and re-scores them under a skill-augmented context. Let H be a deterministic context augmentation function that incorporates the generated skill into the ordinary interaction history. At timestep t in trajectory $\tau_q^{(n)}$, the skill-augmented history is

$$\tilde{h}_{q,n,t} = H\left(h_{q,n,t}, s_q^{(n)}\right).$$

The original sampled action is tokenized as

$$a_{q,n,t} = \left(a_{q,n,t,1}, \ldots, a_{q,n,t,L_{q,n,t}}\right),$$

where $L_{q,n,t}$ denotes the number of tokens in the sampled action $a_{q,n,t}$.

The same policy computes two token-level log-probabilities on the same sampled action tokens. The first is the skill-conditioned teacher branch log-probability, obtained by re-scoring the sampled action under the skill-augmented history:

$$\ell^{\text{skill}}_{q,n,t,\ell} = \log \pi_\theta\left(a_{q,n,t,\ell} \mid \tilde{h}_{q,n,t}, a_{q,n,t,<\ell}\right).$$

The second is the ordinary student branch log-probability under the original interaction history:

$$\ell^{\theta}_{q,n,t,\ell} = \log \pi_\theta \left(a_{q,n,t,\ell} \mid h_{q,n,t}, a_{q,n,t,<\ell}\right).$$

Although both branches share π_θ, they correspond to different input contexts: the teacher observes the hindsight skill, while the student acts only from the ordinary history. The teacher provides a detached training-time signal, and gradients flow exclusively through the ordinary student branch. In summary, during optimization, both branches are evaluated using the current trainable policy π_θ, whereas $\pi_{\theta_{\text{old}}}$ is used for trajectory collection, skill generation, and importance-ratio computation.

We define the detached skill-induced log-probability shift

$$\Delta_{q,n,t,\ell} = \text{sg}\left[\ell^{\text{skill}}_{q,n,t,\ell} - \ell^{\theta}_{q,n,t,\ell}\right],$$

where $\text{sg}[\cdot]$ denotes stop-gradient. Following SDAR (Lu et al., 2026a), SEED maps this shift to a confidence gate:

$$g_{q,n,t,\ell} = \sigma\left(\beta_{\text{opd}} \Delta_{q,n,t,\ell}\right),$$

where $\sigma(\cdot)$ is the logistic sigmoid function and β_{opd} controls the sharpness of the gate. A positive shift indicates that the hindsight skill supports the sampled token and yields a larger gate; a negative shift attenuates that token's auxiliary supervision.

The OPD loss is defined as a confidence-gated sampled-token distillation objective:

$$\mathcal{L}_{\text{opd}}(\theta) = \mathbb{E}_{q,n,t,\ell}\left[m_{q,n,t,\ell} \cdot g_{q,n,t,\ell} \cdot \left(\text{sg}\left[\ell^{\text{skill}}_{q,n,t,\ell}\right] - \ell^{\theta}_{q,n,t,\ell}\right)\right]. \tag{1}$$

Here $m_{q,n,t,\ell} \in \{0,1\}$ is the valid-token mask. Throughout, expectations over (q,n,t,ℓ) are computed as masked means over valid action tokens, normalized by $\sum_{q,n,t,\ell} m_{q,n,t,\ell}$. Because both $g_{q,n,t,\ell}$ and the teacher log-probability are detached, the teacher term is constant with respect to θ, and

$$\nabla_\theta \mathcal{L}_{\text{opd}} = -\mathbb{E}_{q,n,t,\ell}\left[m_{q,n,t,\ell} \cdot g_{q,n,t,\ell} \cdot \nabla_\theta \ell^{\theta}_{q,n,t,\ell}\right].$$

Thus, the objective is gradient-equivalent to a gate-weighted negative log-likelihood. Minimizing it increases the ordinary policy's likelihood of teacher-endorsed on-policy tokens, thereby internalizing the skill's behavioral effect without exposing the skill at inference time.

Joint training objective. In addition to OPD, SEED optimizes the policy with a group-relative RL objective. For each group $\mathcal{G}_q$, we compute the mean and standard deviation of trajectory outcomes:

$$\mu_q = \frac{1}{N}\sum_{n=1}^{N} R\left(\tau_q^{(n)}\right), \quad \sigma_q = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(R\left(\tau_q^{(n)}\right) - \mu_q\right)^2}.$$

The trajectory-level group-relative advantage is

$$A^{\text{rl}}_{q,n} = \frac{R\left(\tau_q^{(n)}\right) - \mu_q}{\sigma_q + \epsilon}.$$

This advantage is broadcast to valid action tokens:

$$A^{\text{rl}}_{q,n,t,\ell} = A^{\text{rl}}_{q,n} m_{q,n,t,\ell}.$$

The token-level probability ratio is

$$\rho_{q,n,t,\ell}(\theta) = \exp\left(\ell^{\theta}_{q,n,t,\ell} - \ell^{\text{old}}_{q,n,t,\ell}\right),$$

where

$$\ell^{\text{old}}_{q,n,t,\ell} = \log \pi_{\theta_{\text{old}}}\left(a_{q,n,t,\ell} \mid h_{q,n,t}, a_{q,n,t,<\ell}\right).$$

The RL loss is

$$\mathcal{L}_{\text{rl}}(\theta) = -\mathbb{E}_{q,n,t,\ell}\left[\min\left(\rho_{q,n,t,\ell}(\theta) A^{\text{rl}}_{q,n,t,\ell}, \text{clip}\left(\rho_{q,n,t,\ell}(\theta), 1-\epsilon_{\text{clip}}, 1+\epsilon_{\text{clip}}\right) A^{\text{rl}}_{q,n,t,\ell}\right)\right] + \beta_{\text{KL}} D_{\text{KL}}.$$

where β_{KL} is the KL regularization coefficient.

The final training objective combines environment-driven agentic RL with hindsight-skill OPD:

$$\mathcal{L}_{\text{SEED}}(\theta) = \mathcal{L}_{\text{rl}}(\theta) + \lambda_{\text{opd}}\mathcal{L}_{\text{opd}}(\theta),$$

where λ_{opd} controls the strength of the auxiliary OPD signal. The RL term optimizes environment outcomes, whereas the OPD loss internalizes the effect of self-generated hindsight skills. The updated policy then becomes $\pi_{\theta_{\text{old}}}$ for the next iteration, closing the self-evolving loop.

At inference time, the deployed agent acts only from the ordinary interaction history:

$$a_t \sim \pi_\theta\left(\cdot \mid h_t\right).$$

Thus, all hindsight skills are used only as training-time guidance. Deployment requires no analyzer, no skill bank, no retrieval module, and no augmented decision prompt.

Algorithm 1 in Appendix B.3 summarizes the complete training procedure.

4 Experiment

4.1 Experimental Setting

Benchmarks. We evaluate SEED across three complementary forms of long-horizon agency. *ALFWorld* (Shridhar et al., 2021) casts household tasks as text-based embodied interaction, requiring an agent to interpret observations and execute extended action sequences. We consider six task families: *Pick*, *Look*, *Clean*, *Heat*, *Cool*, and *Pick2*. *WebShop* (Yao et al., 2022) evaluates interactive web navigation, where an agent searches for products, inspects their attributes, and completes a purchase according to a natural-language request. We use the standard set of 128 test tasks. Finally, following the Search-R1 protocol (Jin et al., 2025), *Search-based QA* requires an agent to gather evidence through search before answering questions from Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), PopQA (Mallen et al., 2023), HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023). Together, these benchmarks cover embodied control, web-based decision making, and tool-augmented information seeking.

Baselines. We compare SEED with prompting, outcome-based RL, and distillation baselines. *Vanilla* evaluates the instruction-tuned backbone without post-training, whereas *Skill-Prompt* provides natural-language skills only in the evaluation context. *GRPO* (Shao et al., 2024) optimizes group-normalized trajectory rewards without auxiliary supervision. *Skill-GRPO* additionally conditions the policy on skills during RL. We also include representative self-distillation and skill-distillation methods: *OPSD* (Zhao et al., 2026), *GRPO+OPSD*, *Skill-SD* (Wang et al., 2026), *RLSD* (Yang et al., 2026a), and *SDAR* (Lu et al., 2026a). These comparisons distinguish the effects of outcome optimization, access to skill context, and dense teacher-derived supervision. An asterisk denotes evaluation with skills; all other methods operate from the ordinary interaction history without privileged skill inputs at test time. We use matched backbones, rollout budgets, and training schedules for all reproduced post-training baselines.

Evaluation Metrics. For ALFWorld, we report the success rate for each task family and their unweighted macro-average. For WebShop, we follow the environment protocol and report both the mean normalized task-completion score and the exact success rate. For Search-based QA, we compute answer accuracy separately on all seven subsets and report their unweighted macro-average. All metrics are expressed as percentages, and higher values indicate better performance.

Implementation Details. We use Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct (Yang et al., 2024), as well as Qwen3-1.7B-Instruct (Yang et al., 2025), as our backbone models. *SFT stage:* For each backbone, we sample $M = 180$ training tasks and collect $K_0 = 8$ rollout trajectories per task, resulting in 1,440 completed trajectories. We then query GLM-5.2 (Z.ai, 2026) as an external trajectory analyzer to extract hindsight skills from these trajectories. After lightweight format validation, the retained trajectory–skill pairs are used to fine-tune the corresponding backbone for three epochs. *RL stage:* The resulting SFT checkpoint initializes both the policy and the synchronized trajectory analyzer. We train each model for 150 policy updates, using a batch size of 16 on ALFWorld and WebShop and 128 on Search-based QA. The rollout group size is set to $N = 8$ for all benchmarks. Additional details on SFT data construction, skill annotation, optimization, and hyperparameter settings are provided in Appendix B.4.

Table 1: **Performance Comparison on the representative long-horizon benchmarks (ALFWorld, Search-based QA, and WebShop).** We report the success rate (%) on ALFWorld, accuracy on Search-based QA, and task-completion score/success rate on WebShop. An asterisk (*) denotes validation with skills. The **best** and second-best results are highlighted.

	ALFWorld							Search-based QA								WebShop	
Method	**Pick**	**Look**	**Clean**	**Heat**	**Cool**	**Pick2**	**Avg**	**NQ**	**Triv**	**Pop**	**Hotp**	**2Wk**	**MuS**	**Bam**	**Avg**	**Score**	**Succ.**
Qwen2.5-3B-Instruct																	
Vanilla	44.4	11.1	6.2	15.4	28.6	12.5	21.9	24.6	48.1	31.0	26.3	25.3	7.2	59.7	31.7	6.7	0.8
Skill-Prompt*	51.7	66.7	48.4	0.0	4.3	10.0	28.9	23.7	46.2	30.6	24.4	22.1	7.5	12.5	23.9	0.2	0.8
OPSD	48.8	41.7	16.7	0.0	15.8	16.7	28.1	0.1	0.1	0.1	0.0	0.0	0.0	0.0	0.0	11.3	3.1
GRPO	91.2	62.5	96.2	61.9	65.0	47.4	75.0	39.3	60.6	41.1	37.4	34.6	15.4	26.4	36.4	79.8	63.3
Skill-GRPO	88.9	71.4	58.8	70.6	40.7	29.2	60.2	43.5	58.8	43.0	36.8	32.2	11.7	12.5	34.1	77.3	60.9
Skill-GRPO*	94.3	57.1	**100.0**	66.7	73.1	57.1	80.5	44.3	59.6	44.3	39.0	36.1	14.5	14.9	36.1	76.3	66.4
GRPO+OPSD	**100.0**	82.4	85.7	75.0	70.0	60.0	81.2	**44.9**	61.2	45.2	40.4	38.5	16.0	66.1	44.6	77.8	66.4
Skill-SD	88.2	50.0	96.2	52.4	65.0	57.9	73.4	44.4	60.4	44.0	39.5	40.4	15.4	64.9	44.1	75.9	64.0
RLSD	87.9	75.0	90.9	75.0	73.1	68.4	79.7	41.5	58.6	42.3	40.4	40.2	16.8	66.9	43.8	84.4	66.4
SDAR	97.1	62.5	**100.0**	61.9	**75.0**	**84.2**	84.4	44.8	58.1	44.3	38.6	36.2	15.7	66.1	43.4	85.0	68.0
SEED (Ours)	**100.0**	**100.0**	**100.0**	**100.0**	70.6	80.0	**91.8**	44.3	**61.9**	**47.2**	**41.0**	**42.0**	**16.9**	**67.7**	**45.7**	**88.5**	**78.9**
Qwen2.5-7B-Instruct																	
Vanilla	36.1	22.2	3.1	0.0	0.0	0.0	12.5	25.2	50.8	29.5	29.0	29.0	10.4	63.7	33.9	5.9	1.6
Skill-Prompt*	51.7	50.0	32.3	5.3	4.3	0.0	23.4	30.9	52.1	32.7	32.7	27.9	12.7	66.1	36.4	1.7	0.8
OPSD	50.0	60.0	22.7	21.4	17.6	9.5	32.8	8.8	8.6	17.5	2.5	4.2	0.5	1.2	6.2	4.5	2.3
GRPO	91.2	87.5	96.2	81.0	65.0	57.9	81.2	45.1	63.7	44.0	43.6	43.2	16.8	37.6	42.0	80.9	72.6
Skill-GRPO	88.5	66.7	65.2	61.1	57.7	73.1	69.5	45.2	63.7	45.7	43.1	43.3	19.6	21.4	40.3	80.4	71.9
Skill-GRPO*	**100.0**	83.3	96.4	83.3	75.0	78.9	88.3	44.8	63.0	45.1	43.7	43.7	20.5	71.4	47.5	87.0	81.2
GRPO+OPSD	91.4	61.5	**100.0**	87.5	76.5	52.2	80.4	**47.3**	64.5	46.9	43.8	39.3	18.0	69.4	47.0	86.8	76.5
Skill-SD	93.9	93.8	90.9	**100.0**	69.2	68.4	85.1	47.1	64.5	47.8	44.2	42.1	20.2	69.0	47.8	86.1	76.5
RLSD	**100.0**	87.5	92.3	58.8	80.0	65.2	82.0	46.8	63.0	44.4	**45.5**	**48.9**	**21.5**	**73.0**	**49.0**	87.4	77.3
SDAR	94.7	75.0	**100.0**	86.7	68.2	78.9	85.9	46.3	63.5	**48.2**	43.8	48.4	19.6	**73.0**	**49.0**	89.4	**82.8**
SEED (Ours)	**100.0**	**100.0**	96.3	80.0	**100.0**	**100.0**	**96.1**	47.0	**64.9**	47.8	45.2	45.3	20.1	70.2	48.6	**89.7**	78.1
Qwen3-1.7B-Instruct																	
Vanilla	25.0	22.2	3.1	0.0	21.4	4.2	12.5	29.4	46.9	37.0	23.5	19.6	6.4	10.5	24.8	46.5	4.7
Skill-Prompt*	10.3	50.0	16.1	0.0	0.0	5.0	9.4	29.4	46.5	36.2	22.9	20.8	4.3	10.1	24.3	23.0	2.3
OPSD	26.3	33.3	9.1	0.0	4.5	5.3	14.1	4.2	8.3	4.6	6.6	15.3	0.7	1.2	5.8	47.4	9.3
GRPO	71.1	41.7	36.4	40.0	31.8	31.6	46.1	40.0	**58.9**	43.5	35.4	30.3	12.0	65.7	40.8	67.3	38.3
Skill-GRPO	27.6	54.5	22.7	27.3	0.0	19.2	21.1	39.2	58.6	43.9	35.2	28.2	11.5	66.1	40.4	73.4	46.1
Skill-GRPO*	31.4	42.9	51.9	8.3	11.5	7.1	28.1	38.0	58.4	43.9	36.3	29.0	12.5	**66.9**	40.7	80.4	50.0
GRPO+OPSD	38.2	50.0	30.8	28.6	30.0	21.1	32.0	40.7	**58.9**	45.0	**37.0**	34.6	**13.3**	65.7	**42.2**	70.7	38.3
Skill-SD	52.9	37.5	69.2	42.9	60.0	36.8	52.3	39.1	57.5	45.4	34.8	34.1	10.7	64.1	40.8	81.8	53.9
RLSD	50.0	37.5	61.5	19.0	50.0	21.1	42.2	38.6	57.3	43.0	34.5	34.1	11.5	65.3	40.6	74.0	50.8
SDAR	73.5	25.0	76.9	33.3	40.0	36.8	53.9	39.7	**58.9**	45.3	35.9	**35.5**	12.6	65.3	41.9	76.8	58.6
SEED (Ours)	**97.6**	**100.0**	**100.0**	**80.0**	**84.2**	**90.0**	**92.0**	**42.0**	**58.9**	**47.1**	36.9	35.2	10.6	64.9	**42.2**	**87.1**	**77.3**

4.2 MAIN RESULTS

Table 1 reports the results across different model scales and agentic domains. Three findings emerge:

SEED consistently outperforms outcome-only RL through dense supervision. Relative to GRPO, SEED improves the ALFWorld macro-average by 14.9-45.9 points, Search-based QA by 1.4-9.3 points, the WebShop task-completion score by 8.7-19.8 points, and the success rate by 5.5-39.0 points across the three backbones. Compared with Skill-GRPO, which conditions exploration on natural-language skills but still broadcasts a single terminal-reward-derived advantage to all valid tokens, SEED further improves the ALFWorld average by 26.6-70.9 points, Search-based QA by 1.8-11.6 points, the WebShop score by 9.3-13.7 points, and the success rate by 6.2-31.2 points. These consistent improvements over both GRPO and Skill-GRPO demonstrate that dense token-level hindsight supervision provides more effective credit assignment than outcome-only optimization, leading to substantially stronger performance across long-horizon agentic tasks.

Internalizing skills is substantially more effective than inserted prompts. Skill-Prompt, which provides skills only during evaluation, underperforms SEED on every aggregate metric across all three backbones. SEED also exceeds Skill-GRPO* in 11 of the 12 aggregate comparisons, despite using no skill context during evaluation. These results indicate that hindsight skills are more effective when distilled into the policy than when supplied as additional context at inference time.

Self-evolving hindsight distillation is stronger than static self-distillation. Across the static distillation baselines, SEED achieves the best or tied best result in 10 of the 12 aggregate comparisons. The advantage is clearest on ALFWorld, where SEED outperforms the strongest static baseline by 7.4 points with Qwen2.5-3B, 10.2 points with Qwen2.5-7B, and 38.1 points with Qwen3-1.7B. This pattern supports the benefit of synchronizing the analyzer with the latest policy, so hindsight supervision adapts to the trajectories and failure modes encountered during training.

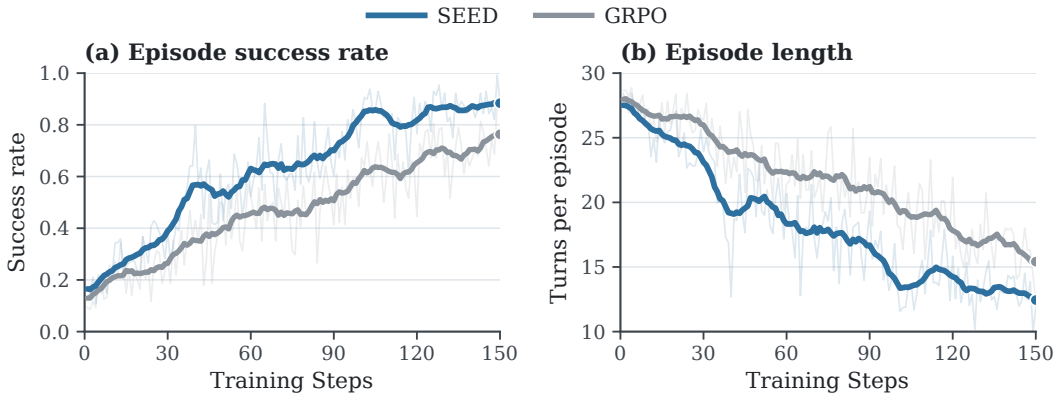

Figure 3: **Training dynamics on ALFWorld.** We compare SEED and GRPO using Qwen2.5-3B-Instruct as the backbone. Translucent curves show raw measurements, while solid curves show 13-point centered moving averages.

4.3 TRAINING DYNAMICS

Figure 3 compares the optimization dynamics of SEED and GRPO on ALFWorld. The success-rate curves diverge early: by training step 40, SEED reaches roughly 57% while GRPO remains near 35%, and the advantage persists thereafter. SEED also reduces the mean episode length more quickly, from approximately 28 turns to 13, compared with about 16 turns for GRPO at the end of training. Because shorter trajectories coincide with higher success, this reduction reflects more efficient task execution rather than premature termination. Together, these trends indicate that SEED improves task completion and interaction efficiency by reducing unproductive exploration and learning more direct solution strategies.

4.4 SAMPLE EFFICIENCY

Figure 4 shows that SEED consistently outperforms GRPO across all data fractions and can match or surpass the performance of GRPO trained with substantially more data. Using only 60% of the training instances, SEED achieves a score of 80.7, exceeding the 75.0 obtained by GRPO with the full training set. Similarly, with 40% of the data, SEED reaches 58.9, closely matching GRPO trained with twice as much data, which achieves 58.6 at the 80% setting. These results demonstrate that SEED extracts more informative and effective supervision from each completed trajectory than outcome-only RL based solely on terminal rewards. Detailed results are provided in Appendix C.1.

4.5 CROSS-DOMAIN GENERALIZATION

We further evaluate the 3B checkpoint trained with SEED and GRPO on the ALFWorld unseen split. As shown in Figure 5, SEED increases the macro-average success rate from 70.9 to 86.2, outperforming GRPO by 15.3 points and achieving better performance in five of the six task families. The largest improvements are observed on *Heat* (+35.0), *Look* (+18.3), and *Pick* (+16.5). These results indicate that the policy trained with SEED acquires reusable behavioral strategies that transfer effectively to unseen environments, rather than merely memorizing the training trajectories. Detailed results are provided in Appendix C.2.

4.6 ABLATION STUDIES AND ANALYSIS

Table 2 evaluates the contribution of the three core components of SEED.

The impact of hindsight-skill SFT. Removing hindsight-skill SFT decreases the ALFWorld average from 91.8 to 86.0, corresponding to a 5.8-point drop. This result shows that equipping the actor

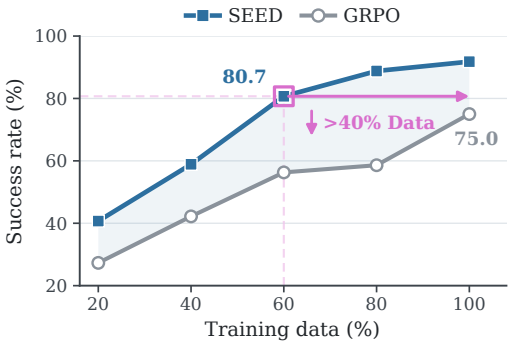

Figure 4: **Sample efficiency analysis.** SEED consistently outperforms GRPO across different data fraction settings and surpasses full-data GRPO using only 60% of the training data.

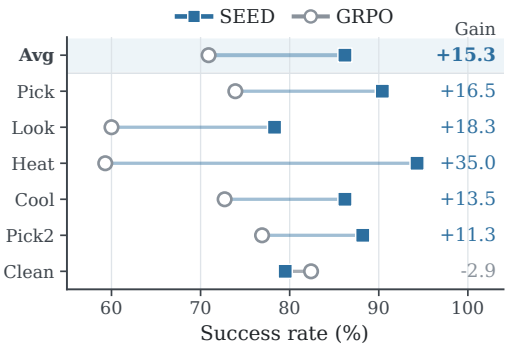

Figure 5: **Cross-domain generalizability on ALFWorld Unseen.** SEED generally outperforms GRPO across unseen task types, demonstrating stronger cross-domain generalizability.

Table 2: **Ablation Results.** We report SEED performance and its ablated variants on ALFWorld.

	ALFWorld						
Method	**Pick**	**Look**	**Clean**	**Heat**	**Cool**	**Pick2**	**Avg.**
SEED	**100.0**	**100.0**	**100.0**	**100.0**	70.6	80.0	**91.8**
w/o Hindsight Skill SFT	97.6	81.8	92.6	90.0	**84.2**	70.0	86.0
w/o Self-Evolving OPD	93.3	70.0	96.9	84.6	76.9	**100.0**	87.0
w/o On-Policy Skill	97.1	62.5	**100.0**	61.9	75.0	84.2	84.4

and analyzer with an initial trajectory-analysis capability provides an important foundation for the subsequent self-evolving training process.

The impact of self-evolving OPD. Removing self-evolving OPD reduces the average performance to 87.0, a decrease of 4.8 points. This indicates that the one-time skill supervision introduced during SFT is insufficient on its own, and that continuously distilling supervision from newly collected trajectories is necessary to keep pace with the evolving policy.

The impact of on-policy skills. Replacing on-policy skills with skills from a static offline library leads to the largest performance degradation, lowering the average to 84.4 by 7.4 points. This finding highlights the importance of deriving guidance from the current policy's own trajectories, as such skills better reflect its evolving behaviors and failure modes.

Qualitative Analysis. Figure 6 presents a representative example of the behavioral differences between GRPO and SEED. GRPO visits the target receptacle before locating the required object, interacts with irrelevant objects, and repeatedly explores unproductive locations until reaching the step limit. In contrast, SEED follows a more coherent and goal-directed trajectory: it searches plausible storage locations, identifies the candle on the second shelf, and completes the task with the correct placement action. The more direct trajectory exhibited by SEED is consistent with its stronger performance and higher interaction efficiency reported in the preceding experiments.

5 CONCLUSION

We introduced SEED, a self-evolving on-policy distillation framework for long-horizon agentic reinforcement learning. SEED bridges sparse trajectory-level outcomes and token-level policy learning by extracting hindsight skills from completed on-policy trajectories and distilling their behavioral effects into dense supervision. The latest policy serves as both the actor and the analyzer, allowing its behavior and the supervision derived from its experience to evolve together while remaining aligned with the current trajectory distribution. Joint optimization with outcome-based RL enables the pol-

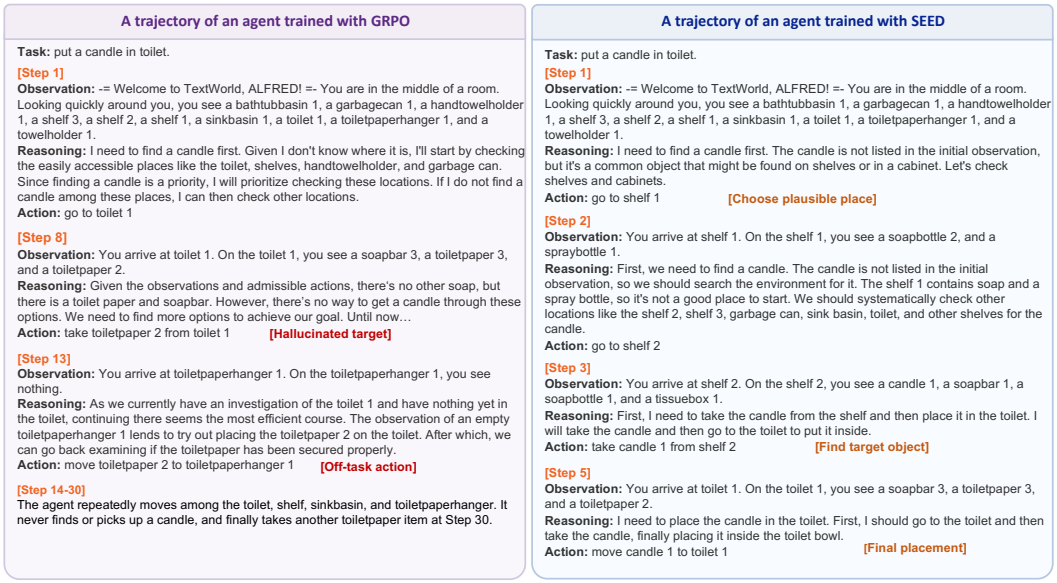

Figure 6: **Qualitative comparison on ALFWorld.** For the task "put a candle in toilet," the GRPO-trained agent searches the target receptacle before locating the candle, takes an irrelevant toilet-paper item, and then enters an off-task loop. In contrast, SEED systematically checks plausible shelves, finds the candle, and completes the required placement in five steps.

icy to internalize reusable guidance without relying on skills at inference time. Experiments across embodied interaction, web navigation, search-based QA, visual reasoning and planning demonstrate consistent performance improvements, sample efficiency, and robustness over powerful baselines.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.13649.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. RUDDER: Return decomposition for delayed rewards. In *Advances in Neural Information Processing Systems*, 2019. URL https://arxiv.org/abs/1806.07857.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Xin Cheng, Xingkai Yu, Chenze Shao, Jiashi Li, Yunfan Xiong, Yi Qian, Jiaqi Zhu, Shirong Ma, Xiaokang Zhang, Jiasheng Ye, et al. Dspark: Confidence-scheduled speculative decoding with semi-autoregressive generation. *arXiv preprint arXiv:2607.05147*, 2026.

Kuofei Fang, Xinyi Che, Haomin Ouyang, Shufan Zhang, Xuehao Wang, Qi Liu, Liyi Liu, Chenqi Zhang, Wenxi Cai, Wenyu Dai, et al. Roboteq: Transitioning from passive intelligence to active intelligence in embodied ai. *arXiv preprint arXiv:2605.06234*, 2026.

Yubin Ge, Salvatore Romeo, Jason Cai, Monica Sunkara, and Yi Zhang. SAMULE: Self-learning agents enhanced by multi-level reflection. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 16591–16610, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.839. URL https://aclanthology.org/2025.emnlp-main.839/.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580/.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation, 2026. URL https://arxiv.org/abs/2601.20802.

Chunyang Jiang, Chi-Min Chan, Wei Xue, Qifeng Liu, and Yike Guo. Importance weighting can help large language models self-improve. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 24257–24265, 2025a. doi: 10.1609/aaai.v39i23.34602.

Dongwei Jiang, Jingyu Zhang, Orion Weller, Nathaniel Weir, Benjamin Van Durme, and Daniel Khashabi. Self-[in]correct: Llms struggle with discriminating self-generated responses. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 24266–24275, 2025b. doi: 10.1609/aaai.v39i23.34603.

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, 2025. URL https://arxiv.org/abs/2503.09516.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 1601–1611, 2017. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, 2016. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139.

Jongwoo Ko, Sara Abdali, Young Jin Kim, Tianyi Chen, and Pashmina Cameron. Scaling reasoning efficiently via relaxed on-policy distillation, 2026. URL https://arxiv.org/abs/2603.11137.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, JD Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. Training language models to self-correct via reinforcement learning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/871ac99fdc5282d0301934d23945ebaa-Abstract-Conference.html.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Zongxia Li, Zhongzhi Li, Yucheng Shi, Ruhan Wang, Junyao Yang, Zhichao Liu, Xiyang Wu, Anhao Li, Yue Yu, Ninghao Liu, Lichao Sun, Haotao Mi, and LeoweiLiang. Long-horizon-terminal-bench: Testing the limits of agents on long-horizon terminal tasks with dense reward-based grading, 2026. URL https://arxiv.org/abs/2607.08964.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2305.20050.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. Agentbench: Evaluating LLMs as agents, 2023. URL https://arxiv.org/abs/2308.03688.

Zhengxi Lu, Zhiyuan Yao, Zhuowen Han, Zi-Han Wang, Jinyang Wu, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Self-distilled agentic reinforcement learning, 2026a. URL https://arxiv.org/abs/2605.15155.

Zhengxi Lu, Zhiyuan Yao, Jinyang Wu, Chengcheng Han, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Skill0: In-context agentic reinforcement learning for skill internalization. *arXiv preprint arXiv:2604.02268*, 2026b.

Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi Chen, Ziyue Qiao, Qingqing Long, et al. Large language model agent: A survey on methodology, applications and challenges. *arXiv preprint arXiv:2503.21460*, 2025a.

Xufang Luo, Yuge Zhang, Zhiyuan He, Zilong Wang, Siyun Zhao, Dongsheng Li, Luna K. Qiu, and Yuqing Yang. Agent lightning: Train ANY AI agents with reinforcement learning, 2025b. URL https://arxiv.org/abs/2508.03680.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2303.17651.

Taslim Mahbub and Shi Feng. Mitigating self-preference by authorship obfuscation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 37701–37708, 2026. doi: 10.1609/aaai.v40i44.41105.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 9802–9822, 2023. doi: 10.18653/v1/2023.acl-long.546. URL https://aclanthology.org/2023.acl-long.546/.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive APIs. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2305.15334.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, 2023. doi: 10.18653/v1/2023.findings-emnlp.378. URL https://aclanthology.org/2023.findings-emnlp.378/.

Zhenting Qi, Susanna Maria Baby, Stefanie Anna Baby, Kan Yuan, Andrew Tomkins, Tu Vu, Da-Cheng Juan, and Cyrus Rashtchian. On the generalization gap in self-evolving language model reasoning. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=mnUidYi5qO.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 627–635, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2302.04761.

Max-Philipp B. Schrader. gym-sokoban. https://github.com/mpSchrader/gym-sokoban, 2018.

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

Yuhao Shen, Tianyu Liu, Junyi Shen, Jinyang Wu, Quan Kong, Huan Li, and Cong Wang. Double: Breaking the acceleration limit via double retrieval speculative parallelism. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19242–19263, San Diego, California, United States, July 2026. Association for Computational Linguistics.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2303.11366.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.03768.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL https://aclanthology.org/2022.tacl-1.31/.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback, 2022. URL https://arxiv.org/abs/2211.14275.

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, 2023. URL https://arxiv.org/abs/2305.16291.

Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, Xiaoxin Chen, and Honggang Qi. Skill-SD: Skill-conditioned self-distillation for multi-turn LLM agents, 2026. URL https://arxiv.org/abs/2604.10674.

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, 2025. URL https://arxiv.org/abs/2504.20073.

Jinyang Wu, Mingkuan Feng, Shuai Zhang, Feihu Che, Zengqi Wen, Chonghua Liao, and Jianhua Tao. Beyond examples: High-level automated reasoning paradigm in in-context learning via mcts. *arXiv preprint arXiv:2411.18478*, 2024.

Jinyang Wu, Shuo Yang, Yuhao Shen, Shuai Zhang, Zhengqi Wen, and Jianhua Tao. SPARK: Strategic policy-aware exploration via dynamic branching for long-horizon agentic learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume*

1: Long Papers), pp. 23981–24004, San Diego, California, United States, July 2026a. Association for Computational Linguistics.

Jinyang Wu, Guocheng Zhai, Ruihan Jin, Jiahao Yuan, Yuhao Shen, Shuai Zhang, Zhengqi Wen, and Jianhua Tao. ATLAS: Orchestrating heterogeneous models and tools for multi-domain complex reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 17503–17535, San Diego, California, United States, July 2026b. Association for Computational Linguistics.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, Zhangyue Yin, Shihan Dou, Rongxiang Weng, Wensen Cheng, Qi Zhang, Wenjuan Qin, Yongyan Zheng, Xipeng Qiu, Xuanjing Huang, and Tao Gui. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 2025. doi: 10.1007/s11432-024-4222-0.

Fangzhi Xu, Hang Yan, Qiushi Sun, Jinyang Wu, Zixian Huang, Muye Huang, Jingyang Gong, Zichen Ding, Kanzhi Cheng, Yian Wang, et al. Odysseyarena: Benchmarking large language models for long-horizon, active and inductive interactions. *arXiv preprint arXiv:2602.05843*, 2026.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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, Jianxin Yang, Jiaxi Yang, Jing Zhou, 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. *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, 2026a. URL https://arxiv.org/abs/2604.03128.

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*, 2026b.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2207.01206.

Z.ai. GLM-5.2: Built for Long-Horizon Tasks. https://z.ai/blog/glm-5.2, June 2026. Accessed: 2026-06-22.

Yuexiang Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Shengbang Tong, Yifei Zhou, Alane Suhr, Saining Xie, Yann LeCun, Yi Ma, and Sergey Levine. Fine-tuning large vision-language models as

decision-making agents via reinforcement learning. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 110935–110971. Curran Associates, Inc., 2024. doi: 10.52202/079017-3522. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/c848b7d3adc08fcd0bf1df3101ba6728-Paper-Conference.pdf.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhong-Zhi Li, Xiangyuan Xue, Yijiang Li, Yifan Zhou, Yang Chen, Chen Zhang, Yutao Fan, Zihu Wang, Songtao Huang, Francisco Piedrahita Velez, Yue Liao, Hongru WANG, Mengyue Yang, Heng Ji, Jun Wang, Shuicheng YAN, Philip Torr, and LEI BAI. The landscape of agentic reinforcement learning for LLMs: A survey. *Transactions on Machine Learning Research*, 2026a. ISSN 2835-8856. Survey Certification.

Yinger Zhang, Shutong Jiang, Renhao Li, Jianhong Tu, Yang Su, Lianghao Deng, Xudong Guo, Chenxu Lv, and Junyang Lin. Deepplanning: Benchmarking long-horizon agentic planning with verifiable constraints. *arXiv preprint arXiv:2601.18137*, 2026b.

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*, 2024. URL https://arxiv.org/abs/2308.10144.

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, 2026. URL https://arxiv.org/abs/2601.18734.

Qiyong Zhong, Mao Zheng, Mingyang Song, Xin Lin, Jie Sun, Houcheng Jiang, Xiang Wang, and Junfeng Fang. SOD: Step-wise on-policy distillation for small language model agents, 2026. URL https://arxiv.org/abs/2605.07725.

Tianyi Zhou, Johanne Medina, and Sanjay Chawla. Can llms detect their confabulations? estimating reliability in uncertainty-aware language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 38164–38172, 2026. doi: 10.1609/aaai.v40i44.41155.

A Theoretical Analysis

This section formalizes three properties of SEED that mirror the requirements identified in Section 1: the hindsight supervision is *on-policy*, *dense*, and *self-evolving*. Specifically, we show that (i) synchronized on-policy hindsight induces an occupancy-matched adaptive target, (ii) the OPD term provides decision-specific credit even when outcome-based group advantages are uninformative, and (iii) refreshing the shared analyzer controls the staleness of the auxiliary gradient as the policy changes. These are local statements about the structure and freshness of the SEED update; they do not by themselves imply monotonic return improvement, which additionally requires the generated hindsight skills to be behaviorally informative.

Notation. Consider outer iteration k. At the beginning of the iteration, the current checkpoint is frozen as the behavior policy $\pi_k = \pi_{\theta_k}$ and also instantiates the trajectory analyzer $\mathcal{A}_k$. Let C denote an ordinary autoregressive context at a valid action-token position, and let $d_k(c)$ be the normalized token-context occupancy induced by π_k and the environment. Conditional on $C = c$, the realized rollout token Y is sampled as

$$Y \sim \pi_k(\cdot \mid c).$$

After the complete trajectory τ is observed, the synchronized analyzer produces the hindsight skill $S = \mathcal{A}_k(x_\tau)$. We overload $H(c, S)$ to denote the skill-augmented version of c, with the same action-token prefix as in the ordinary context. At update initialization, define the detached skill gate

$$g_k(c, v, S) = \sigma(\beta_{\text{opd}} \left[\log \pi_k(v \mid H(c, S)) - \log \pi_k(v \mid c)\right]). \tag{2}$$

Importantly, S is generated from the *complete* trajectory and can therefore depend on the realized token Y, the subsequent rollout, environment feedback, and analyzer decoding. To preserve this dependence, we define the conditional expected gate

$$w_k(c, v) = \mathbb{E}[g_k(C, Y, S) \mid C = c, Y = v], \tag{3}$$

where the expectation is over the future trajectory, environment randomness, and any analyzer randomness. Standard softmax policies have full support, and the sigmoid lies strictly between zero and one, so $0 < w_k(c, v) < 1$. We state the results at $\theta = \theta_k$, where the behavior, analyzer, teacher branch, and student branch are synchronized. The adaptive-target identity below extends to an inner optimization step by replacing Eq. 2 with its current detached numerical value while retaining the same behavior occupancy d_k.

A.1 On-Policy Hindsight Produces an Occupancy-Matched Adaptive Target

The first result characterizes the expected OPD direction and makes explicit why the source trajectories should be generated by the current policy. Rather than imitating every sampled token equally, SEED reweights the current policy according to how strongly its own hindsight skill supports each action.

Proposition 1 (Occupancy-matched hindsight target)**.** *For every context c, define*

$$Z_k(c) = \sum_{v \in \mathcal{V}} \pi_k(v \mid c) w_k(c, v), \qquad r_k(v \mid c) = \frac{\pi_k(v \mid c) w_k(c, v)}{Z_k(c)}. \tag{4}$$

Then $0 < Z_k(c) < 1$, $r_k(\cdot \mid c)$ is a probability distribution, and the expected OPD gradient at update initialization satisfies

$$\nabla_\theta \mathcal{L}_{\mathrm{opd},k}(\theta)|_{\theta=\theta_k} = \mathbb{E}_{c \sim d_k}\left[Z_k(c)\, \nabla_\theta D_{\mathrm{KL}}\left(r_k(\cdot \mid c) \,\|\, \pi_\theta(\cdot \mid c) \right)|_{\theta=\theta_k} \right]. \tag{5}$$

Thus, the OPD update is a Monte Carlo estimate of distillation toward a skill-reweighted target on the current policy's own token-context occupancy.

Moreover, let μ_k be the joint distribution of a completed trajectory and one of its valid token samples, and let $\phi_{k,\theta}(x)$ be the detached per-token OPD gradient obtained by analyzing sample x with the current analyzer $\mathcal{A}_k$. If $\|\phi_{k,\theta}(x)\|_2 \le G$ for all relevant x, then replacing current on-policy samples by samples from an earlier distribution μ_j incurs

$$\left\| \mathbb{E}_{x \sim \mu_k}[\phi_{k,\theta}(x)] - \mathbb{E}_{x \sim \mu_j}[\phi_{k,\theta}(x)] \right\|_2 \le 2G\ \mathrm{TV}(\mu_k, \mu_j). \tag{6}$$

If μ_k is absolutely continuous with respect to μ_j, Pinsker's inequality further gives

$$\left\| \mathbb{E}_{\mu_k}[\phi_{k,\theta}] - \mathbb{E}_{\mu_j}[\phi_{k,\theta}] \right\|_2 \le G\sqrt{2 D_{\mathrm{KL}}(\mu_k \| \mu_j)}. \tag{7}$$

Proof. Because $0 < w_k(c, v) < 1$ and $\pi_k(\cdot \mid c)$ is normalized, Eq. 4 gives $0 < Z_k(c) < 1$ and $\sum_v r_k(v \mid c) = 1$. In Eq. 1, both the teacher log-probability and the gate are detached, so only the ordinary student log-probability contributes to the gradient. Conditioning first on $C = c$ and $Y = v$, and then using Eq. 3, yields

$$\begin{aligned}
&\nabla_\theta \mathcal{L}_{\mathrm{opd},k}(\theta)|_{\theta=\theta_k} \\
&\quad = -\mathbb{E}_{c \sim d_k}\left[\sum_{v \in \mathcal{V}} \pi_k(v \mid c) w_k(c, v)\ \nabla_\theta \log \pi_\theta(v \mid c)|_{\theta=\theta_k} \right] \\
&\quad = -\mathbb{E}_{c \sim d_k}\left[Z_k(c) \sum_{v \in \mathcal{V}} r_k(v \mid c)\ \nabla_\theta \log \pi_\theta(v \mid c)|_{\theta=\theta_k} \right],
\end{aligned}$$

which is Eq. 5, since the entropy of the detached target r_k has zero gradient.

For Eq. 6, use the dual representation of the Euclidean norm and the variational characterization of total variation. For any unit vector u,

$$\left| \mathbb{E}_{\mu_k}[u^\top \phi_{k,\theta}] - \mathbb{E}_{\mu_j}[u^\top \phi_{k,\theta}] \right| \le 2G\ \mathrm{TV}(\mu_k, \mu_j),$$

because $|u^\top \phi_{k,\theta}(x)| \le G$. Taking the supremum over u proves Eq. 6. Equation 7 follows from $\mathrm{TV}(\mu_k, \mu_j) \le \sqrt{D_{\mathrm{KL}}(\mu_k \| \mu_j)/2}$. □

Equation 4 separates the two roles of the first SEED requirement. The factor $d_k(c)$ makes supervision *occupancy matched*: it focuses learning on contexts, actions, and failure modes actually visited by the current policy. Within each such context, $w_k(c, v)$ makes supervision *skill selective*: actions that are more strongly supported by the trajectory-specific hindsight skill receive greater relative mass. In particular,

$$\frac{r_k(u \mid c)}{\pi_k(u \mid c)} > \frac{r_k(v \mid c)}{\pi_k(v \mid c)} \quad \Longleftrightarrow \quad w_k(c, u) > w_k(c, v).$$

The mismatch bounds show the complementary point: even if the same current analyzer is applied to old trajectories, a stale behavior distribution can bias the auxiliary direction in proportion to its divergence from the current trajectory-token distribution. On-policy collection sets this data-distribution mismatch to zero at the rollout stage.

The target also admits a useful value interpretation. Let $Q_k(c, v)$ be the expected task return after choosing token v at context c and following the current policy thereafter. Then

$$\mathbb{E}_{v \sim r_k(\cdot|c)}[Q_k(c, v)] - \mathbb{E}_{v \sim \pi_k(\cdot|c)}[Q_k(c, v)] = \frac{\mathrm{Cov}_{v \sim \pi_k(\cdot|c)}(Q_k(c, v), w_k(c, v))}{Z_k(c)}. \tag{8}$$

Therefore, whenever hindsight support is positively correlated with current-policy action value, the reweighted target has higher local expected value than the unweighted policy. This condition makes explicit what the theorem does and does not assume: SEED converts hindsight support into an on-policy target, while the empirical benefit depends on the generated skills assigning greater support to better decisions.

A.2 DENSE SKILL CREDIT REMAINS INFORMATIVE UNDER SPARSE OR TIED REWARDS

Outcome-based RL assigns the same trajectory-level advantage to every valid action token in a rollout. Consequently, it cannot distinguish a locally useful decision from a harmful one within the same trajectory, and its reward-driven signal disappears completely when all outcomes in a group are tied. The next result shows that the skill-conditioned OPD term can remain non-degenerate in exactly this regime.

Proposition 2 (Decision-specific signal under reward ties)**.** *Consider a rollout group in which every trajectory has the same outcome. Then $A^{\mathrm{rl}}_{q,n,t,\ell} = 0$ for every valid token, and the clipped reward-driven term in Eq. 3.3 has zero gradient. Fix a current-policy context c, parameterize the ordinary student distribution as $p_z(\cdot \mid c) = \mathrm{softmax}(z(c))$, and define the conditional expected OPD loss up to detached constants by*

$$\overline{\mathcal{L}}_{\mathrm{opd},k,c}(z) = -\sum_{v \in \mathcal{V}} \pi_k(v \mid c) w_k(c, v) \log p_z(v \mid c). \tag{9}$$

At update initialization, where $p_z(\cdot \mid c) = \pi_k(\cdot \mid c)$,

$$\frac{\partial \overline{\mathcal{L}}_{\mathrm{opd},k,c}}{\partial z(c, v)} = \pi_k(v \mid c) \left[Z_k(c) - w_k(c, v) \right]. \tag{10}$$

Furthermore, with the inverse-policy weighted norm

$$\|a\|^2_{\pi_k^{-1}} = \sum_{v \in \mathcal{V}} \frac{a(v)^2}{\pi_k(v \mid c)},$$

the gradient magnitude satisfies the exact identity

$$\left\| \nabla_{z(c)} \overline{\mathcal{L}}_{\mathrm{opd},k,c} \right\|^2_{\pi_k^{-1}} = \mathrm{Var}_{v \sim \pi_k(\cdot|c)} \left[w_k(c, v) \right]. \tag{11}$$

Hence the OPD gradient is nonzero if and only if the expected hindsight gate is non-constant over candidate tokens with positive current-policy probability.

Proof. If all group outcomes are identical, then $R(\tau_q^{(n)}) - \mu_q = 0$ for every rollout, so the group-relative advantage in Eq. 3.3 is zero. The clipped surrogate therefore contributes no reward-derived policy gradient; the separate KL regularizer may still contribute a gradient, but it does not provide task-outcome credit.

For the OPD term, the softmax derivative gives

$$\frac{\partial[-\log p_z(u \mid c)]}{\partial z(c,v)} = p_z(v \mid c) - \mathbf{1}\{u = v\}.$$

Applying this identity to Eq. 9 and setting $p_z = \pi_k$ yields

$$\begin{aligned}\frac{\partial \overline{\mathcal{L}}_{\mathrm{opd},k,c}}{\partial z(c,v)} &= \left(\sum_u \pi_k(u \mid c) w_k(c,u)\right) \pi_k(v \mid c) - \pi_k(v \mid c) w_k(c,v) \\ &= \pi_k(v \mid c)\left[Z_k(c) - w_k(c,v)\right],\end{aligned}$$

which proves Eq. 10. Substituting this gradient into the weighted norm gives

$$\begin{aligned}\left\|\nabla_{z(c)} \overline{\mathcal{L}}_{\mathrm{opd},k,c}\right\|^2_{\pi_k^{-1}} &= \sum_v \pi_k(v \mid c)\left[w_k(c,v) - Z_k(c)\right]^2 \\ &= \mathrm{Var}_{v \sim \pi_k(\cdot \mid c)}\left[w_k(c,v)\right],\end{aligned}$$

because $Z_k(c) = \mathbb{E}_{v \sim \pi_k}[w_k(c,v)]$. The variance is zero exactly when $w_k(c,v)$ is constant π_k-almost surely. □

Equations 10–11 formalize the dense credit supplied by SEED. Tokens with $w_k(c,v) > Z_k(c)$ receive a negative logit derivative and are promoted by gradient descent, whereas tokens with $w_k(c,v) < Z_k(c)$ are relatively suppressed. Thus, although every gate value is nonnegative, softmax normalization converts variation in hindsight support into signed relative credit over candidate decisions. The variance in Eq. 11 quantifies the strength of this token-level signal: the more the hindsight skill discriminates among alternatives, the larger the OPD update can be. This establishes *informativeness*, not automatic correctness; whether the signal favors better actions is characterized by the value-alignment covariance in Eq. 8.

The tied-reward case is the sharpest contrast, but the distinction is more general. GRPO broadcasts one scalar $A^{\mathrm{rl}}_{q,n}$ across all valid tokens in a trajectory, while SEED assigns a context- and token-dependent coefficient through $w_k(c,v)$. It can therefore preserve useful local behaviors in failed trajectories and attenuate locally inefficient choices in successful trajectories, even when the terminal reward alone cannot identify those decisions.

A.3 SELF-EVOLVING SYNCHRONIZATION CONTROLS ANALYZER STALENESS

The final requirement concerns the source of the hindsight itself. A fixed analyzer or one-time skill dataset may be informative early in training but can become mismatched to the capabilities, strategies, and failure modes of a later policy. We quantify this mismatch by its effect on the detached OPD gate and hence on the auxiliary gradient.

For a completed token sample $x = (\tau, c, v)$ from the current distribution μ_k, let $s_r(x) = \mathcal{A}_r(x_\tau)$ be the skill produced by an analyzer checkpoint from iteration r. Holding the current policy π_k fixed for this comparison, define

$$\Delta_{k,r}(x) = \log \pi_k(v \mid H(c, s_r(x))) - \log \pi_k(v \mid c), \qquad g_{k,r}(x) = \sigma(\beta_{\mathrm{opd}} \Delta_{k,r}(x)). \tag{12}$$

The first index identifies the policy whose probabilities are re-scored, while the second identifies the analyzer that supplies the skill. For a fixed trainable parameter θ, define the resulting expected auxiliary gradient on current trajectories as

$$U_k(\mathcal{A}_r; \theta) = -\mathbb{E}_{x \sim \mu_k}\left[g_{k,r}(x) \nabla_\theta \log \pi_\theta(v \mid c)\right]. \tag{13}$$

Proposition 3 (Analyzer-staleness bound)**.** *Assume $\|\nabla_\theta \log \pi_\theta(v \mid c)\|_2 \le G$ for all relevant current-policy token samples. Then the discrepancy between the OPD gradient induced by the synchronized analyzer $\mathcal{A}_k$ and that induced by an earlier analyzer $\mathcal{A}_j$ satisfies*

$$\begin{aligned}\left\|U_k(\mathcal{A}_k; \theta) - U_k(\mathcal{A}_j; \theta)\right\|_2 &\le G\, \mathbb{E}_{x \sim \mu_k}\left[|g_{k,k}(x) - g_{k,j}(x)|\right] \\ &\le \frac{\beta_{\mathrm{opd}} G}{4} \mathbb{E}_{x \sim \mu_k}\left[|\Delta_{k,k}(x) - \Delta_{k,j}(x)|\right].\end{aligned} \tag{14}$$

In particular, using the analyzer instantiated from the current checkpoint sets this cross-iteration analyzer mismatch to zero at the rollout-and-analysis stage.

Proof. Subtract Eq. 13 for $r = k$ and $r = j$, apply Jensen's inequality, and use the score-norm bound:

$$\begin{aligned}
&\|U_k(\mathcal{A}_k;\theta) - U_k(\mathcal{A}_j;\theta)\|_2 \\
&\quad \le \mathbb{E}_{x\sim\mu_k}\left[|g_{k,k}(x) - g_{k,j}(x)|\,\|\nabla_\theta \log \pi_\theta(v \mid c)\|_2\right] \\
&\quad \le G\,\mathbb{E}_{x\sim\mu_k}\left[|g_{k,k}(x) - g_{k,j}(x)|\right].
\end{aligned}$$

Because $\sup_z \sigma'(z) = 1/4$, the map $z \mapsto \sigma(\beta_{\mathrm{opd}} z)$ is $\beta_{\mathrm{opd}}/4$-Lipschitz. Applying this fact pointwise to Eq. 12 proves the second inequality. □

Proposition 3 measures analyzer staleness in the quantity that directly affects learning: the change in skill-induced log-probability shift on the current policy's own trajectories. It does not require parameter distance between analyzers to be meaningful, nor does it assume that the updated analyzer is always more accurate. Instead, it states that if an old and a current analyzer produce skills with different behavioral implications, their OPD gradients can differ accordingly. The factor $\beta_{\mathrm{opd}}/4$ also exposes a trade-off: a sharper gate more strongly separates supported and unsupported tokens, but is correspondingly more sensitive to stale skill-induced shifts.

Combining the data and analyzer effects gives a compact decomposition of the two synchronization mechanisms. At a fixed current policy π_k and trainable parameter θ, let $U_{k,\theta}(\mu, \mathcal{A}_r)$ denote Eq. 13 with samples drawn from μ. Under the same score-norm bound,

$$\begin{aligned}
&\|U_{k,\theta}(\mu_k, \mathcal{A}_k) - U_{k,\theta}(\mu_j, \mathcal{A}_j)\|_2 \\
&\qquad \le 2G\ \mathrm{TV}(\mu_k, \mu_j) + \frac{\beta_{\mathrm{opd}} G}{4}\mathbb{E}_{x\sim\mu_j}\left[|\Delta_{k,k}(x) - \Delta_{k,j}(x)|\right].
\end{aligned} \tag{15}$$

The first term is *experience staleness*, controlled by collecting trajectories on-policy; the second is *supervision staleness*, controlled by refreshing the analyzer from the latest shared checkpoint. At each outer iteration, SEED uses the synchronized pair $(\mu_k, \mathcal{A}_k)$, so both cross-iteration mismatch terms are reset at data generation. During the subsequent inner policy optimization, the analyzer remains fixed at $\mathcal{A}_k$; therefore, the claim is not that analyzer lag is identically zero at every optimizer step, but that any within-update lag is discarded when trajectories and skills are regenerated from the next policy checkpoint.

Taken together, Propositions 1, 2, and 3 provide a one-to-one theoretical counterpart to the three design requirements in Section 1. On-policy collection aligns the auxiliary objective with the current policy's visited distribution, dense skill-conditioned re-scoring supplies token-specific credit beyond terminal outcomes, and the self-evolving shared analyzer prevents the hindsight target from remaining permanently tied to an earlier policy checkpoint.

B Additional Experimental Details

This section provides additional experimental details, including the datasets, baseline methods, the complete SEED algorithm and representative extracted skills, and the implementation details of our method.

B.1 Datasets

Our experiments cover three representative forms of long-horizon agentic interaction: embodied household reasoning, web navigation, and search-augmented question answering. Table 3 summarizes the benchmarks and the data sizes used for SFT, RL training, and evaluation.

ALFWorld. ALFWorld (Shridhar et al., 2021) exposes household environments derived from ALFRED through a text-based interaction interface. Given a natural-language instruction and a sequence of textual observations, the agent must select admissible actions to accomplish the specified household goal. We evaluate six task categories: *Pick*, *Look*, *Clean*, *Heat*, *Cool*, and *Pick2*. We report the main results on the 140-task seen split and use the 134-task unseen split for out-of-distribution evaluation.

Table 3: Detailed information on the agentic benchmarks and training data.

Domain	Benchmark(s)	#SFT Train Samples	#RL Train Samples	#Test Samples
Embodied Reasoning	ALFWorld	180	2,400	140 (seen) 134 (unseen)
Web Navigation	WebShop	180	2,400	128
Search-Augmented QA	NQ · TriviaQA · PopQA HotpotQA · 2WikiMultiHopQA MuSiQue · Bamboogle	180	19,200	51,713

WebShop. WebShop (Yao et al., 2022) evaluates agents in an interactive e-commerce environment. For each user request, the agent searches the product catalog, examines candidate items, selects the required attributes, and attempts to complete a purchase satisfying the specified constraints. The benchmark provides a normalized task-completion score for partial constraint satisfaction and a binary success metric for exact completion. We evaluate all methods on 128 test samples.

Search-Augmented QA. Following the Search-R1 protocol (Jin et al., 2025), we construct the search-based QA setting from seven datasets: Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), PopQA (Mallen et al., 2023), HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023). The agent may issue search queries, inspect retrieved documents, and synthesize the collected evidence before returning its final answer. The combined evaluation set contains 51,713 questions.

Training data configuration. For the SFT stage, we select 180 tasks from the training split of each benchmark setting. Each selected task is executed with 8 independent rollouts, yielding 1,440 trajectories for constructing the corresponding trajectory–skill training set. The subsequent RL stage uses 2,400 training instances for ALFWorld, 2,400 for WebShop, and 19,200 for Search-Augmented QA. SFT data construction and RL optimization are performed separately for each benchmark configuration.

B.2 Baselines

We compare SEED against three categories of baselines: prompting-only methods, outcome-based reinforcement learning methods, and self-distillation or skill-distillation methods. Unless a method is marked with an asterisk, evaluation is conducted using only the standard task prompt and the interaction history returned by the environment. The superscript * indicates that a natural-language skill is additionally supplied during validation or testing, while the backbone model and evaluation setting remain unchanged.

Prompting-only methods.

- *Vanilla*. We directly evaluate the original instruction-tuned backbone without performing any additional post-training. The agent receives the default environment prompt together with the observation–action history accumulated during interaction.
- *Skill-Prompt**. This baseline uses the same frozen parameters as *Vanilla*, but appends a retrieved task-relevant skill to the model context during validation and testing. Since no parameter optimization is involved, this comparison measures the benefit of using natural-language skills purely as inference-time guidance.

Outcome-based reinforcement learning.

- *GRPO* (Shao et al., 2024). GRPO is a critic-free reinforcement learning algorithm that samples multiple trajectories for each task and normalizes their scalar outcome rewards within the rollout group to obtain relative advantages. Each valid token in a trajectory is optimized

using the corresponding sequence-level advantage and a clipped importance-ratio objective. In our implementation, GRPO relies solely on terminal environment rewards and does not use process-level annotations, teacher predictions, or skill-conditioned supervision.

- *Skill-GRPO*. This variant retains the standard GRPO optimization objective while introducing a task-relevant natural-language skill into the policy context during both rollout collection and parameter updates. The skill can therefore affect the policy's exploration behavior and the trajectories subsequently reinforced by the outcome reward. It is removed during validation and testing, allowing us to assess whether the training-time guidance has been internalized by the policy.
- *Skill-GRPO**. This baseline follows the same training procedure as *Skill-GRPO*, but continues to provide the skill during validation and testing. It therefore maintains consistent skill conditioning across training and evaluation.

Self-distillation and skill-distillation methods.

- *OPSD* (Zhao et al., 2026). OPSD derives student and teacher branches from the same underlying model while conditioning them on different contexts. The student generates trajectories from the ordinary task context, whereas the teacher additionally observes privileged information available only during training. The teacher then re-scores the student's sampled tokens and produces dense token-level targets through distribution matching. Teacher-side outputs are detached from gradient computation, and the privileged context is not used at inference time.
- *GRPO+OPSD*. This baseline jointly optimizes the trajectory-level GRPO loss and the token-level OPSD objective. Environment rewards provide outcome-based supervision for complete trajectories, while OPSD contributes dense guidance at individual token positions. It serves as a controlled comparison for determining whether a straightforward combination of outcome-based RL and generic on-policy self-distillation is sufficient.
- *Skill-SD* (Wang et al., 2026). Skill-SD adapts self-distillation to multi-turn agent training by representing completed experience as compact natural-language skills. During optimization, a retrieved skill is supplied only to the teacher branch, while the student continues to operate from the ordinary task context. The student is therefore trained to absorb the behavioral guidance conveyed by the skill-conditioned teacher without requiring explicit skills during evaluation.
- *RLSD* (Yang et al., 2026a). RLSD employs a privileged self-teacher to refine token-level credit assignment within reinforcement learning. Rather than introducing an independent distribution-matching objective, it converts the teacher–student log-probability gap into a bounded coefficient that adjusts the magnitude of each token's GRPO update. The sign of the update remains determined by the environment-derived advantage, meaning that the privileged teacher controls update strength but not the reinforcement direction. The self-distillation contribution is emphasized early in training and gradually reduced toward the standard GRPO objective.
- *SDAR* (Lu et al., 2026a). SDAR preserves GRPO as the main outcome-optimization objective and adds a separately gated self-distillation loss. A privileged teacher branch evaluates the student's on-policy tokens under an augmented context, while a bounded gate controls the contribution of each teacher signal. The gate may depend on student uncertainty and the detached teacher–student log-probability difference, strengthening reliable positive guidance and reducing the influence of potentially noisy signals. In contrast to *RLSD*, SDAR leaves the original GRPO advantage unchanged and applies its gating mechanism only to the auxiliary distillation term.

For all reproduced post-training baselines, we use the same backbone models and environment interfaces as SEED. Whenever supported by the corresponding method, we also align the task batch size, rollout budget, rollout group size, number of policy updates, and evaluation protocol. The resulting comparisons primarily differ in their optimization objectives and in whether skills or other privileged information are available during training or evaluation.

Algorithm 1 SEED: Self-Evolving On-Policy Distillation

Require: SFT-initialized policy $\pi_{\theta_{\text{sft}}}$, task set $\mathcal{Q}$, fixed reference policy π_{ref}, context function H, group size N, gate sharpness β_{opd}, KL coefficient β_{KL}, OPD coefficient λ_{opd}, clip parameter ϵ_{clip}, learning rate η
Ensure: Trained policy π_θ

$\theta \leftarrow \theta_{\text{sft}}$
for each policy update **do**
  $\theta_{\text{old}} \leftarrow \theta$
  Sample a task batch $\mathcal{B} \subset \mathcal{Q}$
  // On-policy experience and synchronized skill analysis
  for each task $q \in \mathcal{B}$ **do**
    Sample $\mathcal{G}_q = \{\tau_q^{(n)}\}_{n=1}^N$ with $\tau_q^{(n)} \sim \pi_{\theta_{\text{old}}}(\cdot \mid q)$
    Compute μ_q and σ_q from $\{R(\tau_q^{(n)})\}_{n=1}^N$
    for each trajectory $\tau_q^{(n)} \in \mathcal{G}_q$ **do**
      $A_{q,n}^{\text{rl}} \leftarrow (R(\tau_q^{(n)}) - \mu_q)/(\sigma_q + \epsilon)$
      $s_q^{(n)} \leftarrow A_{\theta_{\text{old}}}(x_{\tau_q^{(n)}})$
      for each action step t in $\tau_q^{(n)}$ **do**
        $\tilde{h}_{q,n,t} \leftarrow H(h_{q,n,t}, s_q^{(n)})$
        Cache $\ell_{q,n,t,\ell}^{\text{old}}$ for all valid sampled tokens ℓ
      end for
    end for
  end for
  // Paired contextual re-scoring and joint optimization
  for each optimization minibatch of valid sampled tokens **do**
    Evaluate $\ell_{q,n,t,\ell}^{\text{skill}}$ and $\ell_{q,n,t,\ell}^{\theta}$ using the current π_θ
    $\Delta_{q,n,t,\ell} \leftarrow \text{sg}[\ell_{q,n,t,\ell}^{\text{skill}} - \ell_{q,n,t,\ell}^{\theta}]$
    $g_{q,n,t,\ell} \leftarrow \sigma(\beta_{\text{opd}} \Delta_{q,n,t,\ell})$
    $\rho_{q,n,t,\ell}(\theta) \leftarrow \exp(\ell_{q,n,t,\ell}^{\theta} - \ell_{q,n,t,\ell}^{\text{old}})$
    Compute the clipped GRPO loss $\mathcal{L}_{\text{rl}}$ from ρ, A^{rl}, and ϵ_{clip}
    $\mathcal{L}_{\text{opd}} \leftarrow \mathbb{E}[m \cdot g \cdot (\text{sg}[\ell^{\text{skill}}] - \ell^{\theta})]$
    $\mathcal{L}_{\text{SEED}} \leftarrow \mathcal{L}_{\text{rl}} + \lambda_{\text{opd}} \mathcal{L}_{\text{opd}}$
    $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{SEED}}$
  end for
end for

B.3 Algorithm and Extracted Skill Examples

Algorithm 1 summarizes the self-evolving training loop. At each update, the frozen snapshot $\pi_{\theta_{\text{old}}}$ both collects trajectories and analyzes the completed interactions. The current trainable policy π_θ then re-scores the same sampled tokens under ordinary and skill-augmented contexts; gradients flow only through the ordinary branch. The resulting OPD loss is optimized jointly with the KL-regularized GRPO objective before the updated policy becomes the next snapshot.

Table 4 provides representative skills extracted from successful and failed trajectories across ALFWorld, WebShop, and Search-based QA.

B.4 Implementation Details

Benchmark metrics. We adopt benchmark-specific measures following the corresponding evaluation protocols. For ALFWorld, let SR_c denote the success rate on task category c. The overall result assigns equal weight to all six categories and is computed as

$$\text{ALFWorld-Avg} = \frac{1}{6} \sum_{c=1}^{6} \text{SR}_c. \tag{16}$$

For Search-based QA, we use a dataset-balanced aggregate rather than pooling all questions together. Denoting the accuracy on dataset d by Acc_d, the reported score is

$$\text{Search-Avg} = \frac{1}{7} \sum_{d=1}^{7} \text{Acc}_d. \tag{17}$$

Table 4: **Episode-level skills extracted by the trajectory analyzer.** For each dataset, we present one successful and one failed trajectory. Skills derived from successful episodes capture reusable workflows, whereas those derived from failed episodes distill actionable failure-avoidance rules.

Outcome	Task	Episode-level skill
ALFWorld		
Success	clean some spatula and put it in diningtable.	Workflow: When tasked to clean and place an object, first locate the target object, take it to a cleaning station, clean it, and finally move it to the target location.
Failure	put a clean spatula in drawer.	Avoid moving an object to the target location before verifying both inventory and required object state. Confirm the object is held and already satisfies required conditions such as cleanliness; avoid repeatedly moving objects without checking state or placement success.
WebShop		
Success	Find me women's jumpsuits, rompers & overalls with button closure, quality polyester, polyester spandex, long sleeve for daily wear with color: hot pink, and size: x-large, and price lower than $40.00 dollars.	Workflow: Use a structured search query covering all required attributes, choose a result matching the core product category, then verify and select required options such as color and size before clicking Buy Now.
Failure	Find me machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: navy, and fit type: youth, and size: 3x-large, and price lower than $50.00 dollars.	Avoid clicking irrelevant search results or purchasing partial matches. If results do not match the target product type, reformulate the query; on the product page, select each required attribute once and verify material, color, size, fit, and price before Buy Now.
Search		
Success	Tie a Yellow Ribbon is the third album by American popular music group Dawn (Michael Anthony Orlando Cassavitas, Telma Hopkins & Joyce Vincent Wilson) released in which year?	Workflow: For a specific factual query, search the exact query, then extract and verify the answer directly from the search results before responding.
Failure	Le Juge is a comic whose story is inspired by a man who called himself what?	Avoid prematurely inferring an answer from search results that do not directly address the core query. First verify the relevant entity/relation, then extract the requested attribute.

For WebShop, we report two complementary metrics. *Score* measures the degree of task completion by reflecting how fully the purchased product satisfies the requirements specified in the user request; the normalized scores are averaged and scaled by 100. *Succ.* denotes the percentage of episodes in which all requirements are satisfied and the task is completed exactly.

External analyzer at the SFT stage. To construct the trajectory–skill supervision for SFT, we serialize each completed rollout into a structured record comprising the task instruction, the full interaction trajectory, and the terminal outcome. The external analyzer then examines the completed episode and produces a natural-language hindsight skill. The analyzer is used exclusively for offline skill annotation and does not participate in trajectory collection, which is carried out by the corresponding backbone model itself. We instantiate the external analyzer with GLM-5.2 (Z.ai, 2026), set the temperature to 0.0, and limit the maximum response length to 4,096 tokens. The complete prompt used for hindsight-skill annotation is provided in Figure 10.

Actor and analyzer prompts at the RL stage. During Stage 2, the actor follows the standard environment-interaction prompt for the corresponding benchmark; the prompt used for ALFWorld

Table 5: Training hyperparameters for SEED.

Hyperparameter	Value
Training steps	150
Training batch size	16 for ALFWorld and WebShop; 128 for Search
Rollout group size N	8
Learning rate	1×10^{-6}
PPO clip parameter ϵ_{clip}	0.2
Sharpness of OPD gate β_{opd}	5.0
OPD loss coefficient λ_{opd}	0.01
KL coefficient β_{KL}	0.01
Maximum prompt length	2,048 for ALFWorld ; 4,096 for WebShop and Search
Response length	512
Maximum interaction steps	30 for ALFWorld, 15 for WebShop, and 4 for Search.

is shown in Figure 11. The analyzer uses the same trajectory-analysis prompt as in the SFT stage, as presented in Figure 10.

Training hyperparameters. Table 5 records the training hyperparameters that are required for exact reproduction.

Computing details. Training is conducted on 8 Nvidia A800 80G GPUs.

C Supplementary Results

C.1 Detailed Sample Efficiency Comparison

Table 6 provides the full comparison across five training data fractions on ALFWorld and WebShop. SEED outperforms GRPO at every fraction on both benchmarks. On ALFWorld, the gains range from 13.4 to 30.2 points. With only 60% of the data, SEED reaches 80.7, exceeding the 75.0 achieved by GRPO with the full dataset. On WebShop, SEED improves performance by 5.5 to 15.6 points and reaches 75.0 with 80% of the data, compared with 63.3 for GRPO trained on the full dataset. These results show that dense hindsight supervision makes more effective use of collected trajectories and remains beneficial as the training set grows.

Table 6: **Sample efficiency comparison on ALFWorld and WebShop.** We report performance using different proportions of the available training data. SEED consistently outperforms GRPO across both benchmarks, while Δ denotes the absolute performance improvement over GRPO.

Method	20%	40%	60%	80%	100%
ALFWorld					
GRPO	27.3	42.2	56.3	58.6	75.0
SEED	40.7	58.9	80.7	88.8	91.8
Δ	+13.4	+16.7	+24.4	+30.2	+16.8
WebShop					
GRPO	31.3	45.3	57.0	63.6	63.3
SEED	37.5	53.1	62.5	75.0	78.9
Δ	+6.2	+7.8	+5.5	+11.4	+15.6

C.2 Cross-Domain Generalization

Table 7 reports results for each task family on ALFWorld Unseen. SEED improves the average success rate from 70.9 to 86.2 and outperforms GRPO on five of the six families. The largest gain

Table 7: **Cross-domain generalization results on ALFWorld Unseen.** Using Qwen2.5-3B-Instruct as the backbone, we evaluate performance on six unseen task categories. SEED outperforms GRPO on five categories and increases the average success rate by 15.3 percentage points.

	ALFWorld Unseen						
Method	**Pick**	**Look**	**Clean**	**Heat**	**Cool**	**Pick2**	**Avg.**
ReAct	17.4	6.7	8.8	7.4	9.1	0.0	8.2
GRPO	73.9	60.0	**82.4**	59.3	72.7	76.9	70.9
SEED	**90.4**	**78.3**	79.5	**94.3**	**86.2**	**88.2**	**86.2**
Δ	+16.5	+18.3	-2.9	+35.0	+13.5	+11.3	+15.3

occurs on *Heat*, where the success rate increases by 35.0 points. Improvements are also substantial on *Look* and *Pick*, reaching 18.3 and 16.5 points, respectively. Although performance on *Clean* decreases by 2.9 points, the broad gains across the remaining families indicate that SEED learns reusable behavioral guidance that transfers beyond the training environments.

C.3 Multimodal Extension

To examine whether SEED extends beyond text-only interaction, we evaluate Qwen2.5-VL-3B-Instruct (Bai et al., 2025) on two vision-based agentic benchmarks. Sokoban (Schrader, 2018) presents each state as a 6×6 visual grid containing the player, boxes, walls, and target locations. The agent must navigate the grid and push every box onto a target. Since boxes cannot be pulled, a poor push may create an irreversible dead end. The task therefore tests visual state tracking and long-horizon spatial planning. EZPoints from Gym Cards (Zhai et al., 2024) presents two playing cards together with a partially constructed expression. At each step, the agent selects a card value or an arithmetic operator to build an expression that evaluates to 12, using each card exactly once. This task couples fine-grained card recognition with sequential arithmetic reasoning.

As shown in Table 8, SEED achieves success rates of 82.0% on Sokoban and 100.0% on EZPoints. It improves over GRPO by 14.9 and 13.1 points, respectively, raising the average success rate from 77.0% to 91.0%. ReAct reaches only 7.4% on average, which highlights the need for policy learning in these visually grounded tasks. The consistent gains across spatial planning and visual arithmetic show that SEED can turn multimodal trajectories into useful hindsight supervision and internalize that guidance through self-evolving OPD, without skill prompts during evaluation. Figure 7 presents a step-by-step visualization of a representative Sokoban trajectory.

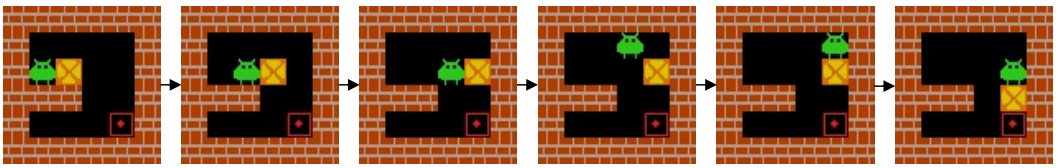

Figure 7: **A representative trajectory on Sokoban.** The sequence shows six consecutive actions executed by the agent. Arrows indicate the temporal progression of the trajectory, and the action taken at each step is displayed below the corresponding observation.

C.4 Additional Training Dynamics

Figure 8 presents the success-rate trajectories for all three backbones on ALFWorld, Search-based QA, and WebShop. Performance improves in every setting, although the convergence pattern varies by domain. ALFWorld shows the largest absolute increase and approaches a success rate of 0.9 for all three backbones. Search-based QA improves rapidly during the early updates before stabilizing between 0.47 and 0.55. WebShop follows a steadier upward trend and finishes between 0.68 and

Table 8: **Results on vision-based agentic benchmarks**. We report success rates (%) on Sokoban [6×6] and EZPoints using Qwen2.5-VL-3B-Instruct as the backbone model.

Method	Sokoban [6×6] ↑	EZPoints ↑	Average ↑
ReAct	11.7	3.1	7.4
GRPO	67.1	86.9	77.0
SEED	**82.0**	**100.0**	**91.0**
Δ	+14.9	+13.1	+14.0

0.75. The consistent progress across model families and environments shows that SEED is not tied to a particular backbone or form of agentic interaction.

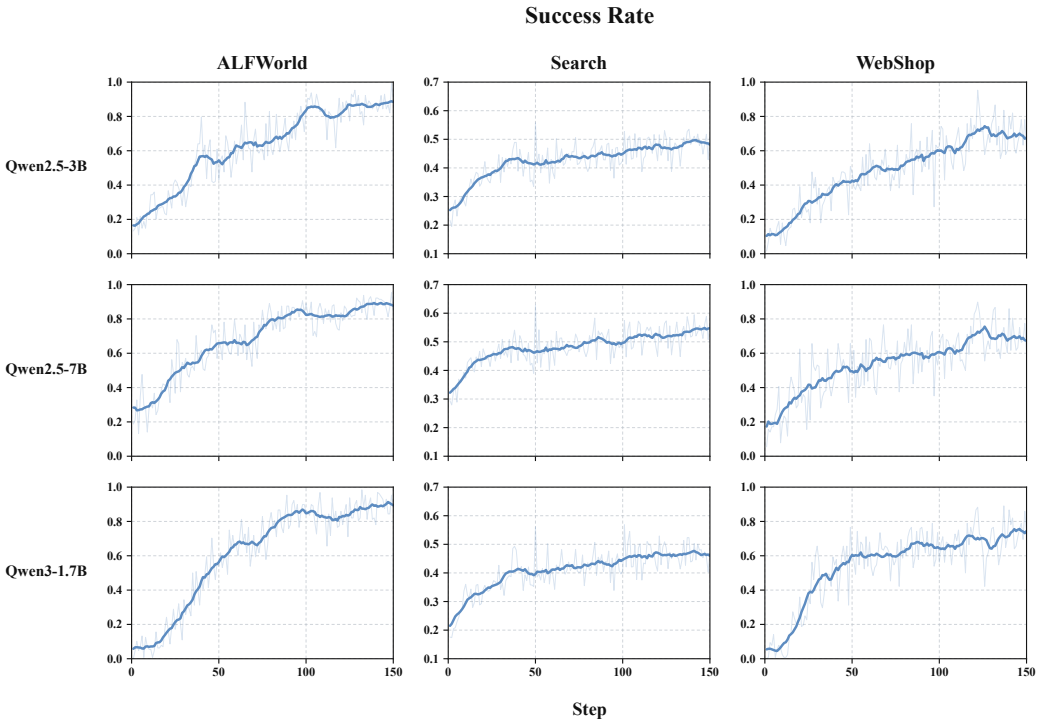

Figure 8: **Success rates across three backbones and three domains.** Success rates increase over training in all nine settings, showing consistent learning across model scales and agentic tasks.

Figure 9 shows the corresponding OPD losses. The loss decreases from its initial value and stabilizes at a lower level in all nine settings. The Qwen2.5 models exhibit gradual convergence, while Qwen3-1.7B shows a sharper early decline, especially on ALFWorld. Lower OPD loss indicates that the ordinary policy increasingly assigns probability to actions favored by hindsight supervision. Together with the rising success rates, this trend supports the stability of the self-evolving loop. The latest policy generates updated experience and hindsight skills, and OPD internalizes that guidance in subsequent updates.

D CASE STUDY

Figures 12–17 illustrate how SEED applies internalized behavioral guidance during evaluation without skill inputs. In the first ALFWorld example, the policy decomposes the task into locating the ladle, cleaning it, opening the drawer, and completing the placement. It correctly handles the drawer as a necessary precondition before the final action. The second example requires two books and repeated navigation between the desk and bed. The policy tracks that the first book has already

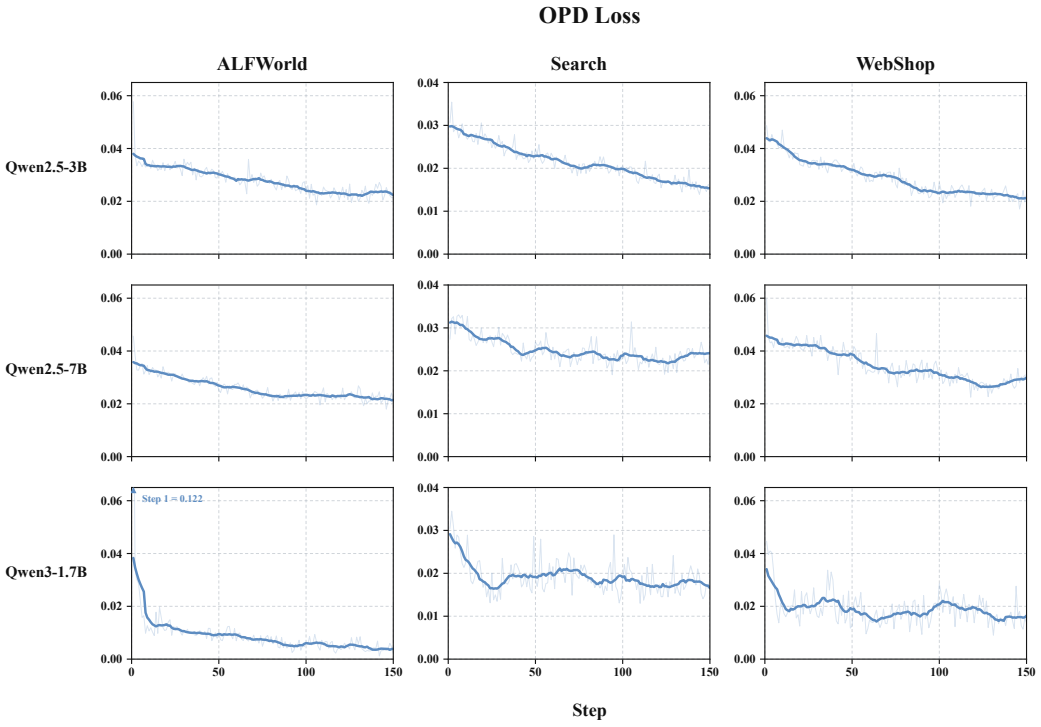

Figure 9: **OPD loss dynamics.** The loss generally decreases and stabilizes during training, indicating that the policy progressively internalizes the behavioral guidance provided by hindsight skills.

been placed, returns for the second, and completes the remaining subgoal without losing progress. These trajectories demonstrate coherent state tracking and precondition management over extended interactions.

The Search-based QA examples show that the policy adapts its information gathering to the available evidence. It answers the first question after one search because the retrieved passages directly establish the shared profession. For the second question, it first identifies *Finding Neverland*, then issues a targeted search for its director before answering *Marc Forster*. In WebShop, the policy retains the requested attributes and price limits from search through purchase. It verifies the product details and selects the required color and size before buying. Together, these cases show that SEED learns to decompose tasks, revise plans from new evidence, and preserve constraints over long trajectories. Because skills are not provided during inference, the observed behavior reflects guidance internalized through self-evolving OPD rather than external prompting.

E ADDITIONAL DISCUSSION

SEED treats completed on-policy experience as an evolving source of supervision. Our experiments cover several forms of long-horizon interaction, but broader benchmarks such as DeepPlanning (Zhang et al., 2026b), Long-Horizon-Terminal-Bench (Li et al., 2026), OdysseyArena (Xu et al., 2026), and RobotEQ (Fang et al., 2026) provide more demanding tests. These settings involve longer workflows, richer state spaces, and greater interaction complexity, often with rare terminal success. Evaluating SEED in such environments would test whether policy-synchronized hindsight can preserve decisive events across extended contexts and remain useful as the policy explores more diverse behaviors. Benchmarks with intermediate or partial-credit grading also make it possible to study how external progress signals interact with the token-level hindsight supervision.

However, longer tasks also expose a central risk of self-evolution: internally generated supervision can inherit model errors and plateau below oracle-supervised training (Jiang et al., 2025b; Qi et al., 2026). Because the actor and analyzer share the same policy, improvements transfer across both roles, but their blind spots can also be shared. An inaccurate analysis may turn a recurring policy er-

ror into an apparently reusable rule, which later updates could reinforce. Evidence of self-preference in model-based evaluation further suggests that a model may systematically favor outputs resembling its own (Mahbub & Feng, 2026). On-policy alignment and confidence gating help reduce distribution mismatch and noisy self-generated supervision (Kumar et al., 2025; Jiang et al., 2025a), but neither confidence nor self-judgment guarantees semantic correctness (Jiang et al., 2025b; Zhou et al., 2026). A more self-correcting version of SEED could anchor skills to verifiable state changes, compare analyses across policy snapshots, and preserve uncertainty alongside each skill. It could also organize hindsight hierarchically by first summarizing local transitions and then deriving trajectory-level rules (Ge et al., 2025). Search-discovered reasoning abstractions (Wu et al., 2024) may provide useful structures for this aggregation. Policy-aware exploration (Wu et al., 2026a) could collect trajectories that distinguish competing rules, while uncertainty-aware self-distillation (Lu et al., 2026a) may offer stronger filters for deciding which guidance to internalize.

Training efficiency is another practical constraint. SEED adds no deployment overhead, but training requires trajectory analysis and paired scoring under ordinary and skill-augmented contexts. The cost grows with interaction length and multimodal context. Speculative decoding methods such as DOUBLE (Shen et al., 2026) and DSPARK (Cheng et al., 2026) could reduce the autoregressive cost of rollout collection and skill generation. Cached representations and batched paired scoring could further reduce the cost of OPD. Another promising direction is selective analysis: the analyzer could focus on trajectories with high uncertainty, novel states, or disagreement between reward and hindsight. This would preserve the self-evolving supervision loop while avoiding repeated analysis of redundant experience.

Analyzer Prompt

Analyze the following agent episode and return ONLY valid JSON.

You need to complete two fields:
1. Write a concise episode_summary.
2. Write one episode_skill that extracts the successful trajectory into workflow: the core decision rule and action ordering that made this trajectory work. / Write one episode_skill that extracts the failed trajectory into avoidance rules: the core mistake and warning signs that agent should avoid.

Important constraints:
- Write episode_skill as one short policy-facing skill, not as retrospective explanation of the trajectory.
- Return only these top-level fields: episode_summary, episode_skill.

Return format:
```
{
 "episode_summary": "string",
 "episode_skill": "string"
}
```

Episode context:
- Task description: <TASK DESCRIPTION>
- episode_success: <success|failure>
- Interaction trajectory: <FORMATTED_TRAJECTORY>

Figure 10: Prompt of analyzer.

Actor Prompt

You are an expert agent operating in the ALFRED Embodied Environment. Your task is to: {task_description}
Prior to this step, you have already taken {step_count} step(s). Below are the most recent {history_length} observations and the corresponding actions you took: {action_history}
You are now at step {current_step} and your current observation is: {current_observation}
Your admissible actions of the current situation are: [{admissible_actions}].

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 admissible action for current step and present it within <action> </action> tags.

Figure 11: Prompt of actor (the policy model) in ALFWorld.

A trajectory of an agent trained with SEED — ALFWorld

Task: *clean some ladle and put it in drawer.*

[Step 1]
Observation: -= Welcome to TextWorld, ALFRED! =- You are in the middle of a room. Looking quickly around you, you see a cabinet 2, a cabinet 1, a coffeemachine 1, a countertop 3, a countertop 2, a countertop 1, a diningtable 1, a drawer 6, a drawer 5, a drawer 4, a drawer 3, a drawer 2, a drawer 1, a fridge 1, a garbagecan 1, a microwave 1, a sinkbasin 1, a stoveburner 4, a stoveburner 3, a stoveburner 2, a stoveburner 1, and a toaster 1.
Reasoning: First, I need to find a ladle. Ladles are often found on dining tables or in cabinets. Let's check the dining table and cabinets.
Action: go to diningtable 1

[Step 2]
Observation: You arrive at diningtable 1. On the diningtable 1, you see a apple 2, a bread 2, a bread 1, a butterknife 3, a cup 2, a dishsponge 1, a fork 2, a knife 2, a ladle 2, a lettuce 1, a mug 2, a potato 3, a potato 2, a soapbottle 1, and a spoon 2.
Reasoning: I need to take the ladle, clean it, and then put it in the drawer. First, I should take the ladle from the dining table. After that, I can go to the sink and use the dishsponge to clean it. Finally, I need to go to a drawer and put the cleaned ladle inside.
Action: take ladle 2 from diningtable 1

[Step 3]
Observation: You pick up the ladle 2 from the diningtable 1.
Reasoning: I need to clean the ladle and then place it in a drawer. First, I should go to the sinkbasin to wash the ladle. After cleaning, I can go to a drawer and put the ladle inside.
Action: go to sinkbasin 1

[Step 4]
Observation: You arrive at sinkbasin 1. On the sinkbasin 1, you see a apple 1, a pot 1, a spatula 3, and a spatula 2.
Reasoning: I need to clean the ladle and then put it in a drawer. First, I should go to the sinkbasin to clean the ladle. After cleaning, I can go to a drawer and put the ladle inside.
Action: clean ladle 2 with sinkbasin 1

[Step 5]
Observation: You clean the ladle 2 using the sinkbasin 1.
Reasoning: I need to place the cleaned ladle in a drawer. I've already cleaned it, so now I just need to go to a drawer and put it inside. The drawers are all nearby, so I'll go to one and put the ladle there.
Action: go to drawer 1

[Step 6]
Observation: You arrive at drawer 1. The drawer 1 is closed.
Reasoning: I need to put the cleaned ladle into the drawer. First, I should open the drawer to put the ladle inside. The drawer is closed, so I need to open it. After opening the drawer, I can place the ladle inside.
Action: open drawer 1

[Step 7]
Observation: You open the drawer 1. The drawer 1 is open. In it, you see nothing.
Reasoning: I've already taken the ladle, cleaned it, and opened the drawer. Now I need to put the cleaned ladle inside the drawer. I should move the ladle to the drawer using the 'move ladle 2 to drawer 1' action.
Action: move ladle 2 to drawer 1

Figure 12: **ALFWorld Example 1.** SEED locates and cleans the ladle, opens the closed drawer, and completes the placement in seven steps, demonstrating task decomposition and correct precondition handling.

A trajectory of an agent trained with SEED — ALFWorld

Task: *put two book in bed.*

[Step 1]
Observation: -= Welcome to TextWorld, ALFRED! =- You are in the middle of a room. Looking quickly around you, you see a bed 1, a desk 1, a drawer 2, a drawer 1, a garbagecan 1, a shelf 1, and a sidetable 1.
Reasoning: I need to find two books to place on the bed. The desk and shelves are good places to start, as they often contain or have access to books. After finding the books, I can 'put' them on the bed.
Action: go to desk 1

[Step 2]
Observation: You arrive at desk 1. On the desk 1, you see a book 2, a book 1, a creditcard 1, a desklamp 1, a keychain 1, a mug 3, a mug 2, a mug 1, a pen 3, a pen 2, a pen 1, a pencil 1, and a tissuebox 1.
Reasoning: I need to take the two books from the desk and place them in the bed. First, I should take book 1 and book 2 from the desk, then go to the bed and put them there.
Action: take book 1 from desk 1

[Step 3]
Observation: You pick up the book 1 from the desk 1.
Reasoning: I need to take the second book and place both books on the bed. I have already picked up the first book, so now I just need to go to the bed and put it there.
Action: go to bed 1

[Step 4]
Observation: You arrive at bed 1. On the bed 1, you see a cellphone 3, a cellphone 2, a laptop 1, a pillow 1, and a teddybear 1.
Reasoning: I need to first place the book I've already taken on the bed, and then find a second book to complete the task. I've already taken book 1, so now I need to go back to the desk to take book 2, and then place both books on the bed.
Action: move book 1 to bed 1

[Step 5]
Observation: You move the book 1 to the bed 1.
Reasoning: I need to take the second book and place it in the bed as well. I've already taken the first book, so now I just need to find the second book and move it to the bed.
Action: go to desk 1

[Step 6]
Observation: You arrive at desk 1. On the desk 1, you see a book 2, a creditcard 1, a desklamp 1, a keychain 1, a mug 3, a mug 2, a mug 1, a pen 3, a pen 2, a pen 1, a pencil 1, and a tissuebox 1.
Reasoning: First, I need to take the second book from the desk, then go to the bed and put both books there.
Action: take book 2 from desk 1

[Step 7]
Observation: You pick up the book 2 from the desk 1.
Reasoning: I need to take book 2 and place it on the bed 1. I've already picked up book 2, so now I just need to go to the bed and put it there.
Action: go to bed 1

[Step 8]
Observation: You arrive at bed 1. On the bed 1, you see a book 1, a cellphone 3, a cellphone 2, a laptop 1, a pillow 1, and a teddybear 1.
Reasoning: I need to place both books on the bed. I already placed book 1, so now I just need to take book 2 and place it on the bed as well.
Action: move book 2 to bed 1

Figure 13: **ALFWorld Example 2.** SEED places two books through repeated trips between the desk and bed while retaining the completed first placement, demonstrating progress tracking over an extended trajectory.

A trajectory of an agent trained with SEED — Search QA

Task: *Lev Pontryagin and Leonty Magnitsky were both what?*

[Step 1]

Observation: Question: Lev Pontryagin and Leonty Magnitsky were both what?
Reasoning: NULL
Action: <search>Lev Pontryagin and Leonty Magnitsky</search>

[Step 2]

Observation: Doc 1: "Leonty Magnitsky" Leonty Magnitsky Leonty Filippovich Magnitsky (), born Telyatin (), (June 9, 1669, Ostashkov – October 19, 1739, Moscow) was a Russian mathematician and educator. Magnitsky was born into a peasant family. According to some accounts, he graduated from the Slavic Greek Latin Academy in Moscow. From 1701 and until his death, he taught arithmetic, geometry and trigonometry at the Moscow School of Mathematics and Navigation, becoming its director in 1716. In 1703, Magnitsky wrote his famous ""Arithmetic"" (Арифметика; 2,400 copies), which was used as the principal textbook on mathematics in Russia until the middle of the 18th century. Mikhail Lomonosov Doc 2: "Lev Pontryagin" Executive Committee of the IMU at the time, vigorously objected. Although the IMU stood by its decision to award Margulis the Fields Medal, Margulis was denied a Soviet exit visa by the Soviet authorities and was unable to attend the 1978 ICM in person. Pontryagin also participated in a few notorious political campaigns in the Soviet Union, most notably, in the Luzin affair. Lev Pontryagin Lev Semyonovich Pontryagin (, also written Pontriagin or Pontrjagin) (3 September 1908 – 3 May 1988) was a Soviet mathematician. He was born in Moscow and lost his eyesight due to a primus stove explosion Doc 3: "Lev Pontryagin" Lev Pontryagin Lev Semyonovich Pontryagin (, also written Pontriagin or Pontrjagin) (3 September 1908 – 3 May 1988) was a Soviet mathematician. He was born in Moscow and lost his eyesight due to a primus stove explosion when he was 14. Despite his blindness he was able to become one of the greatest mathematicians of the 20th century, partially with the help of his mother Tatyana Andreevna who read mathematical books and papers (notably those of Heinz Hopf, J. H. C. Whitehead, and Hassler Whitney) to him. He made major discoveries in a number of fields of mathematics, including algebraic
Reasoning: 1. The question asks what Lev Pontryagin and Leonty Magnitsky were both. 2. From the search results, we see that Leonty Magnitsky was a Russian mathematician and educator. 3. Lev Pontryagin was described as a Soviet mathematician. 4. The common profession they share is mathematician.
Action: <answer>mathematician</answer>

Figure 14: **Search-based QA Example 1.** A single query retrieves evidence that both people are mathematicians, allowing SEED to answer without unnecessary searches.

A trajectory of an agent trained with SEED — Search QA

Task: *Who directed the J. M. Barrie biographical film that Michelle Sy produced?*

[Step 1]
Observation: Question: Who directed the J. M. Barrie biographical film that Michelle Sy produced?
Reasoning: Initial search query; no explicit reasoning was logged before this action.
Action: <search>J. M. Barrie biographical film produced by Michelle Sy</search>

[Step 2]
Observation: Doc 1: "Michelle Sy" Michelle Sy Michelle Sy is an American film producer and executive producer. She started her career as a production assistant, working on films such as ""Good Will Hunting"" and ""The Mighty"". Later on in her career she moved into producing films herself. Her most successful film to date is ""Finding Neverland""; a J. M. Barrie biographical film made in 2004. Other works include 2007 film ""Fan-Demanium"" and 2004's ""My Name Is Modesty: A Modesty Blaise Adventure"". In addition, 2002 release ""Stolen Summer"", which Sy produced, was the first independent film stemming from ""Project Greenlight"", a film competition made as a Doc 2: "Michelle Sy" TV series with Ben Affleck and Matt Damon. Sy appeared in the series, reflecting its ""behind the scenes"" nature. Sy is producing a biopic about late musician Jeff Buckley with the singer's mother, Mary Guibert, which as of May 2008 is in the scripting stage. Michelle Sy Michelle Sy is an American film producer and executive producer. She started her career as a production assistant, working on films such as ""Good Will Hunting"" and ""The Mighty"". Later on in her career she moved into producing films herself. Her most successful film to date is ""Finding Neverland""; a J. M. Barrie Doc 3: "J. M. Barrie" J. M. Barrie Sir James Matthew Barrie, 1st Baronet, (; 9 May 1860 19 June 1937) was a Scottish novelist and playwright, best remembered today as the creator of Peter Pan. He was born and educated in Scotland and then moved to London, where he wrote a number of successful novels and plays. There he met the Llewelyn Davies boys, who inspired him to write about a baby boy who has magical adventures in Kensington Gardens (included in ""The Little White Bird""), then to write ""Peter Pan, or The Boy Who Wouldn't Grow Up"", a ""fairy play"" about an ageless
Reasoning: 1. The question asks who directed the J. M. Barrie biographical film that Michelle Sy produced. 2. From the search results, we see that Michelle Sy produced the film "Finding Neverland", which is a J. M. Barrie biographical film. 3. Now we need to find who directed this film. 4. Conduct a new search query to find the director of "Finding Neverland".
Action: <search>who directed Finding Neverland movie</search>

[Step 3]
Observation: Doc 1: "Finding Neverland (film)" Finding Neverland (film) Finding Neverland is a 2004 historical fantasy drama film directed by Marc Forster and written by David Magee, based on the play ""The Man Who Was Peter Pan"" by Allan Knee. The film is about playwright J. M. Barrie and his relationship with a family who inspired him to create ""Peter Pan"". The film earned several nominations at the 77th Academy Awards, including Best Picture, Best Adapted Screenplay and Best Actor for Johnny Depp's portrayal of J. M. Barrie, and won for Best Original Score. The film was the inspiration for the stage musical of the same Doc 2: "Finding Neverland (film)" the role of Sylvia Llewelyn Davies. Finding Neverland (film) Finding Neverland is a 2004 historical fantasy drama film directed by Marc Forster and written by David Magee, based on the play ""The Man Who Was Peter Pan"" by Allan Knee. The film is about playwright J. M. Barrie and his relationship with a family who inspired him to create ""Peter Pan"". The film earned several nominations at the 77th Academy Awards, including Best Picture, Best Adapted Screenplay and Best Actor for Johnny Depp's portrayal of J. M. Barrie, and won for Best Original Score. The film was the inspiration for Doc 3: "Finding Neverland (film)" his book where he wrote the plays that he ripped apart and that his mother glued back together for him. Barrie sits down and puts his arm around Peter to comfort him. They both fade, and all that is left is the bench. In addition to Johnny Depp as Barrie and Kate Winslet as Sylvia Llewelyn Davies, the film stars Dustin Hoffman as producer Charles Frohman, Julie Christie as Sylvia's mother Emma du Maurier and Radha Mitchell (who had starred in Forster's ""Everything Put Together"") as Barrie's wife Mary. Hoffman had appeared a dozen years earlier in title role of
Reasoning: 1. The question asks who directed the J. M. Barrie biographical film that Michelle Sy produced. 2. From the search results, we see that Michelle Sy produced the film "Finding Neverland". 3. The question then asks who directed this film. 4. In the search results, we find that the film was directed by Marc Forster.
Action: <answer>Marc Forster</answer>

Figure 15: **Search-based QA Example 2.** SEED first identifies *Finding Neverland*, then performs a targeted second search to verify that Marc Forster directed the film.

A trajectory of an agent trained with SEED — WebShop

Task: *Find me easy use bathing accessories for dry hair with color: green, and price lower than 40.00 dollars.*

[Step 1]
Observation: 'Search'
Reasoning: I need to search for easy use bathing accessories for dry hair with green color and price under $40. The search query should include keywords like "bathing accessories dry hair green" to filter the results effectively. I'll use the search action to initiate the search.
Action: search[bathing accessories dry hair green]

[Step 2]
Observation: 'Back to Search' [SEP] 'Page 1 (Total results: 50)' [SEP] 'Next >' [SEP] 'B098824K9T' [SEP] 'Spa Savvy Shower Cap and Twist Hair Turban Duo, Enhance Your Shower Experience with this Complete Set, Fully Lined Shower Cap, Microfiber Turban (Green)' [SEP] '$11.99' [SEP] 'B094QGVQLK' [SEP] 'ZYUN Bathtub Cushion, Full Body Spa Bath Mat with Cushion, 3D Breathable Bathtub Mat, Non-Slip Bath Pillows, Neck and Back Support, for Parents, Children' [SEP] '$32.99' [SEP] 'B08G14B779' [SEP] 'NIUTA 2 Pack Hair Towel Wrap, Microfiber Quick Drying Hair Towels, Super Absorbent Quick Dry Hair Towel, Wrapped Bath Cap (Pink+Light Blue)' [SEP] '$4.99' [SEP] 'B09DD2C1XD' [SEP] 'Beaupretty 8pcs Kids Nightcaps Hair Bonnets Sleeping Caps Hair Protection Wide Rim Hat Bath Shower Accessories for Home' [SEP] '$19.99' [SEP] 'B08BV94TBP' [SEP] 'EASY SPA Natural Bristle Wooden Body Dry Brush for Bath Shower, Shower Scrubber With Long Handle, The Best Dry Brush For Cellulite And Lymphatic' [SEP] '$100.0' [SEP] 'B09LLRSWF4' [SEP] 'Yheakne Metal Hair Slide Clip Barrette Geometric Hair Holder Clip Pin Vintage Hair Slide Pin Bun Holder Alloy Hair Clip Decorative Hair Accessories for Women and Girls (Gold)' [SEP] '$8.5' [SEP] 'B08SKT2TCH' [SEP] 'JYLRX natural loofah sponge,Very Suitable for Kitchen Household use and Personal Skin Care Products with Cleanser Pack of 4 (white)' [SEP] '$12.98' [SEP] 'B09NLZFMRG' [SEP] 'ONGHAHYIS Shower Scrubbing Mitt Gloves Body Brush Take A Shower in 1 Minute Exfoliating Scrub Foot Scrubber Cath Sponge High-Elastic Five-Finger Bath Towel (Color : Random Color 1pc)' [SEP] '$22.16' [SEP] 'B09GF68QPJ' [SEP] 'Electric Shower Brush Set,Deep Cleaning Bath Shower Wash Brush with Long Handle,5 in 1 Spin SPA Brush Back Scrubber Set for Helps Promote Healthier Younger Looking Skin Soothes Tired Muscles' [SEP] '$26.99' [SEP] 'B09FYBZNQL' [SEP] '7pcs Hair Conditioner For Damaged Dry Hair, Hair Mask Repairing Hair Deep Conditioner Conditioning Treatment for Damaged Hair' [SEP] '$11.99'
Reasoning: I need to find easy-use bathing accessories for dry hair, preferably in green color, under $40. The product I found that matches all these criteria is the Spa Savvy Shower Cap and Twist Hair Turban Duo at $11.99. Therefore, I should click on this product to verify details and proceed with the purchase.
Action: click[b098824k9t]

[Step 3]
Observation: 'Back to Search' [SEP] '< Prev' [SEP] 'color' [SEP] 'green' [SEP] 'purple' [SEP] 'Spa Savvy Shower Cap and Twist Hair Turban Duo, Enhance Your Shower Experience with this Complete Set, Fully Lined Shower Cap, Microfiber Turban (Green)' [SEP] 'Price: $11.99' [SEP] 'Rating: N.A.' [SEP] 'Description' [SEP] 'Features' [SEP] 'Reviews' [SEP] 'Buy Now'
Reasoning: I need to find easy use bathing accessories for dry hair with green color option at price under $40. The search query I used earlier successfully returned a product that matches these criteria. I clicked on the Spa Savvy Shower Cap and Twist Hair Turban Duo which shows the green color option at $11.99. The product appears to be a set including a lined shower cap and microfiber turban. It meets the requirement of being easy use and for dry hair as it is a shower cap/turban set. Therefore, I should select the green color option and then purchase the item.
Action: click[green]

[Step 4]
Observation: 'Back to Search' [SEP] '< Prev' [SEP] 'color' [SEP] 'green' [SEP] 'purple' [SEP] 'Spa Savvy Shower Cap and Twist Hair Turban Duo, Enhance Your Shower Experience with this Complete Set, Fully Lined Shower Cap, Microfiber Turban (Green)' [SEP] 'Price: $11.99' [SEP] 'Rating: N.A.' [SEP] 'Description' [SEP] 'Features' [SEP] 'Reviews' [SEP] 'Buy Now'
Reasoning: I need to find easy use bathing accessories for dry hair with color: green, and price lower than $40. From the search results I obtained, I found a Spa Savvy Shower Cap and Twist Hair Turban Duo that matches the criteria. I clicked on the product to see the details and selected the green color option. Now I'm at the product page with the correct color selection and price, so I should click "Buy Now" to complete the purchase.
Action: click[buy now]

Figure 16: **WebShop Example 1.** SEED identifies a green shower cap and hair turban set under $40, verifies the selected variant, and completes the purchase in four steps.

A trajectory of an agent trained with SEED — WebShop

Task: *Find me machine wash men's t-shirts with long sleeve with color: mossy oak country, and size: 5x-large big, and price lower than 60.00 dollars.*

[Step 1]

Observation: 'Search'

Reasoning: I need to search for men's long sleeve t-shirts in mossy oak country color, 5x-large big size, under $60. The search query should be formatted as a natural language sentence describing the product requirements, including size, color, and price constraints. The search function has been clicked already, so I need to now formulate and submit a search query.

Action: search[men's long sleeve t-shirt mossy oak country 5x-large big]

[Step 2]

Observation: 'Back to Search' [SEP] 'Page 1 (Total results: 50)' [SEP] 'Next >' [SEP] 'B00O30JLDK' [SEP] 'Legendary Whitetails Men's Non-Typical Long Sleeve T-Shirt' [SEP] '$10.52 to $40.5' [SEP] 'B09QGK5XHZ' [SEP] 'WENKOMG1 Men's Long Sleeve Undershirt with Mask Turtleneck Hooded T-Shirt Solid Color Workout Tops Zipper Side Slit Shirts Slim Fit Sweatshirt Spring/Summer Tee Shirts(Gray,)' [SEP] '$8.39' [SEP] 'B09QQP3356' [SEP] 'HAUKLIE Men's Sports Waffle Ribbed Polo Shirts Summer Short Sleeve Cotton Muscle Quarter-Zip Henley T-Shirt Tunics Tops' [SEP] '$10.99' [SEP] 'B09Q67H373' [SEP] 'JSPOYOU 2022 Newly T-Shirt for Mens Funny 3D Graphics Pattern Crewneck Short Sleeve Tees Big and Tall Summer Casual Comfy Top' [SEP] '$1.99 to $8.99' [SEP] 'B09S3BN15C' [SEP] 'Mens Linen Shirt,Men's Striped Shirts Casual Short Sleeve Button Down Shirts Regular Fit Hawaiian Shirts Beach Tees Tops' [SEP] '$3.78 to $11.38' [SEP] 'B09NNMV9LN' [SEP] 'JSPOYOU Mens Short Sleeve Crewneck 3D Graphic Tunic Shirts Big & Tall Tie Dye Summer Top Basic Designed Classic Cotton Shirt' [SEP] '$3.99 to $10.99' [SEP] 'B078WSND96' [SEP] 'Clearance Deals Retro T-Shirt Bee Printed Blouse, 2021 Summer Fashion Boys Men 3D Round Neck Short Sleeve Tee Tops' [SEP] '$9.99 to $11.99' [SEP] 'B07HRFSNL4' [SEP] 'Nautica Men's Solid Crew Neck Short-Sleeve Pocket T-Shirt' [SEP] '$16.05 to $40.98' [SEP] 'B07N7TDKXQ' [SEP] 'SOCKS'NBULK Mens Cotton Crew Neck Short Sleeve T-Shirts Mix Colors Bulk' [SEP] '$38.79 to $172.8' [SEP] 'B09T756KQ5' [SEP] 'WENKOMG1 Men's Crewneck Goth Tee Shirts Skull Print Tops Spring/Summer Long Sleeve Sports T-Shirt Baggy Y2K Soft Streetwear' [SEP] '$4.09 to $8.09'

Reasoning: I need to find men's long sleeve t-shirts in mossy oak country color, size 5x-large big, under $60. From the search query I used 'men's long sleeve t-shirt mossy oak country 5x-large big', the items returned include the correct size and color options, and price under $60. The product I'm interested in is 'Legendary Whitetails Men's Non-Typical Long Sleeve T-Shirt' at $10.52. I should click on this product to verify the availability of the correct size and color options on the product page.

Action: click[b00o30jldk]

[Step 3]

Observation: 'Back to Search' [SEP] '< Prev' [SEP] 'color' [SEP] 'army' [SEP] 'big game field camo' [SEP] 'black' [SEP] 'cardinal' [SEP] 'charcoal heather' [SEP] 'inferno' [SEP] 'mossy oak bottomland' [SEP] 'mossy oak country' [SEP] 'mossy oak country dna' [SEP] 'mossy oak coyote' [SEP] 'slate' [SEP] 'swamp' [SEP] 'turkey' [SEP] 'flag' [SEP] 'country' [SEP] 'fish' [SEP] 'size' [SEP] 'small' [SEP] 'medium' [SEP] 'large' [SEP] 'large tall' [SEP] 'x-large' [SEP] 'x-large tall' [SEP] 'xx-large' [SEP] 'xx-large big tall' [SEP] '3x-large big' [SEP] '3x-large big tall' [SEP] '4x-large big' [SEP] '4x-large big tall' [SEP] '5x-large big' [SEP] 'Legendary Whitetails Men's Non-Typical Long Sleeve T-Shirt' [SEP] 'Price: $10.52 to $40.5' [SEP] 'Rating: N.A.' [SEP] 'Description' [SEP] 'Features' [SEP] 'Reviews' [SEP] 'Buy Now'

Reasoning: I need to find men's long sleeve t-shirts in mossy oak country color, 5x-large size, under $60. The current product I'm viewing, Legendary Whitetails Men's Non-Typical Long Sleeve T-Shirt, has a color option I can select (mossy oak country) and a size option I can select (5x-large big). I should select the mossy oak country color and 5x-large big size options to complete the purchase.

Action: click[mossy oak country]

[Step 4]

Observation: 'Back to Search' [SEP] '< Prev' [SEP] 'color' [SEP] 'army' [SEP] 'big game field camo' [SEP] 'black' [SEP] 'cardinal' [SEP] 'charcoal heather' [SEP] 'inferno' [SEP] 'mossy oak bottomland' [SEP] 'mossy oak country' [SEP] 'mossy oak country dna' [SEP] 'mossy oak coyote' [SEP] 'slate' [SEP] 'swamp' [SEP] 'turkey' [SEP] 'flag' [SEP] 'country' [SEP] 'fish' [SEP] 'size' [SEP] 'small' [SEP] 'medium' [SEP] 'large' [SEP] 'large tall' [SEP] 'x-large' [SEP] 'x-large tall' [SEP] 'xx-large' [SEP] 'xx-large big tall' [SEP] '3x-large big' [SEP] '3x-large big tall' [SEP] '4x-large big' [SEP] '4x-large big tall' [SEP] '5x-large big' [SEP] 'Legendary Whitetails Men's Non-Typical Long Sleeve T-Shirt' [SEP] 'Price: $10.52 to $40.5' [SEP] 'Rating: N.A.' [SEP] 'Description' [SEP] 'Features' [SEP] 'Reviews' [SEP] 'Buy Now'

Reasoning: I need to find men's long sleeve t-shirts in mossy oak country color, size 5x-large big, under $60. The current product page allows me to select both the color and size options. I should select the specified size and then click "Buy Now" to complete the purchase.

Action: click[5x-large big]

[Step 5]

Observation: 'Back to Search' [SEP] '< Prev' [SEP] 'color' [SEP] 'army' [SEP] 'big game field camo' [SEP] 'black' [SEP] 'cardinal' [SEP] 'charcoal heather' [SEP] 'inferno' [SEP] 'mossy oak bottomland' [SEP] 'mossy oak country' [SEP] 'mossy oak country dna' [SEP] 'mossy oak coyote' [SEP] 'slate' [SEP] 'swamp' [SEP] 'turkey' [SEP] 'flag' [SEP] 'country' [SEP] 'fish' [SEP] 'size' [SEP] 'small' [SEP] 'medium' [SEP] 'large' [SEP] 'large tall' [SEP] 'x-large' [SEP] 'x-large tall' [SEP] 'xx-large' [SEP] 'xx-large big tall' [SEP] '3x-large big' [SEP] '3x-large big tall' [SEP] '4x-large big' [SEP] '4x-large big tall' [SEP] '5x-large big' [SEP] 'Legendary Whitetails Men's Non-Typical Long Sleeve T-Shirt' [SEP] 'Price: $10.52 to $40.5' [SEP] 'Rating: N.A.' [SEP] 'Description' [SEP] 'Features' [SEP] 'Reviews' [SEP] 'Buy Now'

Reasoning: I need to find men's long sleeve t-shirts in mossy oak country color, 5x-large big size, under $60. I've already selected the correct color and size options from the available choices. The product page displays the title "Legendary Whitetails Men's Non-Typical Long Sleeve T-Shirt," which confirms it's a long sleeve t-shirt. The price range is $10.52-$40.50, which is well under the $60 threshold. Therefore, I should now click "Buy Now" to complete the purchase.

Action: click[buy now]

Figure 17: **WebShop Example 2.** SEED preserves the requested product constraints and selects *mossy oak country* and *5x-large big* before purchasing a matching long-sleeve shirt under $60.