EDGE: Experience-Distillation for Guided Exploration in Agentic Reinforcement Learning

Can Xie[1,2], Yuyi Zhou[2,3], Wen Yang[1,2], Ziyi Zhang[2,3],
Siyao Song[1,2], Yingzhuo Deng[1,2], Shuo Ren[2]*, Jiajun Zhang[1,2,4]* ,

[1] School of Artificial Intelligence, University of Chinese Academy of Sciences
[2] Institute of Automation, Chinese Academy of Sciences
[3] School of Advanced Interdisciplinary Sciences, University of Chinese Academy of Sciences
[4] Wuhan AI Research
{xiecan2024,shuo.ren}@ia.ac.cn, jjzhang@nlpr.ia.ac.cn

Abstract

Reinforcement learning with outcome-based objectives such as GRPO enables LLM-based agents to solve complex, long-horizon tasks, yet the reusable exploration patterns embedded in interaction trajectories are largely discarded after a single policy update. Existing experience-augmented approaches retrieve historical guidance at inference time, but they apply experiences without accounting for the policy's evolving capability and create persistent dependencies on external retrieval. We propose **EDGE** (**E**xperience-**D**istillation for **G**uided **E**xploration), a framework that treats retrieved experiences as temporary training-time scaffolds and progressively internalizes their benefits into the parametric policy. Concretely, EDGE partitions each rollout group into experience-conditioned and experience-free trajectories to estimate and admit only positive marginal gains without extra sampling, then distills the induced behavior into the base policy via a reverse-KL objective on its own empirical support. A co-evolutionary experience bank further synthesizes guidance from emerging failure modes and prunes obsolete entries as the policy evolves. Across embodied, web, and search-based QA tasks, EDGE improves over strong RL baselines by up to 12.5 points and remains effective without inference-time scaffolds or a proprietary reflector. The code is available at https://github.com/xvolcano02/EDGE.

1 Introduction

Reinforcement learning (RL) with outcome-based objectives such as GRPO (Shao et al., 2024) has become a standard post-training paradigm for LLM-based agents that reason, plan, and act through multi-turn interactions (Yao et al., 2023; Shinn et al., 2023; Yao et al., 2022; Shridhar et al., 2021; Feng et al., 2025; Wang et al., 2025b; Jin et al., 2025). Yet current agentic RL uses its own experience poorly. A single rollout may contain reusable patterns—effective task decompositions, recovery strategies, dead-end avoidance, but once a trajectory contributes a scalar advantage to one policy update, these patterns are largely discarded. The agent must therefore rediscover them from scratch, a particularly costly failure mode in sparse-reward, long-horizon environments where agentic RL is most needed.

A growing body of work addresses this inefficiency by augmenting agents with external experience at inference time, through episodic reflections (Shinn et al., 2023; Zhao et al., 2024), persistent memory (Chhikara et al., 2025; Fang et al., 2026), or skill libraries (Xia et al., 2026; Liu et al., 2026). While effective, these approaches share two structural limitations that become apparent as training progresses. Experience utility is inherently policy-dependent: guidance that accelerates an undertrained policy can become redundant—or actively harmful—once the corresponding behavior has been internalized, yet most methods apply experience unconditionally or filter it by static heuristics. Moreover, if the agent must retrieve experience at deployment, part of its competence resides in the context window rather than in the model parameters, incurring persistent token overhead and sensitivity to retrieval noise. [1] These observations suggest that experience reuse in agentic RL should be treated as a dynamic lifecycle rather than a static retrieval mechanism. External experience could first guide exploration, as a temporary training-time scaffold, then be exploited by consolidating its useful behavioral effect into the experience-free policy, and finally be retired once its utility vanishes. Under this view, experience reuse becomes a

*Corresponding authors

[1] These are not hypothetical concerns: in our experiments, naive memory augmentation (e.g., EvolveR, GRPO+Mem0) degrades performance well below vanilla GRPO, confirming that unfiltered experience injection is unreliable.

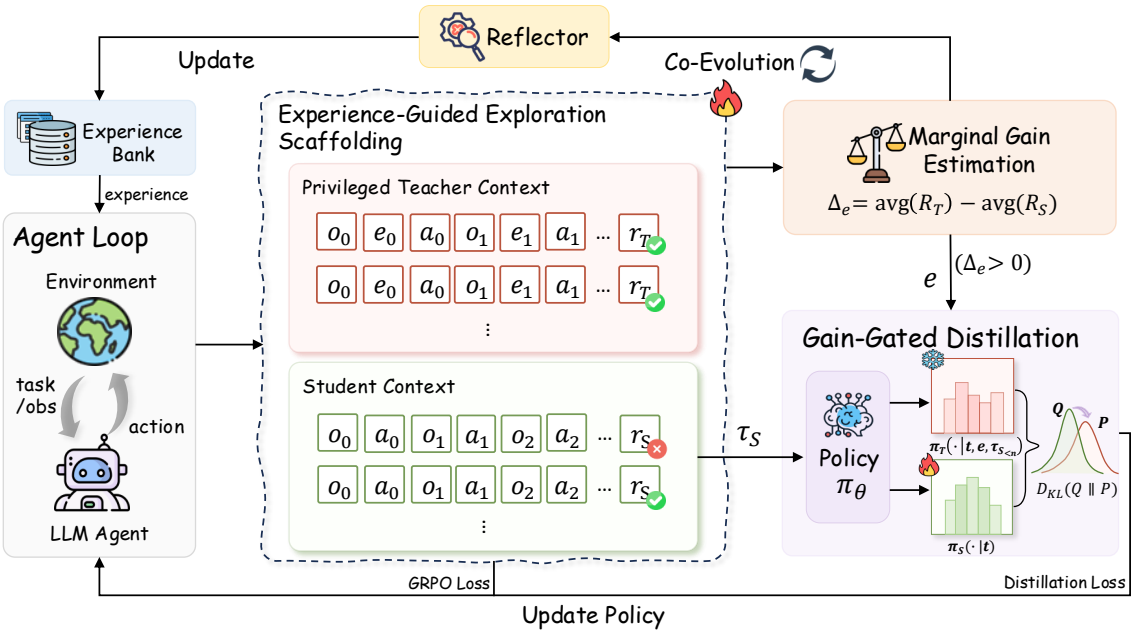

Figure 1: **Overview of the EDGE framework.** (a) Experience-guided exploration scaffolding uses contrastive rollouts to estimate the marginal gain Δ_e of a retrieved experience. (b) Gain-gated distillation internalizes beneficial scaffold behavior into the base policy only when $\Delta_e > 0$. (c) The experience bank co-evolves with the policy via failure-driven expansion and utility-based pruning, yielding a retrieval-free policy at deployment.

scaffold-to-parameter learning problem rather than a retrieve-and-prompt augmentation problem.

In this paper, we instantiate this idea in EDGE (Experience-Distillation for Guided Exploration), a framework that turns retrieved experience from persistent inference-time memory into dynamically validated training-time scaffolding. EDGE uses online marginal-gain estimation to decide when an experience should guide exploration, be distilled into the policy, or be retired as the policy evolves, via three core mechanisms. First, **experience-guided exploration scaffolding** partitions each GRPO rollout group into experience-conditioned and experience-free trajectories, estimating the marginal gain of a retrieved experience without additional environment sampling and retaining only positive-gain instances. Then, **gain-gated privileged distillation** serves as the exploitation step, transferring the behavior induced by the privileged context into the standard policy via reverse-KL divergence computed on the student's own empirical support, activating only when the estimated gain is positive to avoid negative transfer. Finally, **experience bank evolution and management** enables the experience bank to co-evolve with the policy throughout training: new experiences are synthesized from failure-mode analysis, while obsolete ones are pruned based on tracked utility, keeping the scaffold aligned with the agent's evolving capability.

On ALFWorld (Shridhar et al., 2021) and WebShop (Yao et al., 2022), EDGE consistently outperforms skill-free RL and prior experience-augmented methods, with the largest gains on exploration-intensive subtasks (e.g., Heat, Cool, Pick2). When external experiences are withheld at inference time, EDGE retains 96.0% of its scaffolded performance—compared with 82.9% for SkillRL and 92.3% for EMPO^2—confirming that useful exploration priors have been absorbed into the parametric policy. On seven search-based QA benchmarks with Qwen3-4B, EDGE further improves over Search-R1 by 5.9 points on average; using the policy itself as the reflector preserves 97.3% of the performance obtained with GPT-4o.

Our contributions are as follows:

- We identify two failure modes of experience-augmented agentic RL—policy-dependent experience utility and persistent inference-time retrieval dependence—and reframe experience reuse as a dynamic scaffold-to-parameter transition.
- We introduce experience-guided exploration scaffolding, which partitions rollout groups to estimate the marginal value of each retrieved

experience under the current policy without requiring additional environment sampling.

- We propose gain-gated privileged distillation, which internalizes scaffold-induced behavior via reverse-KL on the student's own empirical support, gated by the estimated marginal gain to prevent negative transfer, together with a co-evolutionary experience bank that expands and prunes in response to the policy's evolving needs.
- Experiments across embodied, web, and search-based QA tasks show that EDGE improves task performance and training efficiency, transfers to a newer Qwen3 backbone, and remains effective with a self-reflector and without experience retrieval at deployment.

2 Preliminaries

We formalize the multi-turn decision-making process of an LLM-based agent as a Partially Observable Markov Decision Process (POMDP) (Kaelbling et al., 1998) $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \mathcal{R} \rangle$. The observation space $\mathcal{O}$ consists of natural-language strings emitted by the environment, and the action space $\mathcal{A}$ comprises token sequences generated autoregressively by the policy π_θ. Given a task instruction x, the agent interacts with the environment over a sequence of turns: at step t it receives observation $o_t \in \mathcal{O}$ and produces action $a_t \in \mathcal{A}$ according to

$$a_t \sim \pi_\theta(\cdot \mid x, h_t, o_t), \tag{1}$$

where $h_t = (o_1, a_1, \ldots, o_{t-1}, a_{t-1})$ is the interaction history. An episode terminates upon task completion or at a maximum step limit, yielding a sparse binary reward $R(\tau) \in \{0, 1\}$ and a full trajectory

$$\tau = (x,\ o_1, a_1,\ \ldots,\ o_T, a_T,\ R(\tau)). \tag{2}$$

We build on Group Relative Policy Optimization (GRPO) (Shao et al., 2024), which samples a group of G parallel trajectories $\{\tau_i\}_{i=1}^{G}$ per task and computes group-normalized advantages:

$$\hat{A}_i = \frac{R(\tau_i) - \mathrm{mean}(\{R(\tau_j)\}_{j=1}^{G})}{\mathrm{std}(\{R(\tau_j)\}_{j=1}^{G})}. \tag{3}$$

The policy is updated by maximizing a clipped surrogate objective with KL regularization:

$$\mathcal{L}_{\mathrm{RL}}(\theta) = -\mathbb{E}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\tau_i|}\sum_{t=1}^{|\tau_i|}\Big(\min\big(\rho_{i,t}\,\hat{A}_i,\ \mathrm{clip}(\rho_{i,t}, 1-\epsilon, 1+\epsilon)\,\hat{A}_i\big) - \beta\, D_{\mathrm{KL}}\big(\pi_\theta \| \pi_{\mathrm{ref}}\big)\Big)\Bigg], \tag{4}$$

where $\rho_{i,t} = \pi_\theta(a_t \mid x, h_t, o_t) \,/\, \pi_{\theta_{\mathrm{old}}}(a_t \mid x, h_t, o_t)$ is the importance sampling ratio, π_{ref} is the reference policy, and β controls KL penalty strength. By contrasting outcomes within each group, GRPO steers θ toward successful action sequences without a learned value function. However, in sparse-reward, partially observable environments, unguided exploration often yields groups in which few or no trajectories succeed, rendering the advantage estimate uninformative—a limitation we address in the following section.

3 Method: EDGE

We present **EDGE** (**E**xperience-**D**istillation for **G**uided **E**xploration), a framework that iteratively strengthens the multi-turn reasoning capability of LLM-based agents (Figure 1). Central to our approach is treating retrieved external experience not as a static inference-time prompt—which inflates the context window and creates persistent retrieval dependence—but as a *training-time scaffold* that guides exploration and is discarded at deployment. The agent first explores with privileged access to experience, then distills only the empirically beneficial behavior into its own parameters, so the deployed policy requires no external scaffold. Section 3.1 introduces the experience-guided exploration scaffolding, Section 3.2 details the gain-gated privileged distillation mechanism, and Section 3.3 describes the experience bank evolution and management strategy.

3.1 Experience-Guided Exploration Scaffolding

To provide exploratory guidance in the sparse-reward, partially observable environments typical of agentic tasks, EDGE introduces a controlled information asymmetry within the standard GRPO rollout group. Given a task instruction x and an experience bank $\mathcal{E}$, we retrieve the top-m most relevant experiences by embedding similarity, using the task instruction and initial environmental observations as the query, and select the highest-scoring

entry $e \in \mathcal{E}$. Recall that GRPO samples a group of G parallel trajectories per task to estimate relative advantages (Eq. (3)). We partition this group into two equal subsets without adding extra rollouts:

- **Teacher rollouts** ($\mathcal{T}^{\mathsf{T}}$, $G/2$ trajectories): conditioned on the *privileged context* $c^{\mathsf{T}} = x \oplus e$, where $\oplus$ denotes concatenation under a unified chat template.
- **Student rollouts** ($\mathcal{T}^{\mathsf{S}}$, $G/2$ trajectories): conditioned on the *standard context* $c^{\mathsf{S}} = x$ alone.

Because both subsets share the same policy π_θ and differ only in whether the retrieved experience is visible, any performance gap can be attributed to the informational advantage provided by e. We quantify this gap via the *instantaneous marginal gain*:

$$\Delta_e = \frac{1}{|\mathcal{T}^{\mathsf{T}}|} \sum_{\tau \in \mathcal{T}^{\mathsf{T}}} R(\tau) - \frac{1}{|\mathcal{T}^{\mathsf{S}}|} \sum_{\tau \in \mathcal{T}^{\mathsf{S}}} R(\tau), \quad (5)$$

where $R(\tau)$ is the binary outcome reward defined in §2. A positive Δ_e signals that the experience provides useful guidance beyond the agent's current capability; a non-positive value indicates it is redundant or harmful. This estimate gates the distillation objective (§3.2) and drives experience bank updates (§3.3).

Beyond gating distillation, Δ_e controls which rollouts enter the RL loss itself. When $\Delta_e > 0$, advantages (Eq. (3)) are computed over all G trajectories. The pooled baseline therefore reflects the performance level achievable under the scaffold, giving student rollouts a stronger calibrated reference than a within-subset baseline would provide. When $\Delta_e \leq 0$, teacher-conditioned trajectories are excluded from the RL loss entirely: advantages are computed over $\mathcal{T}^{\mathsf{S}}$ alone, reducing the update to a standard GRPO step. Without this masking, non-beneficial teacher rollouts would still shift the group baseline and distort student advantages—contaminating the policy gradient even though distillation is gated off.

3.2 Gain-Gated Privileged Distillation

The scaffolding in §3.1 exposes the agent to successful reasoning patterns it may not discover on its own, but because the retrieved experience is unavailable at deployment, these gains remain ephemeral unless consolidated into the policy itself. This motivates a complementary mechanism: distilling the scaffold-induced improvements into the base policy parameters so that the agent can reproduce them from the standard context alone.

Our approach realizes this through asymmetric self-distillation. Rather than relying on a separate, unconditionally superior teacher, we use the same policy π_θ under two informational conditions: the teacher is π_θ evaluated with the privileged context c^{T} (gradients stopped), while the student operates under c^{S}. The asymmetry therefore lies in the information available to each role, not in model capacity. The gain gate $\mathbb{I}(\Delta_e > 0)$ further ensures that distillation is triggered only when the scaffold yields a verified performance advantage, allowing the policy to selectively internalize reusable reasoning patterns while avoiding negative transfer.

To guard against covariate shift, we construct the distillation target on the student's own empirical support. For a gain-gated task instance ($\Delta_e > 0$) and a student trajectory $\tau \in \mathcal{T}^{\mathsf{S}}$ with token sequence $(y_1, \ldots, y_m)$ generated under c^{S}, we perform a no-gradient forward pass of π_θ over the same tokens conditioned on c^{T}. Because both passes share identical actions, the comparison isolates the informational advantage of e without introducing out-of-distribution transitions.

We adopt the reverse KL divergence $D_{\mathrm{KL}}(\pi_{\text{student}} \| \pi_{\text{teacher}})$ as the distillation objective. Unlike the forward KL, which compels the student to cover the full teacher support and can induce mode-covering artifacts, the reverse KL is mode-seeking: it encourages the policy to concentrate on the most effective reasoning mode under the scaffold. The token-level scaffold internalization loss is:

$$\mathcal{L}_{\text{distill}}(\theta) = \mathbb{E}_{\tau \sim \mathcal{T}^{\mathsf{S}}} \Bigg[\mathbb{I}(\Delta_e > 0) \sum_{t=1}^{m} \sum_{y \in \mathcal{V}} \pi_\theta^{\mathsf{S}}(y)\, \delta_t(y) \Bigg],$$

$$\delta_t(y) = \log \frac{\pi_\theta(y \mid c^{\mathsf{S}}_{<t})}{\pi_{\mathrm{sg}(\theta)}(y \mid c^{\mathsf{T}}_{<t})}, \quad (6)$$

where $\pi_\theta^{\mathsf{S}}(y) \triangleq \pi_\theta(y \mid c^{\mathsf{S}}_{<t})$ for brevity, $\delta_t(y)$ is the per-token log-ratio between the student and the privileged teacher, and $\mathrm{sg}(\cdot)$ denotes stop-gradient. This loss is combined with the GRPO objective (Eq. (4)) to form the joint actor loss:

$$\mathcal{L}_{\text{actor}}(\theta) = \mathcal{L}_{\text{RL}}(\theta) + \lambda\, \mathcal{L}_{\text{distill}}(\theta), \quad (7)$$

where λ controls the relative weight of scaffold internalization versus the RL signal. Through this

joint optimization, the deployed agent reproduces privileged reasoning without any external scaffold at test time.

3.3 Experience Bank Evolution and Management

A static experience library cannot adequately support the agent throughout training: as π_θ improves, it encounters situations where existing experiences provide insufficient guidance; conversely, previously beneficial experiences may become redundant once the policy has internalized the corresponding behavior, or harmful if they conflict with newly discovered strategies. EDGE therefore treats $\mathcal{E}$ as a living repository that co-evolves with the policy through both expansion and pruning.

Following Xia et al. (2026), we generate new experiences from the agent's own rollout trajectories. After each training step, we identify task categories whose success rate falls below a threshold ξ and collect representative failed and successful trajectories from these categories. A reflector LLM then analyzes the contrast between them to synthesize new experiential guidance:

$$e_{\text{new}} = f_{\text{reflect}}\left(\tau^+, \tau^-\right), \tag{8}$$

where τ^+ and τ^- denote a successful and a failed trajectory from the same task category, respectively. The generated experiences are deduplicated against existing entries and inserted into $\mathcal{E}$ for retrieval in subsequent training iterations.

Meanwhile, the utility of existing experiences is continuously tracked via an Exponential Moving Average (EMA) score for each experience e:

$$U_e^{(t)} = (1-\mu)\, U_e^{(t-1)} + \mu\, \Delta_e, \tag{9}$$

where $\mu \in (0,1)$ is the momentum coefficient and Δ_e is the instantaneous marginal gain from Eq. (5). The EMA smooths over stochastic fluctuations while remaining responsive to genuine shifts in experience utility. Experiences whose tracked utility $U_e^{(t)}$ falls below a threshold η are removed from $\mathcal{E}$, retiring scaffolds that have been absorbed into the policy and reducing overhead during retrieval and rollout. This yields a co-evolutionary dynamic: the policy improves by internalizing useful experiences, exposing new failure modes that drive bank expansion, while obsolete experiences are simultaneously pruned away.

4 Experiments

We evaluate EDGE to examine whether experience scaffolding can improve agentic RL while being progressively internalized by the policy. Our experiments address five questions: (1) Does EDGE improve over prompting, vanilla RL, and prior experience-augmented methods, particularly on exploration-intensive tasks? (2) Does the trained policy retain its performance when external experiences are removed at inference time? (3) How much does each component—gain gating, distillation, and pruning—contribute, and how do they interact? (4) Is experience utility truly non-stationary, and does the co-evolutionary bank adapt accordingly during training? (5) Does EDGE generalize to a newer backbone and search-based QA, and remain effective without a stronger proprietary reflector?

4.1 Experiment Setup

Environments. We evaluate EDGE across two complementary agentic settings: interactive decision making and search-based QA. For the former, we use two environments with sparse outcome-level feedback. *ALFWorld* (Shridhar et al., 2021) comprises 3,827 text-based household tasks across six types (Pick, Look, Clean, Heat, Cool, and Pick2), while *WebShop* (Yao et al., 2022) requires agents to search and purchase products matching user instructions from a catalog of over 1.1M items. To test generalization beyond interactive environments, we further evaluate search-based QA on NQ, TriviaQA, PopQA, HotpotQA, 2WikiMultiHopQA, MuSiQue, and Bamboogle, covering single- and multi-hop evidence retrieval and reasoning.

Baselines. We compare against three families of methods: closed-source LLM agents (GPT-4o (OpenAI et al., 2024) and Gemini-2.5-Pro (Comanici et al., 2025)), prompting agents (ReAct (Yao et al., 2023) and Reflexion (Shinn et al., 2023)), and training-based agents. The post-training baselines include GRPO (Shao et al., 2024), OPSD (Zhao et al., 2026), EvolveR (Wu et al., 2026), GRPO augmented with Mem0 (Chhikara et al., 2025), SkillRL (Xia et al., 2026), and EMPO^2 (Liu et al., 2026). Detailed descriptions are provided in Appendix B.1.

Training details. We use Qwen2.5-1.5B/7B-Instruct (Qwen et al., 2025) as the base models for

Type	Method	ALFWorld							WebShop	
		Pick	Look	Clean	Heat	Cool	Pick2	All	Score	Succ.
Closed-Source Model										
Prompting	GPT-4o	75.3	60.8	31.2	56.7	21.6	49.8	48.0	31.8	23.7
Prompting	Gemini-2.5-Pro	92.8	63.3	62.1	69.0	26.6	58.7	60.3	42.5	35.9
Qwen2.5-1.5B-Instruct										
Prompting	Base Model	5.9	5.5	3.3	9.7	4.2	0.0	4.1	23.1	5.2
Prompting	ReAct	17.4	20.5	15.7	6.2	7.7	2.0	12.8	40.1	11.3
Prompting	Reflexion	35.3	22.2	21.7	13.6	19.4	3.7	21.8	55.8	21.9
Post-Training	GRPO	85.3	53.7	**84.5**	78.2	59.7	53.5	72.8	75.8	56.8
Post-Training	SkillRL	**91.2**$_{\pm 4.3}$	64.3$_{\pm 4.6}$	78.1$_{\pm 5.4}$	76.9$_{\pm 6.3}$	70.8$_{\pm 6.5}$	54.6$_{\pm 6.1}$	74.2$_{\pm 4.7}$	77.3$_{\pm 3.5}$	60.9$_{\pm 3.7}$
Post-Training	EMPO^2	86.9$_{\pm 3.3}$	66.2$_{\pm 5.2}$	79.3$_{\pm 4.9}$	79.1$_{\pm 4.5}$	75.3$_{\pm 5.7}$	64.8$_{\pm 6.6}$	76.8$_{\pm 3.7}$	78.2$_{\pm 3.7}$	63.2$_{\pm 3.3}$
Post-Training	**EDGE**	80.6$_{\pm 2.2}$	**73.7**$_{\pm 5.5}$	76.0$_{\pm 4.3}$	**87.3**$_{\pm 4.0}$	**85.6**$_{\pm 6.1}$	**68.4**$_{\pm 5.6}$	**79.7**$_{\pm 2.3}$	**78.8**$_{\pm 2.1}$	**65.6**$_{\pm 3.2}$
Qwen2.5-7B-Instruct										
Prompting	Base Model	33.4	21.6	19.3	6.9	2.8	3.2	14.8	26.4	7.8
Prompting	ReAct	48.5	35.4	34.3	13.2	18.2	17.6	31.2	46.2	19.5
Prompting	Reflexion	62.0	41.6	44.9	30.9	36.3	23.8	42.7	58.1	28.8
Post-Training	EvolveR†	64.9	33.3	46.4	13.3	33.3	33.3	43.8	42.5	17.6
Post-Training	GRPO+Mem0†	78.1	54.8	56.1	31.0	65.0	26.9	54.7	58.1	37.5
Post-Training	OPSD†	50.0	60.0	22.7	21.4	17.6	9.5	32.8	4.5	2.3
Post-Training	GRPO	92.8	85.7	89.3	75.7	74.5	67.7	82.1	80.3	70.1
Post-Training	GRPO+OPSD†	91.4	61.5	**100**	87.5	76.5	52.2	80.4	86.8	76.5
Post-Training	SkillRL	94.1$_{\pm 2.4}$	83.3$_{\pm 4.3}$	88.4$_{\pm 3.6}$	85.6$_{\pm 5.1}$	90.2$_{\pm 5.6}$	78.6$_{\pm 4.9}$	88.2$_{\pm 3.7}$	86.2$_{\pm 3.1}$	76.7$_{\pm 3.3}$
Post-Training	EMPO^2	93.3$_{\pm 3.7}$	**88.9**$_{\pm 3.9}$	91.2$_{\pm 4.4}$	88.5$_{\pm 5.3}$	89.4$_{\pm 4.6}$	79.8$_{\pm 5.1}$	89.1$_{\pm 2.8}$	88.3$_{\pm 2.6}$	77.1$_{\pm 4.0}$
Post-Training	**EDGE**	**96.0**$_{\pm 1.9}$	85.1$_{\pm 4.8}$	93.4$_{\pm 3.2}$	**90.0**$_{\pm 2.6}$	**92.9**$_{\pm 4.5}$	**81.6**$_{\pm 6.0}$	**90.4**$_{\pm 2.2}$	**89.6**$_{\pm 2.8}$	**82.6**$_{\pm 3.8}$

Table 1: Performance on ALFWorld and WebShop. We report the average success rate (%) per subtask and overall for ALFWorld, and both the average score and success rate (%) for WebShop. Results of EDGE and other experience-augmented training methods are averaged over 3 random seeds (mean±std). † denotes results replicated from (Xia et al., 2026) and (Lu et al., 2026a).

ALFWorld and WebShop, and Qwen3-4B-Instruct-2507 (Yang et al., 2025) for search-based QA. For ALFWorld and WebShop, all post-training methods use exactly the same hyperparameter configurations. The rollout group size G for group-based RL methods is set to 8. For experience retrieval, we encode experiences with Qwen3-Embedding-0.6B (Zhang et al., 2025b) and rank candidates by cosine similarity against the task instruction and initial observations. We use GPT-4o (OpenAI et al., 2024) as the default reflector. To test whether EDGE depends on a stronger proprietary model, the QA experiments also replace GPT-4o with the Qwen3 policy itself while leaving the remaining framework unchanged. Full training settings and hyperparameter details are provided in Appendix B.2, with a wall-clock breakdown in Appendix B.3.

4.2 Main Results

Table 1 presents results on ALFWorld and WebShop across two model scales. At 7B, EDGE achieves 90.4% success rate on ALFWorld and 82.6% on WebShop, improving over GRPO by 8.3 and 12.5 points and over the strongest experience-augmented baseline EMPO^2 by 1.3 and 5.5 points, with consistent gains at 1.5B.

The subtask breakdown reveals that EDGE's advantage concentrates on exploration-heavy tasks—Heat, Cool, and Pick2—which demand longer action sequences with fewer intermediate rewards. On WebShop, the success-rate improvement over GRPO (+12.5) exceeds the score improvement (+9.3), indicating that EDGE helps agents complete full decision chains rather than merely accumulate partial credit.

These results highlight that experience is most effective as selective exploration guidance rather than unconditional augmentation. Naive memory approaches (EvolveR, GRPO+Mem0) degrade well below vanilla GRPO, and standalone self-distillation (OPSD) provides insufficient signal without effective exploration. Combining the two (GRPO+OPSD) recovers competitive aggregate performance but remains brittle across subtasks, whereas EDGE's marginal-gain gating ensures only beneficial experiences contribute, yielding both higher overall success and more uniform subtask coverage.

Method	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	Avg.
Qwen3-4B-Instruct-2507	17.8	42.3	27.2	23.6	21.5	5.6	7.4	20.8
Search-R1	40.6	61.2	43.7	35.2	36.3	12.4	41.6	38.7
SkillRL (GPT-4o)	42.4	59.6	44.3	42.1	43.4	15.9	43.2	41.5
EDGE (GPT-4o)	**48.0**	63.4	**46.9**	42.4	**48.1**	**17.8**	**45.6**	**44.6**
EDGE (self)	44.8	**64.3**	45.4	**43.6**	45.5	16.2	44.2	43.4

Table 2: Performance (%) on seven search-based QA benchmarks with Qwen3-4B-Instruct-2507. "GPT-4o" and "self" denote the reflector used to synthesize experiences. Best results are bolded.

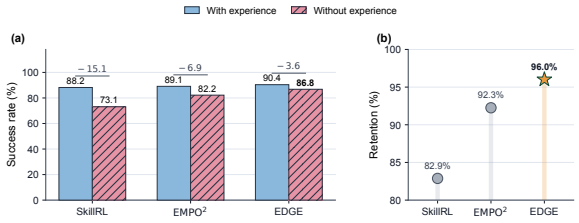

Figure 2: **Performance retention after scaffold removal at inference time.** EDGE preserves 96.0% of its scaffolded performance without external experiences, compared with 82.9% for SkillRL and 92.3% for EMPO^2, indicating more effective internalization into the parametric policy at the 7B scale.

Generalization to Search-Based QA and Self-Reflection. Table 2 tests whether EDGE transfers to a newer backbone and search-based QA without relying on external reflection. With GPT-4o, EDGE leads on five of seven benchmarks. More importantly, self-reflection retains 97.3% of this performance while surpassing both baselines on every benchmark. Its particularly strong results on TriviaQA and HotpotQA further suggest that self-generated experience remains effective across both fact-oriented and multi-hop search settings. Together, these findings indicate that EDGE's gains arise primarily from its experience construction mechanism rather than privileged access to a stronger reflector; GPT-4o improves the quality of the resulting experience, but is not essential for successful transfer.

Inference without External Scaffolds. Figure 2 poses a stricter test: how much performance survives when all external experiences are withheld at inference time? EDGE preserves 96.0% of its scaffolded performance, compared with 92.3% for EMPO^2 and 82.9% for SkillRL, confirming that the reverse-KL distillation stage successfully transfers scaffold-induced behavior into the parametric policy. This near-complete retention supports the intended scaffold-to-parameter transition:

Method	Pick	Look	Clean	Heat	Cool	Pick2	All
GRPO	92.8	**85.7**	89.3	75.7	74.5	67.7	82.1
EDGE	**96.0**	85.1	**93.4**	**90.0**	**92.9**	**81.6**	**90.4**
w/o ExpPruning	90.5	82.2	90.1	84.3	87.6	76.8	86.7
w/o Distillation	91.2	81.5	91.8	78.7	77.2	71.1	83.6
w/o Gain-Gating	83.1	72.5	79.7	65.3	61.8	61.2	72.3

Table 3: **Ablation results** on ALFWorld with Qwen2.5-7B-Instruct. We report success rate (%) per subtask and overall.

experience-guided behaviors are largely internalized by the policy and remain available without inference-time retrieval.

4.3 Analysis

Ablation Studies. Table 3 isolates each component on ALFWorld with Qwen2.5-7B-Instruct. The most striking finding is that removing the gain gate drops overall success to 72.3%—9.8 points below vanilla GRPO—revealing that unfiltered experience injection does not merely fail to help but actively harms the policy, particularly on exploration-heavy subtasks where misleading guidance compounds over long horizons. This result also addresses a natural concern about the $G/2{+}G/2$ rollout partition: since all other ablated variants retain the same split yet outperform GRPO, the partition itself does not dilute the RL signal; the degradation is attributable entirely to distilling low-quality experiences.

The remaining two components contribute complementary benefits. Without distillation, performance falls to 83.6%, only 1.5 points above GRPO, with losses concentrated on the same exploration-heavy subtasks—confirming that scaffolded exploration provides transient guidance but does not durably reshape the policy without explicit behavioral transfer. Without experience pruning, performance declines more modestly to 86.7% with losses spread evenly across subtasks, indicating that pruning acts as a maintenance mechanism that keeps the experience bank aligned with the pol-

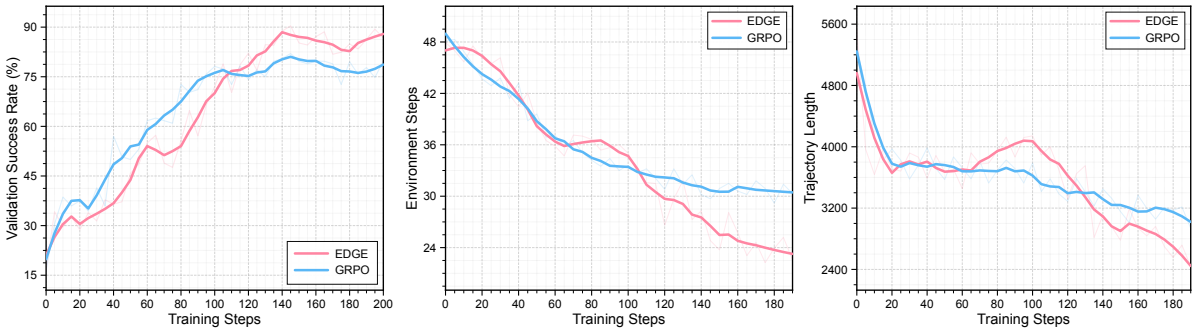

Figure 3: **Training dynamics of EDGE vs. GRPO on ALFWorld with Qwen2.5-7B-Instruct.** EDGE achieves higher validation success while reducing both environment steps and trajectory length more rapidly, indicating that scaffolded exploration is progressively internalized into more efficient experience-free behavior.

icy's evolving capability rather than targeting any specific failure mode.

Training Dynamics. Figure 3 reveals a clear divergence between EDGE and GRPO after approximately step 100: GRPO plateaus and begins to regress, whereas EDGE continues to improve steadily toward 90% validation success. We attribute GRPO's decline to an exploration–exploitation collapse: once the policy commits to locally successful strategies, it loses the diversity needed to solve the remaining hard tasks, and further optimization erodes earlier gains. EDGE's experience scaffolds counteract this by continually injecting structured exploration guidance, while gain gating ensures that this guidance remains beneficial as the policy strengthens. The efficiency panels corroborate this interpretation—EDGE reduces both environment steps and trajectory length more rapidly than GRPO, indicating that the policy internalizes increasingly direct action sequences rather than relying on extended trial-and-error, consistent with the scaffold-removal results in Figure 2.

Experience Gain Tracking. Figure 4 tracks the mean marginal gain Δ_e (Eq. (5)) across training to test a key premise of §3.3: that experience utility is non-stationary. With a static bank, the initially positive gain decays and turns negative after roughly step 100—confirming that once-useful experiences become actively harmful as the policy outgrows them, and explaining why removing gain gating in Table 3 degrades performance below vanilla GRPO. Experience evolution delays this decay, but unchecked bank growth (reaching over 650 entries) introduces retrieval noise that keeps the gain signal volatile. The full co-evolutionary configuration sustains the most stable positive gain: pruning not

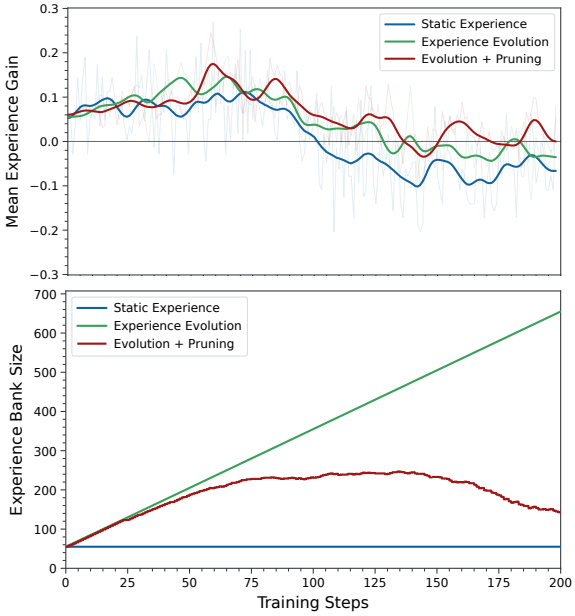

Figure 4: Experience gain dynamics during training.

only curbs bank growth but causes the bank to shrink after step 100, indicating that the policy absorbs existing experiences faster than new failure modes generate replacements—a direct signature of successful internalization. Further details on bank evolution dynamics appear in Appendix C.2.

5 Related Work

Reinforcement Learning for LLM Agents. RL has become a standard post-training paradigm for LLM agents in multi-turn environments (Yao et al., 2022; Shridhar et al., 2021; Feng et al., 2025; Wang et al., 2025b; Jin et al., 2025; Li et al., 2026), progressing from PPO-based methods (Schulman et al., 2017) to critic-free objectives such as GRPO (Shao et al., 2024) and RLOO (Ahmadian et al., 2024), with further improvements

in multi-turn credit assignment through turn-level or stepwise signals (Feng et al., 2025; Wei et al., 2025; Wang et al., 2025a; Xie et al., 2026; Zhang et al., 2026b). However, the reusable exploration patterns within trajectories are still consumed once and discarded. EDGE retains the group-sampling backbone of GRPO but repurposes it to estimate which retrieved experiences currently improve exploration and to transfer their effects into the policy.

Experience-Augmented LLM Agents. External memory and experience reuse have been explored through prompting-based reflections (Shinn et al., 2023; Zhao et al., 2024; Yang et al., 2024; Fang et al., 2026; Ouyang et al., 2026) and persistent retrieval during interaction (Chhikara et al., 2025; Xia et al., 2026; Wu et al., 2026; Ma et al., 2026; Wang et al., 2026; Zhang et al., 2026a). Recent methods selectively replay past reasoning traces (Yan et al., 2025; Zhang et al., 2025a; Qin et al., 2025; Zhan et al., 2026), but primarily target single-turn tasks without the multi-turn, partially observable interaction loops of agentic settings. EMPO^2 (Liu et al., 2026) pursues memory-free behavior through heuristic experience selection and off-policy distillation. Concurrent SKILL0 (Lu et al., 2026b) progressively withdraws a preconstructed skill inventory under a decaying curriculum. In contrast, EDGE uses paired-rollout gains to validate and distill online experience while co-evolving its bank through utility-based pruning.

Privileged Information and Self-Distillation. Leveraging training-time information unavailable at deployment spans LUPI (Vapnik and Vashist, 2009), asymmetric actor-critic methods (Pinto et al., 2017), and context distillation in LLMs (Snell et al., 2022; Choudhury and Sodhi, 2025; Liu et al., 2026; Cheng et al., 2026), though these approaches are largely off-policy and suffer from distribution mismatch with the student's own visitation. On-policy distillation (Agarwal et al., 2024; Ye et al., 2026) addresses this by supervising the student on its own sequences, and on-policy self-distillation (OPSD) (Zhao et al., 2026; Hübotter et al., 2026; Lu et al., 2026a) further removes the need for a separate teacher. EDGE shares this structure but adds gain-based distillation gating to activate distillation only under verified positive gain and co-evolves the experience bank through utility-driven expansion and pruning.

6 Conclusion

We presented EDGE, a framework for using retrieved experience as a temporary training-time scaffold rather than a persistent inference-time dependency. EDGE estimates the marginal utility of retrieved experiences under the current policy, admits only positive-gain scaffolds, and distills their behavioral effect into the standard experience-free policy. Across ALFWorld, WebShop, and seven search-based QA benchmarks, EDGE improves over RL and experience-augmented baselines, with especially large gains on exploration-intensive subtasks and strong retention after scaffold removal. The Qwen3 results further show that these gains transfer to a newer backbone and largely persist when the policy itself replaces GPT-4o as the reflector. These results suggest a practical principle for agentic RL: external experience is most useful when it is dynamically validated, selectively applied, and ultimately internalized.

Limitations

While EDGE requires no extra environment rollouts, it introduces training-time overhead for maintaining the experience bank, computing teacher–student comparisons, and invoking the reflector LLM. The effectiveness of gain-gating also depends on reward quality; as with all outcome-based RL methods, noisy or misspecified rewards would reduce the reliability of the marginal-gain estimates. In terms of empirical scope, our evaluation covers two interactive environments and seven search-based QA benchmarks with Qwen2.5 models at the 1.5B and 7B scales and Qwen3 at the 4B scale; broader validation across model families, larger scales, and multimodal or real-world environments remains future work. More broadly, EDGE assumes experiences are discrete textual artifacts; extending the scaffold-to-parameter principle to latent memory representations is an open problem.

Acknowledgements

This work is supported by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDA04080400 and Beijing Natural Science Foundation L259016.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist,

and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 12248–12267. Association for Computational Linguistics.

Zihao Cheng, Zeming Liu, Yingyu Shan, Xinyi Wang, Xiangrong Zhu, Yunpu Ma, Hongru Wang, Yuhang Guo, Wei Lin, and Yunhong Wang. 2026. Mem^{2}evolve: Towards self-evolving agents via co-evolutionary capability expansion and experience distillation. *Preprint*, arXiv:2604.10923.

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

Sanjiban Choudhury and Paloma Sodhi. 2025. Better than your teacher: LLM agents that learn from privileged AI feedback. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Pim de Haan, Dinesh Jayaraman, and Sergey Levine. 2019. Causal confusion in imitation learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Runnan Fang, Yuan Liang, Xiaobin Wang, Jialong Wu, Shuofei Qiao, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. 2026. Memp: Exploring agent procedural memory. *Preprint*, arXiv:2508.06433.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. 2025. Group-in-group policy optimization for llm agent training. In *Advances in Neural Information Processing Systems*, volume 38, pages 46375–46408. Curran Associates, Inc.

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

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

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134.

Songze Li, Xiaoke Guo, Tianqi Liu, Biao Yi, Zhaoyan Gong, Zhiqiang Liu, Huajun Chen, and Wen Zhang. 2026. What's missing in screen-to-action? towards a UI-in-the-loop paradigm for multimodal GUI reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 17674–17690, San Diego, California, United States. Association for Computational Linguistics.

Zeyuan Liu, Jeonghye Kim, Xufang Luo, Dongsheng Li, and Yuqing Yang. 2026. Exploratory memory-augmented llm agent via hybrid on- and off-policy optimization. *Preprint*, arXiv:2602.23008.

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

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

Weiyu Ma, Yongcheng Zeng, Yan Song, Xinyu Cui, Jian Zhao, Xuhui Liu, and Mohamed Elhoseiny. 2026. Freshness-aware prioritized experience replay for llm/vlm reinforcement learning. *Preprint*, arXiv:2604.16918.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T. Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. 2026. Reasoningbank: Scaling agent self-evolving with reasoning memory. *Preprint*, arXiv:2509.25140.

Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. 2017. Asymmetric actor critic for image-based robot learning. *Preprint*, arXiv:1710.06542.

Yulei Qin, Xiaoyu Tan, Zhengbao He, Gang Li, Haojia Lin, Zongyi Li, Zihan Xu, Yuchen Shi, Siqi Cai, Renting Rui, Shaofei Cai, Yuzheng Cai, Xuan Zhang, Sheng Ye, Ke Li, and Xing Sun. 2025. Learn the ropes, then trust the wins: Self-imitation with progressive exploration for agentic reinforcement learning. *Preprint*, arXiv:2509.22601.

Qwen, :, 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, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. 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*, volume 15 of *Proceedings of Machine Learning Research*, pages 627–635, Fort Lauderdale, FL, USA. PMLR.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

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

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew J. Hausknecht. 2021. Alfworld: Aligning text and embodied environments for interactive learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Charlie Snell, Dan Klein, and Ruiqi Zhong. 2022. Learning by distilling context. *Preprint*, arXiv:2209.15189.

Vladimir Vapnik and Akshay Vashist. 2009. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5):544–557. Advances in Neural Networks Research: IJCNN2009.

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

Jiongxiao Wang, Qiaojing Yan, Yawei Wang, Yijun Tian, Soumya Smruti Mishra, Zhichao Xu, Megha Gandhi, Panpan Xu, and Lin Lee Cheong. 2026. Reinforcement learning for self-improving agent with skill library. *Preprint*, arXiv:2512.17102.

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. 2025b. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *Preprint*, arXiv:2504.20073.

Quan Wei, Siliang Zeng, Chenliang Li, William Brown, Oana Frunza, Wei Deng, Anderson Schneider, Yuriy Nevmyvaka, Yang Katie Zhao, Alfredo Garcia, and Mingyi Hong. 2025. Reinforcing multi-turn reasoning in llm agents via turn-level reward design. *Preprint*, arXiv:2505.11821.

Rong Wu, Xiaoman Wang, Jianbiao Mei, Pinlong Cai, Daocheng Fu, Cheng Yang, Licheng Wen, Xuemeng Yang, Yufan Shen, Yuxin Wang, and Botian Shi. 2026. Evolver: Self-evolving llm agents through an experience-driven lifecycle. *Preprint*, arXiv:2510.16079.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, Zeyu Zheng, Cihang Xie, and Huaxiu Yao. 2026. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. *Preprint*, arXiv:2602.08234.

Can Xie, Ruotong Pan, Xiangyu Wu, Zhang Yunfei, Jiayi Fu, Tingting Gao, and Guorui Zhou. 2026. Unlocking exploration in RLVR: Uncertainty-aware advantage shaping for deeper reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 19057–19076, San Diego, California, United States. Association for Computational Linguistics.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. 2025. Learning to reason under off-policy guidance. *Preprint*, arXiv:2504.14945.

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, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Ling Yang, Zhaochen Yu, Tianjun Zhang, Shiyi Cao, Minkai Xu, Wentao Zhang, Joseph E. Gonzalez, and Bin Cui. 2024. Buffer of thoughts: Thought-augmented reasoning with large language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. 2026. On-policy context distillation for language models. *Preprint*, arXiv:2602.12275.

Runzhe Zhan, Yafu Li, Zhi Wang, Xiaoye Qu, Dongrui Liu, Jing Shao, Derek F. Wong, and Yu Cheng. 2026. Exgrpo: Learning to reason from experience. *Preprint*, arXiv:2510.02245.

Haozhen Zhang, Quanyu Long, Jianzhu Bao, Tao Feng, Weizhi Zhang, Haodong Yue, and Wenya Wang. 2026a. Memskill: Learning and evolving memory skills for self-evolving agents. *Preprint*, arXiv:2602.02474.

Kaiyi Zhang, Ang Lv, Jinpeng Li, Yongbo Wang, Feng Wang, Haoyuan Hu, and Rui Yan. 2025a. Stephint: Multi-level stepwise hints enhance reinforcement learning to reason. *Preprint*, arXiv:2507.02841.

Siwei Zhang, Yun Xiong, Xi Chen, Zi'an Jia, Renhong Huang, Jiarong Xu, and Jiawei Zhang. 2026b. Rapo: Expanding exploration for llm agents via retrieval-augmented policy optimization. *Preprint*, arXiv:2603.03078.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025b. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *Preprint*, arXiv:2506.05176.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. Expel: LLM agents are experiential learners. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 19632–19642. AAAI Press.

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

A Theoretical Analysis of EDGE

This section provides an analytical justification for the four core components of EDGE, following the pipeline through which a retrieved experience e influences the policy update: its marginal utility is estimated (§A.1), smoothed across training steps (§A.2), used to gate and calibrate the RL advantage (§A.3), and channeled through on-support distillation (§A.4).

Throughout, let c^{S} denote the standard context, $c^{\mathsf{T}} = c^{\mathsf{S}} \oplus e$ the privileged context, π_θ the current policy, and $R(\tau) \in \{0, 1\}$ the binary outcome reward. Teacher and student rollout subsets $\mathcal{T}^{\mathsf{T}}, \mathcal{T}^{\mathsf{S}}$ each have size $K = G/2$. All expectations and probabilities condition on fixed π_θ and e.

A.1 Marginal-Gain Estimation and Gating

Define the *value of privileged information* as the expected return gap between the privileged and standard contexts:

$$\mathcal{V}_e(\theta) \;=\; \mathbb{E}_{\tau\sim\pi_\theta(\cdot|c^{\mathsf{T}})}\big[R(\tau)\big] - \mathbb{E}_{\tau\sim\pi_\theta(\cdot|c^{\mathsf{S}})}\big[R(\tau)\big]. \tag{10}$$

The empirical marginal gain is

$$\Delta_e \;=\; \frac{1}{K}\sum_{\tau\in\mathcal{T}^{\mathsf{T}}} R(\tau) \;-\; \frac{1}{K}\sum_{\tau\in\mathcal{T}^{\mathsf{S}}} R(\tau). \tag{11}$$

Under the assumption that the two subsets are conditionally independent and identically distributed up to the presence of e, Δ_e is unbiased for $\mathcal{V}_e(\theta)$ with variance $(p_T(1-p_T)+p_S(1-p_S))/K$, where p_T and p_S are the respective success probabilities. Decomposing Δ_e as a sum of $2K$ independent terms each bounded in an interval of length $1/K$, Hoeffding's inequality yields

$$\Pr\big(|\Delta_e - \mathcal{V}_e(\theta)| \geq \epsilon\big) \;\leq\; 2\exp\big(-K\epsilon^2\big). \tag{12}$$

EDGE converts this estimate into a binary gate:

$$M_e \;=\; \mathbb{I}(\Delta_e > 0). \tag{13}$$

The gate serves as a probabilistic risk-control mechanism. By the one-sided form of (12), if an experience is truly harmful with margin $\gamma > 0$ ($\mathcal{V}_e(\theta) \leq -\gamma$), then $\Pr(M_e{=}1) \leq \exp(-K\gamma^2)$; the symmetric bound holds for missed activations when $\mathcal{V}_e(\theta) \geq \gamma$. When $M_e = 0$, teacher rollouts are masked from both the RL and distillation losses, and the update falls back to standard GRPO on the student subset.

A.2 EMA Smoothing

Because Δ_e is high-variance for small K and the true utility $\mathcal{V}_e(\theta)$ drifts as the policy evolves, EDGE tracks each experience's utility with an exponential moving average:

$$U_e^{(t)} \;=\; (1-\mu)\,U_e^{(t-1)} + \mu\,\Delta_e^{(t)}, \qquad \mu \in (0, 1). \tag{14}$$

Under a local-stationarity approximation (successive $\Delta_e^{(t)}$ treated as i.i.d. with variance σ_e^2), the steady-state variance is $\mathrm{Var}(U_e) = \frac{\mu}{2-\mu}\,\sigma_e^2$, which is strictly less than σ_e^2 for any $\mu < 1$. Smaller μ yields greater smoothing at the cost of slower adaptation to genuine utility shifts. The pruning threshold η thus operates on a smoothed estimate of recent marginal utility rather than a single noisy contrast.

A.3 Pooled Advantage Calibration

When the gate activates, teacher rollouts enter the RL update and alter the advantage baseline. Let b_S, b_T be the student and teacher mean returns, so $\Delta_e = b_T - b_S$. The active set is

$$\mathcal{A} \;=\; \begin{cases} \mathcal{T}^{\mathsf{S}} \cup \mathcal{T}^{\mathsf{T}}, & M_e = 1, \\ \mathcal{T}^{\mathsf{S}}, & M_e = 0, \end{cases} \tag{15}$$

with baseline $b_{\mathcal{A}} = \mathrm{mean}_{j\in\mathcal{A}}(R_j)$. When $M_e = 1$, the pooled baseline $b_{\mathrm{pool}} = \frac{1}{2}(b_S + b_T)$ shifts the raw advantage numerators (prior to GRPO's standard-deviation normalization, which rescales uniformly without changing signs) as follows:

$$\widetilde{A}_S^{\mathrm{EDGE}} = \widetilde{A}_S^{\mathrm{within}} - \tfrac{1}{2}\Delta_e, \tag{16}$$

$$\widetilde{A}_T^{\mathrm{EDGE}} = \widetilde{A}_T^{\mathrm{within}} + \tfrac{1}{2}\Delta_e, \tag{17}$$

where the superscript "within" denotes baselines computed from each subset alone. Since $M_e = 1$ implies $\Delta_e > 0$, student advantages are shifted *downward* and teacher advantages *upward*: privileged successes receive stronger reinforcement, while unguided successes are tempered when the scaffold demonstrates superior performance. When $M_e = 0$, the teacher subset is excluded and no shift is applied.

A.4 On-Support Reverse-KL Distillation

Directly imitating teacher-generated trajectories risks covariate shift (Ross et al., 2011), as the teacher may visit prefixes whose rationale depends on information absent from the student context (causal misidentification; de Haan et al., 2019).

EDGE mitigates this by evaluating the teacher only on student-generated prefixes. For a student trajectory $\tau = (y_1, \ldots, y_m) \in \mathcal{T}^{\mathsf{S}}$, let $h_t^{\mathsf{S}} = (c^{\mathsf{S}}, y_{<t})$ and $h_t^{\mathsf{T}} = (c^{\mathsf{T}}, y_{<t})$. The on-support distillation loss is

$$\widehat{\mathcal{L}}_{\text{distill}}(\theta) = M_e \sum_{\tau \in \mathcal{T}^{\mathsf{S}}} \sum_{t=1}^{|\tau|} D_{\text{KL}}\big(\pi_\theta(\cdot \mid h_t^{\mathsf{S}}) \,\big\|\, \pi_{\text{sg}(\theta)}(\cdot \mid h_t^{\mathsf{T}})\big), \tag{18}$$

where $\text{sg}(\cdot)$ denotes stop-gradient. Unlike forward-KL imitation on teacher rollouts, this objective compares distributions only at prefixes the student has actually reached, avoiding training on teacher-only states. The reverse-KL direction is mode-seeking with respect to the teacher, concentrating student mass on teacher-preferred actions rather than spreading to cover the full teacher distribution, and thereby keeping the update conservative on the student support.

The final actor loss combines both pathways:

$$\mathcal{L}_{\text{actor}}(\theta) = \mathcal{L}_{\text{RL}}(\theta; \mathcal{A}) + \lambda \widehat{\mathcal{L}}_{\text{distill}}(\theta). \tag{19}$$

When $M_e = 0$, both terms reduce to unprivileged-only updates ($\mathcal{A} = \mathcal{T}^{\mathsf{S}}$, $\widehat{\mathcal{L}}_{\text{distill}} = 0$). When $M_e = 1$, the teacher subset raises the RL baseline (§A.3) and the reverse KL transfers privileged behavior into the standard-context policy.

B Implementation Details

B.1 Baselines

- **GPT-4o (OpenAI et al., 2024)**: A closed-source multimodal LLM from OpenAI, used as a strong proprietary agent baseline with standard prompting.
- **Gemini-2.5-Pro (Comanici et al., 2025)**: A closed-source reasoning model from Google DeepMind, serving as another proprietary agent baseline with standard prompting.
- **ReAct (Yao et al., 2023)**: A prompting framework that interleaves chain-of-thought reasoning with environment actions, enabling LLMs to plan and act in a synergistic loop.
- **Reflexion (Shinn et al., 2023)**: Extends ReAct by appending verbal self-reflection after task failures, allowing the agent to refine its strategy across successive trials without weight updates.
- **GRPO (Shao et al., 2024)**: A group-relative policy optimization algorithm that estimates advantages from a group of sampled rollouts, eliminating the need for a separate critic network.
- **OPSD (Zhao et al., 2026)**: An on-policy self-distillation method that distills the model's own high-quality rollouts back into itself to improve reasoning without external supervision.
- **GRPO+OPSD**: A hybrid baseline that combines GRPO's group-relative advantage estimation with OPSD's on-policy self-distillation objective.
- **EvolveR (Wu et al., 2026)**: An experience-augmented training method that iteratively evolves a retrieval-augmented memory of past trajectories to guide policy learning.
- **GRPO+Mem0 (Chhikara et al., 2025)**: Augments GRPO with Mem0, a memory module that stores and retrieves past interaction experiences as additional context during both training and inference.
- **Search-R1 (Jin et al., 2025)**: Trains language models with reinforcement learning to interleave reasoning and search, serving as the standard search-agent baseline in our QA evaluation.
- **SkillRL (Xia et al., 2026)**: A skill-based RL framework that extracts reusable skills from successful trajectories and conditions policy optimization on retrieved skill demonstrations.
- **EMPO^2 (Liu et al., 2026)**: An exploratory memory-augmented policy optimization method that leverages curated past experiences to enhance exploration during RL training.

These baselines span closed-source LLM agents, prompting-based reasoning frameworks, standard RL algorithms, self-distillation methods, and experience-augmented training approaches, enabling a comprehensive evaluation of EDGE from multiple perspectives.

B.2 RL-Training Configuration

We implement EDGE on top of the verl-agent framework (Feng et al., 2025) and train the model with the joint optimization objective in Eq. 19. To reduce the computational cost of reverse-KL distillation, we approximate the vocabulary-level loss using only the top-k student tokens with the highest log probabilities. The experience bank is initialized as empty, and each newly inserted experience is assigned an initial utility score of 0. For retrieval, we use the task instruction and initial environment observations as the query, encode experiences with Qwen3-Embedding-0.6B (Zhang et al., 2025b), and rank candidates by cosine similarity. We retrieve a top-m candidate pool and use the highest-scoring experience as the scaffold for the current rollout.

After each training step, we update the experience bank through both expansion and pruning. For expansion, if the success rate of a task category falls below the expansion threshold ξ, the reflector contrasts successful and failed trajectories from the same category and synthesizes up to three new experiences. We use GPT-4o (OpenAI et al., 2024) by default; in the EDGE (self) QA variant, the Qwen3 policy itself serves as the reflector, with the remaining framework unchanged. For pruning, we update the EMA utility score of each experience according to Eq. 14 and remove experiences whose utility falls below the pruning threshold η. For both ALFWorld and WebShop, we use the hyperparameters in Table 4. All training and profiling experiments are conducted on $8\times$ A100 80GB GPUs.

B.3 Computational Cost

We profile all methods under the same hardware and training configuration using Qwen2.5-7B-Instruct on ALFWorld. Table 5 reports wall-clock seconds per training step. Rollout time includes experience retrieval and prompt construction, while reflector latency is reported separately.

Summing all components, EDGE requires 417.72 seconds per step, a 20.3% overhead over GRPO, mainly from retrieval, privileged-context forward computation, and reflection. Nevertheless, it is 7.8% faster than SkillRL and 6.0% faster than EMPO^2. All methods use the same rollout budget; EDGE partitions the original group into experience-conditioned and experience-free trajectories without additional environment interactions.

Configuration	**Value**
RL-Training	
Actor learning rate	$1e^{-6}$
Maximum prompt length	4096
Maximum response length	512
Training batch size	16
Rollout group size (G)	8
Training mini-batch size	128
Rollout temperature	1.0
Training steps	200
EDGE configuration	
Distillation weight λ	0.1
Distillation top-k tokens	20
Utility pruning threshold η	-0.1
Retrieval pool size (top-m)	6
Expansion success threshold ξ	0.4
EMA momentum μ	0.5
Maximum new experiences per step	3

Table 4: RL Hyperparameters

Method	Rollout	Old Prob.	Ref. Prob.	Update	Reflector
GRPO	263.41	14.10	14.23	55.38	–
SkillRL	334.13	18.45	18.51	64.74	17.12
EMPO^2	318.69	17.37	17.65	63.17	27.56
EDGE	304.56	17.26	17.67	61.45	16.78

Table 5: Per-step wall-clock time (seconds) on ALFWorld with Qwen2.5-7B-Instruct.

C Further Analysis

C.1 Sensitivity to Distillation Weight λ.

Figure 5 sweeps the distillation coefficient λ on Qwen2.5-1.5B-Instruct, revealing a clear trade-off between the RL and distillation objectives. At $\lambda = 1$, the distillation term dominates the gradient and effectively freezes the policy near its initial performance ($\sim$15%), preventing autonomous exploration beyond the scaffold-prescribed behavior. At $\lambda = 0.01$, the policy recovers RL-driven improvement but internalizes scaffold-induced patterns too slowly, converging roughly 5 points below the best setting. The moderate value $\lambda = 0.1$ balances both pressures, sustaining steady improvement to approximately 80%—enough distillation to accelerate internalization without suppressing the RL objective's exploratory signal. We adopt this value for all remaining experiments.

Case	Failure mode	Reflected experience	Later evidence
1. Clean bowl	Navigates to destination before finding object	Locate object before placing	Retrieved on a related task at step 155; U=0.125
2. Pillow on sofa	Issues `take` from source lacking target	Verify source before taking	Utility rises to 0.578 by step 200 (78 retrievals)
3. Pick2 search	Unstructured search before collecting targets	Search systematically before storing	Retrieved on a related Pick2 task at step 166
4. No-op movement	Repeats navigation after arrival	Avoid repeated no-op moves	Near-threshold utility; recovers to U=0.452 by step 200

Table 6: **Experience-bank co-evolution examples.** Each case is extracted from saved failure trajectories, LLM reflection logs, retrieval logs, and utility traces.

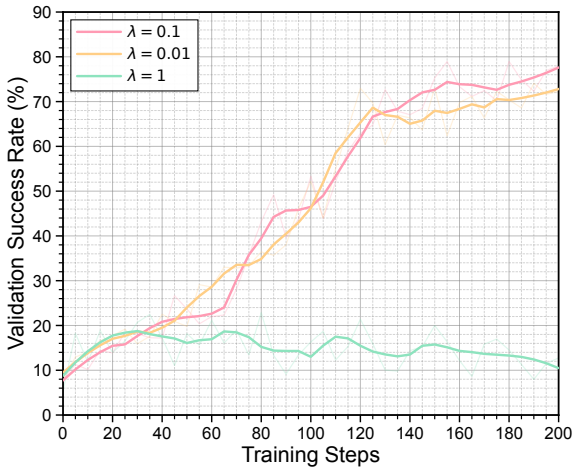

Figure 5: Sensitivity of validation success rate to distillation weight λ.

C.2 Case Study: Experience Bank Evolution Dynamics

This section complements the aggregate gain-tracking curves in Figure 4 with entry-level evidence from ALFWorld training logs (Qwen2.5-7B-Instruct). We trace four representative cases (Table 6), each linking a rollout failure to the experience it produces, its later retrieval, and the resulting change in agent behavior. Cases 1–2 (Figure 6) illustrate the expansion phase, while Cases 3–4 (Figure 7) illustrate late-stage refinement and non-stationary utility.

The four cases reveal a natural curriculum driven by the policy's evolving failure distribution. Early failures are structural and broadly shared: destination-first wandering (Case 1) and hallucinated object presence (Case 2) affect many task variants, producing generic scaffolds with sustained high utility—these entries drive the initial bank growth visible in Figure 4. As the policy internalizes these broad strategies, the remaining failures narrow in scope. By step 165, single-object manipulation is reliable but coordinating two targets remains fragile; Case 3's search-then-collect scaffold addresses precisely this gap, and its moderate final utility (U=0.375) is consistent with Pick2 remaining the hardest subtask in Table 3.

Case 4 provides the most direct evidence for non-stationary utility. Entry `step_914` ("avoid repeating no-op movements") hovers near the pruning boundary for roughly 20 steps (U=−0.026 at step 150, −0.085 at step 165) before recovering to U=0.452 by step 200. The mechanism is interpretable: once destination-first errors (Case 1) are resolved, no-op navigation loops become the dominant Pick2 failure mode, re-activating a previously marginal entry. A positive pruning threshold would have discarded it; the adopted η=−0.1 retains such latent-value entries until the policy's distribution shifts in their favor.

Together, these cases explain the aggregate bank trajectory in Figure 4—initial growth as generic scaffolds accumulate, followed by contraction as the policy absorbs broad patterns and the bank converges to a compact set of specialized, currently relevant entries.

D Pseudocode

Algorithm 1 outlines the full EDGE training loop, which alternates among three stages per iteration: experience-guided exploration scaffolding (§3.1), gain-gated privileged distillation (§3.2), and experience bank evolution (§3.3).

E Prompts

This section provides the prompt templates used in our experiments. We include both rollout prompts and experience update prompts for ALFWorld and WebShop. The rollout prompts are used for action generation, while the update prompts are used to generate new state-aware experiences from con-

Case 1: From destination-first wandering to object-first execution

Failure (step 150). The task is `put a clean bowl in shelf`. The failed rollout immediately navigates to the destination and loops around empty shelves:

`go to shelf 3` → `go to cabinet 1` → `go to shelf 3` → `go to cabinet 3` → `examine shelf 3`.

Successful contrast. The agent first searches likely sources, finds a bowl in the fridge, takes and cleans it, then navigates to a shelf:

`go to fridge 1` → `open fridge 1` → `take bowl 1 from fridge 1` → `clean bowl 1 with sinkbasin 1` → `go to shelf 1`.

Reflected experience. Entry `task_943`: *"First identify where the target bowl is, retrieve it, clean it if the goal requires a clean bowl, and only then move it to a shelf."*

Later retrieval. At step 155, retrieved for `put a clean bowl in diningtable` (similarity 0.886). The rollout no longer starts by visiting the table; it locates, cleans, and places the bowl. Utility: U=0.125.

Case 2: From hallucinated source to source verification

Failure (step 58). The task is `put some pillow on sofa`. The agent goes to the sofa, observes `box`, `creditcard`, `keychain`, and `newspaper`—but no pillow. It issues `take pillow from sofa 1`, receives `Nothing happens`, and repeats similar invalid source assumptions.

Successful contrast. The paired trajectory searches alternative sources, reaches `armchair 1`, observes `pillow 1`, takes it, and moves it to the sofa—only issuing `take` when the observation lists the target.

Reflected experience. Entry `step_1067`: *"Only use* `take <object> from <container>` *when the object is actually listed at that location; if not observed, do not repeat the same invalid action."*

Later retrieval. At step 60, retrieved for `put some keychain on ottoman`, where the observation lacks the keychain—the same structural error. Utility evolves from 0.000 at creation to 0.578 at step 200 after 78 retrievals.

Figure 6: **Experience bank evolution: early-stage expansion.** Case 1 illustrates destination-first wandering corrected by an object-first scaffold; Case 2 illustrates hallucinated source actions corrected by an observation-gated rule.

trasted trajectories.

E.1 ALFWorld

Figure 8 shows the ALFWorld rollout prompts. The first prompt provides retrieved experiences as additional context, while the second prompt removes external experiences and asks the agent to act only based on the current task, history, observation, and admissible actions.

Figure 9 shows the prompt used to update the ALFWorld experience bank. Given a failed trajectory and a successful reference trajectory for the same task, the model generates compact state-aware experiences in JSON format.

E.2 WebShop

Figure 10 shows the WebShop rollout prompts. The experience-conditioned prompt includes retrieved memories, while the experience-free prompt relies only on the shopping instruction, interaction history, current observation, and admissible actions.

Figure 11 shows the prompt used to update the WebShop experience bank. The model compares failed and successful shopping trajectories and produces new state-aware experiences tied to visible state cues and available actions.

F Dataset License

Our experiments are based on the publicly available ALFWorld (Shridhar et al., 2021) and WebShop (Yao et al., 2022) environments. Training, evaluation, and experience-bank construction are performed using trajectories generated from the task instances and interaction interfaces provided by these environments. We strictly follow the licenses and usage terms of ALFWorld and WebShop, and use the resulting data only for academic research purposes. No private, personally identifiable, or proprietary user data is used in any stage of training, evaluation, or experience construction.

Case 3: From unstructured Pick2 search to systematic collection

Failure (step 165). For `find two ladle and put them in drawer`, the policy spends actions on destination drawers or repeated navigation before confirming where the two targets are. Unlike early scaffolds, this failure involves managing object count, source search, and final placement simultaneously.

Reflected experience. Entry `task_995`: *"Identify likely locations of the target item, inspect those locations methodically, pick up each required object, and then place them into the specified container. Avoid random movement or opening unrelated containers before locating the targets."*

Later retrieval. At step 166, retrieved for `find two spatula and put them in drawer` (similarity 0.854). The rollout locates `spatula 3` on a countertop, picks it up, checks other locations, and deposits it in a drawer. Utility reaches $U{=}0.375$ by steps 195–200.

Case 4: Utility tracking separates useful retrieval from harmful repetition

Creation (step 144). Entry `step_914`, *"Avoid repeating a no-op movement,"* is reflected from a `find two tissuebox and put them in drawer` failure. The rule: if the agent is already at the target location, it should inspect or act rather than re-issuing the same navigation action.

Non-stationary utility. The entry is retrieved frequently but its utility is initially unstable: $U{=}-0.026$ at step 150 and -0.085 at step 165—hovering near the pruning threshold $\eta{=}-0.1$ but remaining above it. Frequency-based retention would treat this entry as important despite its near-zero or negative marginal gain; conversely, a positive threshold would have discarded it prematurely.

Recovery. As the policy reaches more states where repeated no-op movements are the dominant error, the entry becomes useful: utility turns positive at $U{=}0.048$ by step 180 and rises to $U{=}0.452$ at step 200 after 142 retrievals. This illustrates why EDGE combines smoothed EMA tracking (Eq. (14)) with a mildly negative pruning threshold: experience utility can shift as the policy distribution changes, and premature pruning would discard entries whose value has not yet materialized.

Figure 7: **Experience bank evolution: late-stage refinement and non-stationary utility.** Case 3 illustrates specialized multi-object coordination guidance; Case 4 illustrates an entry whose utility is initially near the pruning threshold but recovers as the policy's failure distribution shifts.

G LLMs Usage Statement

We employed a Large Language Model (LLM) to assist exclusively in the editorial stage of manuscript preparation. Its role was limited to refining phrasing, correcting grammar, and enhancing clarity and readability across different sections. The LLM had no involvement in formulating research ideas, designing experiments, or conducting analyses. All scientific contributions and findings are entirely the work of the authors. The authors have ensured that the use of the LLM complies with ethical standards, avoiding plagiarism and scientific misconduct.

Algorithm 1 EDGE: Experience-Distillation for Guided Exploration

Input:

Policy π_θ; experience bank $\mathcal{E}$;
Rollout group size G; distillation weight λ;
EMA momentum μ; pruning threshold η;
Success-rate threshold ξ.

for each training iteration **do**
  for each task x in batch **do**
    Stage 1: Experience-Guided Exploration Scaffolding
    Retrieve top experience $e \in \mathcal{E}$ by embedding similarity.
    Sample G trajectories from π_θ:
    $\mathcal{T}^{\mathsf{T}}$ $(G/2)$: conditioned on $c^{\mathsf{T}} = x \oplus e$
    $\mathcal{T}^{\mathsf{S}}$ $(G/2)$: conditioned on $c^{\mathsf{S}} = x$
    Compute marginal gain Δ_e via Eq. (5).
    Stage 2: Gain-Gated Privileged Distillation
    if $\Delta_e > 0$ **then**
      Compute advantages over all G trajectories.
      for each $\tau \in \mathcal{T}^{\mathsf{S}}$ **do**
        Forward $\pi_{\mathrm{sg}(\theta)}$ on τ with c^{T}.
        Compute $\mathcal{L}_{\text{distill}}$ via Eq. (18).
      end for
    else
      Compute advantages over $\mathcal{T}^{\mathsf{S}}$ only.
      $\mathcal{L}_{\text{distill}} \leftarrow 0$.
    end if
    $\mathcal{L}_{\text{actor}} \leftarrow \mathcal{L}_{\text{RL}} + \lambda\, \mathcal{L}_{\text{distill}}$.
    Stage 3: Experience Bank Evolution
    Update EMA utility: $U_e^{(t)} \leftarrow (1-\mu)\, U_e^{(t-1)} + \mu\, \Delta_e$.
  end for
  Update π_θ with $\mathcal{L}_{\text{actor}}$.
  Prune experiences with $U_e^{(t)} < \eta$ from $\mathcal{E}$.
  Identify task categories with success rate $< \xi$.
  Generate new experiences via $f_{\text{reflect}}(\tau^+, \tau^-)$; insert into $\mathcal{E}$.
end for

Output: Trained policy π_θ (deployed without $\mathcal{E}$).

Prompt: ALFWorld Agent Execution with Experience

System Prompt:

You are an expert agent operating in the ALFRED Embodied Environment. Your task is to: `{task_description}`

 # Retrieved Relevant Experience

`{retrieved_experiences}`

Warning: These experiences may be outdated. Use them only if they align with your current observation.

 # Current Progress

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 is 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 have finished your reasoning, you should choose an admissible action for the current step and present it within `<action> </action>` tags.

Prompt: ALFWorld Agent Execution without Experience

System Prompt:

You are an expert agent operating in the ALFRED Embodied Environment. Your task is to: `{task_description}`

 # Retrieved Relevant Experience

No external general and task-specific experiences are provided. Use your learned strategy and current observation.

 # Current Progress

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 is 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 have finished your reasoning, you should choose an admissible action for the current step and present it within `<action> </action>` tags.

Figure 8: ALFWorld rollout prompts with and without retrieved experience.

Prompt: ALFWorld Experience Update

System Prompt:

You are an expert updating an ALFWorld state-aware experience bank. You are given one failed trajectory and one successful trajectory for the same task. Analyze the contrasted rollout snippets below and propose **NEW or revised experiences** that help the agent act better in the same environment state.

 # Task Information

Task: `{task_text}`

Task Type: `{task_type}`

The task was `{successfully/unsuccessfully}` completed.

 # Contrasted Rollout Snippets

Failed trajectory: `{failed_text}`

Successful trajectory (reference): `{success_text}`

 # Requirements

- Each experience must be **state-aware**, not a generic tip.
- Focus on what the successful no-experience rollout did, and what the retrieved experience may have caused the agent to do incorrectly.
- Prefer compact trigger language that can be matched at retrieval time.
- Use actionable wording tied to admissible actions and visible state cues.
- Return only JSON.

Generate 1–`{max_new_skills_per_update}` new experiences.

 # Output Format Example

```
{
  "title": "Plan object location before acting",
  "principle": "For this task type, first identify where the object is, then plan the sequence of actions.",
  "when_to_apply": "When the task involves finding or moving a specific object"
}
```

Figure 9: Prompt for updating the ALFWorld state-aware experience bank from contrasted failed and successful trajectories. The model is instructed to generate compact, state-aware experiences that can guide future retrieval and decision making.

Prompt: WebShop Agent Execution with Experience

System Prompt:

You are an expert autonomous agent operating in the WebShop e-commerce environment. Your task is to: `{task_description}`.

 # Retrieved Relevant Experience

`{retrieved_memories}`

Warning: These experiences may be outdated. Use them only if they align with your current observation.

 # Current Progress

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: `{available_actions}`

Now it is your turn to take one action for the current step. You should first reason step-by-step about the current situation, then think carefully which admissible action best advances the shopping goal. This reasoning process **MUST** be enclosed within `<think> </think>` tags. Once you have finished your reasoning, you should choose an admissible action for current step and present it within `<action> </action>` tags.

Prompt: WebShop Agent Execution without Experience

System Prompt:

You are an expert autonomous agent operating in the WebShop e-commerce environment. Your task is to: `{task_description}`.

 # Retrieved Relevant Experience

No external general and task-specific experiences are provided. Use your learned strategy and current observation.

 # Current Progress

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: `{available_actions}`

Now it is your turn to take one action for the current step. You should first reason step-by-step about the current situation, then think carefully which admissible action best advances the shopping goal. This reasoning process **MUST** be enclosed within `<think> </think>` tags. Once you have finished your reasoning, you should choose an admissible action for current step and present it within `<action> </action>` tags.

Figure 10: WebShop rollout prompts with and without retrieved experience.

Prompt: WebShop Experience Update

System Prompt:

You are an expert updating a WebShop shopping state-aware experience bank. You are given one failed trajectory and one successful trajectory for the same task. Analyze the contrasted rollout snippets below and propose **NEW or revised experiences** that help the agent act better in the same environment state.

 # Task Information

Task: `{task_text}`

Task Type: `{task_type}`

The task was `{successfully/unsuccessfully}` completed.

 # Contrasted Rollout Snippets

Failed trajectory: `{failed_text}`

Successful trajectory (reference): `{success_text}`

 # Requirements

- Each experience must be **state-aware**, not a generic tip.
- Focus on what the successful no-experience rollout did, and what the retrieved experience may have caused the agent to do incorrectly.
- Prefer compact trigger language that can be matched at retrieval time.
- Use actionable wording tied to available actions and visible state cues.
- Return only JSON.

Generate 1–`{max_new_skills_per_update}` new experiences.

 # Output Format Example

```
{
  "title": "Verify Early, Abort Fast",
  "principle": "On the product page, immediately check category, core attributes, and price; if a key constraint is violated,
       leave the page at once.",
  "when_to_apply": "Within the first observation on every product detail page."
}
```

Figure 11: Prompt for updating the WebShop shopping state-aware experience bank from contrasted failed and successful trajectories. The model is instructed to generate compact, state-aware experiences that can guide future retrieval and shopping decisions.