RICE: RESPONSE-INFORMED CONTEXT EVOLUTION FOR AGENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Interactive agents increasingly summarize completed trajectories as natural-language skills and reuse them across policy updates. This persistence allows past experience to support later decisions, but it also creates a mismatch: stored skill content can remain available while the policy that interprets it continues to change. Task outcomes evaluate the quality of sampled behavior, yet they do not record how retrieved context changes the current policy's predictions. On ALFWORLD/Qwen3-1.7B, 52.1% of the recorded response histories associated with stored skills have tied outcome differences; within this subset, the 95th percentile of the response magnitude is 3.22 times the 5th percentile. This observation shows that outcomes and policy responses describe different properties of the same retrieval event. We define the *local policy response* as the difference between next-token distributions with and without retrieved context at the same decision prefix. We then introduce **Response-Informed Context Evolution (RICE)**, which accumulates these responses across retrieval events and associates the resulting history with the participating skills. Returns select which observed behavior provides supervision, response history determines how strongly accepted examples contribute, and response history together with past outcome evidence controls which skills remain available for future retrieval. Paused skills remain stored and can return to the available set when their outcome utility recovers. Across three model sizes on ALFWORLD, WEBSHOP, and SEARCH-QA, RICE achieves the highest listed overall success on ALFWORLD and exact success on WEBSHOP, while the SEARCH-QA comparisons are mixed. In a controlled ALFWORLD/Qwen3-1.7B comparison, validation success increases from 92.4% for the Base configuration to 94.5% with response weighting and 96.9% with response-informed retrieval control. These results support treating retrieval as both context provision and feedback about how stored skills affect an evolving policy.

1 INTRODUCTION

Large language models are increasingly trained as interactive agents that reason, act, use tools, and revise their behavior over extended interactions. Reinforcement learning provides task-level feedback, but a final outcome must guide many intermediate decisions (Feng et al., 2025; Zeng et al., 2025; Yang et al., 2026a). Completed trajectories provide an additional source of learning signal: after an interaction ends, an agent can identify useful procedures, decisive observations, and recurring mistakes, summarize them as natural-language skills, and reuse these skills to guide or supervise later behavior (Wang et al., 2026; Lu et al., 2026; Wu et al., 2026; Tu et al., 2026a). Once stored, these skills may persist across many subsequent policy updates.

Persistent skill reuse couples stored context to a policy that continues to evolve. The same skill can induce different local prediction changes across policy states, even when sampled task outcomes coincide. Each retrieval therefore provides information about both the observed outcome and the current policy's response to the retrieved context, as illustrated in Figure 1. Outcome statistics record behavioral quality, but they do not record how strongly the retrieved context changes the current learner.

We call the difference between next-token distributions at the same decision prefix with and without retrieved context the *local policy response*. Task outcomes and local responses provide comple-

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

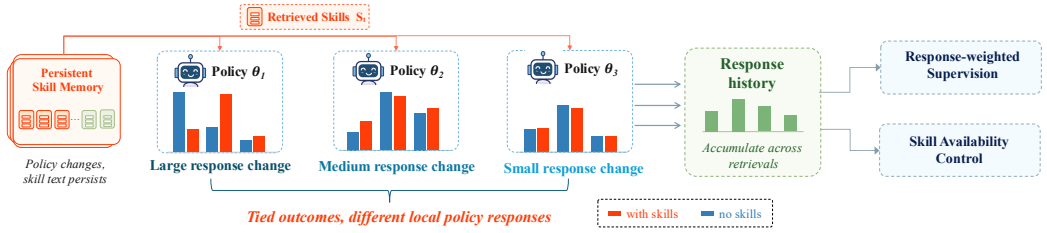


Figure 1: **Motivation for response-informed skill reuse.** A stored skill can induce different local policy responses as the learner changes. RICE accumulates these observations and uses the resulting history for response-weighted supervision and retrieval availability control.

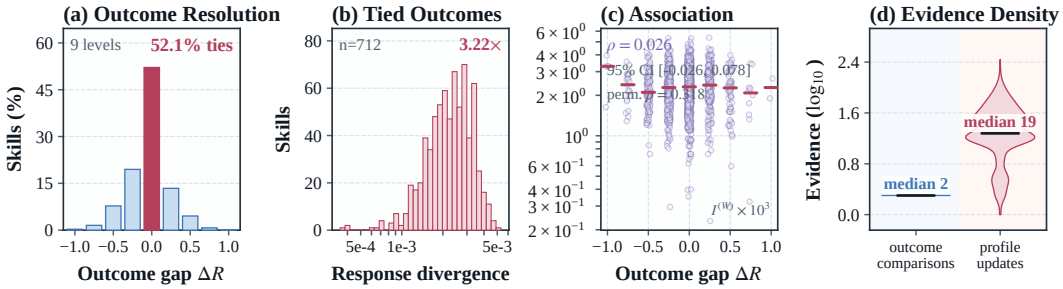


Figure 2: **Equal observed outcomes can accompany different local policy responses.** Panel (a) shows the distribution of observed outcome differences. Panel (b) shows local policy responses among tied outcomes. Panel (c) reports the association between outcome differences and policy responses. Panel (d) compares the number of outcome comparisons and response observations; these counts reflect recording density.

mentary evidence. Returns indicate which observed behavior is better suited to provide supervision, whereas the response measures how strongly retrieved context changes the current learner. Figure 2 illustrates this distinction on ALFWORLD/Qwen3-1.7B. Among 1,366 skill-associated response histories, 52.1% correspond to tied observed outcome differences, yet within this subset the 95th percentile of local response is 3.22 times the 5th percentile. In the same run, the measured association between outcome differences and local responses is near zero ($\rho = 0.026$, permutation $p = 0.318$). Thus, task outcomes and local responses distinguish different properties of the same retrieval event.

Recent work has addressed complementary parts of this learning process. Hindsight and self-distillation methods turn completed experience or privileged skill context into additional policy supervision, using trajectory comparison, token-level preferences, or context-induced probability changes to refine the current learner (Wang et al., 2026; Lu et al., 2026; Wu et al., 2026; Yang et al., 2026b; Zhang et al., 2026a; Lv et al., 2026; Pan et al., 2026). A parallel line of work maintains reusable skills across training, using relevance, return-based credit, downstream utility, validation performance, or explicit lifecycle rules to retrieve, revise, retain, or suppress stored experience (Xia et al., 2026; Tu et al., 2026b; Shi et al., 2026; Tu et al., 2026a; He et al., 2026; Shen et al., 2026; Zhang et al., 2026b; Lin et al., 2026; Zhu et al., 2026). Together, these methods show how hindsight can supervise the current policy and how persistent skills can be evaluated and evolved over time. What remains unmodeled in the representative methods we consider is a persistent record of how retrieved context changes the learner when it is used.

Our central idea is to retain the learner's response to retrieved skills as feedback for later reuse. We introduce **Response-Informed Context Evolution (RICE)**, which maintains an accumulated response history across repeated retrieval events. This history summarizes how successive policy states respond when a skill participates in retrieval. At each measured decision, RICE compares next-token distributions with and without the complete retrieved set. Returns determine which observed behavior provides supervision. Response history determines how strongly each accepted ex-

ample contributes and, together with past outcome utility, whether skills remain available for future retrieval. Each retrieval therefore supplies context for the current decision and produces evidence that guides subsequent reuse.

Sequential configurations directly examine the two uses of response history. On ALFWORLD/Qwen3-1.7B, validation success is 92.4%, 94.5%, and 96.9% for the Base, Policy, and Full configurations, respectively; the retrieval-eligible set decreases from $3{,}084$ in Policy to $1{,}630$ in Full. Across three model sizes on ALFWORLD, WEBSHOP, and SEARCH-QA, RICE-Full achieves the highest listed overall success on ALFWORLD and exact success on WEBSHOP, while the SEARCH-QA comparisons are mixed. We also evaluate RICE-RL, which uses response-informed retrieval without auxiliary distillation in the primary actor update and reaches 90.6%, 96.1%, and 98.4% ALFWORLD success with retrieved skills across the three model sizes. Together, these experiments evaluate system performance, the contribution of the two uses of response history, and the use of response evidence under a different actor objective.

The main contributions are:

1. We show that equal observed outcome differences can accompany different local policy responses, demonstrating that outcomes do not fully describe how retrieved context affects the current learner.
2. We introduce RICE, which records responses to the complete retrieved set across retrievals. The resulting history weights examples selected by returns and, together with outcome utility, controls whether skills remain available for later retrieval.
3. We evaluate RICE across three agent tasks and three model sizes. The results include controlled comparisons of response weighting and retrieval control, system-level performance, and analyses of supervision allocation, retrieval availability, and temporal variation.

2 WHY OUTCOMES ARE INCOMPLETE FOR PERSISTENT SKILL REUSE

We study an agent that learns with persistent natural-language skills while its policy continues to change. At training iteration n, the policy is π_{θ_n}, the skill bank $\mathcal{B}_n$ contains every stored skill, and the available set $\mathcal{A}_n \subseteq \mathcal{B}_n$ contains the skills that may be retrieved. Given the decision context x_t, comprising the task, observation, and interaction history, the retriever selects $S_t = \mathcal{R}(x_t; \mathcal{A}_n) \subseteq \mathcal{A}_n$; the observation also serves as the anchor $\tilde{s}_t$ for grouping rollout records. The selected set S_t conditions the current action distribution, whereas $\mathcal{A}_n$ determines which skills can enter later trajectories. Learning from retrieval therefore requires deciding how strongly the resulting examples contribute to the current policy update and whether the participating skills remain available. Outcomes evaluate observed behavior but do not reveal how retrieved context changes current policy predictions, since equal outcomes can accompany different local responses.

To distinguish behavioral quality from the policy's local response to retrieved context, we consider two prompts under the same policy parameters. The prompt $P_t^+ = \mathrm{Prompt}(x_t, S_t)$ contains the retrieved skills, whereas $P_t^0 = \mathrm{Prompt}(x_t, \emptyset)$ retains the same task, observation, and interaction history but removes them. Let G_t denote the return from decision t onward, and let $p_t^c(\cdot) = \pi_{\theta_n}(\cdot \mid P_t^c)$ be the next-token distribution for prompt $c \in \{+, 0\}$. These prompts define two complementary quantities:

$$\Delta_t \coloneqq \mathbb{E}[G_t \mid P_t^+] - \mathbb{E}[G_t \mid P_t^0], \qquad D_t \coloneqq D_{\mathrm{JS}}(p_t^+ \| p_t^0). \tag{1}$$

The return difference Δ_t compares the expected behavioral quality of the two conditions, and its sign identifies which condition provides supervision. The local policy response D_t instead measures the magnitude of the prediction change. Evaluating both distributions under the same policy parameters and at the same decision prefix isolates the change in retrieved context. Because D_t is unsigned, it cannot determine which behavior is better or replace the return difference. It also describes the complete retrieved set rather than the marginal contribution of an individual skill. Repeated measurements are associated with the participating skills to form response histories. Returns therefore determine the supervision direction, while response history informs relative training weight and, together with outcome utility, future retrieval availability. Section 3 specifies these updates.

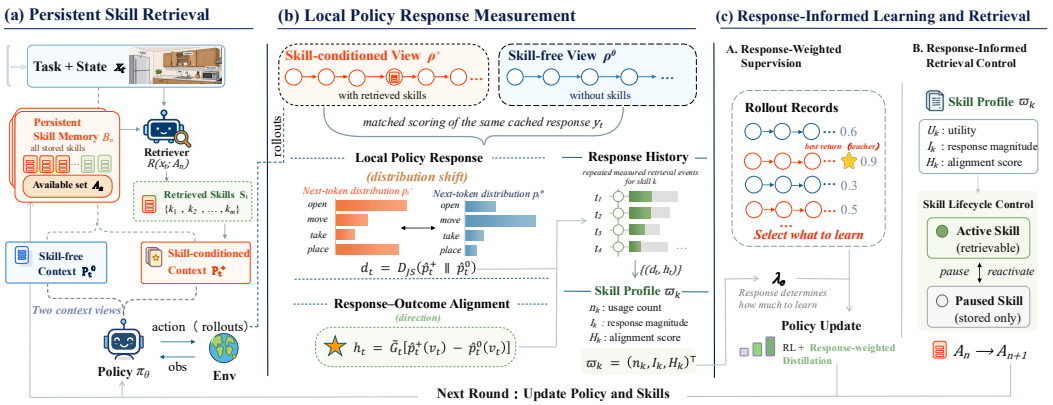


Figure 3: **Overview of RICE.** (a) The agent retrieves task and state skills from a persistent bank. (b) Skill-conditioned and skill-free prompts are scored at the same decision prefix to measure the local policy response and update the response history. (c) Returns select the supervision target, while response history weights accepted examples and, together with outcome utility, updates retrieval eligibility for subsequent interactions.

3 RICE: Persistent Policy-Response Feedback

RICE accumulates the context comparison in Eq. (1) as response history for later skill reuse. The method separates three decisions: which observed behavior provides supervision, how strongly each accepted example contributes, and which skills remain available for retrieval. Figure 3 summarizes the complete workflow. For each task, we collect rollouts with and without skills. At each turn, the acting prompt generates a reply, and the record stores that reply, both prompt versions, the anchor observation, and the observed reward. Response scoring and supervision selection use these records after rollout, before the policy update. Its data flow is $(P_t^+, P_t^0) \to (d_t, h_t) \to \boldsymbol{\varpi}_k \to (\lambda_e, A_{n+1})$: the two prompts at a fixed decision prefix produce a local policy-response observation, repeated observations form an accumulated response history, and this history affects both current supervision and future retrieval. Returns retain a separate role: they select which observed behavior provides supervision, while response history modulates accepted supervision and, together with past outcomes, controls retrieval eligibility. Appendix B.3 gives the complete update order.

3.1 Local Policy Response

After rollout, we select valid records from skill-conditioned trajectories for response measurement. For each selected record, $y_t = (y_{t,1}, \ldots, y_{t,L_t})$ is the reply already sampled at that turn under P_t^+. We score this same reply under the stored P_t^+ and P_t^0, holding the task, observation, and interaction history fixed and changing only the retrieved skill context. Both scores reuse the cached reply and decision prefix. Let $v_t = y_{t,1}$ and $\ell_t^c(v)$ be the logit predicting the first reply token under condition $c \in \{+, 0\}$. We use the shared support $\Omega_t = \mathrm{TopK}(\ell_t^+, K_D) \cup \mathrm{TopK}(\ell_t^0, K_D) \cup \{v_t\}$ and $\hat{p}_t^c = \mathrm{softmax}(\ell_t^c|_{\Omega_t})$. With $\widetilde{G}_t$ denoting the return centered over active records with the same task and anchor observation, we record

$$d_t := D_{\mathrm{JS}}(\hat{p}_t^+ \| \hat{p}_t^0), \qquad h_t := \widetilde{G}_t\big[\hat{p}_t^+(v_t) - \hat{p}_t^0(v_t)\big]. \tag{2}$$

Here $d_t \in [0, \log 2]$ measures the magnitude of the local next-token response, while h_t records whether the probability change on the sampled token agrees in sign with the centered return. Thus, d_t measures response size and h_t provides outcome-conditioned directional evidence for the complete retrieved set.

Let $S_t^{\mathrm{meas}} \subseteq S_t$ contain the stored skills participating in the measurement. For skill k, define $\mathcal{T}_k = \{t : k \in S_t^{\mathrm{meas}}\}$ and $n_k = |\mathcal{T}_k|$. Its response history is $\boldsymbol{\varpi}_k = (n_k, I_k, H_k)^\top$, where $(I_k, H_k) = n_k^{-1} \sum_{t \in \mathcal{T}_k} (d_t, h_t)$. The profile is a cumulative, skill-indexed history of joint-context responses observed across retrieval events containing k. The skill index supports later supervision and

eligibility decisions, while the measurement unit remains the complete retrieved set. Appendix B.1 details the cumulative statistics.

3.2 Response-Weighted Supervision

Response history affects supervision only after outcomes determine its direction. After rollout, we group valid records by task u and anchor observation $\tilde{s}$, following the grouping criterion used by GiGPO (Feng et al., 2025), to form $\mathcal{G}(u, \tilde{s})$. Records in a group may have different interaction histories: response measurement fixes the complete decision prefix, whereas supervision selection groups records by task and anchor observation. We select $\rho^\star = \arg\max_{\rho \in \mathcal{G}(u,\tilde{s})} G_\rho$ as the detached teaching record and use its sampled reply to supervise the opposite prompt condition. A distinct target record ρ^{tar} in the same group supplies that prompt from its stored versions. The pair $e = (\rho^\star, \rho^{\mathrm{tar}})$ is accepted only when its margin $\Delta_e = G_{\rho^\star} - G_{\rho^{\mathrm{tar}}} > \epsilon_{\mathrm{CB}}$; Appendix B.2 specifies the fallback when no distinct target record is available. Hence outcome evidence determines *what to imitate* before response history changes its contribution.

For accepted pair e, let S_e^{valid} contain retrieved skills with valid response histories. We compute $\phi_e = |S_e^{\mathrm{valid}}|^{-1} \sum_{k \in S_e^{\mathrm{valid}}} \mathrm{clip}(I_k/\tau_I, 0, \phi_{\max})$ and normalize it as $\lambda_e = \phi_e / \bar{\phi}$, where $\bar{\phi}$ is the mean valid score in the batch. Pairs without valid response history use $\lambda_e = 1$. The resulting auxiliary objective is

$$\mathcal{L}_{\mathrm{RICE}} = \bar{Z}_{\mathrm{CB}}^{-1} \sum_{e,j} \bar{\omega}_{e,j} \bar{\lambda}_e D_{\mathrm{KL}}(q_{e,j} \| p_{e,j}) \,, \tag{3}$$

where $q_{e,j}$ and $p_{e,j}$ are the detached teacher and trainable actor distributions, $\bar{\omega}_{e,j}$ contains the return-margin, confidence, validity-mask, and action-token factors, $\bar{Z}_{\mathrm{CB}} = \sum_{e,j} \bar{\omega}_{e,j}$, and $\bar{\lambda}_e = \mathrm{sg}[\lambda_e]$. Thus, Δ_e determines the supervision direction, while λ_e allocates supervision among accepted pairs.

The primary update uses $\mathcal{L}_{\mathrm{actor}} = \mathcal{L}_{\mathrm{RL}} + \beta_{\mathrm{ref}} \mathcal{L}_{\mathrm{ref}} + \lambda_{\mathrm{BD}} \mathcal{L}_{\mathrm{RICE}}$, followed by the independent reflection update. Appendix B.2 and Appendix B give the exact normalization, fallback cases, and optimizer order.

3.3 Response-Informed Retrieval Control

Response history also changes which stored skills can enter later trajectories. This operates above the underlying retriever: relevance, outcome utility, and exploration continue to rank currently available skills, while RICE controls which stored skills remain eligible for that ranking.

For skill k, U_k summarizes past outcome evidence, while I_k and H_k summarize the magnitude and alignment of its recorded response history. Let $r(k)$ denote the skill type, $f_{r(k)}$ its utility threshold, and $\mathrm{SE}(H_k)$ the nominal dispersion scale defined in Appendix B.1. A currently available skill becomes eligible for pausing only when

$$\mathcal{E}_{\mathrm{pause}}^{(n)} = \left\{ k \in A_n : n_k \geq n_{\min},\ U_k \leq f_{r(k)},\ (I_k \leq \tau_I \ \vee \ H_k < -c_{\mathrm{SE}}\, \mathrm{SE}(H_k)) \right\}. \tag{4}$$

The outcome condition prevents low response alone from suppressing a skill with favorable historical utility. Conditional on weak utility, pausing is considered when the recorded response is small or its alignment falls below the dispersion-scaled boundary $-c_{\mathrm{SE}}\, \mathrm{SE}(H_k)$.

Unavailable skills whose utility recovers above the threshold are reactivated first. Among the remaining pause candidates, RICE prioritizes smaller I_k and applies only a bounded number of transitions. Writing $\mathcal{E}_n^{\mathrm{re}}$ for reactivated skills and $\widehat{\mathcal{E}}_n^{\mathrm{pause}}$ for the selected pause subset, the available set becomes $A_n^{\mathrm{res}} = (A_n \cup \mathcal{E}_n^{\mathrm{re}}) \setminus \widehat{\mathcal{E}}_n^{\mathrm{pause}}$ before newly written skills are added.

Pausing changes retrieval eligibility while retaining each skill in the persistent bank for later reactivation. The controller limits transitions per update and suspends further pausing when the inventory or skill-writing health guard activates. Appendix B specifies the transition budget, active-set floor, health guard, and reactivation rule. Section 4.6 summarizes the rule through four complementary decision patterns, with the corresponding final-state records in Appendix E.3. RICE-Full uses response history for both supervision weighting and retrieval control, whereas RICE-RL uses it for retrieval without auxiliary distillation under the configuration in Appendix C.3.

4 EXPERIMENTS

The experiments first evaluate whether RICE improves agent performance across tasks and model sizes. We then isolate the contributions of response-weighted supervision and response-informed retrieval control, and examine whether the measured responses affect supervision and retrieval as intended. Finally, we test whether response-informed retrieval remains useful when the primary actor update does not include auxiliary distillation.

4.1 EXPERIMENTAL SETUP

Benchmarks and metrics. We evaluate three forms of long-horizon agent behavior. ALFWORLD (Shridhar et al., 2020) presents household tasks as text interactions and includes Pick, Look, Clean, Heat, Cool, and Pick2. We report success for each task family and across all evaluation episodes. WEBSHOP (Yao et al., 2022) requires an agent to search for products, verify requested attributes, and complete a purchase. We report normalized task score and exact success on the 128-task evaluation set. SEARCH-QA (Jin et al., 2025) evaluates information seeking with search tools on NQ, TriviaQA, PopQA, HotpotQA, 2Wiki, MuSiQue, and Bamboogle. We report accuracy for each dataset and their unweighted average. Together, these benchmarks cover embodied control, web interaction, and single-hop and multi-hop search.

Models, baselines, and protocol. We use Qwen2.5-7B-Instruct, Qwen2.5-3B-Instruct (Qwen Team, 2024), and Qwen3-1.7B (Yang et al., 2025). Table 1 compares three groups of methods. **RL without skills** includes GRPO and GiGPO (Feng et al., 2025). **Methods with persistent skills** include SkillRL (Xia et al., 2026), D2Skill (Tu et al., 2026b), SkillC (Lin et al., 2026), SAPO (Zhang et al., 2026b), and Skill1 (Shi et al., 2026). **Self-distillation methods** include RLSD (Yang et al., 2026a), Skill-SD (Wang et al., 2026), SDAR (Lu et al., 2026), StepOPSD (Zhang et al., 2026a), OPID (Yang et al., 2026b), PCSD (Lv et al., 2026), and BCSD (Pan et al., 2026). These groups represent learning directly from task outcomes, learning with stored skills, and adding supervision derived from the learner's own behavior. The symbols in Table 1 identify results taken from cited sources. All WebShop results use the same 128-task evaluation protocol, allowing direct comparison of score and success.

RICE-Full retains the environment return objective and collects rollouts with and without retrieved skills within each task group. At fixed decision prefixes, their next-token distributions provide the observations accumulated in response history. This history weights selected training examples and helps determine which skills remain available for later retrieval. The response measurement and retrieval settings are shared across benchmarks and model sizes; the base learner settings and environment interface follow each benchmark. Appendix C gives the full protocol, and Appendix E presents representative interactions and prompts.

4.2 MAIN RESULTS

Table 1 reports the performance of the complete RICE system across the three agent settings. The clearest gains occur on the two interactive benchmarks, while the SEARCH-QA comparisons are mixed.

ALFWorld. RICE-Full reaches 96.9%, 93.0%, and 93.0% overall success with the 1.7B, 3B, and 7B backbones, exceeding the strongest listed comparator by 17.2, 2.4, and 0.8 points. The largest margin occurs with the 1.7B backbone, and the gains are distributed across the six task families.

WebShop. On the shared 128-task evaluation set, RICE obtains score and success pairs of $86.7/75.8$, $91.8/86.7$, and $87.9/84.4$ at 1.7B, 3B, and 7B. The corresponding success margins over the strongest listed comparator are 11.0, 11.7, and 1.6 points. The normalized score measures partial progress toward the requested product, while success requires exact task completion.

Search-QA. The average accuracies across the seven datasets are 42.3%, 44.4%, and 48.0% at 1.7B, 3B, and 7B. RICE ties the strongest listed average at 1.7B but does not exceed the strongest listed baselines at 3B and 7B. The individual results also vary across the single-hop and multi-hop datasets.

Table 1: **Overall results across agent tasks and model scales.** The table reports task-level and overall ALFWORLD success, WEBSHOP score and success on the shared 128-task evaluation set, and SEARCH-QA per-dataset accuracy with their macro-average. "–" denotes an unreported result, † a result from the cited source, and * a result reported by SDAR (Lu et al., 2026).

	ALFWORLD							WEBSHOP		SEARCH-QA							
Method	**Pick**	**Look**	**Clean**	**Heat**	**Cool**	**Pick2**	**Succ**	**Score**	**Succ**	**NQ**	**Triv**	**Pop**	**Hotp**	**2Wk**	**MuS**	**Bam**	**Avg**
Qwen2.5-7B-Instruct																	
GRPO*	91.2	87.5	96.2	81.0	65.0	57.9	81.2	80.9	72.6	45.1	63.7	44.0	43.6	43.2	16.8	37.6	42.0
GiGPO†	97.7	82.7	98.8	83.7	89.3	79.2	90.8	84.4	72.8	46.4	64.7	46.1	41.6	43.6	18.9	68.9	47.2
SkillRL†	97.9	71.4	90.0	90.0	95.5	87.5	89.9	85.2	72.7	45.9	63.3	45.9	43.2	40.3	20.2	73.8	47.5
D2Skill†	93.8	77.8	94.7	95.0	95.5	72.0	87.8	91.1	80.5	48.7	63.4	44.9	47.5	43.7	21.0	67.7	48.1
SkillC†	88.5	78.6	91.2	94.7	100.0	88.9	90.6	85.6	74.0	–	–	–	–	–	–	–	–
SAPO†	98.7	73.9	98.1	92.6	85.0	89.2	92.2	90.5	78.1	48.4	62.9	46.7	45.0	45.2	18.3	46.4	44.7
RLSD*	100.0	87.5	92.3	58.8	80.0	65.2	82.0	87.4	77.3	46.8	63.0	44.4	45.5	48.9	21.5	73.0	49.0
Skill-SD*	93.9	93.8	90.9	100.0	69.2	68.4	85.1	86.1	76.5	47.1	64.5	47.8	44.2	42.1	20.2	69.0	47.8
SDAR*	94.7	75.0	100.0	86.7	68.2	78.9	85.9	89.4	82.8	46.3	63.5	48.2	43.8	48.4	19.6	73.0	49.0
BCSD†	89.7	66.7	100.0	93.3	74.1	71.4	83.6	85.9	78.1	–	–	–	–	–	–	–	–
OPID†	100.0	81.8	97.1	100.0	80.8	80.0	90.0	85.3	79.7	48.8	65.6	46.8	46.1	42.7	21.7	72.6	**49.2**
Skill1	100.0	83.3	89.5	100.0	88.5	85.0	92.2	86.2	77.3	46.8	46.3	47.8	43.7	39.3	18.2	70.6	44.7
RICE-FULL	100.0	100.0	100.0	92.9	82.1	81.2	**93.0**	87.9	84.4	47.8	65.9	46.9	43.2	44.3	17.8	70.2	48.0
Qwen2.5-3B-Instruct																	
GRPO*	91.2	62.5	96.2	61.9	65.0	47.4	75.0	79.8	63.3	39.3	60.6	41.1	37.4	34.6	15.4	26.4	36.4
GiGPO	94.3	72.7	93.8	100.0	64.3	89.5	85.9	83.8	70.3	43.0	60.5	46.9	36.2	35.5	12.1	64.1	42.6
RLSD*	87.9	75.0	90.9	75.0	73.1	68.4	79.7	84.4	66.4	41.5	58.6	42.3	40.4	40.2	16.8	66.9	43.8
Skill-SD*	88.2	50.0	96.2	52.4	65.0	57.9	73.4	75.9	64.0	44.4	60.4	44.0	39.5	40.4	15.4	64.9	44.1
SDAR*	97.1	62.5	100.0	61.9	75.0	84.2	84.4	85.0	68.0	44.8	58.1	44.3	38.6	36.2	15.7	66.1	43.4
StepOPSD†	97.1	66.7	87.0	79.1	78.9	95.0	83.6	–	–	45.0	61.6	46.2	39.5	39.5	14.4	65.3	44.5
BCSD†	93.1	50.0	100.0	66.7	77.8	75.0	82.0	82.3	66.4	–	–	–	–	–	–	–	–
PCSD†	100.0	63.6	82.1	83.3	94.4	100.0	90.6	85.0	67.2	–	–	–	–	–	–	–	–
OPID†	92.7	100.0	88.9	70.0	84.2	70.0	84.3	85.0	74.2	45.9	61.4	45.7	40.7	38.8	16.4	66.1	**45.0**
Skill1	96.7	76.9	96.3	93.8	77.3	95.0	90.6	84.1	75.0	44.2	60.3	44.8	39.2	35.9	13.1	64.9	43.2
RICE-FULL	100.0	90.0	96.3	100.0	86.4	83.3	**93.0**	91.8	86.7	45.5	61.7	46.0	39.3	37.6	14.9	66.1	44.4
Qwen3-1.7B																	
GRPO*	71.1	41.7	36.4	40.0	31.8	31.6	46.1	67.3	38.3	40.0	58.9	43.5	35.4	30.3	12.0	65.7	40.8
GiGPO	96.7	69.3	88.9	62.5	59.1	85.0	79.7	70.0	45.3	41.1	59.3	45.1	34.1	32.7	10.5	64.5	41.0
RLSD*	50.0	37.5	61.5	19.0	50.0	21.1	42.2	74.0	50.8	38.6	57.3	43.0	34.5	34.1	11.5	65.3	40.6
Skill-SD*	52.9	37.5	69.2	42.9	60.0	36.8	52.3	81.8	53.9	39.1	57.5	45.4	34.8	34.1	10.7	64.1	40.8
SDAR*	73.5	25.0	76.9	33.3	40.0	36.8	53.9	76.8	58.6	39.7	58.9	45.3	35.9	35.5	12.6	65.3	41.9
StepOPSD†	64.7	44.4	56.5	60.9	42.1	55.0	56.3	–	–	40.5	59.4	44.4	37.1	32.0	11.6	64.1	41.3
BCSD†	51.7	33.3	69.6	53.3	44.4	25.0	46.9	76.8	50.8	–	–	–	–	–	–	–	–
PCSD†	63.3	68.8	50.0	69.2	48.1	63.6	59.4	78.9	58.6	–	–	–	–	–	–	–	–
OPID†	65.9	72.7	66.7	40.0	63.2	45.0	58.9	79.6	64.8	38.1	58.1	43.4	35.5	31.7	11.7	64.5	40.4
Skill1	71.4	54.5	59.4	72.7	60.0	73.7	65.6	83.2	61.7	42.7	59.8	46.0	35.4	34.8	12.0	65.7	**42.3**
RICE-FULL	100.0	88.9	100.0	100.0	90.0	96.2	**96.9**	86.7	75.8	43.9	59.9	44.3	37.6	36.3	11.9	62.5	**42.3**

The main results establish system-level performance across the evaluated settings. We next use a controlled configuration comparison to examine the two roles assigned to response history.

4.3 CONTRIBUTION OF RESPONSE-WEIGHTED SUPERVISION AND RETRIEVAL CONTROL

We compare three configurations that share the same ALFWORLD/Qwen3-1.7B training setup, including supervision selected by returns and skill retrieval based on utility. RICE-Base provides this shared setup, RICE-Policy adds response weighting through I_k, and RICE-Full additionally combines (U_k, I_k, H_k) to control retrieval availability. Appendix C.4 gives the exact definitions of the three variants.

Each successive use of response history improves validation success in this controlled comparison. Validation is 92.4% for Base, 94.5% for Policy, and 96.9% for Full (Figure 4, panels (a)–(b)). Policy differs from Base by response weighting, while Full additionally applies response-informed retrieval control. At the reported endpoints, $3{,}084$ skills are eligible for retrieval in Policy and $1{,}630$ in Full, while all stored records are retained. The comparison therefore links the two uses of response history to successive changes in performance and retrieval availability within the same training setup.

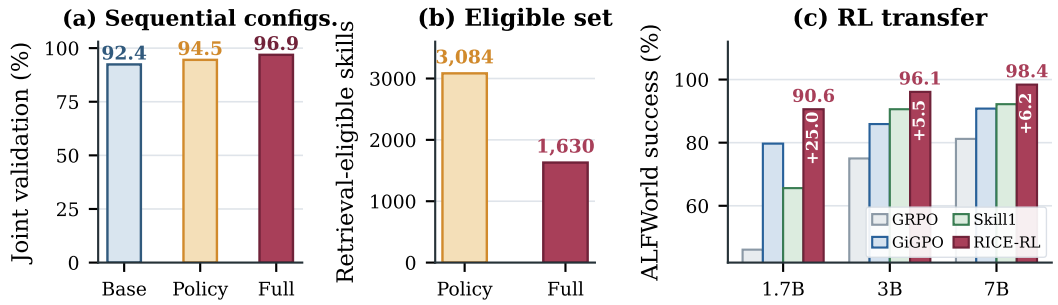


Figure 4: **Performance of the supervision and retrieval configurations and RICE-RL.** Panels (a)–(b) compare the Base, Policy, and Full configurations, which differ in response weighting and retrieval control. Panel (c) shows RICE-RL alongside reported Skill1 results and other RL references without auxiliary distillation in the primary actor update.

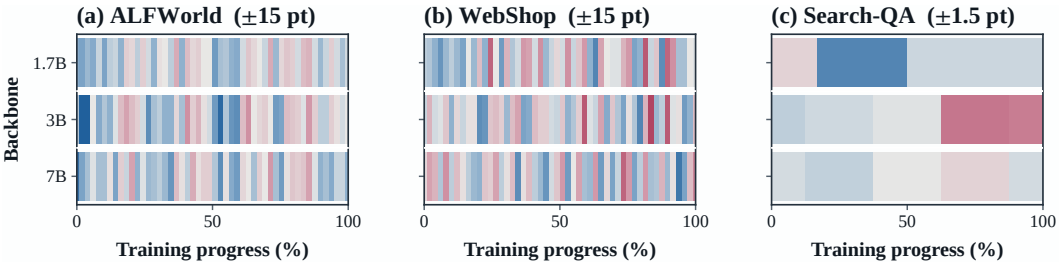


Figure 5: **The preferred context condition changes across training.** Each cell is the signed validation difference between rollouts with and without retrieved skills at a recorded checkpoint. Red favors retrieval, blue favors no retrieval, and gray indicates parity. Interactive tasks use a ± 15 point range, whereas SEARCH-QA uses ± 1.5 points. The sign changes support selecting the training target from observed returns.

4.4 WHY RETURNS SELECT THE SUPERVISION DIRECTION

Figure 5 tracks the signed validation difference between rollouts with and without retrieved skills across tasks and model sizes. The preferred context condition alternates across checkpoints, supporting the use of observed returns within each task and anchor group to select the training target.

4.5 RESPONSE SEPARATION AND VARIATION IN TRAINING WEIGHTS

We next examine whether the response measurement distinguishes retrieved skills from unrelated context and whether its accumulated history changes supervision allocation. On the ALFWORLD/Qwen3-1.7B measurement run, the mean response normalized by the entropy of the distribution without skills is 0.00924 higher for retrieved context than for sham context at the same decision prefixes (Figure 6, left). Across the nine combinations of task and model size, the resulting weights have nonzero coefficients of variation and effective sample size ratios below one (Figure 6, right). These observations show that the measured response differs from sham context in this run and that response history produces non-uniform relative weights over selected training examples. Appendix D.1 gives the resampling procedure and comparisons across response estimators and settings.

4.6 CASE STUDY: RESPONSE-INFORMED RETRIEVAL AVAILABILITY

Equation (4) applies three stages of evidence before changing retrieval availability. A skill first accumulates at least $n_{\min}$ response observations and reaches its type-specific utility floor. The low-

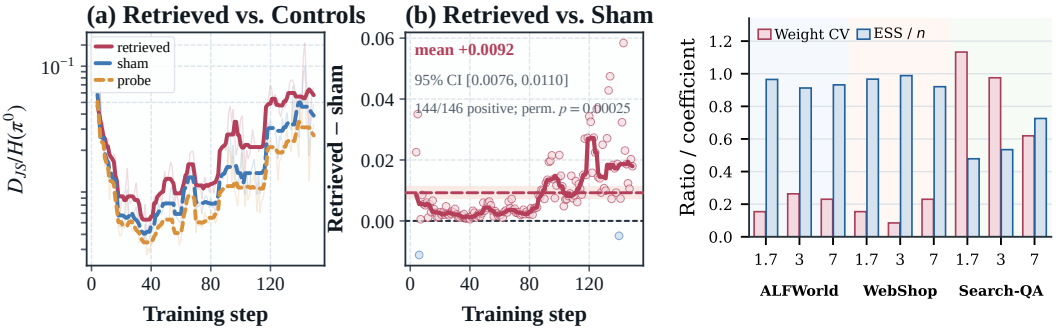


Figure 6: **Response measurements and resulting training-weight variation.** Left: response normalized by policy entropy for retrieved, sham, and probe contexts at the same decision prefixes on ALFWORLD/Qwen3-1.7B, including the response difference (retrieved minus sham); resampling details are in Appendix D.1. Right: coefficient of variation of response weights and the effective sample size ratio across tasks and model sizes. Nonzero variation with $\mathrm{ESS}/n < 1$ shows that response history changes the relative contribution of measured pairs.

response and negative-alignment conditions then determine pause eligibility, with favorable utility protecting the skill. The transition budget and health safeguards are applied after this eligibility test.

The four final-state records in Appendix E.3 instantiate these paths on ALFWORLD/Qwen3-1.7B. The utility-protection case remains available because $U_k = 0.1625$ exceeds its type-specific floor, despite a low response magnitude of $I_k = 0.0010$. Among records with weak utility, a low response magnitude or sufficiently negative alignment can make a skill eligible for pausing, whereas a record satisfying neither response condition remains available. These cases show how response history distinguishes among records that reach the same utility gate without overriding favorable outcome.

4.7 REVERSIBILITY OF RETRIEVAL AVAILABILITY

Pausing retrieval retains the skill record for later reactivation. At the end of training, 68% of stored skills remain available on ALFWORLD and 70% remain available on WEBSHOP. Figure 10 shows the retrieval-eligible fraction and cumulative readmissions during training. The observed readmissions confirm that availability can recover rather than changing through permanent deletion. Appendix E provides representative supervision and availability decisions, while Appendix D.5 reports the lifecycle trajectories.

4.8 RESPONSE-INFORMED RETRIEVAL WITHOUT AUXILIARY DISTILLATION

We finally test whether response history can guide retrieval when the primary actor update does not include the auxiliary distillation loss. RICE-RL combines response history with outcome utility to determine which skills remain available for retrieval, while using group relative reinforcement learning for the primary actor update. It reaches 90.6%, 96.1%, and 98.4% success with retrieved skills on the 1.7B, 3B, and 7B models, compared with 65.6%, 90.6%, and 92.2% for the reported Skill1 results. The corresponding differences are 25.0, 5.5, and 6.2 percentage points (Figure 4, panel (c)). Appendix C.3 gives the skill initialization and retrieval settings for this configuration.

5 CONCLUSION

Stored skills can outlast the policy state in which they were written. RICE therefore complements outcome-based evaluation with a persistent record of the local policy response to retrieved context. Returns select the behavior that provides supervision, while response history determines how strongly accepted examples contribute and, together with outcome evidence, which skills remain available for later retrieval. Across three model sizes, RICE achieves the highest listed overall success on ALFWORLD and exact success on WEBSHOP; the SEARCH-QA comparisons are mixed. These results support treating retrieval as both context provision and feedback for adapting how persistent skills are reused.

AI USE STATEMENT

Generative AI tools assisted with manuscript language editing and result analysis. The authors verified all outputs and accept full responsibility for the final manuscript.

REFERENCES

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *arXiv preprint arXiv:2505.10978*, 2025. URL https://arxiv.org/abs/2505.10978.

Zelin He, Haotian Lin, Boran Han, Wei Zhu, Haoyang Fang, Bernie Wang, Xuan Zhu, Runze Li, and Matthew Reimherr. ReSkill: Reconciling skill creation with policy optimization in agentic rl. *arXiv preprint arXiv:2606.01619*, 2026. URL https://arxiv.org/abs/2606.01619.

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

Hongxiang Lin, Zhirui Kuai, Erpeng Xue, and Lei Wang. Skillc: Learning autonomous skill internalization in llm agents via contrastive credit assignment. *arXiv preprint arXiv:2605.27899*, 2026. URL https://arxiv.org/abs/2605.27899.

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. *arXiv preprint arXiv:2605.15155*, 2026. URL https://arxiv.org/abs/2605.15155.

Chunji Lv, Yangguang Wei, Junlin Liu, Yang Gao, Ming Liu, Xinming Wang, Jinyang Wu, Guoren Wang, and Changsheng Li. PCSD: Persistent consistency for self-distillation in agentic reinforcement learning. *arXiv preprint arXiv:2608.01837*, 2026. URL https://arxiv.org/abs/2608.01837.

Tianjun Pan, Yuan Li, Hongda Wang, Linbo Jin, Mengfei Song, Lei Gao, Qiming Shi, Shaokang Fu, Jiarong Zhao, Chengyu Wang, and Chengfu Huo. Bidirectional context self-distillation for reinforcement learning of skill-based llm agents. *arXiv preprint arXiv:2608.09555*, 2026. URL https://arxiv.org/abs/2608.09555.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Junhao Shen, Teng Zhang, Xiaoyan Zhao, and Hong Cheng. Dynamic skill lifecycle management for agentic reinforcement learning. *arXiv preprint arXiv:2605.10923*, 2026. URL https://arxiv.org/abs/2605.10923.

Yaorui Shi, Yuxin Chen, Zhengxi Lu, Yuchun Miao, Shugui Liu, Qi Gu, Xunliang Cai, Xiang Wang, and An Zhang. Skill1: Unified evolution of skill-augmented agents via reinforcement learning. *arXiv preprint arXiv:2605.06130*, 2026. URL https://arxiv.org/abs/2605.06130.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*, 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. *arXiv preprint arXiv:2010.03768*, 2020. URL https://arxiv.org/abs/2010.03768.

Songjun Tu, Chengdong Xu, Qichao Zhang, Yiwen Ma, Yaocheng Zhang, Linjing Li, Dong Li, Xiangyuan Lan, and Dongbin Zhao. UCOB: Learning to utilize and evolve agentic skills via credit-aware on-policy bidirectional self-distillation. *arXiv preprint arXiv:2606.29502*, 2026a. URL https://arxiv.org/abs/2606.29502.

Songjun Tu, Chengdong Xu, Qichao Zhang, Yaocheng Zhang, Xiangyuan Lan, Linjing Li, Dong Li, and Dongbin Zhao. Dynamic dual-granularity skill bank for agentic rl. *arXiv preprint arXiv:2603.28716*, 2026b. URL https://arxiv.org/abs/2603.28716.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 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. *arXiv preprint arXiv:2604.10674*, 2026. URL https://arxiv.org/abs/2604.10674.

Jinyang Wu, Shuo Yang, Zhengxi Lu, Fan Zhang, Yuhao Shen, Lang Feng, Haoran Luo, Zheng Lian, Shuai Zhang, Zhengqi Wen, and Jianhua Tao. Seed: Self-evolving on-policy distillation for agentic reinforcement learning. *arXiv preprint arXiv:2607.14777*, 2026. URL https://arxiv.org/abs/2607.14777.

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. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*, 2026. URL https://arxiv.org/abs/2602.08234.

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. URL https://arxiv.org/abs/2505.09388.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled rlvr. *arXiv preprint arXiv:2604.03128*, 2026a. 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, and Jianhua Tao. OPID: On-policy skill distillation for agentic reinforcement learning. *arXiv preprint arXiv:2606.26790*, 2026b. URL https://arxiv.org/abs/2606.26790.

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

Siliang Zeng, Quan Wei, William Brown, Oana Frunza, Yuriy Nevmyvaka, Yang Zhao, and Mingyi Hong. Reinforcing multi-turn reasoning in llm agents via turn-level reward design. *arXiv preprint arXiv:2505.11821*, 2025. URL https://arxiv.org/abs/2505.11821.

Yanfei Zhang, Xu Lin, and Chenglin Wu. Stepopsd: Step-aware online preference distillation for agent reinforcement learning. *arXiv preprint arXiv:2605.27140*, 2026a. URL https://arxiv.org/abs/2605.27140.

Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. A survey on the memory mechanism of large language model based agents. *arXiv preprint arXiv:2404.13501*, 2024. URL https://arxiv.org/abs/2404.13501.

Zhiwei Zhang, Yudi Lin, Nikki Lijing Kuang, Linlin Wu, Xiaomin Li, Songtao Liu, and Fenglong Ma. Co-evolving skill generation and policy optimization. *arXiv preprint arXiv:2606.08755*, 2026b. URL https://arxiv.org/abs/2606.08755.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. *arXiv preprint arXiv:2308.10144*, 2023. URL https://arxiv.org/abs/2308.10144.

Jiapeng Zhu, Jianxiang Yu, Yibo Zhao, Chengcheng Han, Qi Gu, Xunliang Cai, Xiang Li, and Weining Qian. Skill0.5: Joint skill internalization and utilization for out-of-distribution generalization in agentic reinforcement learning. *arXiv preprint arXiv:2605.28424*, 2026. URL https://arxiv.org/abs/2605.28424.

Appendix

The appendix follows the same progression as the main paper. Appendix A first positions RICE by asking what evidence methods retain from retrieved context. Appendix B then traces how RICE measures local policy response, accumulates response history, selects supervision by returns, allocates training weight, and updates retrieval availability. Appendix C fixes the evaluation protocol and the boundaries among the reported configurations. Appendix D examines the observed response measurements, their role in retrieval eligibility, their evolution during training, and the reversibility of retrieval availability. Appendix E instantiates the same sequence with concrete supervision, weighting, and availability decisions. Appendix F provides the exact prompts used for interaction, retrieved context, and skill writing.

A Related Work and Comparison with Methods that Reuse Skills **14**

B Constructing and Using Response History **15**

B.1 Retrieval and Response History 15

B.2 Using Returns and Response History for Supervision and Retrieval 17

B.3 Complete Update Sequence 18

C Experimental Protocol and Configurations **19**

C.1 Data, Evaluation, and Baseline Protocol 19

C.2 Shared Base Learner and Skill Learning 20

C.3 Response-Informed Retrieval under Group Relative RL 20

C.4 Controlled Supervision and Retrieval Configurations 21

D Evidence from Response Histories during Training **22**

D.1 Response Separation Across Contexts 22

D.2 Response Conditions for Retrieval Eligibility 23

D.3 Temporal Variation in Response History 23

D.4 Performance over Training across Tasks 24

D.5 Reversibility of Retrieval Availability 24

E Worked Examples of Supervision and Retrieval Decisions **25**

E.1 Returns Determine the Supervision Direction 25

E.2 Response History Determines Relative Training Weight 26

E.3 Outcome Utility and Response History Control Retrieval Availability 26

F Complete Prompt Templates **27**

F.1 Environment Interaction Prompts 27

F.2 Retrieved Skill Context and Writing Prompts 28

A Related Work and Comparison with Methods that Reuse Skills

The comparison follows the distinction established in Section 2: task outcomes describe behavioral quality, whereas local policy response describes how retrieved context changes the current learner. We first review methods that turn experience into supervision and methods that retain skills across policy updates. We then compare whether and how methods with persistent skills retain evidence from the context used at retrieval time.

Hindsight supervision for agent learning. Agent learning methods optimize behavior from grouped returns or local credit within a group (Feng et al., 2025). Self-distillation methods add supervision from the learner's own behavior (Yang et al., 2026a; Wu et al., 2026). Skill-SD and SDAR use skills as additional context for the teacher (Wang et al., 2026; Lu et al., 2026); SEED repeatedly extracts and distills hindsight skills; StepOPSD uses preferences defined at individual steps (Zhang et al., 2026a); and OPID, PCSD, and BCSD provide other ways to select and calibrate supervision across contexts (Yang et al., 2026b; Lv et al., 2026; Pan et al., 2026). These methods study how experience becomes supervision. RICE extends this question to persistent use by recording the local next-token distribution difference caused by retrieved context and using the resulting history in later training and retrieval.

Persistent skills in agent learning. Language agents store experience as verbal feedback, memories, and reusable skills (Shinn et al., 2023; Zhao et al., 2023; Wang et al., 2023; Zhang et al., 2024). SkillRL integrates learned skills into reinforcement learning (Xia et al., 2026); D2Skill maintains separate collections of task and state skills (Tu et al., 2026b); SkillC learns skills autonomously (Lin et al., 2026); SAPO updates skills alongside the policy (Zhang et al., 2026b); and Skill1 combines skill learning with reinforcement learning (Shi et al., 2026). These methods provide several ways to create, store, and reuse experience. RICE studies how observations collected when skills are retrieved can adjust their contribution to later training and their availability for future retrieval.

Positioning RICE against the evaluated baselines. Existing methods generate skills from experience, train one context using behavior observed under another, and update stored skills using credit or utility. RICE additionally records the difference between local next-token distributions with and without retrieved context at the same decision prefix. Returns still select the supervision direction, and the retriever continues to rank available skills by relevance, utility, and exploration. The recorded responses provide complementary evidence for weighting selected training examples and controlling which skills remain eligible for retrieval. UCOB uses the return difference at a shared state to select supervision and update skill utility. RICE-RL uses a group relative actor objective and response-informed retrieval under the configuration in Appendix C.3.

Comparison setup. Table 2 compares seven representative methods that retain skills across training iterations, focusing on how each method evaluates retrieved context and uses that evidence in later learning (Tu et al., 2026b;a; He et al., 2026; Shen et al., 2026; Zhang et al., 2026b; Lin et al., 2026; Zhu et al., 2026).

Comparison axes. Table 2 compares six properties. A cell marked ✓ indicates that the paper explicitly reports the property, △ indicates a related but non-equivalent design, and – indicates that the property is not reported. The properties are defined below.

- **Context at each decision** indicates whether reusable experience is attached to a state, anchor, or decision point rather than only to a task or episode. D2Skill and UCOB explicitly provide this granularity.
- **Task/state comparison** indicates whether the same task or state is evaluated under different context conditions, such as skills versus no skills, an old versus revised skill, or a base context versus one containing a candidate skill.
- **Adaptive supervision** indicates whether evidence about the context changes the learning objective or the contribution of rollout data. An explicit weight for an example or group receives ✓, while selection or filtering without reweighting receives △.

Table 2: How methods with persistent skills evaluate retrieved context and use the resulting evidence. The first four columns summarize how each method uses and updates skills. The fifth indicates whether model-response evidence is retained across retrieval events and used in later training or retrieval. The final column identifies which context is changed by the comparison. ✓ denotes an explicit design, △ a related but non-equivalent design, and – no explicit report. Appendix A provides detailed definitions.

Method	Use of stored skills				Effect of retrieved context	
	Context at each decision	Task/state comparison	Adaptive supervision	Skill updates	Evidence from retrieved context	Compared context
D2Skill	✓	✓	△	✓	–	Joint context
UCOB	✓	✓	✓	△	–	Joint context
ReSkill	–	✓	△	✓	–	Targeted version
SkillC	–	✓	✓	✓	–	Joint context
SAPO	–	✓	△	✓	–	Targeted candidate
SLIM	–	✓	–	✓	–	One skill removed
Skill0.5	–	–	△	△	–	Joint skill set
RICE	✓	✓	✓	✓	✓	**Joint retrieved set**

- **Skill updates** indicates whether the method explicitly creates, revises, retains, pauses, removes, or adds skills. RICE makes changes to retrieval availability reversible by retaining each stored record and allowing it to return.
- **Evidence from retrieved context** records whether observations of how retrieved context changes the model's action or token distribution are retained for use in subsequent training or retrieval. RICE implements this feedback as a running response history.
- **Compared context** identifies what changes between the two conditions. A *joint context* comparison changes the complete retrieved context. A *targeted version* comparison changes one version of a skill, while a *targeted candidate* comparison adds one candidate to an otherwise fixed set. Removing one skill compares the retrieved set with and without that skill. RICE compares the complete retrieved set with its absence at the same decision prefix.

Takeaway. The methods differ in both the context they compare and the evidence they retain. ReSkill, SAPO, and SLIM isolate a skill version, candidate, or individual skill, whereas D2Skill, UCOB, SkillC, Skill0.5, and RICE evaluate a complete context condition. Among the methods in Table 2, task outcomes or validation performance commonly guide later skill use. RICE additionally records the local next-token distribution difference caused by the complete retrieved set, while returns continue to determine the preferred behavior. Each observation is associated with the skills included in that retrieval, producing the response history formalized next.

B Constructing and Using Response History

This section follows one response observation from collection to later use. It first defines retrieval, response measurement, and the statistics accumulated for each participating skill. It then specifies how returns select the supervision direction, how response history determines relative training weight, and how response history combines with outcome utility to update retrieval availability. The final algorithm places these operations in their executed order.

At iteration n, the system state is $\Xi_n = (\theta_n, \mathcal{B}_n, \mathcal{A}_n, \mathcal{H}_n, z_n)$. Here, θ_n denotes the actor parameters, $\mathcal{B}_n$ contains all stored skills, $\mathcal{A}_n \subseteq \mathcal{B}_n$ contains the skills available for retrieval, $\mathcal{H}_n$ stores the response and retrieval histories, and z_n records the recent reliability of skill writing. Each stored skill $k \in \mathcal{B}_n$ also maintains its outcome utility U_k, usage count ν_k, sufficient statistics $\mathbf{s}_k$, and availability history. RICE-Full uses response history to weight the auxiliary distillation loss and combines it with outcome utility to update retrieval availability. RICE-RL uses response history only for retrieval availability under the group relative actor objective specified in Appendix C.3.

B.1 Retrieval and Response History

Response history begins with a retrieved set and a record that preserves both context conditions at the same decision. The retrieval score determines which available skills enter the prompt, and the paired record then supports response measurement after rollout. Repeated measurements update the statistics associated with every skill in the retrieved set.

Rollout records and measurement selection. Each turn records the task group ID, anchor observation, acting context condition, sampled reply, prompt versions with and without skills, reward, and retrieved skill IDs. The two prompt versions share the task, observation, and interaction history of that turn. After rollout, response measurement scans each skill-conditioned trajectory in temporal order and selects at most four active records with valid actions, a stored reply, and both prompt versions available. Both scoring inputs reuse the stored reply, and the response statistics use the distributions predicting its first token.

Partition the bank by skill type $r \in \{\text{task}, \text{state}\}$, and let $\mathbf{q}_r(x) \in \mathbb{R}^d$ and $\mathbf{e}_k \in \mathbb{R}^d$ denote the query and skill embeddings used by the similarity function sim. For candidate k in pool r, retrieval combines embedding relevance, outcome utility U_k, and UCB exploration. Let $\nu_k > 0$ be its usage count initialized with a pseudocount, and let $N_{\text{all}}^{\text{use}} \geq 1$ be the total usage across stored skills. The retrieval score is

$$R_r(k; x) = \alpha \,\mathrm{sim}(\mathbf{q}_r(x), \mathbf{e}_k) + (1 - \alpha) \left(U_k + \xi_{\text{ucb}} \sqrt{\frac{\log N_{\text{all}}^{\text{use}}}{\nu_k}} \right). \tag{5}$$

For each pool, the highest-scoring available skills under $R_r(k; x_t)$ form the retrieved set $S_t \subseteq \mathcal{A}_n$. At each measured decision, $\ell_t^c(v)$ denotes the next-token logit under prompt c. Both prompts are restricted to the shared token set $\Omega_t = \mathrm{TopK}(\ell_t^+, K_D) \cup \mathrm{TopK}(\ell_t^0, K_D) \cup \{v_t\}$. Explicitly, $\hat{p}_t^c(v) = \exp \ell_t^c(v) / \sum_{u \in \Omega_t} \exp \ell_t^c(u)$ for $v \in \Omega_t$, yielding the response magnitude $d_t = D_{\text{JS}}(\hat{p}_t^+ \| \hat{p}_t^0)$ defined in the main text. Distractor probes estimate the threshold τ_I but are neither retrieved nor stored as skills.

Let $\mathcal{G}(t)$ contain the active rollout records sharing decision t's task group ID and anchor observation, regardless of context condition, and center its return as $\widetilde{G}_t = G_t - |\mathcal{G}(t)|^{-1} \sum_{t' \in \mathcal{G}(t)} G_{t'}$. The grouping key is the task group ID and the environment's anchor observation, not the full prompt. We then define $h_t = \widetilde{G}_t [\hat{p}_t^+(v_t) - \hat{p}_t^0(v_t)]$. The pair (d_t, h_t) describes the complete retrieved set at this decision. With $S_t^{\text{meas}} \subseteq S_t$ denoting the stored skills included in the measurement, every $k \in S_t^{\text{meas}}$ records the same observation through

$$\mathbf{s}_k \leftarrow \mathbf{s}_k + (1, d_t, d_t^2, h_t, h_t^2)^\top, \qquad \mathbf{s}_k = (n_k, S_{d,k}, Q_{d,k}, S_{h,k}, Q_{h,k})^\top. \tag{6}$$

$$\boldsymbol{\varpi}_k = (n_k, I_k, H_k)^\top, \qquad I_k = \frac{S_{d,k}}{n_k}, \quad H_k = \frac{S_{h,k}}{n_k}, \quad \mathrm{SE}(H_k) = \sqrt{\frac{\max(Q_{h,k}/n_k - H_k^2, 0)}{n_k}}. \tag{7}$$

The means become available after the first observation, and $\mathrm{SE}(H_k)$ is used only once $n_k \geq n_{\min}$. Outcome utility is updated separately as $U_k \leftarrow (1 - \beta_U) U_k + \beta_U \delta_k$, where δ_k is the task- or trajectory-level outcome credit relative to the skill-free baseline within the same rollout group. The quantities retain distinct roles: I_k determines training weights, whereas (U_k, I_k, H_k) jointly determine whether skill k remains available for retrieval.

Baseline-relative outcome credit. For a mixed rollout group g, let J_g^+ and J_g^0 denote its skill-conditioned and skill-free trajectories, and let $s_i \in \{0, 1\}$ indicate terminal task success. The skill-free baseline and the two pool-specific credits are

$$b_g = \frac{1}{|J_g^0|} \sum_{j \in J_g^0} s_j, \qquad \delta_g^{\text{task}} = \frac{1}{|J_g^+|} \sum_{i \in J_g^+} s_i - b_g, \qquad \delta_i^{\text{state}} = s_i - b_g, \quad i \in J_g^+.$$

Each used task-skill ID receives δ_g^{task} once per group; each used state-skill ID receives δ_i^{state} once per skill-conditioned trajectory. Repeated uses within the same assignment unit are deduplicated, and the utility EMA is applied to each resulting credit assignment.

Measurement unit. Response observations are measured for the complete retrieved set and indexed by each participating skill. Co-retrieved skills share an observation in Eq. (6), so each profile represents context-associated evidence from the retrieval events containing that skill. These profiles support supervision and eligibility decisions at the joint-context level.

Temporal scope. The response profile uses cumulative means over observations from successive policy states. After m new measurements, their combined weight is $m/(n_k + m)$, where n_k is the previous count. New evidence therefore changes a longer history more slowly. Online probe calibration updates the current threshold τ_I, while recorded profiles remain cumulative.

Response thresholds and coverage. Equation (7) defines a nominal dispersion scale from the recorded moments under an independent-observation approximation. The count n_k measures accumulated observations, and $n_{\min}$ sets the coverage requirement. Together with $c_{\mathrm{SE}} = 2$, these quantities define the coverage-aware pausing boundary, while the probe percentile defines the empirical response threshold.

B.2 Using Returns and Response History for Supervision and Retrieval

This subsection applies the accumulated evidence to supervision and future retrieval. Returns first select the supervision direction within each task and anchor group. Response history then determines relative weight among accepted examples and, together with outcome utility, updates which skills remain available for later retrieval.

After rollout, valid supervision records are grouped by task group ID and anchor observation to form $\mathcal{G}(u, \tilde{s})$. Within each group, $\rho^\star = \arg\max_{\rho \in \mathcal{G}(u,\tilde{s})} G_\rho$ supplies the detached teacher. A distinct target record ρ^{tar} supplies a stored prompt under the condition opposite to the source record's acting condition. The target record itself may have been sampled under either condition, since its alternative prompt is cached. We retain the pair $e = (\rho^\star, \rho^{\mathrm{tar}})$ when

$$\Delta_e = G_{\rho^\star} - G_{\rho^{\mathrm{tar}}} > \epsilon_{\mathrm{CB}}, \qquad w_e = \mathrm{clip}(\Delta_e/\tau_{\mathrm{CB}}, 0, w_{\max}). \tag{8}$$

If no distinct record supplies the target prompt, we set $\rho^{\mathrm{tar}} = \rho^\star$, use its cached alternative prompt, and replace $G_{\rho^{\mathrm{tar}}}$ in Eq. (8) with the mean return of the group. All return comparisons use observed rollout outcomes; scoring an alternative prompt generates no new outcome. Accepted pairs are ordered by decreasing return margin and capped at two per group.

The source record provides both the teacher prompt and the reply used as supervision; the target record provides the actor prompt. At token j, both prompts score the same prefix of the source reply on the teacher's fixed top-K_{tok} token set. The prompts may retain different interaction histories within the same anchor group. Here, $q_{e,j}$ is the detached teacher distribution and $p_{e,j}$ is the trainable actor distribution. The detached base weight is $\bar{\omega}_{e,j} = \mathrm{sg}[w_e m_{e,j} a_{e,j} g_{e,j}]$, where $m_{e,j} \in \{0, 1\}$ masks invalid tokens, $a_{e,j} \geq 1$ gives additional weight to action tokens, and $g_{e,j} \in (0, 1)$ is computed from the probability difference between the two prompts on the credited token. The exact normalization of the response weight is

$$\begin{aligned}
\phi_e &\coloneqq \frac{1}{\max\{1, |S_e^{\mathrm{valid}}|\}} \sum_{k \in S_e^{\mathrm{valid}}} \mathrm{clip}\left(\frac{I_k}{\tau_I}, 0, \phi_{\max}\right), \\
\bar{\phi} &\coloneqq \frac{1}{\max\{1, |\mathcal{P}_{\mathrm{meas}}|\}} \sum_{e \in \mathcal{P}_{\mathrm{meas}}} \phi_e, \qquad \mathcal{P}_{\mathrm{meas}} \coloneqq \{e : 0 < \phi_e < \infty\}, \\
\lambda_e &\coloneqq \begin{cases} \phi_e/\bar{\phi}, & e \in \mathcal{P}_{\mathrm{meas}},\ \bar{\phi} > 0, \\ 1, & \text{otherwise.} \end{cases}
\end{aligned} \tag{9}$$

Equation (9) defines the fallback for pairs without valid response history and keeps the mean weight equal to one across measured pairs. With $\bar{Z}_{\mathrm{CB}} = \sum_{e,j} \bar{\omega}_{e,j}$ and $\bar{\lambda}_e = \mathrm{sg}[\lambda_e]$, Eq. (3) is evaluated only when $\bar{Z}_{\mathrm{CB}} > 0$; otherwise the auxiliary loss is zero.

The full actor update is

$$\mathcal{L}_{\mathrm{actor}} = \mathcal{L}_{\mathrm{RL}} + \beta_{\mathrm{ref}} \mathcal{L}_{\mathrm{ref}} + \lambda_{\mathrm{BD}} \mathcal{L}_{\mathrm{RICE}}, \tag{10}$$

$$\theta_{n+\frac{1}{2}} = \mathrm{Step}_{\mathrm{actor}}(\theta_n; \nabla_\theta \mathcal{L}_{\mathrm{actor}}),$$

$$\theta_{n+1} = \mathrm{Step}_{\mathrm{refl}}\left(\theta_{n+\frac{1}{2}}; \nabla_\theta \mathcal{L}_{\mathrm{refl}}\right). \tag{11}$$

Here $\mathcal{L}_{\mathrm{RL}}$ is the environment-return objective, $\mathcal{L}_{\mathrm{ref}}$ is reference-policy regularization, and $\mathcal{L}_{\mathrm{refl}}$ is the subsequent reflection self-training loss.

The following rule identifies skills whose retrieval may be paused:

$$\mathcal{E}^{(n)}_{\text{pause}} \coloneqq \left\{ k \in \mathcal{A}_n : \begin{array}{l} n_k \geq n_{\min}, \\ I_k \leq \tau_I \end{array} \vee \begin{array}{l} U_k \leq f_{r(k)}, \\ H_k < -c_{\text{SE}}\,\text{SE}(H_k) \end{array} \right\}. \tag{12}$$

Here $r(k) \in \{\text{task}, \text{state}\}$ identifies the skill pool and $f_{r(k)}$ is its current utility threshold. Stored skills whose utility is above that threshold form $\mathcal{E}^{\text{re}}_n \subseteq \mathcal{B}_n \setminus \mathcal{A}_n$ and return to the available set first. The remaining candidates for pausing are ordered by increasing I_k, and the number paused in one update is limited by

$$\mathfrak{b}(\Xi_n) = \mathbf{1}\{\text{healthy}(z_n)\} \min\{C, \lfloor c_{\max} N \rfloor, \max(0, L - \lfloor \rho_{\min} N \rfloor)\}. \tag{13}$$

Here $N = |\mathcal{B}_n|$ excludes probes, $L = |\mathcal{A}_n \cup \mathcal{E}^{\text{re}}_n|$ is the number available after reactivation, and $C = |\mathcal{E}^{(n)}_{\text{pause}}|$ is the number eligible for pausing. Let $\widehat{\mathcal{E}}^{\text{pause}}_n$ contain the first $\mathfrak{b}(\Xi_n)$ eligible skills under increasing I_k. The updated available set is $\mathcal{A}^{\text{res}}_n = (\mathcal{A}_n \cup \mathcal{E}^{\text{re}}_n) \setminus \widehat{\mathcal{E}}^{\text{pause}}_n$; skills written later in the iteration may then extend the bank and the available set. Pausing is reversible because every affected skill remains in $\mathcal{B}_n$ with its statistics intact. If the safeguards for skill writing or the size of the available set fail, $\mathfrak{b}(\Xi_n) = 0$ while skill writing and reactivation remain enabled. This update changes which skills can enter future trajectories without altering the actor loss.

B.3 Complete Update Sequence

Algorithm 1 gives the complete RICE-Full procedure. It makes the separation of roles explicit: returns select supervision, response history determines relative training weight, and response history together with outcome utility updates retrieval availability.

Algorithm 1. RICE: Complete Update Sequence

Require: initial state $\Xi_0 = (\theta_0, \mathcal{B}_0, \mathcal{A}_0, \mathcal{H}_0, z_0)$, reference policy π_{ref}, task distribution $\mathcal{D}$
Ensure: final joint state Ξ_T

for iteration $n = 0, \ldots, T-1$ **do**
  Phase I: Collect rollouts with and without skills
  Sample a batch of task groups $\{u\} \sim \mathcal{D}$
  for each task group u **do**
    Retrieve task/state skills from $\mathcal{A}_n$ by relevance, utility, and exploration
    Roll out a balanced group under P^+ and P^0
    During rollout, cache both prompt versions, replies, rewards, anchors, and skill IDs
  end for
  After rollout, compute returns and group records by task group ID and anchor observation
  Phase II: Measure policy response
  for each selected valid decision t from a trajectory with retrieved skills **do**
    Reuse the sampled reply under its two cached prompts; compute (d_t, h_t) at the first token
    Update $\mathbf{s}_k$ for every $k \in S^{\text{meas}}_t$ via Eq. (6)
  end for
  Refresh distractor probes and update the response threshold τ_I
  Phase III: Update retrieval availability
  Compute utility thresholds and reactivate stored skills whose utility exceeds them
  Form $\mathcal{E}^{(n)}_{\text{pause}}$ from the updated response history, outcome utility, and safeguards
  Pause at most $\mathfrak{b}(\Xi_n)$ candidates by increasing I_k; skip pausing if safeguards fail
  Update outcome utilities, write or revise skills, and update the skill-writing reliability state
  Phase IV: Weight supervision and update the actor
  Initialize $\mathcal{P} \leftarrow \emptyset$
  for each group of valid records sharing task u and anchor observation $\tilde{s}$ **do**
    Select $\rho^\star$ by return and target records supplying the opposite prompt condition
    Apply the target fallback in Appendix B.2; retain pairs with $\Delta_e > \epsilon_{\text{CB}}$
  end for
  Reuse source replies; build detached targets and weights; compute $\mathcal{L}_{\text{RICE}}$
  Apply the primary actor optimizer using $\mathcal{L}_{\text{actor}}$ in Eq. (10)
  Apply the independent reflection optimizer to the same actor
  Persist $\Xi_{n+1} = (\theta_{n+1}, \mathcal{B}_{n+1}, \mathcal{A}_{n+1}, \mathcal{H}_{n+1}, z_{n+1})$
end for
return Ξ_T

Update order. After rollout at iteration n, RICE converts the paired observations $\{(d_t, h_t)\}$ into updated sufficient statistics $\mathbf{s}_k$ and response histories $\boldsymbol{\varpi}_k$. It combines these histories with the current outcome utility U_k and safeguard state z_n to produce the set available for later retrieval, $\mathcal{A}_n^{\text{res}}$. The same rollout then updates U_k, writes or revises skills, and produces z_{n+1} for the next iteration. Finally, the response histories produce normalized pair weights λ_e for the primary actor update $\theta_n \mapsto \theta_{n+\frac{1}{2}}$, followed by the reflection update $\theta_{n+\frac{1}{2}} \mapsto \theta_{n+1}$ in Eq. (11). This order ensures that the same response observations inform both the current training weights and the skills available to subsequent trajectories. The next section fixes the experimental conditions under which these updates and their controlled variants are evaluated.

C Experimental Protocol and Configurations

This section defines the protocol needed to interpret the reported evidence. It first specifies datasets, metrics, baseline provenance, and the paired rollout protocol. It then gives the shared learner and skill-learning settings before separating the RICE-RL configuration from the controlled Base, Policy, and Full comparison. Across benchmarks, RICE-Full uses the same response history and rule for updating the available set; only the base learner settings and environment interface change with the task.

Table 3: The same response history and rule for updating the available set are used across the three agent settings. The table reports each benchmark's agent setting, metric, interaction horizon, and limits on context and response length.

Benchmark	Agent setting	Metric	Horizon	Context / response
ALFWorld	Textual embodied household planning	Success rate (%)	50	8192 / 512
WebShop	Goal-conditioned web interaction	Score and success (%)	15	8192 / 512
Search-QA	Single- and multi-hop web search over seven QA sets	Accuracy (%)	4	4096 / 512

C.1 Data, Evaluation, and Baseline Protocol

Data and evaluation. ALFWorld evaluation aggregates episode success over the six task families listed in Table 3; the columns in Table 1 report both individual families and the aggregate. WebShop reports normalized task score and completed-purchase success on the 128-task evaluation set. For Search-QA, Table 1 reports exact answer accuracy on NQ, TriviaQA, PopQA, HotpotQA, 2Wiki, MuSiQue, and Bamboogle, together with their unweighted average. The first three datasets require single-hop search, and the remaining four require multi-hop search. In the skill-conditioned evaluation view, RICE retrieves from the skills currently available and uses the task-specific limits in Table 3. For each single RICE-Full training run in Table 1, we report the best validation result across checkpoints and the skill-conditioned and skill-free views, using success for the interactive tasks and macro-average accuracy for Search-QA.

Baselines and reported results. The main table includes RL methods without skills, methods with persistent skills, and self-distillation methods, covering learning from task outcomes, reusable external skills, and additional supervision derived from the learner's own behavior. † marks values reported by the cited original paper, and * marks values reported by SDAR (Lu et al., 2026). The RICE WebShop results use the 128-task evaluation set described above.

Rollout pairing and response measurement. For each task group, RICE draws rollouts from the same current policy: half use the prompt with retrieved skills, P^+, and half use the prompt without them, P^0. Supervision records are grouped after rollout by task and anchor observation, following Section 3.2. Response measurement reuses a selected record's reply under its two cached prompt versions with the complete decision prefix fixed, as detailed in Appendix B.1. The resulting response histories weight accepted training pairs and, together with outcome utility, determine which skills remain available through Section 3.3.

C.2 SHARED BASE LEARNER AND SKILL LEARNING

Optimization and rollout. RICE uses the same base learner within each benchmark for RICE-Base, RICE-Policy, and RICE-Full. For ALFWORLD and WEBSHOP, the local-credit PPO-style learner uses rollout groups of eight, training/validation batches of $16/128$, actor learning rate 10^{-6}, PPO mini-batch size 64, per-GPU micro-batch size 8, reference-KL coefficient 0.01, entropy coefficient 0, invalid-action penalty 0.1, and discount $\gamma = 0.95$. Rollouts are generated with vLLM using top-$p = 1.0$ and maximum model length 10240; validation uses temperature 0.4 and top-$p = 1.0$. The task-specific prompt, response, and horizon limits appear in Table 3.

Training schedule and compute. All reported training schedules target 250 policy-update steps and use eight NVIDIA A800 GPUs. The ALFWORLD and WEBSHOP runs validate every five steps and retain at most five checkpoints.

SEARCH-QA configuration. SEARCH-QA uses rollout groups of eight, training/validation batches of $128/512$, actor learning rate 10^{-6} with warmup ratio 0.1, PPO mini-batch size 256, per-GPU micro-batch size 16, reference-KL coefficient 0.001, and invalid-action penalty 0.01. Rollout and validation decoding again use top-$p = 1.0$ and temperature 0.4, respectively; the search history length is four. Search rollouts call a local retrieval server. Training runs for 250 steps, with validation at the final step.

Distillation configuration. Returns select the teacher distribution, and response history determines the relative weight of each accepted pair. Distillation is applied at the turn level over response tokens. The three RICE variants use a shared support of $K_{\text{tok}} = 32$ tokens, loss coefficient $\lambda_{\text{BD}} = 0.1$, return margin $\epsilon_{\text{CB}} = 0.05$, margin temperature $\tau_{\text{CB}} = 0.2$, maximum margin weight 2.0, and at most two accepted teacher and student pairs per task and anchor group. The coefficient of the confidence term is 5.0, and action tokens receive weight 2.0.

Skill writing and retrieval configuration. RICE-Base, RICE-Policy, and RICE-Full use the same procedure to write and retrieve skills. Each RICE system retrieves three task skills and three state skills after selecting the ten nearest candidates with sentence-transformer similarity. Retrieval uses relevance weight $\alpha = 0.1$, relevance threshold 0.4, a 4096-token budget for skill context, and UCB exploration scale $\xi_{\text{ucb}} = 0.1$. Outcome utility uses EMA rate $\beta_U = 0.2$. The writer produces one task reflection per rollout group and one state reflection per qualifying contrast with outcome gap at least 0.05, using a 2048-token response limit and temperature/top-$p = 1.0/1.0$. An independent optimizer applies the reflection self-training loss, with weight 1.0, after the primary policy update. In the Qwen3-1.7B runs, reflection self-training begins after a 30-step warmup. Each reflection update samples at most 16 task and 16 state memories whose absolute utility is at least 0.02, and clips pool-normalized utility advantages to $[-2, 2]$. These settings are shared by RICE-Base, RICE-Policy, and RICE-Full.

Response history and retrieval configuration. The following response configuration is used for RICE-Full across the reported benchmarks and model sizes; Appendix C.3 gives the RICE-RL settings. Response measurement evaluates at most four valid decisions per skill trajectory over a shared set of 64 next-token candidates. The response threshold is calibrated online as the 95th percentile of 64 probe observations per update, with an absolute fallback of 0.01. Training weights clip the normalized response magnitude at 2.0 and are renormalized to have mean one across measured pairs. Decisions about retrieval availability require at least 20 response observations and use a utility threshold equal to the median for each skill type, capped at zero. The negative-alignment boundary uses $c_{\text{SE}} = 2$ with the nominal scale in Eq. (7). Each update pauses retrieval for at most 2% of stored skills while keeping at least 30% available. Further pauses are suspended when fewer than 50% of skills remain available or when fewer than half of the three most recent writing attempts succeed. Task-dependent changes are limited to the base learner settings and environment interface.

C.3 RESPONSE-INFORMED RETRIEVAL UNDER GROUP RELATIVE RL

RICE-RL configuration. RICE-RL uses response-informed retrieval without auxiliary distillation. It measures local policy responses under the two prompts at a fixed decision prefix, associates

them with the retrieved skills, and combines the resulting histories with outcome utility to update retrieval availability. The actor is optimized by group relative RL followed by reflection self-training, using the `gigpo` local credit estimator. Both RICE-Full and RICE-RL start with empty task and state skill banks and use the same retrieval settings, including response thresholds and safeguards. Table 4 summarizes the two configurations.

Table 4: RICE-RL uses response history for retrieval without auxiliary distillation. The table compares actor optimization, response measurement, supervision weighting, retrieval, and updates to retrieval availability in the two evaluated RICE configurations.

Component	RICE-Full	RICE-RL
Primary actor update	RL + response-weighted distillation + reference regularization	Group relative RL with local credit (`gigpo`)
Reflection actor update	Sequential self-training	Same
Context comparison	Response history + training pairs	Response history only
Supervision weighting	Mean-normalized response weights	–
Skill retrieval	Relevance + utility + exploration	Same
Retrieval availability	Reversible pausing + safeguards	Same
Initial memory	Empty; skills written online	Same

Same: identical to RICE-Full. – disabled.

Training loop. For every task group, RICE-RL generates mixed skill-conditioned and skill-free trajectories, retrieving from the currently available set for the skill-conditioned branch. Their terminal outcomes supply the baseline-relative utility credits defined in Appendix B.1. An additional scoring pass removes the skills while holding the decision prefix fixed to record the response observations in Eq. (2). This pass updates response statistics, while the RL batch and actor objective remain unchanged. The retrieval update combines response history with outcome utility to control availability. Subsequent retrieval uses the resulting available set and the shared relevance, utility, and exploration score. The actor update uses the `gigpo` group relative advantage estimator and environment return, followed by the independent reflection update in Eq. (11). Thus, response history affects this RL configuration through the skills supplied to later trajectories.

Reported results. Skill1 provides an RL reference in Figure 4(c). Its reported success with retrieved skills is 65.6%, 90.6%, and 92.2% at 1.7B, 3B, and 7B. RICE-RL reaches 90.6%, 96.1%, and 98.4% under the initialization and retrieval settings above. These RL values are the highest recorded skill-conditioned validation success within each run.

C.4 Controlled Supervision and Retrieval Configurations

Table 5 defines the sequential RICE variants used in Section 4.3. All three use task and state skill banks, rollouts with and without retrieved skills, return-directed distillation between the two context conditions, retrieval based on utility and relevance, skill writing, and reflection self-training. RICE-Base omits response history, RICE-Policy uses it to weight accepted training pairs, and RICE-Full also combines it with outcome utility to update retrieval availability. The comparison between Base and Policy varies response weighting in this setup, and the comparison between Policy and Full varies response-informed retrieval control.

Table 5: Comparison of RICE supervision and retrieval configurations. The table reports the components included in RICE-Base, RICE-Policy, and RICE-Full.

Component	RICE-Base	RICE-Policy	RICE-Full
Task/state skill bank	✓	✓	✓
Rollouts under P^+ and P^0 + distillation	✓	✓	✓
Utility-based retrieval + reflection writing	✓	✓	✓
Response history for each skill	–	✓	✓
Mean-normalized response weights	–	✓	✓
Reversible update to retrieval availability	–	–	✓
Safeguards on paused skills	–	–	✓

✓ enabled; – disabled. The first three rows define the shared skill learning procedure. RICE-Policy adds response history and response-based training weights; RICE-Full further combines that history with outcome utility to update retrieval availability under the listed safeguards.

The three configurations isolate the two uses of response history in the order presented in the main paper: Policy adds supervision weighting to Base, and Full adds retrieval availability control to Policy. The next section examines the measurements and retrieval behavior produced by these mechanisms during training.

D Evidence from Response Histories during Training

This section follows the evidential roles assigned to response history. It first tests whether retrieved context produces measurements distinct from sham or distractor context. It then examines how the response conditions affect retrieval eligibility and how cumulative histories compare with recent observations. Finally, it reports performance over training and verifies that paused skills can return to retrieval. Section 4.3 provides the corresponding controlled comparison of Base, Policy, and Full.

D.1 Response Separation Across Contexts

Comparison with sham context. Figure 6 summarizes the paired difference in entropy-normalized response, computed as the response to retrieved context minus that to sham context, over 146 recorded training updates. Its mean is 0.00924, and the 95% percentile bootstrap interval is $[0.00761, 0.01100]$ over 4,000 update-level resamples. Table 6 complements this interval with contiguous block resampling.

Response estimators and settings. Alternative response measures preserve similar orderings, with Spearman rank correlations of 0.973–0.992 across the evaluated measures. Across the nine task–scale settings in Figure 7, retrieved-versus-probe AUC ranges from 0.32 to 0.96. The range shows that separation from distractor probes varies by task and model size rather than holding uniformly across settings.

Block-bootstrap sensitivity. For the series of paired response differences, Table 6 uses 4,000 circular block-bootstrap resamples with block lengths 1, 5, 10, and 20. All four percentile intervals remain above zero, with wider intervals for longer blocks. This analysis tests whether the retrieved-minus-sham difference remains positive when the resampling procedure preserves progressively longer contiguous segments of training.

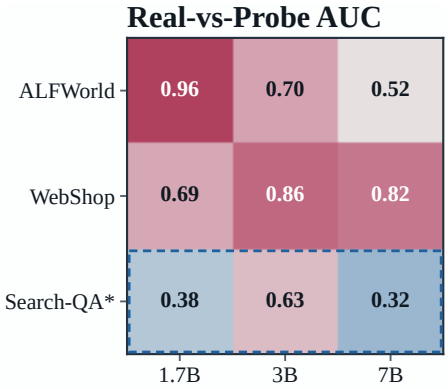


Figure 7: Separation between retrieved skill context and distractor probes varies across settings. The matrix reports AUC for every task and model-size combination and complements the comparison with sham context in Figure 6. An AUC of 0.5 denotes chance-level ranking under the retrieved-versus-probe ordering.

Table 6: Block-bootstrap sensitivity of the retrieved-minus-sham response difference. The table reports 95% percentile intervals on ALFWORLD/Qwen3-1.7B; block length 1 resamples individual updates.

Block length	Lower bound	Upper bound	Interval width
1	0.00760	0.01097	0.00337
5	0.00655	0.01229	0.00574
10	0.00571	0.01326	0.00755
20	0.00457	0.01462	0.01005

D.2 RESPONSE CONDITIONS FOR RETRIEVAL ELIGIBILITY

We examine the saved response profiles from the ALFWORLD and WEBSHOP Qwen3-1.7B runs. The condition counts use the final banks at update 250 and the recorded response thresholds, $\tau_I = 0.00582$ and 0.02017, respectively. Utility floors are computed over all real records within each skill type, as in the retrieval rule. The tabulations evaluate the response conditions on saved profiles before applying availability masks, transition budgets, and health guards.

Contribution of the response conditions. Table 7 partitions the sufficiently measured, weak-utility records into four disjoint groups. Low response alone accounts for $738/1{,}306$ records on ALFWORLD and $697/1{,}371$ on WEBSHOP; negative alignment adds another 23 and 57 records whose response magnitude exceeds the threshold. Conversely, 39.3% and 41.9% of the weak-utility records satisfy neither response condition. Thus the utility threshold leaves a substantial set of records whose response histories determine whether the additional conditions are met.

Table 7: Response conditions among weak-utility records. Final Qwen3-1.7B banks at update 250. N counts real records with $n_k \geq 20$; W is the subset with $U_k \leq f_{r(k)}$. The four disjoint groups show count (percentage of W), using $I_k \leq \tau_I$ and $H_k < -2\,\mathrm{SE}(H_k)$ for the low-response and negative-alignment conditions.

Setting	Pool	N	W	Low only	Alignment only	Both	Neither
ALFWORLD	Task	1,449	803	447 (55.7)	10 (1.2)	8 (1.0)	338 (42.1)
ALFWORLD	State	1,051	503	291 (57.9)	13 (2.6)	24 (4.8)	175 (34.8)
WEBSHOP	Task	1,857	983	521 (53.0)	36 (3.7)	31 (3.2)	395 (40.2)
WEBSHOP	State	789	388	176 (45.4)	21 (5.4)	11 (2.8)	180 (46.4)

D.3 TEMPORAL VARIATION IN RESPONSE HISTORY

At each saved five-update snapshot b, we recover the mean response over the preceding $w \in \{10, 25, 50\}$ updates as $I_k^{(b-w,b]} = (n_k^{(b)} I_k^{(b)} - n_k^{(b-w)} I_k^{(b-w)})/(n_k^{(b)} - n_k^{(b-w)})$. We compare it

with the cumulative $I_k^{(b)}$ on records having at least 20 observations both before and within the window. Figure 8 reports rank correlation, agreement on the low-response condition under the same current τ_I, and the fraction of measured records meeting this support requirement. For the 25-update window, the median snapshot-level rank correlations are 0.164 and 0.046, while the corresponding low-response agreement rates are 96.1% and 78.8%. Median coverage is 18.4% and 19.2%. The cumulative and recent means therefore differ in fine-grained ordering while often producing the same binary response condition on records with continued measurement.

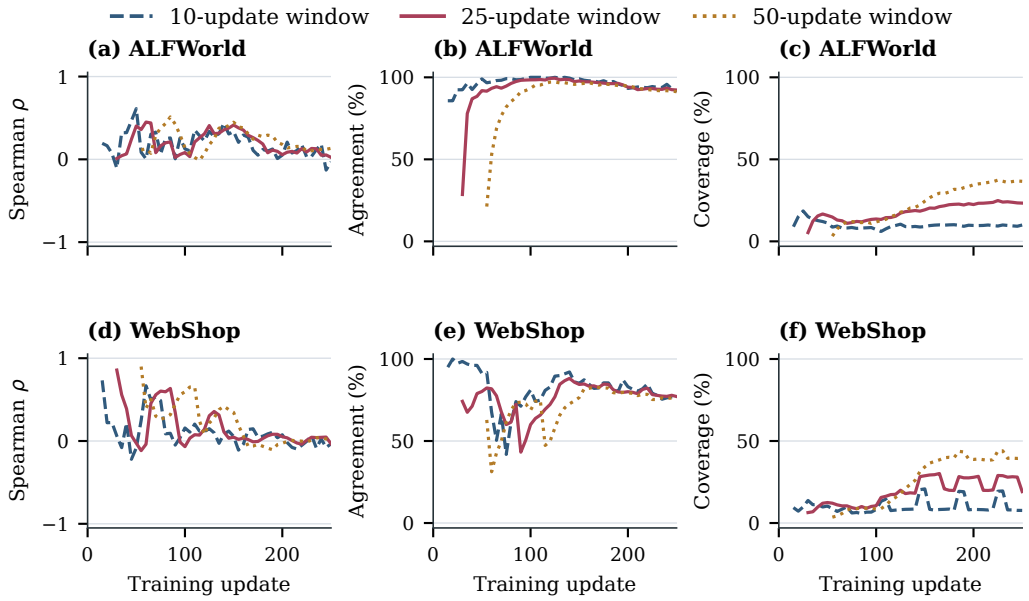


Figure 8: Cumulative and recent response at three temporal resolutions. Qwen3-1.7B on ALFWORLD (top) and WEBSHOP (bottom). Panels (a,d) show Spearman correlation across supported real records; (b,e) show agreement on $I_k \leq \tau_I$; (c,f) show support coverage among all real records with at least 20 cumulative observations. A supported record has at least 20 observations before the window and 20 within it.

D.4 PERFORMANCE OVER TRAINING ACROSS TASKS

Across all three model sizes, performance improves over training in all nine RICE runs, with larger median gains on the interactive benchmarks than on SEARCH-QA (Figure 9). This temporal view complements the endpoint results in Table 1.

D.5 REVERSIBILITY OF RETRIEVAL AVAILABILITY

RICE changes whether stored skills are available for retrieval while retaining every record. Figure 10 shows the retrieval-eligible fraction and cumulative readmissions during training. The recorded readmissions show that the recovery path defined by outcome utility is exercised in the observed runs.

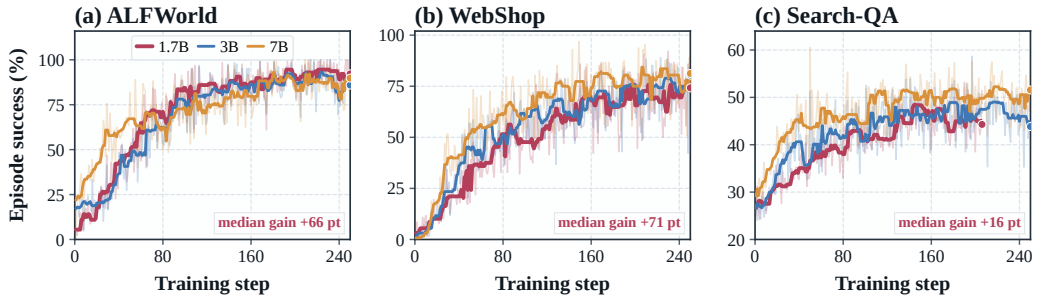


Figure 9: Learning trajectories across three tasks and three model sizes. Each panel shows RICE episode success for the 1.7B, 3B, and 7B runs. Faint curves are raw measurements, and saturated curves are smoothed trends. The callout reports the median improvement from the first 20 to the last 20 training steps across the three model sizes.

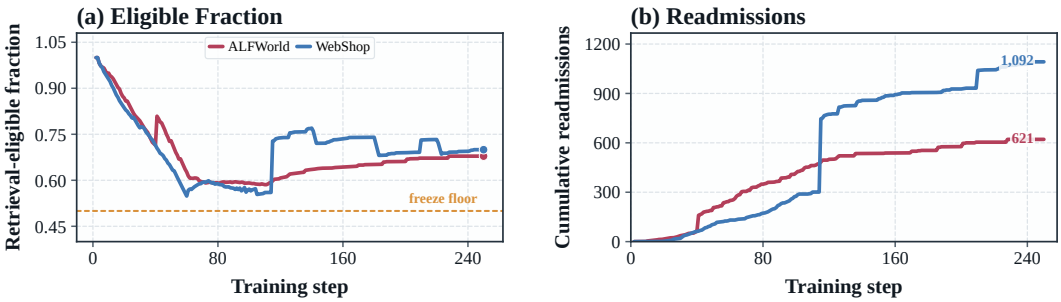


Figure 10: Paused skills can return to retrieval. The panels show the retrieval-eligible fraction and cumulative readmissions during training.

E Worked Examples of Supervision and Retrieval Decisions

The examples follow the three decisions used throughout the main text and appendix. We first show how returns determine the supervision direction and how response history subsequently redistributes weight among accepted pairs. We then illustrate how response magnitude, response alignment, and outcome utility determine retrieval availability. All three cases use the response measurement for the complete retrieved set defined in Appendix B.1.

E.1 Returns Determine the Supervision Direction

The two WebShop rollouts in Case 1 reach the same product page but differ when selecting the required options. The higher-return rollout supplies the detached training target, independently of which context condition produced it.

Case 1: Returns determine the supervision direction

Task	Find a loose-fit short-sleeve top in `a2-blue`, size `medium`, priced below $40.
Shared prefix	Both rollouts issue nearly identical searches, open product `B09NPML43M`, and reach the same option page.
Successful rollout	Select `a2-blue`, select `medium`, then click `buy now`.
Failed rollout	Click `buy now` immediately without resolving the requested color and size.
Behavioral difference	The decisive difference is whether the required color and size are selected before purchase.
RICE update	The successful rollout becomes the teacher record because it has the higher return. Response history is applied after the pair passes the return-margin test.

The return comparison fixes the supervision direction. The next example shows the separate role of response history after accepted pairs have been formed.

E.2 Response History Determines Relative Training Weight

Case 2 records the response-weight allocation at the final update of the ALFWorld/Qwen3-1.7B run. Returns had already selected the teacher in each accepted pair. The profile score ϕ_e then changed the pair's relative contribution through the pair-normalized multiplier λ_e in Eq. (9).

Case 2: Response history redistributes an accepted training batch

Accepted pairs	The update contains 76 pairs; 57 pairs (75%) have valid response histories, while the remaining pairs use the neutral multiplier $\lambda_e = 1$.
Applied weights	Mean normalization gives $\mathrm{mean}(\lambda_e) = 1.0000$. The multipliers range from 0.4214 to 1.4327, with coefficient of variation 0.3535.
Effect on supervision	The return-selected targets are unchanged. Response history reallocates their relative loss contributions while preserving the pair-level mean multiplier.

This batch-level example complements the aggregate weight distributions in Figure 6. It shows the realized separation between direction and allocation: returns choose the target before response history changes how much that accepted pair contributes.

E.3 Outcome Utility and Response History Control Retrieval Availability

Table 8 illustrates four decision paths at the final update of the ALFWorld/Qwen3-1.7B run. At this update, $\tau_I = 0.00582$, the utility floors are $f_{\mathrm{state}} = -0.3233$ and $f_{\mathrm{task}} = -0.0400$, and every displayed record exceeds the minimum of 20 response observations. The table compares each stored value directly with the thresholds in Eq. (4).

Table 8: Four final-state records instantiate the complementary paths through the retrieval availability rule. Each comparison uses the final calibrated threshold and type-specific utility floor. Amber denotes paused retrieval; blue denotes continued availability.

Record and decision path	n_k	U_k vs. $f_{r(k)}$	I_k vs. τ_I	H_k vs. $-2\,\mathrm{SE}(H_k)$	Status
Utility protection Place Object in Container	140	$0.1625 > -0.3233$	$0.0010 \le 0.0058$	$0.0082 > -0.0041$	Available
Low-response branch heat_mug	232	$-0.4380 \le -0.3233$	$0.0007 \le 0.0058$	$-0.0002 > -0.0016$	Paused
Negative-alignment branch Directly place cooled item	388	$-0.4661 \le -0.3233$	$0.0534 > 0.0058$	$-0.0165 < -0.0128$	Paused
Response protection Place Object in Container	224	$-0.0500 \le -0.0400$	$0.0176 > 0.0058$	$0.0024 > -0.0020$	Available

The first row shows the outer utility guard: favorable outcome history protects a low-response record. The two paused records satisfy the low-response and negative-alignment conditions, respectively: the former has $I_k \le \tau_I$ but $H_k \ge -2\,\mathrm{SE}(H_k)$, whereas the latter has $I_k > \tau_I$ but $H_k < -2\,\mathrm{SE}(H_k)$. In the final row, weak utility is insufficient for pausing because neither response condition is met. Their final profiles illustrate how outcome utility and recorded joint-context responses enter the availability rule.

Figure 10 complements these final-state decision paths with aggregate pausing and readmission trajectories over training.

Together, the examples instantiate the same sequence used throughout the paper: returns determine the supervision direction, response history determines relative training weight, and response history combines with outcome utility to control future retrieval.

F COMPLETE PROMPT TEMPLATES

This section provides the exact templates that instantiate the two context conditions and produce the stored skills used by RICE. The interaction templates define the task, observation, and history shared by the prompts with and without retrieved skills. The remaining templates define the inserted skill context and the task-level and state-level skill-writing procedures. Braced placeholders are instantiated with the current task, observation, interaction history, retrieved skills, or trajectory pair.

F.1 ENVIRONMENT INTERACTION PROMPTS

```
ALFWORLD_TEMPLATE_NO_HIS = """
You are an expert agent operating in the ALFRED Embodied Environment.
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.
{thinking_instruction}
Once you've finished your reasoning, you should choose an admissible action for current step
     and present it within <action> </action> tags.
"""

ALFWORLD_TEMPLATE = """
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.
{thinking_instruction}
Once you've finished your reasoning, you should choose an admissible action for current step
     and present it within <action> </action> tags.
"""
```

Listing 1: **ALFWorld initial-decision and history-conditioned action prompts.**

```
WEBSHOP_TEMPLATE_NO_HIS = """
You are an expert autonomous agent operating in the WebShop e-commerce environment.
Your task is to: {task_description}.
Your current observation is: {current_observation}.
Your admissible actions of the current situation are:
[
{available_actions}
].

Now it's your turn to take one action for the current step.
{thinking_instruction}
Once you've finished your reasoning, you should choose an admissible action for current step
     and present it within <action> </action> tags.
"""

WEBSHOP_TEMPLATE = """
You are an expert autonomous agent operating in the WebShop e-commerce 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:
[
{available_actions}
].

Now it's your turn to take one action for the current step.
{thinking_instruction}
Once you've finished your reasoning, you should choose an admissible action for current step
     and present it within <action> </action> tags.
"""
```

Listing 2: **WebShop initial-decision and history-conditioned action prompts.**

```
SEARCH_TEMPLATE_NO_HIS = """
You are an expert agent tasked with answering the given question step-by-step.
```

```
Your question: {task_description}

Now it's your turn to respond for the current step.
{thinking_instruction}
After completing your reasoning, choose only one of the following actions (do not perform
     both):
(1) If you find you lack some knowledge, you can call a search engine to get more external
     information using format: <search> your query </search>.
(2) If you have enough knowledge to answer the question confidently, provide your final
     answer within <answer> </answer> tags, without detailed illustrations. For example, <
     answer>Beijing</answer>.
"""

SEARCH_TEMPLATE = """
You are an expert agent tasked with answering the given question step-by-step.
Your question: {task_description}

Prior to this step, you have already taken {step_count} step(s). Below is the interaction
     history where <search> </search> wrapped your past search queries and <information> </
     information> wrapped the corresponding search results returned by the external search
     engine. History:
{memory_context}

Now it's your turn to respond for the current step.
{thinking_instruction}
After completing your reasoning, choose only one of the following actions (do not perform
     both):
(1) If you find you lack some knowledge, you can call a search engine to get more external
     information using format: <search> your query </search>.
(2) If you have enough knowledge to answer the question confidently, provide your final
     answer within <answer> </answer> tags, without detailed illustrations. For example, <
     answer>Beijing</answer>.
"""
```

Listing 3: **Search-QA initial-decision and history-conditioned action prompts.**

F.2 Retrieved Skill Context and Writing Prompts

```
THINKING INSTRUCTIONS

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

WebShop:
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.

Search-QA:
You should first conduct reasoning process. This process MUST be enclosed within <think> </
     think> tags.


RETRIEVED-SKILL CONTEXT

Relevant category-specific skills:
Task-level skills:
- Skill: {title}
  Principle: {principle}
  When to apply: {when_to_apply}
Step-level skills:
- Skill: {title}
  Principle: {principle}
  When to apply: {when_to_apply}
Use these skills only when they fit the current task category and observation.
```

Listing 4: **Retrieved skill context inserted into the prompt with skills.**

```
ALFWORLD_VLLM_TASK_SKILL_REFLECT_TEMPLATE = """
You are an expert skill writer for the ALFRED Embodied Environment.
Your job is to convert task trajectories into one concise, reusable memory skill.

Task category: {task_category}
Task: {task_description}
```

```
The attempt being evaluated was {success} completed.

{reference_trajectory}

Trajectory of the attempt being evaluated:
{current_trajectory}

Write one high-level skill that would help a future agent solve similar ALFWorld tasks.
The skill should be abstract enough to transfer across different objects and layouts, but
     concrete enough to guide behavior in this environment.

Think about:
1. What reusable principle separates the better trajectory from the worse one?
2. What should the agent remember for future tasks of this kind?
3. When during an episode should this memory be used?

Output ONLY valid JSON with this schema:

{{
"task_success": false,
"task_category": "{task_category}",
"title": "[short title, 2-6 words]",
"principle": "[one reusable rule for future similar tasks]",
"when_to_apply": "[specific situation in which this skill should be applied]",
"evidence": "[one short note grounding the skill in the trajectory comparison]"
}}

Guidelines:
- Do not summarize the whole trajectory.
- Do not write a skill for only this exact scene.
- Avoid exact object ids, location ids, and repeated action strings.
- Keep useful task concepts such as cooling, heating, cleaning, placing, searching,
     containers, receptacles, appliances, or light sources when they matter.
- Write one reusable lesson, not a trajectory summary.
"""

ALFWORLD_VLLM_STATE_SKILL_REFLECT_TEMPLATE = """
You are an expert skill writer for the ALFRED Embodied Environment.
Your job is to convert a better/worse step comparison into one concise state-level memory
     skill.

Task category: {task_category}
Task: {task_description}

The worse step being evaluated was {success} completed.

{reference_trajectory}

Step being evaluated:
{current_trajectory}

Write one state-level skill explaining the local decision principle that made the better step
      preferable.
The skill should help a future agent choose a better next action in a similar observation.

Think about:
1. What observation clue mattered?
2. What local action choice was better?
3. When should this memory be used again?

Output ONLY valid JSON with this schema:

{{
"task_success": false,
"task_category": "{task_category}",
"title": "[short title, 2-6 words]",
"principle": "[one reusable local action-choice rule]",
"when_to_apply": "[specific observation condition where this state skill applies]",
"evidence": "[one short note explaining why the better step had higher return than the worse
     step]"
}}

Guidelines:
- Do not summarize the whole episode.
- Do not copy exact object ids or location ids.
- Keep the skill abstract, but not vague.
- The memory should sound like a reusable agent lesson, not a command copied from the
     trajectory.
"""
```

Listing 5: **ALFWorld task-level and state-level skill-writing prompts.**

```
WEBSHOP_VLLM_TASK_SKILL_REFLECT_TEMPLATE = """
You are an expert skill writer for the WebShop e-commerce environment.
Your job is to convert shopping trajectories into one concise, reusable memory skill.

Task category: {task_category}
Task: {task_description}

The attempt being evaluated was {success} completed.

{reference_trajectory}

Trajectory of the attempt being evaluated:
{current_trajectory}

Write one high-level shopping skill that would help a future agent solve similar WebShop
    tasks.
The skill should be abstract enough to transfer across different products, but concrete
    enough to guide search, comparison, option selection, and purchase decisions.

Think about:
1. What reusable principle separates the better trajectory from the worse one?
2. What should the agent remember for future shopping tasks of this kind?
3. When during a shopping episode should this memory be used?

Output ONLY valid JSON with this schema:

{{
"task_success": false,
"task_category": "{task_category}",
"title": "[short title, 2-6 words]",
"principle": "[one reusable rule for future similar shopping tasks]",
"when_to_apply": "[specific situation in which this skill should be applied]",
"evidence": "[one short note grounding the skill in the trajectory comparison]"
}}

Guidelines:
- Do not summarize the whole trajectory.
- Do not write a skill for only this exact product.
- Avoid exact item ids, prices, product names, and repeated action strings.
- Keep useful shopping concepts such as product category, required attributes, variants,
    color, size, filters, search terms, price constraints, product pages, and Buy Now when
    they matter.
- Write one reusable lesson, not a trajectory summary.
"""

WEBSHOP_VLLM_STATE_SKILL_REFLECT_TEMPLATE = """
You are an expert skill writer for the WebShop e-commerce environment.
Your job is to convert a better/worse page-step comparison into one concise state-level
    memory skill.

Task category: {task_category}
Task: {task_description}

The worse step being evaluated was {success} completed.

{reference_trajectory}

Step being evaluated:
{current_trajectory}

Write one state-level shopping skill explaining the local decision principle that made the
    better step preferable.
The skill should help a future agent choose a better next action on a similar search result
    page, product page, or option-selection page.

Think about:
1. What page or option clue mattered?
2. What local action choice was better?
3. When should this memory be used again?

Output ONLY valid JSON with this schema:

{{
"task_success": false,
"task_category": "{task_category}",
```

```
"title": "[short title, 2-6 words]",
"principle": "[one reusable local shopping action-choice rule]",
"when_to_apply": "[specific observation condition where this state skill applies]",
"evidence": "[one short note explaining why the better step had higher return than the worse
    step]"
}}

Guidelines:
- Do not summarize the whole shopping episode.
- Do not copy exact item ids, product names, prices, or option strings.
- Keep the skill abstract, but not vague.
- The memory should sound like a reusable shopping lesson, not a command copied from the
    trajectory.
"""
```

Listing 6: **WebShop task-level and state-level skill-writing prompts.**

```
SEARCH_VLLM_TASK_SKILL_REFLECT_TEMPLATE = """
You are an expert skill writer for a search-based question answering agent.
Your job is to convert search trajectories into one concise, reusable memory skill.

Task category: {task_category}
Question: {task_description}

The attempt being evaluated was {success} completed.

{reference_trajectory}

Trajectory of the attempt being evaluated:
{current_trajectory}

Write one high-level search skill that would help a future agent answer similar questions.
The skill should be abstract enough to transfer across questions, but concrete enough to
    guide search behavior, evidence use, and final-answer timing.

Think about:
1. What reusable principle separates the better trajectory from the worse one?
2. What should the agent remember before issuing the next query or final answer?
3. When during a search episode should this memory be used?

Output ONLY valid JSON with this schema:

{{
"task_success": false,
"task_category": "{task_category}",
"title": "[short title, 2-6 words]",
"principle": "[one reusable search rule]",
"when_to_apply": "[specific question or evidence condition where this skill applies]",
"evidence": "[one short note grounding the skill in the trajectory comparison]"
}}

Guidelines:
- Do not summarize the whole trajectory.
- Do not write a skill for only this exact question.
- Avoid copying exact retrieved snippets or a full query sequence.
- Keep useful search concepts such as decomposition, exact entity names, attribute lookup,
    query refinement, source evidence, comparison, ambiguity, and answer timing when they
    matter.
- Write one reusable lesson, not a trajectory summary.
"""

SEARCH_VLLM_STATE_SKILL_REFLECT_TEMPLATE = """
You are an expert skill writer for a search-based question answering agent.
Your job is to convert a better/worse step comparison into one concise state-level memory
    skill.

Task category: {task_category}
Question: {task_description}

The worse step being evaluated was {success} completed.

{reference_trajectory}

Step being evaluated:
{current_trajectory}

Write one state-level search skill explaining the local decision principle that made the
    better step preferable.
```

```
The skill should help a future agent choose a better next search query or final answer in a
    similar question/history state.

Think about:
1. What clue in the question or search history mattered?
2. Was it better to search, refine the query, compare evidence, or answer?
3. When should this memory be used again?

Output ONLY valid JSON with this schema:

{{
"task_success": false,
"task_category": "{task_category}",
"title": "[short title, 2-6 words]",
"principle": "[one reusable local search-decision rule]",
"when_to_apply": "[specific question/history condition where this state skill applies]",
"evidence": "[one short note explaining why the better step had higher return than the worse
    step]"
}}

Guidelines:
- Do not summarize the whole episode.
- Do not copy exact retrieved snippets or a full query sequence.
- Keep the skill abstract, but not vague.
- The memory should sound like a reusable search lesson, not a command copied from the
    trajectory.
"""
```

Listing 7: **Search-QA task-level and state-level skill-writing prompts.**