SKILLCRED: ONLINE VALUE ESTIMATION FOR EVOLVING AGENT SKILLS

Anonymous authors
Paper under double-blind review

ABSTRACT

Agent skills encapsulate reusable strategies and procedural knowledge to improve agent task performance. Recent work has begun to evolve such skills continually from historical interaction trajectories. However, skill evolution need not yield monotonic improvements. Newly generated versions may encode erroneous strategies, and blindly replacing an existing version with the latest one can degrade agent performance. To address this problem, we present SKILLCRED, an online framework that maintains competing versions within each skill cluster and estimates their values from trajectories generated during normal task execution. For each skill version invoked in an episode, a local progress assessor evaluates the agent behavior observed across all of its invocation intervals and produces one ternary local progress signal. SKILLCRED uses these signals to update a Dirichlet posterior over each version's local progress probabilities and applies Thompson sampling to route among competing versions while accounting for estimated value and uncertainty. It further uses posterior-based lifecycle control to promote, retain, or archive challengers according to posterior evidence and evaluation budgets. Because all value estimates are learned from ongoing task trajectories, SKILLCRED requires neither additional evaluation rollouts nor a held-out validation set. Experiments with three executor models on ALFWorld and WebShop show that SKILLCRED consistently achieves the highest mean success rate among the compared methods. Averaged across the three executors, its relative success-rate gains over the strongest baseline are 14.5% on ALFWorld and 38.2% on WebShop. The code is available at https://anonymous.4open.science/r/SkillCred.

1 INTRODUCTION

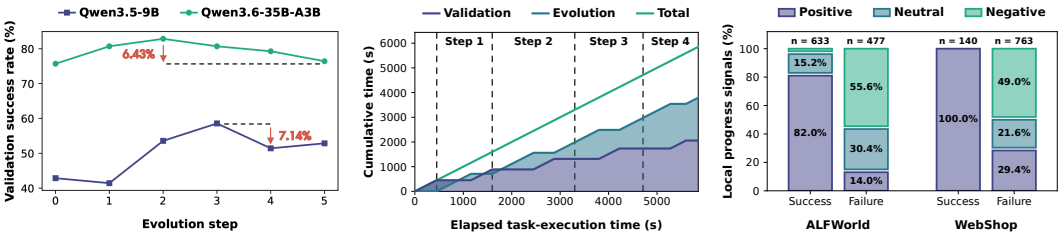
LLM-based agents are increasingly used to solve complex interactive tasks (Wang et al., 2024b; Yao et al., 2023; Liu et al., 2024). Their performance depends on both the capabilities of the underlying model and mechanisms for retaining and reusing experience across interactions (Shinn et al., 2023; Zhao et al., 2024; Wang et al., 2025; Madaan et al., 2023; Zhong et al., 2024; Zhou et al., 2024). Agent skills provide one such mechanism by packaging procedural knowledge and reusable strategies into modules that can be loaded on demand (Anthropic, 2026; Wang et al., 2024a). These skills allow agents to reuse relevant guidance rather than solve every task from scratch.

Even carefully designed skills may be incomplete or suboptimal for a given executor and task distribution, and their limitations may emerge only through continued interaction. Manually revising skills as interaction experience accumulates is difficult to scale. Recent work therefore studies skill evolution, which uses execution trajectories to create or revise reusable skills and adapt the skill library over time (Yang et al., 2026b; Ni et al., 2026; Ouyang et al., 2026a).

However, skill evolution does not guarantee that each new version will outperform its predecessor. As shown in Figure 1a, the validation success rates of successive skill versions generated by Trace2Skill (Ni et al., 2026) fluctuate substantially on ALFWorld (Shridhar et al., 2021), with later versions falling more than 6% below earlier ones under both executor models. This non-monotonicity motivates reliable skill value estimation to identify higher-value versions and limit performance regressions.

Existing mechanisms for evaluating evolved skill versions often rely on additional evaluation rollouts and task-level outcomes. For example, SkillOpt (Yang et al., 2026b) optimizes a single skill

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.



(a) Non-monotonic skill evolution (b) Costly rollout-based validation (c) Outcomes obscure local progress

Figure 1: Motivation for online skill value estimation. (a) Non-monotonic evolution makes reliable skill value estimation necessary. (b) Separate rollout-based validation incurs substantial overhead. (c) Terminal task outcomes provide overly coarse evidence for estimating the value of individual skill versions. n is the number of local progress signals per group.

for a target benchmark and evaluates candidate versions on held-out validation tasks. In our measurement under its reported ALFWorld configuration, validation accounts for 35.2% of the total evolution runtime (Figure 1b). Such evaluation also requires representative validation tasks, which may be difficult to obtain in open-ended or changing environments. In this setting, terminal task outcomes provide a natural signal for evaluating candidate versions. The situation changes when an episode invokes selected skill versions from multiple clusters. A terminal outcome summarizes agent behavior across the entire episode and therefore does not directly distinguish the local progress observed during each version's invocation intervals. As shown in Figure 1c, failed trajectories may still contain positive local progress, while successful ones may contain neutral or negative progress. This motivates estimating the relative values of skill versions within each cluster using local progress evidence from invocation intervals in normal task trajectories, without separate evaluation rollouts.

We propose SKILLCRED, a framework for online skill value estimation that learns from local progress observed during normal task execution. It maintains competing versions within each skill cluster and uses this evidence to estimate their relative values. For each skill version invoked in an episode, a local progress assessor evaluates the observed agent behavior across all of its invocation intervals and produces one ternary local progress signal. These signals update a Dirichlet posterior over each version's local progress probabilities, yielding a posterior over its value. The resulting value posteriors guide Thompson sampling to balance selecting versions with high estimated value and exploring versions whose value remains uncertain. SKILLCRED further uses posterior-based lifecycle control to promote, retain, or archive challengers according to posterior evidence and evaluation budgets, supporting continued evolution within a bounded live set. By obtaining value evidence from ongoing task trajectories, SKILLCRED requires neither additional validation rollouts nor a held-out validation set.

We summarize our contributions as follows:

- We formulate online skill value estimation as estimating the relative value of competing versions within each skill cluster from trajectories generated during normal task execution.
- We introduce SKILLCRED, which combines local progress assessment with Bayesian value estimation, posterior-sampling routing, and lifecycle control within a bounded live set. Its Competitive Retention Value guides challenger retention through posterior-predictive comparisons.
- Experiments with three executor models on ALFWorld and WebShop show that SKILLCRED achieves the highest mean success rate among the compared methods for all three executors on both benchmarks, without additional validation rollouts or a held-out validation set for online value estimation.

2 PRELIMINARIES

2.1 AGENT SKILLS

In episode e, an agent interacts with an environment to complete a task, producing an interaction trajectory τ_e. A *skill* is a reusable unit of procedural knowledge that can be loaded into the agent's

context on demand. Formally, we represent a skill $\mathcal{S}$ as

$$\mathcal{S} = (D, B), \tag{1}$$

where D is the capability description used to determine the skill's applicability, and B is the skill content specifying procedural guidance for task execution. Following the Agent Skills format (Anthropic, 2026), D is stored in the YAML metadata of a `SKILL.md` file, and B forms its Markdown body.

2.2 Skill Evolution

Skill evolution uses historical interaction trajectories to revise existing skills (Yang et al., 2026b; Ni et al., 2026; Ouyang et al., 2026a). The system maintains a skill library $\mathcal{L}$ comprising M skill clusters, each containing versions of the same underlying capability. Formally,

$$\begin{aligned} \mathcal{L} &= \{\mathcal{C}_c\}_{c=1}^{M}, \\ \mathcal{C}_c &= \{\mathcal{S}_c^k = (D_c, B_c^k)\}_{k=1}^{K_c}, \end{aligned} \tag{2}$$

where K_c is the number of versions generated in cluster c. All versions share the capability description D_c and differ in their skill content B_c^k. Given a parent version $\mathcal{S}_c^p$ and a set of associated historical trajectories $\mathcal{T}_c^p$, a skill-evolution operator Φ_{evo} produces a new candidate version as

$$\mathcal{S}_c^{\text{new}} = \Phi_{\text{evo}}(\mathcal{S}_c^p, \mathcal{T}_c^p)\,. \tag{3}$$

The parent index p need not equal K_c. The new candidate receives index $K_c + 1$ when added to the cluster. Appendix A details parent selection, trajectory collection, and candidate generation. However, newly generated versions need not outperform their predecessors, motivating online skill value estimation to guide version selection for subsequent tasks.

3 SkillCred

To estimate the value of evolving skill versions online, we design SkillCred, as illustrated in Figure 2. *Fine-grained online skill value estimation* (§3.1) derives local progress signals from invocation intervals and updates value posteriors. *Posterior-sampling skill routing* (§3.2) balances exploration and exploitation among skill versions, while *predictive skill lifecycle control* (§3.3) promotes, retains, or archives challengers using posterior-predictive comparisons and evaluation-budget criteria.

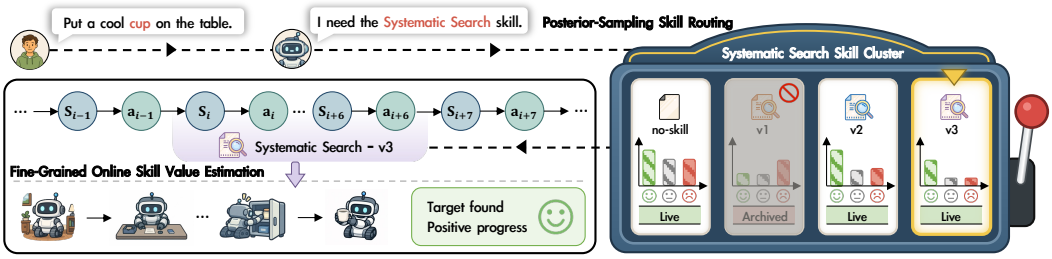
3.1 Fine-Grained Online Skill Value Estimation

As shown in Figure 1c, trajectory-level outcomes provide only coarse supervision because an episode may invoke multiple skills whose invocation intervals exhibit different local progress. Assigning the final outcome to every invoked skill version therefore conflates these local signals and can yield misleading value estimates. As illustrated in Figure 2a, SkillCred derives local progress signals from the agent's behavior within each selected skill version's invocation intervals to obtain finer-grained evidence and uses these signals to estimate the relative value of versions within the corresponding cluster.

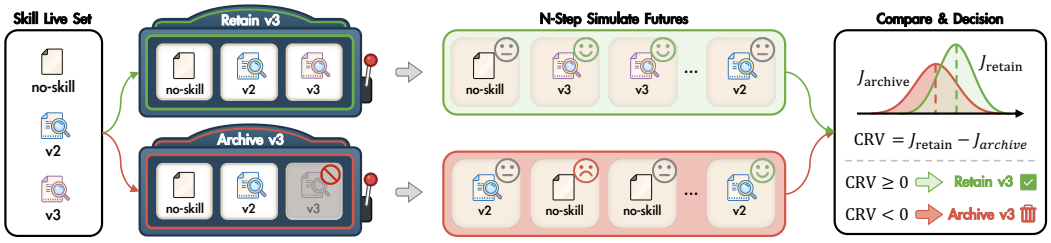
Local progress assessment. For each episode e in which skill version $\mathcal{S}_c^k$ is invoked, a local progress assessor evaluates the agent behavior observed across all of its invocation intervals in the context of the task goal and interaction trajectory τ_e. The assessor may use deterministic environment evidence, model-based judgment, or a combination of both, depending on the available feedback sources. It aggregates the observed behavior into one ternary local progress signal

$$y_e(\mathcal{S}_c^k) \in \{-1, 0, +1\}, \tag{4}$$

where -1, 0, and $+1$ denote negative, neutral, and positive progress, respectively. Negative progress denotes an attributable setback or substantially ineffective behavior, neutral progress denotes no reliable material effect, and positive progress denotes material task progress without an offsetting setback. All invocations of $\mathcal{S}_c^k$ within episode e are aggregated into this single signal. Further details of local progress assessment are provided in Appendix B.



(a) Online value estimation and posterior-sampling routing.



(b) Competitive Retention Value for lifecycle control.

Figure 2: Overview of SKILLCRED. (a) Posterior sampling selects an arm from a routed cluster's live set, and the local progress signal updates its posterior. (b) Paired posterior-predictive simulations estimate each challenger's Competitive Retention Value for retention or archiving.

Dirichlet posterior over local progress probabilities. When an explicit skill version $\mathcal{S}_c^k$ enters routing and lifecycle control, we treat it as an arm and use a as the generic arm index. The cluster-specific no-skill arm has no version index k. For any arm a, let $Y_{c,a} \in \{-1, 0, +1\}$ denote its random local progress signal. For an explicit arm corresponding to $\mathcal{S}_c^k$, the signal $y_e(\mathcal{S}_c^k)$ is an observed realization of $Y_{c,a}$. For each $o \in \{-1, 0, +1\}$, let $\theta_{c,a,o} = \Pr(Y_{c,a} = o)$, where Pr denotes probability, and collect these probabilities in $\boldsymbol{\theta}_{c,a} = (\theta_{c,a,-}, \theta_{c,a,0}, \theta_{c,a,+})$. Let r index the cluster-specific routing rounds, with a new round beginning when $\mathcal{C}_c$ is first routed in an episode. At cluster round r, we maintain the posterior

$$\begin{aligned} \boldsymbol{\theta}_{c,a} \mid \mathcal{D}_{c,r} &\sim \text{Dirichlet}(\boldsymbol{\alpha}_{c,a,r}), \\ \boldsymbol{\alpha}_{c,a,r} &= \alpha_0 \mathbf{1} + \left(n^r_{c,a,-}, n^r_{c,a,0}, n^r_{c,a,+}\right), \end{aligned} \tag{5}$$

where $\alpha_0 > 0$ is the concentration of the symmetric prior, $\mathcal{D}_{c,r}$ contains all valid local progress signals for cluster c through round r, and $n^r_{c,a,o}$ counts the signals assigned to arm a with outcome o. We define the value of arm a as its expected local progress

$$\mu_{c,a} = \mathbb{E}[Y_{c,a}] = \sum_o o\, \theta_{c,a,o} = \theta_{c,a,+} - \theta_{c,a,-}. \tag{6}$$

Because $\mu_{c,a}$ is a deterministic function of $\boldsymbol{\theta}_{c,a}$, the Dirichlet posterior induces a posterior over the value. Arms with similar posterior mean values can therefore have different posterior variances when supported by different amounts of evidence.

3.2 POSTERIOR-SAMPLING SKILL ROUTING

Online skill value estimation requires routing to favor versions with high estimated value while allocating interactions to uncertain versions. We therefore use Thompson sampling with a Dirichlet posterior to balance task performance and evidence collection during online task execution (Thompson, 1933; Chapelle & Li, 2011; Agrawal & Goyal, 2013).

Two-stage routing. Given the current task context, the agent first routes to an applicable cluster $\mathcal{C}_c$ using its shared description D_c and then selects an arm within that cluster. Because cluster eligibility does not depend on the version-specific content B_c^k, versions receive contexts through the same cluster-level routing process, reducing version-specific selection bias and supporting value comparison within the cluster. Once an arm is selected for $\mathcal{C}_c$ in episode e, it remains fixed throughout the

episode. Cluster invocation and switching depend only on D_c and are therefore independent of the selected arm. When the no-skill arm is selected, an unassisted assessment interval begins when $\mathcal{C}_c$ is invoked and ends when the router switches away from it, following the same cluster-level boundaries as an explicit skill invocation but without supplying any skill content. The intervals associated with the selected arm are aggregated into one local progress signal that updates only that arm's posterior.

Dirichlet Thompson sampling. At round r, $\mathcal{A}_{c,r}$ denotes the live set of arms eligible for selection in cluster c, including the arms corresponding to live skill versions and the cluster-specific no-skill arm whenever it is live. The no-skill arm uses the same posterior form as an explicit skill version. We independently sample the local progress probabilities of every $a \in \mathcal{A}_{c,r}$, compute its sampled value, and denote the selected maximizer by $\hat{a}_{c,r}$.

$$\begin{aligned} \widetilde{\boldsymbol{\theta}}_{c,a,r} &\sim \mathrm{Dirichlet}(\boldsymbol{\alpha}_{c,a,r}), \qquad a \in \mathcal{A}_{c,r}, \\ \widetilde{\mu}_{c,a,r} = \widetilde{\theta}_{c,a,r,+} - \widetilde{\theta}_{c,a,r,-}, &\qquad \hat{a}_{c,r} = \arg\max_{a \in \mathcal{A}_{c,r}} \widetilde{\mu}_{c,a,r}. \end{aligned} \tag{7}$$

This rule favors arms with high estimated value while preserving opportunities to evaluate uncertain arms, allowing normal task trajectories to provide evidence for value estimation without additional evaluation rollouts.

The no-skill arm. The no-skill arm is a special arm that is not itself a member of $\mathcal{C}_c$. When selected, its unassisted assessment intervals are evaluated by the same local progress assessor and used to update its cluster-specific posterior. While it is live, it allows routing to learn both whether to use a skill and which version to use. Like an explicit arm, it may be archived and removed from the live set. If its posterior value is comparable to or higher than those of the explicit versions, the skill cluster provides limited incremental value for the current model under the observed task distribution.

3.3 Predictive Skill Lifecycle Control

Skill evolution continually introduces new versions into each cluster, but finite online routing opportunities make it impractical to evaluate an ever-growing live set. Posterior sampling determines which live arm is selected at each round, but not which arms should remain eligible for future routing. We therefore bound concurrent evaluation and use lifecycle control to determine which challengers should be promoted, retained, or archived. The capacity bound leaves the posterior-sampling selection rule unchanged while limiting the cumulative exploration burden of continual version generation.

Bounded lifecycle and promotion. For every cluster and round, we constrain the live set by $|\mathcal{A}_{c,r}| \leq \kappa$, where κ is the concurrent evaluation capacity and includes the no-skill arm whenever it is live. We designate one live arm as the incumbent $b_{c,r}$ and the remaining live arms as challengers. A new evolved version enters as a challenger only when a slot is available. Otherwise, evolution for that cluster waits until lifecycle control creates a vacancy. Each challenger first undergoes a protected initial value estimation period, as detailed in Appendix C.1. For a challenger $a \in \mathcal{A}_{c,r} \setminus \{b_{c,r}\}$, promotion tests whether it should replace the incumbent using

$$p^{\mathrm{prom}}_{c,a,r} = \Pr\left(\mu_{c,a} > \mu_{c,b_{c,r}} \mid \mathcal{D}_{c,r}\right). \tag{8}$$

A challenger is promoted when its evaluation age after the initial value estimation period satisfies $\rho_{c,a,r} \geq R_{\mathrm{prom}}$ and its quadrature approximation $\widehat{p}^{\mathrm{prom}}_{c,a,r}$ exceeds a fixed confidence threshold. Promotion makes it the new incumbent and archives the previous incumbent. Any challenger that remains live without promotion at $\rho_{c,a,r} = R_{\max}$ is archived under the fixed evaluation budget. Appendix C.2 details the deterministic computation of the promotion probability and the lifecycle thresholds.

Competitive Retention Value (CRV). Agent interactions are costly, and evaluating challengers with low posterior value consumes routing opportunities while occupying scarce capacity in the live set. Lifecycle control must therefore distinguish promotion from retention, since a challenger may retain some chance of surpassing the incumbent yet contribute little to exploration when alternatives with higher posterior value are available, whereas an uncertain challenger may still be worth evaluating because further evidence could reveal high value. As illustrated in Figure 2b, we formulate retention as a counterfactual comparison of the future routing policy with and without the challenger.

Let $S_{c,r}$ contain the posterior parameters, incumbent, live set, and lifecycle counters of cluster c. For focal challenger a, let $H_{c,a,r} = R_{\max} - \rho_{c,a,r}$ be its remaining evaluation budget. Given candidate live set $\mathcal{A}$, policy $\pi^H_{\mathcal{A}}$ simulates posterior-sampling routing, posterior updates, promotion, and maximum-age archiving for H rounds, without recursive CRV archiving or new versions, and then uses the terminal incumbent b_H thereafter. Its discounted posterior-predictive value is

$$J_H(S_{c,r}; \mathcal{A}) = \mathbb{E}\left[\sum_{h=0}^{H-1} \gamma^h \mu_{c,\hat{a}_h} + \sum_{h=H}^{\infty} \gamma^h \mu_{c,b_H} \;\middle|\; S_{c,r}, \pi^H_{\mathcal{A}}\right], \tag{9}$$

where $\gamma \in (0,1)$ discounts later cluster rounds. The second sum is the exact infinite discounted tail under terminal commitment and equals $\frac{\gamma^H}{1-\gamma}\mu_{c,b_H}$. We define

$$\mathrm{CRV}_{c,a,r} = J_{H_{c,a,r}}(S_{c,r}; \mathcal{A}_{c,r}) - J_{H_{c,a,r}}(S_{c,r}; \mathcal{A}_{c,r} \setminus \{a\}). \tag{10}$$

The first branch retains a for routing and possible promotion, whereas the second archives it immediately. CRV is therefore a forward-looking, policy-level retention criterion rather than an estimate of the challenger's standalone value. It balances the value of resolving uncertainty about a against the opportunity cost of routing to it, while terminal commitment accounts for the persistent value of the resulting incumbent. Unlike promotion probability, CRV depends on the entire live set because retaining a changes the routing and learning opportunities available to competing arms. It can therefore decrease when a higher-value challenger enters even if the posteriors of a and the incumbent remain unchanged.

We estimate Equation 10 from current posteriors using paired retain and archive simulations with shared randomness. Each return difference is a Monte Carlo sample of this receding-horizon CRV surrogate, and their average estimates CRV without additional environment interactions. The surrogate is recomputed at each review. Sustained negative estimates can trigger early archiving, while challengers above the promotion threshold are protected. Appendices C.3 and C.4 give the simulation and decision details.

4 Experiments

4.1 Experimental Setup

Benchmarks. We conduct experiments on two established benchmarks for interactive agents. ALFWorld (Shridhar et al., 2021) is a text-based environment aligned with the ALFRED embodied AI benchmark, in which agents complete household tasks by navigating and interacting with objects through text commands. WebShop (Yao et al., 2022) is a simulated web-shopping environment in which agents find and purchase products that match user specifications.

Baselines. We compare SkillCred against five representative baselines. No Skill directly solves each task without access to reusable skills, while Human Skill equips the agent with a fixed set of human-authored skills. ReasoningBank (Ouyang et al., 2026b) stores and reuses reasoning experiences distilled from previous trajectories. Trace2Skill (Ni et al., 2026) extracts reusable skills from agent execution traces, and SkillOpt (Yang et al., 2026b) iteratively optimizes skills and accepts updates only when they improve a held-out validation score. Together, these baselines cover unassisted execution, manually specified skills, experience-based reasoning reuse, and automatic skill induction and refinement.

Implementation and evaluation details. We evaluate all methods with Qwen3.5-9B, Qwen3.6-35B-A3B, and DeepSeek-V4-Flash as executor models. We use DeepSeek-V4-Flash for all skill evolution steps, independent of the executor. All methods use the same stream of 500 adaptation episodes, with method-specific update schedules following their official configurations. For ALFWorld, the adaptation episodes come from the train split, and final evaluation uses all 140 tasks in the valid_seen split. For WebShop, the adaptation episodes and the fixed set of 500 evaluation goals are sampled from separate partitions of the synthetic goal pool. No task instance used for adaptation appears in the final evaluation set. After adaptation, we freeze each resulting system and report the mean and standard deviation over 3 seeds. Appendix D provides the complete implementation and evaluation details.

Table 1: Results across three models. Each entry reports the mean ± standard deviation over 3 seeds. ALFWorld reports success rates (SR) by task type and overall average (Avg. SR); WebShop reports average score and SR.

Method	ALFWorld							WebShop	
	Pick	Look	Clean	Heat	Cool	Pick2	Avg. SR	Score	SR
Qwen3.5-9B									
No Skill	73.3 $_{\pm 1.6}$	61.5 $_{\pm 7.7}$	22.2 $_{\pm 3.7}$	8.3 $_{\pm 9.5}$	0.0 $_{\pm 0.0}$	36.1 $_{\pm 10.5}$	35.5 $_{\pm 2.9}$	18.2 $_{\pm 1.7}$	8.9 $_{\pm 1.9}$
Human Skill	88.6 $_{\pm 0.0}$	69.2 $_{\pm 7.7}$	56.8 $_{\pm 2.1}$	25.0 $_{\pm 6.2}$	8.0 $_{\pm 4.0}$	70.8 $_{\pm 7.2}$	56.0 $_{\pm 0.4}$	21.9 $_{\pm 0.9}$	13.1 $_{\pm 0.8}$
ReasoningBank	80.0 $_{\pm 2.9}$	48.7 $_{\pm 4.4}$	59.3 $_{\pm 3.7}$	**45.8** $_{\pm 3.6}$	13.3 $_{\pm 2.3}$	62.5 $_{\pm 4.2}$	54.3 $_{\pm 1.4}$	20.8 $_{\pm 0.6}$	12.9 $_{\pm 0.5}$
Trace2Skill	86.7 $_{\pm 3.3}$	61.5 $_{\pm 0.0}$	54.3 $_{\pm 4.3}$	37.5 $_{\pm 0.0}$	22.7 $_{\pm 2.3}$	41.7 $_{\pm 7.2}$	53.3 $_{\pm 1.1}$	18.9 $_{\pm 1.0}$	11.6 $_{\pm 0.7}$
SkillOpt	82.9 $_{\pm 0.0}$	**76.9** $_{\pm 7.7}$	49.4 $_{\pm 4.3}$	22.9 $_{\pm 9.5}$	41.3 $_{\pm 2.3}$	61.1 $_{\pm 2.4}$	57.9 $_{\pm 0.0}$	24.0 $_{\pm 0.6}$	18.4 $_{\pm 0.7}$
SKILLCRED	**93.3** $_{\pm 3.3}$	71.8 $_{\pm 8.9}$	**90.1** $_{\pm 2.1}$	35.4 $_{\pm 3.6}$	**62.7** $_{\pm 6.1}$	**87.5** $_{\pm 11.0}$	**77.6** $_{\pm 1.5}$	**34.2** $_{\pm 1.0}$	**23.6** $_{\pm 0.9}$
Qwen3.6-35B-A3B									
No Skill	99.0 $_{\pm 1.6}$	74.4 $_{\pm 4.4}$	71.6 $_{\pm 2.1}$	20.8 $_{\pm 9.5}$	68.0 $_{\pm 6.9}$	98.6 $_{\pm 2.4}$	76.9 $_{\pm 3.4}$	35.1 $_{\pm 1.6}$	27.7 $_{\pm 1.0}$
Human Skill	**100.0** $_{\pm 0.0}$	82.1 $_{\pm 4.4}$	82.7 $_{\pm 5.7}$	29.2 $_{\pm 7.2}$	70.7 $_{\pm 12.9}$	98.6 $_{\pm 2.4}$	81.4 $_{\pm 2.1}$	37.1 $_{\pm 0.1}$	29.8 $_{\pm 0.4}$
ReasoningBank	99.0 $_{\pm 1.6}$	76.9 $_{\pm 0.0}$	81.5 $_{\pm 0.0}$	79.2 $_{\pm 3.6}$	58.7 $_{\pm 2.3}$	98.6 $_{\pm 2.4}$	84.0 $_{\pm 1.1}$	53.0 $_{\pm 1.4}$	43.9 $_{\pm 0.7}$
Trace2Skill	98.1 $_{\pm 1.6}$	74.4 $_{\pm 4.4}$	75.3 $_{\pm 2.1}$	79.2 $_{\pm 3.6}$	78.7 $_{\pm 6.1}$	91.7 $_{\pm 4.2}$	84.8 $_{\pm 1.1}$	32.2 $_{\pm 0.2}$	26.8 $_{\pm 0.8}$
SkillOpt	**100.0** $_{\pm 0.0}$	87.2 $_{\pm 4.4}$	85.2 $_{\pm 0.0}$	70.8 $_{\pm 7.2}$	88.0 $_{\pm 6.9}$	**100.0** $_{\pm 0.0}$	90.5 $_{\pm 0.4}$	44.5 $_{\pm 1.0}$	37.6 $_{\pm 1.6}$
SKILLCRED	**100.0** $_{\pm 0.0}$	**97.4** $_{\pm 4.4}$	**100.0** $_{\pm 0.0}$	**83.3** $_{\pm 3.6}$	**93.3** $_{\pm 2.3}$	**100.0** $_{\pm 0.0}$	**96.7** $_{\pm 0.8}$	**63.3** $_{\pm 2.2}$	**55.8** $_{\pm 2.0}$
DeepSeek-V4-Flash									
No Skill	**100.0** $_{\pm 0.0}$	94.9 $_{\pm 4.4}$	77.8 $_{\pm 3.7}$	33.3 $_{\pm 13.0}$	78.7 $_{\pm 2.3}$	**100.0** $_{\pm 0.0}$	83.8 $_{\pm 1.5}$	10.9 $_{\pm 0.5}$	9.9 $_{\pm 0.5}$
Human Skill	99.0 $_{\pm 1.6}$	**100.0** $_{\pm 0.0}$	87.7 $_{\pm 2.1}$	54.2 $_{\pm 3.6}$	82.7 $_{\pm 2.3}$	98.6 $_{\pm 2.4}$	88.8 $_{\pm 0.8}$	16.7 $_{\pm 0.6}$	15.7 $_{\pm 0.3}$
ReasoningBank	98.1 $_{\pm 1.6}$	**100.0** $_{\pm 0.0}$	76.5 $_{\pm 4.3}$	68.8 $_{\pm 0.0}$	77.3 $_{\pm 2.3}$	**100.0** $_{\pm 0.0}$	87.4 $_{\pm 1.1}$	23.5 $_{\pm 0.2}$	22.1 $_{\pm 0.5}$
Trace2Skill	98.1 $_{\pm 1.6}$	**100.0** $_{\pm 0.0}$	69.1 $_{\pm 5.7}$	72.9 $_{\pm 9.5}$	69.3 $_{\pm 2.3}$	94.4 $_{\pm 2.4}$	84.0 $_{\pm 2.1}$	8.9 $_{\pm 1.3}$	8.8 $_{\pm 1.1}$
SkillOpt	99.0 $_{\pm 1.6}$	**100.0** $_{\pm 0.0}$	97.5 $_{\pm 2.1}$	87.5 $_{\pm 0.0}$	84.0 $_{\pm 0.0}$	**100.0** $_{\pm 0.0}$	95.0 $_{\pm 0.7}$	33.6 $_{\pm 2.1}$	29.6 $_{\pm 1.7}$
SKILLCRED	**100.0** $_{\pm 0.0}$	**100.0** $_{\pm 0.0}$	**100.0** $_{\pm 0.0}$	**91.7** $_{\pm 3.6}$	**90.7** $_{\pm 4.6}$	**100.0** $_{\pm 0.0}$	**97.4** $_{\pm 1.1}$	**51.1** $_{\pm 1.1}$	**47.1** $_{\pm 1.1}$

4.2 MAIN RESULTS

ALFWorld. Table 1 shows that SKILLCRED achieves the highest average success rate with all three executors. With Qwen3.5-9B, it improves on the strongest baseline from 57.9 to 77.6, a 34.0% relative gain. The absolute gains are 6.2 and 2.4 points with Qwen3.6-35B-A3B and DeepSeek-V4-Flash, respectively, consistent with their smaller remaining headroom. Across task types, SKILLCRED achieves the best or tied-best result in 16 of the 18 model and task type combinations, indicating that the improvements extend across diverse household tasks.

WebShop. On WebShop, SKILLCRED achieves the highest score and success rate with all three executors. Relative success-rate gains over the strongest baseline are 28.3%, 27.1%, and 59.1% for Qwen3.5-9B, Qwen3.6-35B-A3B, and DeepSeek-V4-Flash, respectively. The corresponding score improvements are 10.2, 10.3, and 17.5 points. The largest gain occurs with DeepSeek-V4-Flash, whose success rate increases from 29.6 to 47.1. Together, these results show gains in both graded task performance and complete task success.

4.3 FURTHER ANALYSIS

Agreement with held-out evaluation. We examine online value estimates for the ALFWorld `systematic-search` cluster with Qwen3.5-9B. Following the validation paradigm of SkillOpt (Yang et al., 2026b), we evaluate each version with at least 10 valid local progress signals on held-out tasks while fixing all other ALFWorld clusters to their final incumbents. This post-hoc reference is unavailable to SKILLCRED during online execution.

Figure 3a shows that online posterior mean value estimates broadly track held-out performance. Because versions accumulate evidence on different task samples, we assess agreement on lifecycle decisions rather than exact global rankings. Across all six skill clusters on both benchmarks, 28/30 completed decisions agree with the held-out reference, including 14/14 promotions and 14/16



(a) Value-estimation diagnostic. (b) Online performance. (c) Exploration efficiency.

Figure 3: Online value estimation and performance during skill evolution with Qwen3.5-9B on ALFWorld. (a) Online posterior mean value estimates versus SkillOpt-style held-out performance for the `systematic-search` cluster; bubble area indicates valid local progress signals. (b) Online performance of SKILLCRED, Hindsight, and Latest. Faint curves show iteration-level results and solid curves show smoothed trends. (c) Cumulative invocations of never-promoted challengers with and without CRV-based early archiving.

Table 2: Ablation results using Qwen3.5-9B. LPA denotes Local Progress Assessment.

Method	ALFWorld							WebShop	
	Pick	Look	Clean	Heat	Cool	Pick2	Avg. SR	Score	SR
No Skill	73.3 $\pm$1.6	61.5 $\pm$7.7	22.2 $\pm$3.7	8.3 $\pm$9.5	0.0 $\pm$0.0	36.1 $\pm$10.5	35.5 $\pm$2.9	18.2 $\pm$1.7	8.9 $\pm$1.9
Latest	73.3 $\pm$4.4	**92.3** $\pm$7.7	18.5 $\pm$6.4	6.2 $\pm$0.0	1.3 $\pm$2.3	56.9 $\pm$2.4	41.2 $\pm$2.2	17.1 $\pm$0.3	11.8 $\pm$0.7
w/o LPA	88.6 $\pm$0.0	74.4 $\pm$4.4	35.8 $\pm$10.7	29.2 $\pm$7.2	22.7 $\pm$4.6	77.8 $\pm$8.7	56.7 $\pm$2.7	26.1 $\pm$1.3	18.4 $\pm$1.2
w/o CRV	89.5 $\pm$1.6	79.5 $\pm$4.4	79.0 $\pm$5.7	**45.8** $\pm$3.6	58.7 $\pm$4.6	70.8 $\pm$12.5	72.9 $\pm$1.2	30.9 $\pm$0.9	18.5 $\pm$0.6
SKILLCRED	**93.3** $\pm$3.3	71.8 $\pm$8.9	**90.1** $\pm$2.1	35.4 $\pm$3.6	**62.7** $\pm$6.1	**87.5** $\pm$11.0	**77.6** $\pm$1.5	**34.2** $\pm$1.0	**23.6** $\pm$0.9

archives (Appendix F.1). This agreement supports the use of online local progress signals for within-cluster value estimation and lifecycle control.

Online performance during evolution. We next examine whether the evidence collection required for online value estimation incurs a substantial performance cost during skill evolution, using Qwen3.5-9B on ALFWorld. We compare SKILLCRED with two reference trajectories. *Latest* performs evolution without SKILLCRED, always continuing from the most recently generated version. *Hindsight* is an idealized reference that retrospectively removes the evaluation delay by adopting each version from the moment it is generated if that version is eventually promoted by SKILLCRED. Figure 3b shows that SKILLCRED closely tracks Hindsight while substantially outperforming Latest, indicating that online value estimation incurs only a modest delay cost while preventing regressive updates from propagating through subsequent evolution.

Exploration efficiency of CRV. To quantify the exploration savings from CRV-based early archiving, we compare the full lifecycle policy against a matched reference policy that follows the same routing, promotion, and maximum evaluation-age rules but retains challengers until promotion or the fixed evaluation limit. As shown in Figure 3c, CRV-based early archiving reduces the cumulative number of invocations spent on challengers that are ultimately never promoted from 954 to 658, saving 296 invocations, or 31.0%. This reduction shows that the lifecycle policy can stop evaluating non-promoted challengers earlier, freeing routing opportunities and capacity in the live set for subsequent skill evolution.

4.4 ABLATION STUDIES

Table 2 compares SKILLCRED with No Skill, Latest, and two component ablations. *w/o LPA* replaces local progress signals with binary terminal task outcomes, mapping success to $+1$ and failure to -1, while *w/o CRV* disables early archiving and retains each challenger until promotion or the fixed maximum evaluation age.

Removing LPA reduces ALFWorld average success rate from 77.6 to 56.7, compared with 72.9 without CRV. On WebShop, success rate falls from 23.6 to 18.4 and 18.5, respectively. The LPA

results support assessments over invocation intervals as evidence for within-cluster value estimation, since terminal outcomes conflate progress across multiple invoked skills with subsequent agent behavior. The CRV results complement Figure 3c, showing performance gains alongside reduced exploration of never-promoted challengers. Early archiving also releases capacity in the live set, allowing subsequent versions to enter evaluation sooner when evolution would otherwise wait for a vacancy. The gains over Latest further support value-based version management during continual skill evolution. *(See Appendix F for additional experimental results.)*

5 Related Work

Experience Reuse and Agent Skills. LLM agents can improve from past interactions by externalizing useful experience as reusable knowledge, including reflections, episodic memories, reasoning strategies, workflows, tools, and executable skills (Shinn et al., 2023; Zhao et al., 2024; Wang et al., 2025; Ouyang et al., 2026b; Cai et al., 2024; Wang et al., 2024a; Madaan et al., 2023; Zhong et al., 2024; Zhou et al., 2024). A parallel line of work augments language models with callable tools or pretrained action repertoires, emphasizing selection and composition through descriptions or API documentation (Schick et al., 2023; Shen et al., 2023; Patil et al., 2024; Qin et al., 2024; Ahn et al., 2022; Huang et al., 2022). Recent work increasingly organizes reusable procedural knowledge into structured skill libraries that can be retrieved, composed, and reused across tasks without modifying the underlying model (Ni et al., 2026; Ouyang et al., 2026a; Zhou et al., 2026; Ding et al., 2026). Other approaches jointly optimize skill-augmented policies through reinforcement learning (Shi et al., 2026; Xia et al., 2026; Wang et al., 2026; Yang et al., 2026a). Together, these directions establish skill-based adaptation as an effective mechanism for reusing interaction experience. Our work focuses on a complementary problem that becomes important when skills are continually updated, namely how to determine the value of competing skill versions during ongoing interaction.

Skill Evolution and Evaluation. Skill evolution methods use task outcomes, execution feedback, replay, or behavioral verification to guide skill refinement and curation (Ouyang et al., 2026a; He & Yang, 2026; Yang et al., 2026a). Closely related evaluation methods differ in both what they assess and how they obtain evidence. SkillOpt (Yang et al., 2026b) evaluates candidate revisions on held-out validation tasks. SkillCoach (Zhu et al., 2026) uses process-level rubrics to assess whether agents select and follow skills appropriately, focusing on the agent's skill-use process rather than the relative value of competing versions. SkillAudit (Gao et al., 2026) revises individual skills through paired executions with and without the candidate skill, obtaining targeted diagnostic evidence through dedicated comparisons. SkillCred addresses online skill value estimation during ongoing task execution. It estimates the relative value of competing versions within a skill cluster from local progress signals in normal task trajectories, without dedicated evaluation rollouts or a held-out validation set. These estimates and their posterior uncertainty guide routing and lifecycle control as new versions become available.

Trajectory Evaluation and Posterior Sampling. Intermediate progress metrics and process supervision provide finer-grained feedback than terminal outcomes alone (Ma et al., 2024; Uesato et al., 2022; Lightman et al., 2024; Wang et al., 2024d; Zhuge et al., 2025). LLM-based evaluators make such assessments scalable, with rubric design and evaluator bias as key considerations (Zheng et al., 2023; Liu et al., 2023; Wang et al., 2024c; Lan et al., 2024). Posterior sampling and best-arm identification study how to allocate observations among uncertain alternatives (Thompson, 1933; Chapelle & Li, 2011; Agrawal & Goyal, 2013; Kaufmann et al., 2016; Garivier & Kaufmann, 2016). These lines of work inform our local progress assessment and adaptive allocation of routing opportunities.

6 Conclusion

We introduced SkillCred, a framework for online skill value estimation that uses local progress signals from normal task trajectories to guide posterior-sampling routing and lifecycle control. Across ALFWorld and WebShop with three executors, it achieves the highest mean success rates among the compared methods without dedicated validation rollouts. Further analyses show agreement with post-hoc held-out lifecycle decisions and reduced exploration of never-promoted challengers, supporting online value estimation as a practical basis for continual skill evolution.

AI USE STATEMENT

Generative AI models were used as components of the experimental methodology. Specifically, they served as executor models, the skill-evolution model, and, where specified, local progress assessors, as described in Section 4.1 and Appendices A and B.2. Separately, generative AI tools assisted with software implementation, including drafting, debugging, and editing experimental code, and with the analysis and interpretation of experimental results. The authors independently developed the research methodology and experimental design, and configured and executed all reported experiments. Generative AI tools also assisted with manuscript drafting, language editing, and formatting. We reviewed and tested all AI-assisted code, independently checked AI-assisted analyses against the underlying experimental records, verified the reported claims and quantitative results, and take responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

Section 4.1 specifies the benchmarks, baselines, backbone models, and evaluation protocol. Appendices A–C document skill evolution, local progress assessment, and lifecycle control, including the assessor prompt and decision rules. Appendix D reports the implementation settings, hyperparameters, random seeds, task splits, and adaptation procedure, while Appendix E provides the initial skill library. Appendix F reports the held-out comparisons, evaluation overhead, and robustness analyses. The main experimental results report the mean and standard deviation over three seeds. The anonymized code repository is available at `https://anonymous.4open.science/r/SkillCred`.

REFERENCES

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, pp. 127–135. PMLR, 2013.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Anthropic. Agent Skills. Claude Platform Documentation, 2026. URL `https://platform.claude.com/docs/en/agents-and-tools/agent-skills/overview`.

Tianle Cai, Xuezhi Wang, Tengyu Ma, Xinyun Chen, and Denny Zhou. Large language models as tool makers. In *International Conference on Learning Representations*, pp. 54067–54089, 2024.

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. *Advances in Neural Information Processing Systems*, 24, 2011.

Kexin Ding, Yang Zhou, Can Jin, Feng Tong, Mu Zhou, and Dimitris N Metaxas. Agent skill evaluation and evolution: Frameworks and benchmarks. *arXiv preprint arXiv:2606.11435*, 2026.

Haowen Gao, Haoran Chen, Can Wang, Shasha Guo, Liang Pang, Zhaoyang Liu, Huawei Shen, and Xueqi Cheng. Skillaudit: Ground-truth-free skill evolution via paired trajectory auditing. *arXiv preprint arXiv:2606.14239*, 2026.

Aurélien Garivier and Emilie Kaufmann. Optimal best arm identification with fixed confidence. In *Conference on Learning Theory*, pp. 998–1027. PMLR, 2016.

Yu He and Weikai Yang. Skillcommit: Evolving agent skills through behaviorally validated scope expansion. *arXiv preprint arXiv:2608.15165*, 2026.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pp. 9118–9147. PMLR, 2022.

Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 17(1):1–42, 2016.

Tian Lan, Wenwei Zhang, Chen Xu, Heyan Huang, Dahua Lin, Kai Chen, and Xian-Ling Mao. Criticeval: Evaluating large-scale language model as critic. *Advances in Neural Information Processing Systems*, 37:66907–66960, 2024.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, pp. 39578–39601, 2024.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. AgentBench: Evaluating LLMs as Agents. In *International Conference on Learning Representations*, pp. 52989–53046, 2024.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 2511–2522, 2023.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. Agentboard: An analytical evaluation board of multi-turn llm agents. *Advances in Neural Information Processing Systems*, 37:74325–74362, 2024.

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

Jingwei Ni, Yihao Liu, Xinpeng Liu, Yutao Sun, Mengyu Zhou, Pengyu Cheng, Dexin Wang, Erchao Zhao, Xiaoxi Jiang, and Guanjun Jiang. Trace2Skill: Distill trajectory-local lessons into transferable agent skills. *arXiv preprint arXiv:2603.25158*, 2026.

Siru Ouyang, Jun Yan, Yanfei Chen, Rujun Han, Zifeng Wang, Bhavana Dalvi Mishra, Rui Meng, Chun-Liang Li, Yizhu Jiao, Kaiwen Zha, et al. SkillOS: Learning skill curation for self-evolving agents. *arXiv preprint arXiv:2605.06614*, 2026a.

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

Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. Gorilla: Large language model connected with massive apis. *Advances in Neural Information Processing Systems*, 37:126544–126565, 2024.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *International Conference on Learning Representations*, pp. 9695–9717, 2024.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face. *Advances in Neural Information Processing Systems*, 36:38154–38180, 2023.

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.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *International Conference on Learning Representations*, 2021.

William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 1933.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

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. *Transactions on Machine Learning Research*, 2024a. ISSN 2835-8856.

Jiongxiao Wang, Qiaojing Yan, Yawei Wang, Yijun Tian, Soumya Smruti Mishra, Zhichao Xu, Megha Gandhi, Panpan Xu, and Lin Lee Cheong. Reinforcement learning for self-improving agent with skill library. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1529–1550, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.69. URL https://aclanthology.org/2026.acl-long.69/.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of computer science*, 18(6):186345, 2024b.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, et al. Large language models are not fair evaluators. In *Proceedings of the 62nd annual meeting of the Association for Computational Linguistics (volume 1: Long papers)*, pp. 9440–9450, 2024c.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, 2024d.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In *International Conference on Machine Learning*, pp. 63897–63911, 2025.

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, 2026. URL https://arxiv.org/abs/2602.08234.

Shidong Yang, Ziyu Ma, Tongwen Huang, Xucong Wang, Renda Li, Yiming Hu, Yong Wang, and Xiangxiang Chu. Skillforge: Evolving verifiable skills for reinforcement learning agents. *arXiv preprint arXiv:2608.24747*, 2026a.

Yifan Yang, Ziyang Gong, Weiquan Huang, Qihao Yang, Ziwei Zhou, Zisu Huang, Yan Li, Xuemei Gao, Qi Dai, Bei Liu, et al. SkillOpt: Executive strategy for self-evolving agent skills. *arXiv preprint arXiv:2605.23904*, 2026b.

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.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19632–19642, 2024. doi: 10.1609/aaai.v38i17.29936.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623, 2023.

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

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning, acting, and planning in language models. In *International Conference on Machine Learning*, pp. 62138–62160. PMLR, 2024.

Huichi Zhou, Siyuan Guo, Anjie Liu, Zhongwei Yu, Ziqin Gong, Bowen Zhao, Zhixun Chen, Menglong Zhang, Yihang Chen, Jinsong Li, et al. Memento-skills: Let agents design agents. *arXiv preprint arXiv:2603.18743*, 2026.

Jiayin Zhu, Kelong Mao, Yudong Guo, Dengbo He, Sulong Xu, Simiu Gu, and Yutao Yue. Skill-coach: Self-evolving rubrics for evaluating and enhancing agentic skill-use. *arXiv preprint arXiv:2607.01874*, 2026.

Mingchen Zhuge, Changsheng Zhao, Dylan R Ashley, Wenyi Wang, Dmitrii Khizbullin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, et al. Agent-as-a-judge: evaluate agents with agents. In *International Conference on Machine Learning*, pp. 80569–80611, 2025.

Appendix Overview

- Appendix A Skill Evolution Procedure
- Appendix B Local Progress Assessment Details
- Appendix C Lifecycle Control Details
- Appendix D Implementation and Evaluation Details
- Appendix E Initial Skill Library
- Appendix F Additional Experimental Results
- Appendix G Limitations and Future Work

A Skill Evolution Procedure

General formulation. A common pattern in recent skill-evolution methods is to revise reusable skill content from accumulated interaction experience (Yang et al., 2026b; Ni et al., 2026; Ouyang et al., 2026a). Given a parent skill version and a set of associated trajectories, an LLM-based evolver extracts reusable evidence from successful and unsuccessful behavior and produces a revised version for subsequent execution. We instantiate the skill-evolution operator in Equation 3 by selecting a parent version p and constructing its associated trajectory set $\mathcal{T}_c^p$. Importantly, p need not be the most recently generated version; version indices indicate creation order rather than a strict parent chain.

Parent selection and evolution trigger. We deliberately use a simple instantiation of Φ_{evo} so that the gains of SkillCred primarily reflect online value estimation and version management rather than a specialized candidate generator. Evolution for cluster c is attempted only when the cluster has available live-set capacity and the required trajectory evidence is available. Among eligible explicit skill versions currently in the live set, we select as the parent the version with the highest empirical local-progress value,

$$\bar{\mu}_{c,k} = \frac{1}{N_{c,k}} \sum_{e=1}^{N_{c,k}} y_e(\mathcal{S}_c^k), \qquad p = \arg\max_{k \in \mathcal{P}_c} \bar{\mu}_{c,k}, \tag{11}$$

where $\mathcal{P}_c$ contains sufficiently evaluated explicit live versions. The cluster-specific no-skill arm is never used as an evolution parent. Thus, unlike a latest-version evolution policy, SkillCred can continue evolution from an older version when its accumulated online evidence indicates higher value.

Trajectory construction. For the selected parent $\mathcal{S}_c^p$, we construct $\mathcal{T}_c^p$ from eight recent eligible episodes in which that version was selected for cluster c. Each trajectory contains at least three action steps within the parent's marked invocation intervals. We use positive and negative anchor cases together with up to two neutral boundary cases. A positive anchor is a successful episode with $y_e(\mathcal{S}_c^p) = +1$, whereas a negative anchor is a failed episode with $y_e(\mathcal{S}_c^p) = -1$. A neutral case satisfies $y_e(\mathcal{S}_c^p) = 0$, irrespective of the terminal task outcome. Depending on the available neutral evidence, the numbers of positive, neutral, and negative trajectories are respectively $(4, 0, 4)$, $(3, 1, 4)$, or $(3, 2, 3)$. Requiring agreement between terminal outcome and local progress for the positive and negative anchors provides unambiguous evidence of effective and ineffective behavior. Episodes with mismatched terminal and local signals remain valid observations for online value estimation but are not used as evolution anchors. Neutral cases instead help delimit behaviors for which the evidence does not justify a revision.

Revision and candidate admission. The evolver receives the shared capability description D_c, the complete parent body B_c^p, the task goals, and the selected trajectories with all invocation intervals of the parent explicitly marked. It is instructed to identify the highest-impact decision supported by the evidence, make the smallest correction needed to address it, preserve unrelated guidance, and introduce no capability unsupported by the observed environment or executor interface. The

output is a complete replacement body containing an executable procedure and observable stopping conditions, while the shared description D_c remains unchanged.

After structural and length validation, the generated version is added to $\mathcal{C}_c$ and admitted to the live set as a new challenger. It does not automatically replace its parent or inherit the parent's lifecycle status. Subsequent routing and lifecycle control determine whether the challenger is selected, retained, promoted, or archived. Consequently, candidate generation and candidate evaluation remain separated: the evolver only proposes a revision, whereas SKILLCRED determines its value from trajectories obtained during normal task execution. No additional environment rollouts or held-out validation tasks are used for evolution.

B LOCAL PROGRESS ASSESSMENT DETAILS

B.1 SIGNAL SEMANTICS

The ternary local progress signal uses the following semantics:

- **Positive progress** ☺ denotes observable material progress with no material attributable setback.
- **Neutral progress** 😐 denotes no material net effect, or mixed evidence with no reliable dominant contribution.
- **Negative progress** ☹ denotes a material attributable setback, unresolved key error, lost progress, or substantial ineffective waste.

The local progress assessment evaluates observed agent behavior within invocation intervals rather than estimating the counterfactual causal effect of the skill content. For each skill version $\mathcal{S}_c^k$ invoked in episode e, the assessor aggregates all of its invocation intervals into a single local progress signal $y_e(\mathcal{S}_c^k)$. It uses the remainder of τ_e only to interpret the observed trajectory context, including novelty, recovery or reversal, persistent errors, and downstream effects, while treating the terminal task outcome as context rather than an assigned label. If the judgment is invalid, or if the assessor abstains for that version, no posterior update is made.

B.2 ASSESSOR IMPLEMENTATIONS

The framework does not require a particular implementation of local progress assessment, provided that the assessor applies the same ternary semantics to observed behavior. Local progress assessment does not require a ground-truth terminal success label. To align the supervision strength of the assessment signals with that available to methods such as SkillOpt (Yang et al., 2026b) that perform label-based evaluation, we use benchmark-specific assessors in our main experiments. These assessors operate on the task instruction and action-observation trajectory from the same online episode and require no additional environment rollouts. Deterministic extraction identifies exact trajectory events that an LLM judge may infer unreliably.

ALFWorld. We use a lightweight hybrid assessor that combines a deterministic script with an LLM judge. The script extracts task-relevant state changes and common failure patterns from the trajectory, including wrong-object manipulations and repeated ineffective actions. It also preserves the exact object categories appearing in the task instruction and observations, avoiding semantic conflation such as treating `mug` and `cup` as equivalent. It maps unambiguous cases directly to local progress signals. When both positive and negative evidence are present, an LLM judge implemented with DeepSeek-V4-Flash aggregates the evidence across all anonymized invocation intervals associated with the same skill version, using the prompt template in Figure 4. We disable the judge's native thinking mode and instead request a brief evidence-grounded rationale in the explicit `thinking` field of the structured output.

WebShop. We use a lightweight scripted assessor. The script parses product attributes and option selections from observed page contents and uses the action-observation history to determine whether an opened product satisfies the request, whether a purchase succeeds, and whether the agent repeats ineffective actions or issues invalid actions. It maps these trajectory-derived signals to the same positive, neutral, and negative progress semantics.

Prompt Template for ALFWorld Local Progress Assessment

[SYSTEM INSTRUCTION]
You are a blind progress judge for interactive task trajectories. Score the net task contribution of anonymous skills across their marked intervals without inferring their identities or selection.

[TASK GOAL]
{goal}

[TASK OUTCOME]
{outcome}

[INTERACTION BUDGET]
{interaction_budget}

[ANONYMOUS ACTION INTERVALS]
{calls}

[IMPORTANT EVENTS]
{important_events}
These events are deterministic directional evidence, not scores. Verify them against the full trajectory, do not count them mechanically, and consider relevant unlisted evidence.

[JUDGMENT TASK]
For each SKILL ID, aggregate all of its marked intervals into one net local score. Use the rest of the trajectory only to assess novelty, causal attribution, recovery or undo, persistent errors, and downstream effects.

[SCORING]
- 1: observable material progress with no material attributable setback.
- -1: a material attributable setback, unresolved key error, lost progress, or substantial ineffective waste.
- 0: no material net effect, or mixed evidence with no reliable dominant contribution.

[RULES]
- Follow the environment's observed action outcomes rather than assumptions about how an action should work.
- Apply causal attribution first, then the budget-impossibility test, then net-evidence balancing. A satisfied budget test mandates -1 and must agree with the returned score.
- Attribute errors and waste only to the SKILL whose marked intervals caused them. Do not charge inherited or later blockers; once a blocker is clear, charge only continued ineffective behavior by the current SKILL. Prolonged stagnation is negative only when that SKILL caused it.
- If the episode exhausts its recorded limit and, when a SKILL yields control, its attributable step use leaves fewer steps than the visibly necessary minimum for all still-unmet goal constraints, score it -1. Count every explicit goal requirement; late or partial progress cannot offset making completion impossible. Require explicit budget and trajectory evidence: never infer this from failure alone or charge another SKILL's steps.
- Score the overall causal contribution, not intervals or events separately. Count each root event and milestone once; evaluate repetition, recovery, and undo-redo only through their final net effect.
- Novel task-relevant information first revealed after the initial state is material progress when it enables the task or visible downstream progress. Initial, known, irrelevant, or repeatedly discovered information is 0.
- Discovery alone does not outweigh substantial repeated invalid actions, ineffective waste, or prolonged control. If the budget test did not apply, mixed material progress and substantial waste default to 0; use 1 only for necessary, non-redundant progress, and -1 only when attributable waste clearly dominates through regression or an unresolved critical error.
- Treat task requirements as exact constraints. Related categories such as cup and mug are not interchangeable unless the environment explicitly says so. Encountering or inspecting a wrong object is 0; its first deliberate manipulation starts one regression chain.
- The final task outcome is context, not a score. Never automatically reward every SKILL on success or punish every SKILL on failure.
- Return one integer score (-1, 0, or 1) for each SKILL ID, in this order: {skill_ids}.

[OUTPUT FORMAT]
Output only one JSON object with exactly these top-level fields:
- "thinking": a concise judgment trace grounded only in observed trajectory evidence. Use at most one short sentence per SKILL, do not restate the trajectory, and keep the whole string under 500 words.
- "scores": an array with one item per SKILL. Each item must contain only an integer "skill_id" and an integer "score".

Figure 4: Prompt template for the LLM judge in the ALFWorld hybrid assessor. When deterministic evidence is inconclusive, the judge aggregates each skill version's invocation intervals into a single local progress signal.

Alternative feedback sources. These implementations instantiate rather than constrain the framework. In deployment, automated trajectory evidence may be replaced by or combined with user feedback. Explicit feedback can include confirmation, rejection, or correction of the agent's behavior or result, while implicit feedback can be inferred from observable follow-up behavior such as accepting a result, requesting a revision, reversing an action, or abandoning the current course of action. Regardless of source, the feedback is associated with the relevant invocation intervals and mapped to the same positive, neutral, and negative progress semantics, yielding at most one local

progress signal for each invoked skill version in an episode. Multiple feedback channels may be combined by a deterministic rule, a learned assessor, or human review. When the available evidence is insufficient or conflicting, the assessor may abstain and leave the posterior unchanged. Thus, the framework does not prescribe a particular feedback source or mapping mechanism as long as the resulting signal preserves the assessment granularity and ternary interface used for online value estimation.

C Lifecycle Control Details

C.1 Live Set Policy

We use a concurrent evaluation capacity of $\kappa = 3$ and allow at most one challenger per cluster to undergo the protected initial value estimation period at a time. Each cluster begins with the no-skill arm as its incumbent, and each newly admitted version enters as a challenger. A version is admitted only when the live set has an available slot. Retaining a challenger leaves its role and routing eligibility unchanged. Promoting it makes it the new incumbent and archives the previous incumbent, including the no-skill arm when it is the incumbent. Archiving removes an arm from the live set but does not remove the corresponding explicit skill version from its cluster or the skill library.

A new challenger remains in the protected initial value estimation period until it accumulates $N_{\min} = 8$ valid local progress signals. During this period, if it is not selected in three consecutive routing opportunities for its cluster, the next opportunity is assigned to that challenger. We denote its subsequent evaluation age by $\rho_{c,a,r}$ and initialize it to zero when the initial value estimation period ends. This age increases at every subsequent routing opportunity for cluster c, including rounds in which a competing arm is selected.

C.2 Deterministic Promotion-Probability Computation

We compute Equation 8 by deterministic quadrature. For the focal challenger a and incumbent $b = b_{c,r}$, let $z \in \{a, b\}$ index either arm and define

$$\eta_z = \theta_{c,z,-} + \theta_{c,z,+}, \qquad t_z = \frac{\theta_{c,z,+}}{\theta_{c,z,-} + \theta_{c,z,+}}. \tag{12}$$

The aggregation and neutrality properties of the Dirichlet distribution give

$$\begin{aligned} \eta_z &\sim \mathrm{Beta}(\alpha_{c,z,r,-} + \alpha_{c,z,r,+}, \alpha_{c,z,r,0}), \\ t_z &\sim \mathrm{Beta}(\alpha_{c,z,r,+}, \alpha_{c,z,r,-}), \qquad \eta_z \perp t_z, \end{aligned} \tag{13}$$

and $\mu_{c,z} = \eta_z(2t_z - 1)$. For fixed η_a, t_a, and η_b, the event $\mu_{c,a} > \mu_{c,b}$ is equivalent to $t_b < \bar{\tau}$, where

$$\bar{\tau} = \mathrm{clip}\left(\frac{1}{2}\left[1 + \frac{\eta_a(2t_a - 1)}{\eta_b}\right], 0, 1\right). \tag{14}$$

Let $I_x(u, v)$ denote the regularized incomplete Beta function. Analytically integrating over t_b reduces the promotion probability to an expectation of $I_{\bar{\tau}}(\alpha_{c,b,r,+}, \alpha_{c,b,r,-})$ over η_a, t_a, and η_b. Let $\eta_{a,i}$, $t_{a,j}$, and $\eta_{b,g}$ denote the Gauss–Jacobi nodes for the corresponding Beta distributions after mapping the raw nodes from $[-1, 1]$ to $[0, 1]$ by $x \mapsto (x+1)/2$. We obtain $\bar{\tau}_{ijg}$ by substituting these nodes into Equation 14, namely

$$\bar{\tau}_{ijg} = \mathrm{clip}\left(\frac{1}{2}\left[1 + \frac{\eta_{a,i}(2t_{a,j} - 1)}{\eta_{b,g}}\right], 0, 1\right). \tag{15}$$

Using n-point Gauss–Jacobi nodes and normalized weights for the three Beta variables gives

$$\widehat{p}^{\,\mathrm{prom}}_{c,a,r} = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{g=1}^{n} w_i^{\eta_a} w_j^{t_a} w_g^{\eta_b} I_{\bar{\tau}_{ijg}}(\alpha_{c,b,r,+}, \alpha_{c,b,r,-}). \tag{16}$$

This computation has cost $\mathcal{O}(n^3)$ and introduces no Monte Carlo sampling variance, although the fixed-order quadrature can incur deterministic discretization error. We use $n = 16$ for online lifecycle decisions and $n = 8$ for simulated promotion checks within CRV estimation.

C.3 PAIRED CRV ESTIMATION

For each Monte Carlo replication s, we draw a latent progress distribution for every live arm z

$$\boldsymbol{\theta}^s_{c,z} \sim \mathrm{Dirichlet}(\boldsymbol{\alpha}_{c,z,r}), \qquad z \in \mathcal{A}_{c,r}, \tag{17}$$

and treat it as the fixed data-generating distribution for that replication. Its value is $\mu^s_{c,z} = \theta^s_{c,z,+} - \theta^s_{c,z,-}$. We then clone $S_{c,r}$ into a retain branch containing the full live set and an archive branch in which challenger a is immediately removed. The two branches share the latent arm distributions. Random draws are indexed deterministically by replication, simulated step, event type, and arm, so both branches use the same draw for each corresponding event and arm even when their selected arms differ.

The simulation horizon is the challenger's remaining evaluation budget

$$H_{c,a,r} = R_{\max} - \rho_{c,a,r}. \tag{18}$$

This horizon covers the focal challenger's complete remaining opportunity for evaluation before maximum-age archiving. At each simulated round, each branch performs the following updates.

1. Select a live arm by Dirichlet Thompson sampling, applying the guaranteed-routing rule only to a challenger that is still within its initial value estimation period.
2. Draw one categorical local progress signal from the selected arm's latent distribution and update only its simulated Dirichlet posterior.
3. Advance challenger ages and apply the same promotion and maximum-age archiving rules as the online lifecycle policy.

The simulated futures do not recursively invoke CRV-based archiving or admit new skill versions. This keeps each comparison tied to the single intervention under evaluation, namely retaining or immediately archiving a, over the current competitive live set.

For fixed (c, a, r) and branch $u \in \{\mathrm{ret}, \mathrm{arc}\}$, let $\hat{a}^s_{u,h}$ be the arm selected at simulated step h and let $b^s_{u,H}$ be the terminal incumbent. Its discounted return is

$$Z^s_{c,a,r,u} = \sum_{h=0}^{H_{c,a,r}-1} \gamma^h \mu^s_{c,\hat{a}^s_{u,h}} + \frac{\gamma^{H_{c,a,r}}}{1-\gamma} \mu^s_{c,b^s_{u,H}}. \tag{19}$$

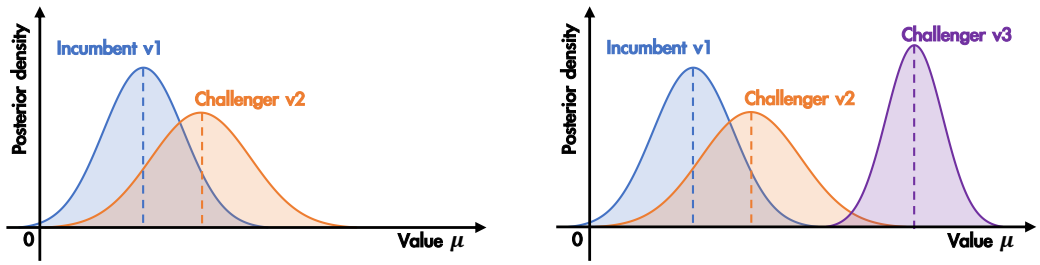
The final term equals $\sum_{t=H_{c,a,r}}^{\infty} \gamma^t \mu^s_{c,b^s_{u,H}}$ and is therefore the exact infinite discounted tail under the terminal policy that permanently selects the terminal incumbent. This terminal commitment is part of the CRV evaluation surrogate rather than a claim that online routing stops after $H_{c,a,r}$ rounds. At every real review, the surrogate is recomputed from the updated lifecycle state. Define the paired difference as $\Delta^s_{c,a,r} = Z^s_{c,a,r,\mathrm{ret}} - Z^s_{c,a,r,\mathrm{arc}}$. We estimate

$$\widehat{\mathrm{CRV}}_{c,a,r} = \frac{1}{N_{\mathrm{MC}}} \sum_{s=1}^{N_{\mathrm{MC}}} \Delta^s_{c,a,r}. \tag{20}$$

Shared randomness reduces the variance of the paired difference and isolates the effect of retaining a. For a directional agreement diagnostic, we estimate the conditional frequency of a negative paired difference as

$$\widehat{q^-_{c,a,r}} = \frac{\sum_{s=1}^{N_{\mathrm{MC}}} \mathbf{1}[\Delta^s_{c,a,r} < 0]}{\sum_{s=1}^{N_{\mathrm{MC}}} \mathbf{1}[\Delta^s_{c,a,r} \neq 0]}. \tag{21}$$

We set $\widehat{q^-_{c,a,r}} = 0$ when all paired differences are tied. We first generate $N_{\mathrm{MC}} = 256$ paired simulations and form a 99% normal interval for the mean difference and a 99% Wilson interval for $q^-_{c,a,r}$. We stop at 256 simulations only when there are at least 32 non-tied differences and both intervals support the same direction. The archive direction requires the upper endpoint of the mean interval to be below zero and the lower endpoint of the Wilson interval to exceed $1/2$. The retain direction requires the lower endpoint of the mean interval to exceed zero and the upper endpoint of the Wilson interval to be below $1/2$. Otherwise, we extend the same random streams to $N_{\mathrm{MC}} = 1024$ simulations. At the maximum simulation budget, the lifecycle decision uses the sign of $\widehat{\mathrm{CRV}}_{c,a,r}$; $\widehat{q^-_{c,a,r}}$ remains a simulation diagnostic and does not define a separate lifecycle threshold.



(a) Live set with incumbent v1 and challenger v2.

(b) Live set additionally containing challenger v3.

Figure 5: Illustration of CRV's dependence on the full live set. The value posteriors of v1 and v2 are held fixed across panels, while adding v3 changes the future routing alternatives in the hypothetical branch that immediately archives v2. The marginal policy value of retaining v2 can therefore change. Posterior shapes are schematic.

C.4 Lifecycle Decisions

CRV reviews begin when $\rho_{c,a,r} \geq R_{\text{stage}} = 16$ and occur at every subsequent cluster round while the challenger remains live. Promotion becomes available when $\rho_{c,a,r} \geq R_{\text{prom}} = 64$ and requires $\widehat{p}^{\text{prom}}_{c,a,r} > 0.80$. If multiple challengers qualify, we prioritize their estimated promotion probabilities, followed by their empirical mean differences from the incumbent and their evidence counts. Promotion makes the selected challenger the new incumbent and archives the previous incumbent.

A challenger whose estimated promotion probability exceeds the promotion threshold is protected from CRV-based archiving. Otherwise, CRV-based archiving requires $\widehat{\text{CRV}}_{c,a,r} < 0$ in two consecutive reviews. A challenger not promoted by $\rho_{c,a,r} = R_{\max} = 96$ is archived under the fixed evaluation budget. All other challengers are retained. We use $\gamma = 0.95$ throughout the paired simulations.

C.5 Rationale for a Policy-Level Retention Criterion

Promotion and retention answer different questions. Promotion asks whether the posterior evidence supports replacing the incumbent with a challenger, whereas retention asks whether preserving the challenger as a future routing option improves posterior-predictive policy value. For example, a simpler retention policy could threshold the challenger's posterior mean or variance, its posterior mean gap from the incumbent, or its probability of outperforming the incumbent. Such quantities provide reasonable proxies for current value or uncertainty, but they do not explicitly represent how retaining the challenger changes future routing and learning under the full live set. This distinction matters under bounded evaluation capacity because a retained challenger occupies live-set capacity and may be selected in rounds that would otherwise evaluate other live arms. CRV therefore evaluates retention through a marginal policy-value comparison rather than an arm-level or incumbent-pairwise decision.

Figure 5 holds the value posteriors of v1 and v2 fixed while varying whether v3 is live. Arm-local and incumbent-pairwise quantities for v2 are therefore unchanged, whereas its retention value can differ because the available uses of future routing opportunities differ. When v3 is also live, rounds in which v2 is selected may replace selections that would otherwise provide value or evidence for v3. If the posterior of v3 assigns substantial probability to high value, archiving v2 can redirect future selections toward v3, accelerate evidence accumulation for it, and potentially alter subsequent promotion decisions.

Let S_{A} and S_{B} denote the states in Figures 5a and 5b, respectively, and let H_{A} and H_{B} be the corresponding remaining evaluation horizons of v2. For this focal challenger, the two instances of Equation 10 are

$$\begin{aligned} \text{CRV}(S_{\text{A}}; \texttt{v2}) &= J_{H_{\text{A}}}(S_{\text{A}}; \{\texttt{v1}, \texttt{v2}\}) - J_{H_{\text{A}}}(S_{\text{A}}; \{\texttt{v1}\}), \\ \text{CRV}(S_{\text{B}}; \texttt{v2}) &= J_{H_{\text{B}}}(S_{\text{B}}; \{\texttt{v1}, \texttt{v2}, \texttt{v3}\}) - J_{H_{\text{B}}}(S_{\text{B}}; \{\texttt{v1}, \texttt{v3}\}). \end{aligned} \tag{22}$$

Here, $\mathrm{CRV}(S; a)$ denotes the CRV of challenger a evaluated from state S. These two quantities need not be equal because their archive branches expose different future routing alternatives. Thus, CRV depends on the competitive context of the full live set, not only on the focal challenger's posterior. The example does not imply that adding another challenger necessarily decreases CRV. The direction depends on the joint posterior state and the resulting future routing policy.

The same counterfactual comparison gives uncertainty an endogenous role. A challenger whose current posterior mean is below that of the incumbent may still be worth retaining when additional observations can reveal that its value is higher than currently estimated and thereby change a future routing or promotion decision. Conversely, uncertainty alone need not justify continued evaluation when the same finite routing opportunities have more valuable alternative uses elsewhere in the live set. CRV therefore does not introduce a separate exploration bonus. Uncertainty matters insofar as resolving it can improve future policy decisions.

D Implementation and Evaluation Details

Implementation. The initial skill library contains six task-specific skill clusters, listed in Appendix E. For SkillCred, each skill cluster maintains at most $\kappa = 3$ live arms at a time. We set the symmetric Dirichlet prior concentration to $\alpha_0 = 0.1$, which assigns 0.1 pseudo-count to each outcome and limits the influence of the prior on value estimates as evidence accumulates. Each challenger undergoes an initial value estimation period until it collects $N_{\min} = 8$ valid local progress signals, after which its evaluation age $\rho_{c,a,r}$ counts subsequent routing opportunities for cluster c. CRV reviews begin when $\rho_{c,a,r} \geq R_{\text{stage}} = 16$, promotion becomes available when $\rho_{c,a,r} \geq R_{\text{prom}} = 64$, and any challenger that remains unpromoted at $\rho_{c,a,r} = R_{\max} = 96$ is archived. All value estimates use only trajectories from normal task execution, without additional evaluation rollouts or a held-out validation set.

Adaptation procedure. During adaptation, SkillCred processes a stream of 500 task episodes online. We organize the stream as 50 consecutive batches of 10 tasks and handle the tasks in each batch sequentially. All methods receive the same adaptation tasks in the same order. For competing methods, we follow their official adaptation and evolution configurations, including their method-specific update schedules and batch construction, while keeping the total interaction budget fixed at 500 task episodes.

For ALFWorld, the adaptation episodes are drawn from the official `train` split, whereas final evaluation uses the disjoint `valid_seen` split. For WebShop, we use the 1,000-product WebShop-Small configuration distributed with WebShop. Product metadata and option combinations yield 6,910 synthetic shopping goals. The released implementation deterministically shuffles these goals with seed 233. We reserve the first 500 goals in this ordering as the fixed evaluation set and treat the remaining 6,410 goals as the training pool. Using seed 42, we shuffle the training indices and select the first 500 without replacement for adaptation. Thus, the adaptation and final evaluation tasks are strictly disjoint in both benchmarks. Additional validation rollouts required by a baseline are excluded from the shared interaction budget and reported separately as evaluation overhead.

E Initial Skill Library

The initial skill library used in our experiments contains six task-specific skill clusters, with one initial skill version per cluster. Table 3 summarizes their capability descriptions.

Figure 6 shows an initial skill version from the ALFWorld `systematic-search` cluster. It specifies the routing condition, procedure, and stopping condition.

F Additional Experimental Results

F.1 Lifecycle Decision Agreement with a Held-Out Reference

We extend the decision-level comparison in Figure 3a to all six skill clusters under Qwen3.5-9B. For each cluster, we use the same SkillOpt-style post-hoc held-out evaluation as an independent ref-

Table 3: Initial skill versions in the skill library.

Skill cluster	Capability description
ALFWorld	
`systematic-search`	Locate the next required exact entity through systematic search.
`object-relocation`	Pick up, transport, and place an exact target object.
`state-transformation`	Apply a required clean, heat, or cool state change to an exact target object.
`look-at-object-in-light`	Examine an exact held target with the required lamp.
WebShop	
`product-discovery`	Find and open a promising product candidate.
`product-evaluation-and-purchase`	Verify a product candidate, select required options, and purchase it.

`systematic-search`

description: Use when the next required exact entity is not located. Finish an actionable exact target before searching for another entity.

Procedure

1. Search open surfaces first. Continue searching other available locations until the exact required entity is found.

When to Stop Using This Skill

Stop when the sought exact entity is observed or an exact target becomes actionable.

Figure 6: An initial skill version from the ALFWorld `systematic-search` cluster.

erence and include versions with at least 10 valid local progress signals. A promotion agrees with the reference if the promoted challenger achieves higher held-out performance than the incumbent it replaces, while an archive agrees if the archived challenger does not outperform its incumbent. Eligible versions without a completed promotion or archive decision are excluded from the agreement denominator.

As shown in Table 4, 28/30 completed lifecycle decisions agree with the held-out reference. All 14/14 promotions agree, while 14/16 archives agree. Both disagreements are archives rather than promotions. Because the held-out evaluations are themselves finite-sample estimates, these results measure decision consistency with validation-based evaluation rather than ground-truth correctness. Nevertheless, the high agreement, particularly across all promotion decisions, indicates that SKILLCRED's online value estimates capture most of the information needed for lifecycle decisions without dedicated validation rollouts.

F.2 COMPARISON OF RETENTION CRITERIA

We compare CRV with two simpler early-archiving criteria while holding all other components fixed. At each review, we first apply the unchanged promotion rule to eligible challengers. Regardless of promotion eligibility, a challenger with $\widehat{p}^{\mathrm{prom}}_{c,a,r} > 0.80$ is protected from early archiving. We then apply one of the following criteria to each remaining challenger.

- **Posterior Probability Retention (PPR)** reuses the estimated promotion probability $\widehat{p}^{\mathrm{prom}}_{c,a,r}$ from Equation 16 and archives challenger a when $\widehat{p}^{\mathrm{prom}}_{c,a,r} < 0.20$. PPR therefore makes an incumbent-pairwise decision from the current posterior without accounting for other challengers.

Table 4: Agreement between SKILLCRED lifecycle decisions and a SkillOpt-style post-hoc held-out reference under Qwen3.5-9B. Eligible versions have at least 10 valid local progress signals, and fractions report agreeing decisions over completed decisions evaluated.

Skill cluster	Promotion agreement	Archive agreement	Overall agreement
ALFWorld			
`look-at-object-in-light`	1/1	1/1	2/2
`object-relocation`	3/3	1/2	4/5
`state-transformation`	2/2	1/1	3/3
`systematic-search`	2/2	5/5	7/7
WebShop			
`product-discovery`	4/4	2/3	6/7
`product-evaluation-and-purchase`	2/2	4/4	6/6
Overall	**14/14**	**14/16**	**28/30**

Table 5: Comparison of early-archiving criteria using Qwen3.5-9B. Each entry reports the mean ± standard deviation over 3 seeds. ALFWorld reports success rates by task type and overall success rate across all 140 evaluation tasks. WebShop reports average score and success rate.

Criterion	ALFWorld							WebShop	
	Pick	Look	Clean	Heat	Cool	Pick2	Avg. SR	Score	SR
PPR	89.5 $_{\pm 4.4}$	64.1 $_{\pm 8.9}$	72.8 $_{\pm 5.7}$	37.5 $_{\pm 6.3}$	49.3 $_{\pm 10.1}$	81.9 $_{\pm 2.4}$	69.5 $_{\pm 0.4}$	31.2 $_{\pm 0.8}$	19.9 $_{\pm 0.5}$
PoB	90.5 $_{\pm 4.4}$	64.1 $_{\pm 4.4}$	65.4 $_{\pm 2.1}$	**39.6** $_{\pm 3.6}$	52.0 $_{\pm 6.9}$	83.3 $_{\pm 8.3}$	69.3 $_{\pm 2.1}$	25.9 $_{\pm 1.7}$	18.1 $_{\pm 0.8}$
CRV	**93.3** $_{\pm 3.3}$	**71.8** $_{\pm 8.9}$	**90.1** $_{\pm 2.1}$	35.4 $_{\pm 3.6}$	**62.7** $_{\pm 6.1}$	**87.5** $_{\pm 11.0}$	**77.6** $_{\pm 1.5}$	**34.2** $_{\pm 1.0}$	**23.6** $_{\pm 0.9}$

- **Probability of Being Best (PoB)** defines

$$q^{\text{best}}_{c,a,r} = \Pr\left(\mu_{c,a} = \max_{z \in \mathcal{A}_{c,r}} \mu_{c,z} \,\middle|\, \mathcal{D}_{c,r}\right). \tag{23}$$

  PoB estimates this probability as $\widehat{q}^{\text{best}}_{c,a,r}$ through joint posterior sampling over the values of all live arms and archives challenger a when $\widehat{q}^{\text{best}}_{c,a,r} < 0.20$. Unlike PPR, PoB accounts for posterior uncertainty across the full live set. It remains a static posterior criterion, however, and does not simulate future routing, learning, or promotion dynamics.

Table 5 reports the resulting end-to-end performance. PPR trails CRV by 8.1 points in ALFWorld average success rate, 3.0 points in WebShop score, and 3.7 points in WebShop success rate. PoB obtains a similar ALFWorld average success rate to PPR but lower results on both WebShop metrics. Across the six ALFWorld task types, neither posterior-probability criterion uniformly dominates the other. CRV obtains the highest success rate on five task types. CRV's advantage on all three aggregate metrics suggests that a snapshot posterior probability does not fully capture the consequences of future routing, posterior updates, and promotion decisions.

F.3 EVALUATION OVERHEAD

We measure evaluation overhead as the additional computation required for value estimation beyond the shared online task workload. Dedicated evaluation episodes and executor calls count only extra environment rollouts and agent model invocations used to assess candidate skill versions. Incoming task episodes are excluded because all methods must solve them. Token and wall clock overheads also include local progress assessment, routing, and lifecycle control.

Table 6 reports the measured overhead of our benchmark implementations and compares it with SkillOpt's separate validation rollouts. SKILLCRED obtains value evidence from ongoing task trajectories and therefore requires no dedicated evaluation episodes or executor calls. On ALFWorld, a hybrid assessor derives local progress signals from deterministic trajectory evidence and calls an LLM judge only when that evidence is insufficient to determine the signal. On WebShop, a fully scripted assessor produces the signals without model calls. These implementations incur 1.19% to-

Table 6: Evaluation overhead under matched online workloads. Dedicated evaluation episodes and executor calls include only work introduced for candidate value assessment. Token and wall clock overheads are percentage increases relative to the same workload without candidate evaluation.

Method	Dedicated evaluation		Relative overhead	
	Episodes	Executor calls	Tokens	Wall clock
ALFWorld				
SkillOpt	700	13,949	188%	21.7%
SKILLCRED	**0**	**0**	**1.19%**	**1.24%**
WebShop				
SkillOpt	500	6,534	122%	9.2%
SKILLCRED	**0**	**0**	**0%**	**≈ 0%**

ken overhead and 1.24% wall clock overhead on ALFWorld, and zero token overhead with negligible wall clock overhead on WebShop. Section B.2 provides the assessor details.

In practical deployments, even fully LLM-based local progress assessment is expected to incur modest overhead relative to multi-step agent execution. A single batched judge request can take the completed trajectory as input and return one local progress signal for each invoked skill version in one response. The trajectory is therefore supplied only once, and all signals are returned together, whereas rollout-based validation requires repeated model calls interleaved with environment interactions.

This cost distinction also applies when ground-truth outcome labels or automated task verifiers are unavailable. SkillOpt's validation gate requires task-level scores on separate validation rollouts (Yang et al., 2026b). Adapting such a gate to these settings would require an alternative evaluator, such as an LLM judge. If both approaches use LLM judges with comparable per-trajectory scoring costs, rollout-based validation still incurs the additional cost of generating validation trajectories, while SKILLCRED assesses trajectories already produced by the online workload. The saving from avoiding dedicated rollouts therefore persists when scoring itself requires an LLM.

F.4 Robustness to Assessor Choice and Signal Corruption

F.4.1 End-to-End Performance with LLM-only Assessors

Our main experiments use deterministic trajectory extractors because direct LLM inference can be unreliable for benchmark-specific constraints. In ALFWorld, observations expose canonical object names, but an LLM may nevertheless treat semantically related categories such as `mug` and `cup` as equivalent. The extractor preserves this symbolic distinction. In WebShop, the extractor parses attributes from product detail pages in the trajectory and applies exact request matching. Thus, the main assessors improve the reliability of interpreting observed trajectory evidence rather than introducing a different source of task information.

To evaluate robustness without deterministic extraction, we rerun SKILLCRED with Qwen3.5-9B as the executor and DeepSeek-V4-Flash as the sole local progress assessor. The judge applies the same ternary semantics directly to the task instruction and action-observation history. We otherwise follow the main evaluation setup and the LLM judge configuration described in Section B.2.

Table 7: End-to-end task performance when an LLM-only judge is the sole local progress assessor. SkillOpt is the strongest baseline under the same Qwen3.5-9B setting. Each entry reports the mean ± standard deviation over 3 seeds.

Method	ALFWorld							WebShop	
	Pick	Look	Clean	Heat	Cool	Pick2	Avg. SR	Score	SR
SkillOpt	82.9 $_{\pm 0.0}$	**76.9** $_{\pm 7.7}$	49.4 $_{\pm 4.3}$	22.9 $_{\pm 9.5}$	41.3 $_{\pm 2.3}$	61.1 $_{\pm 2.4}$	57.9 $_{\pm 0.0}$	24.0 $_{\pm 0.6}$	18.4 $_{\pm 0.7}$
LLM-only assessor	91.4 $_{\pm 4.9}$	**76.9** $_{\pm 0.0}$	82.7 $_{\pm 4.3}$	**39.6** $_{\pm 3.6}$	49.3 $_{\pm 4.6}$	80.6 $_{\pm 10.5}$	73.1 $_{\pm 2.2}$	28.0 $_{\pm 1.1}$	19.5 $_{\pm 1.4}$
SKILLCRED	**93.3** $_{\pm 3.3}$	71.8 $_{\pm 8.9}$	**90.1** $_{\pm 2.1}$	35.4 $_{\pm 3.6}$	**62.7** $_{\pm 6.1}$	**87.5** $_{\pm 11.0}$	**77.6** $_{\pm 1.5}$	**34.2** $_{\pm 1.0}$	**23.6** $_{\pm 0.9}$

As shown in Table 7, relative to the benchmark assessors used in the main experiments, LLM-only assessment reduces the average success rate on ALFWorld from 77.6 to 73.1. On WebShop, the score decreases from 34.2 to 28.0 and the success rate decreases from 23.6 to 19.5. Nevertheless, the LLM-only variant exceeds SkillOpt, the strongest baseline in Table 1, by 15.2 points in ALFWorld average success rate, 4.0 points in WebShop score, and 1.1 points in WebShop success rate. These results suggest that direct LLM-only assessment preserves substantial gains from online skill value estimation, while deterministic extraction of the same observable trajectory evidence improves aggregate performance. We next examine whether replacing the benchmark assessors with the LLM-only assessor changes lifecycle decisions.

F.4.2 Lifecycle Decision Stability under LLM-only Assessment

To test whether LLM-only local progress assessment supports reliable lifecycle control, we replay the 30 completed lifecycle-decision checkpoints from Section F.1. At each checkpoint, we fix the live set, incumbent, challenger, evaluation age, and observation history. We replace the accumulated benchmark-assessor signals with LLM-only signals, recompute the Dirichlet posteriors, and apply the same promotion and archiving criteria. Each checkpoint is replayed independently, so a replayed decision does not alter later checkpoints.

Table 8: Lifecycle decision stability under the LLM-only assessor. Preserved denotes unchanged promotion or archive decisions, retain denotes deferred decisions, and reversal denotes direct promotion-to-archive or archive-to-promotion changes. Held-out agreement is computed over replay decisions that remain decisive.

Benchmark	Preserved	→ Retain	Reversal	Held-out agreement
ALFWorld	9/17	8/17	0/17	9/9
WebShop	10/13	3/13	0/13	10/10
Overall	**19/30**	**11/30**	**0/30**	**19/19**

As shown in Table 8, 19 of the 30 original lifecycle decisions are preserved, while the remaining 11 are deferred to retain. None reverse direction. All 19 replay decisions that remain decisive agree with the post-hoc held-out reference. In these fixed-checkpoint replays, LLM-only assessment therefore reduces decision coverage rather than producing opposite or held-out-inconsistent decisive actions. This pattern is consistent with posterior aggregation and evidence thresholds producing conservative lifecycle decisions when the assessor changes. Because each checkpoint is replayed independently, the analysis measures immediate decision stability and held-out agreement rather than the downstream effects of deferred decisions.

F.4.3 Sensitivity to Assessor Noise

Two-stage routing first identifies an applicable skill cluster from its shared capability description and then selects an arm within that cluster. This exposes arms to contexts induced by the same cluster-level routing process and reduces version-specific selection bias. We therefore approximate residual assessor noise using a common arm-independent process. For controlled sensitivity analysis, we instantiate this approximation by independently corrupting every accumulated signal under the same label-symmetric transition rule and corruption rate p for all arms. Label symmetry is a simplifying assumption for this stress test rather than a claim that real assessor errors are exactly symmetric.

Under this model, each accumulated local progress signal $y \in \{-1, 0, +1\}$ is independently replaced according to

$$\widetilde{y} = \begin{cases} y, & \text{with probability } 1-p, \\ \mathrm{Uniform}(\{-1,0,+1\} \setminus \{y\}), & \text{with probability } p, \end{cases} \tag{24}$$

where $p \in \{0.05, 0.10, 0.20, 0.30, 0.40\}$. We replay the same 30 fixed lifecycle-decision checkpoints used in Section F.4.2. Each noise level uses 10 independent corruption seeds, giving 300 checkpoint replays per level. The condition $p = 0$ is deterministic, while the remaining conditions report the mean and standard deviation across seeds.

As shown in Table 9, decision coverage decreases from 100.0% to 59.0% as p increases, while held-out agreement conditional on decisive actions remains between 91.8% and 92.8%. Thus, most of

Table 9: Lifecycle decision sensitivity to simulated assessor noise. Overall held-out agreement counts held-out-consistent decisive actions over all 30 checkpoints, while decisive agreement conditions on promotion and archive decisions. Preservation measures agreement with the original decision, coverage measures the fraction not changed to retain, and direct reversals aggregate all checkpoint replays at each noise level. Rates report the mean and standard deviation over 10 seeds.

Noise p	Held-out agreement (%)		Decision rates (%)		Direct reversals
	Overall	Decisive	Preservation	Coverage	
0	**93.3**	**93.3**	**100.0**	**100.0**	**0/30**
0.05	78.7 $_{\pm 5.2}$	91.8 $_{\pm 2.0}$	85.0 $_{\pm 5.2}$	85.7 $_{\pm 5.6}$	2/300
0.10	72.0 $_{\pm 5.4}$	92.4 $_{\pm 3.0}$	77.3 $_{\pm 5.9}$	78.0 $_{\pm 6.0}$	2/300
0.20	63.3 $_{\pm 5.4}$	92.6 $_{\pm 2.8}$	67.7 $_{\pm 4.5}$	68.3 $_{\pm 4.0}$	2/300
0.30	58.3 $_{\pm 7.8}$	92.4 $_{\pm 4.9}$	62.0 $_{\pm 7.5}$	63.0 $_{\pm 6.7}$	3/300
0.40	55.0 $_{\pm 8.2}$	92.8 $_{\pm 5.9}$	58.0 $_{\pm 6.7}$	59.0 $_{\pm 6.0}$	3/300

the degradation in overall held-out agreement comes from additional retain decisions rather than less reliable decisive actions.

Direct reversals remain rare. Only 12 of 1,500 noisy checkpoint replays, or 0.8%, reverse the original decision direction, all from archive to promotion. Increasing corruption therefore primarily produces additional retain decisions rather than reversals.

F.4.4 Effect of Assessor Noise on Decision Confidence

The preceding experiment shows that increasing corruption mainly reduces decision coverage. Under the symmetric corruption model, the following result explains this behavior. Corruption preserves the population value ordering for $p < 2/3$, but shrinks value gaps and increases the evidence required to reach a fixed promotion confidence threshold.

Proposition. Fix a cluster, a challenger a, and an incumbent b before observing the signals, and suppress the cluster and round indices. Suppose each arm $z \in \{a, b\}$ produces independent local progress signals $Y_z \in \{-1, 0, +1\}$ from a fixed distribution with mean μ_z, under comparable invocation contexts. Assume a positive gap $\Delta = \mu_a - \mu_b > 0$. Apply the simulated noise process in Equation 24 independently to every signal with the same $0 \le p < 2/3$, and let $\lambda_p = 1 - 3p/2$. The resulting population means satisfy

$$\widetilde{\mu}_z = \lambda_p \mu_z, \qquad \widetilde{\mu}_a - \widetilde{\mu}_b = \lambda_p \Delta > 0. \tag{25}$$

For a fixed sample size of n signals per arm, fit independent Dirichlet posteriors with prior $\alpha_0 \mathbf{1}$ to the noisy signals, as in the unmodified online update. Let $\widetilde{\mu}^{\mathrm{post}}_{z,n}$ denote the posterior random value of arm z under the fitted model. Denote the resulting promotion probability by $\widetilde{p}^{\mathrm{prom}}_n$ and the noisy observations by $\widetilde{\mathcal{D}}_n$. For any confidence threshold $q \in (0, 1)$ and failure probability $\delta \in (0, 1)$, a sufficient condition for $\widetilde{p}^{\mathrm{prom}}_n > q$ with sampling probability at least $1 - \delta$ is

$$n > \max\left\{ 3\alpha_0,\ \frac{32}{\lambda_p^2 \Delta^2} \max\left[\log \frac{2}{\delta}, \frac{q}{1-q} \right] \right\}. \tag{26}$$

Proof. For a given signal y, the other two outcomes sum to $-y$. Hence

$$\mathbb{E}[\widetilde{Y}_z \mid Y_z = y] = (1-p)y - \frac{p}{2} y = \lambda_p y, \tag{27}$$

which proves Equation 25. Write $d = \lambda_p \Delta$ and let $\overline{\widetilde{Y}}_z$ be the noisy sample mean for arm z. Since every signal lies in $[-1, 1]$, Hoeffding's inequality and a union bound give

$$\Pr\left(\overline{\widetilde{Y}}_a - \overline{\widetilde{Y}}_b < d/2 \right) \le 2 \exp(-n d^2 / 32). \tag{28}$$

On the complementary event, the posterior mean gap g_n satisfies

$$g_n = \frac{n}{n + 3\alpha_0} \left(\overline{\widetilde{Y}}_a - \overline{\widetilde{Y}}_b \right) \ge \frac{d}{4}, \qquad n \ge 3\alpha_0. \tag{29}$$

For either arm, the Dirichlet posterior variance of its value is at most $1/(n+3\alpha_0+1)$. Independence of the arm posteriors therefore bounds the posterior variance v_n of their value difference by $2/n$. Applying the one-sided Chebyshev inequality to this difference yields

$$1-\widetilde{p}_n^{\mathrm{prom}} = \Pr\Big(\widetilde{\mu}_{a,n}^{\mathrm{post}} - \widetilde{\mu}_{b,n}^{\mathrm{post}} \le 0 \;\Big|\; \widetilde{\mathcal{D}}_n\Big) \le \frac{v_n}{v_n+g_n^2} \le \frac{32}{32+nd^2}. \tag{30}$$

Equation 26 makes this bound smaller than $1-q$ and the failure probability in Equation 28 smaller than δ, completing the proof.

Implications. Under the symmetric corruption model, the population value ordering is preserved for $p < 2/3$, but the evidence-dependent term in the sufficient sample bound grows by $(1-3p/2)^{-2}$, reaching $6.25\times$ at $p = 0.40$. Corruption therefore reduces the effective separation between arms and requires more observations to reach the same promotion-confidence threshold. Together with the declining decision coverage, stable held-out agreement among decisive actions, and rare direct reversals in Table 9, this suggests that assessor noise primarily slows confidence accumulation rather than changing decision direction. A genuinely higher-value challenger may consequently require more routing opportunities before it can be promoted. Because challenger evaluation is capped by $R_{\max}$, it may instead remain unresolved and be archived before sufficient evidence is collected, potentially preventing a beneficial skill version from being adopted and thereby reducing downstream task performance. The bound is conservative and applies to a fixed arm pair under the synthetic corruption model; this interpretation describes a plausible mechanism for the observed loss of decision coverage rather than arbitrary assessor errors.

G Limitations and Future Work

A key limitation of SkillCred is its dependence on the quality of local progress signals. When deterministic trajectory extractors are unavailable, these signals must instead be inferred by an LLM judge or a more general evaluator agent, potentially introducing additional assessment noise and uncertainty. Our robustness analyses suggest that, under moderate noise, the primary effect is slower confidence accumulation rather than a change in the underlying population value ordering. Consistent with this observation, across 30 fixed-checkpoint replays using an LLM-only assessor, lifecycle decisions were either preserved or deferred, with no direct reversals.

This delay nevertheless has practical consequences. Noisier signals require more evidence to reliably distinguish challengers, and under a fixed evaluation budget, a potentially higher-value challenger may be archived before its value is sufficiently resolved. Moreover, the fixed-checkpoint replays do not capture feedback effects in which altered lifecycle decisions change subsequent routing and evidence collection, while our controlled corruption analysis does not cover arbitrary or systematic assessor errors. Future work should therefore develop evaluator agents that can better verify evidence under incomplete observations, together with lifecycle mechanisms that adapt the evaluation horizon to estimated signal quality.