SKILL-V: VERIFIABLE SELF-EVOLVING SKILL LIBRARY FOR INTERACTIVE AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Interactive agents can turn experience into reusable skills, yet existing self-evolving skill libraries primarily improve by accumulating new knowledge. Failures may lead to new skills, while previously stored skills are less often revisited as new evidence arrives. However, growth alone does not ensure reliability, as a retrieved skill may be inapplicable under the current task conditions, and an existing skill may encode a mis-specified operational boundary. Reliable skill evolution therefore requires not only adding knowledge, but also testing and revising what is already stored. We introduce **Skill-V**, a verifiable self-evolving skill library. To make stored knowledge testable, we propose representing skills as versioned, falsifiable contracts that link semantic intent to observable behavioral criteria. We use environment outcomes to drive continuous library evolution. Specifically, task failures motivate skill addition, while disagreements between contract evaluations and task outcomes guide revisions to existing skill boundaries. To validate these revisions, we require them to preserve protected semantic constraints and satisfy non-regression criteria for rubric–outcome metrics on historical replay evidence. Finally, we employ an applicability-aware filter to exclude candidates judged confidently inapplicable to the current task. Across ALFWorld and WebShop, Skill-V achieves success rates of 95.3% and 85.9%, respectively, while maintaining a more compact skill library than growth-oriented baselines. Applicability-aware filtering reduces incorrect skill invocations, and outcome-grounded revisions correct mis-specified skill boundaries without degrading performance on previously observed evidence. These results show that reliable skill evolution requires more than accumulating experience: the library must learn which knowledge to retain, when to revise it, and when it should be applied.

1 INTRODUCTION

Interactive agents (OpenAI, 2023; Team, 2023; Anthropic; DeepSeek-AI, 2025; Team, 2025) are increasingly capable of autonomous learning, a critical step toward recursive self-improvement (Liu et al., 2026b; Yin et al., 2025). However, retaining that experience in a form that can be reused, inspected, and corrected remains challenging (Yao et al., 2023; Shinn et al., 2023; Zhao et al., 2024). Reinforcement learning absorbs feedback into model parameters, making individual strategies difficult to isolate or revise. External skill libraries offer a more explicit alternative. They store reusable procedural knowledge outside the policy, allowing agents to retrieve it to guide future decisions (Wang et al., 2024a; Xia et al., 2026). Recent methods (Xia et al., 2026; Shi et al., 2026; Vishe et al., 2026; Ouyang et al., 2026; Wang et al., 2026) further allow such libraries to evolve during training by distilling interaction trajectories into new skills. This turns experience into an expanding external memory, but expansion alone does not make that memory reliable.

As a skill library grows, the main challenge is no longer only whether it contains enough useful strategies, but whether the strategies already stored remain valid (Ouyang et al., 2026; Pu et al., 2026). Existing self-evolving skill libraries (Xia et al., 2026; Shi et al., 2026) primarily improve by accumulating new content. While failures lead to new skills, previously stored skills are rarely revisited (Figure 1(a)). A retrieved skill may be semantically relevant yet inapplicable under the current task conditions (Chen et al., 2026; Zheng et al., 2026; Shi et al., 2026). A stored skill may also encode an incorrect operational boundary that is either too broad and admits ineffective behavior, or too narrow and excludes a valid strategy. These failures stem not merely from missing

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

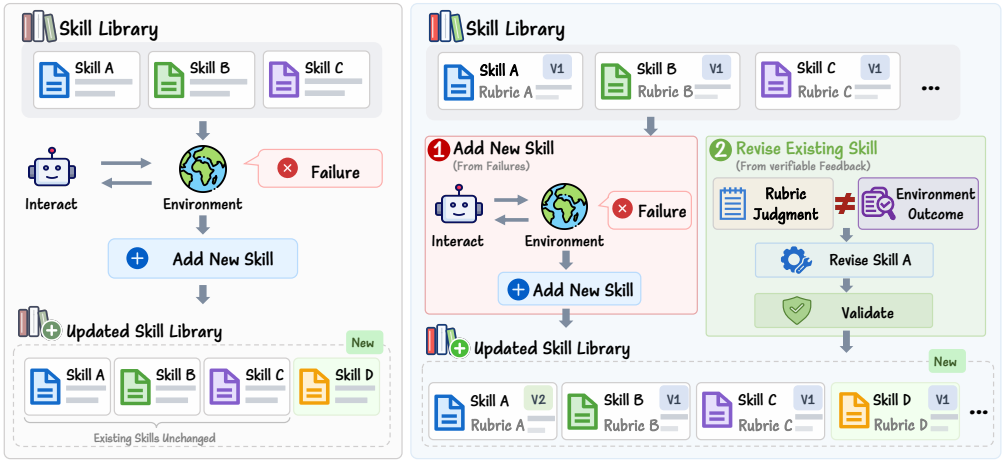

Figure 1: **Comparison of self-evolving skill library paradigms.** **(a)** Existing methods rely on growth-oriented evolution. Failures prompt the addition of new skills, but previously stored skills are rarely questioned, which often leads to boundary conflicts. **(b) Skill-V** treats skills as versioned, falsifiable contracts. It adds missing skills from failures (Path 1) and revises existing operational boundaries when rubric judgments disagree with task outcomes (Path 2). Through validated revisions, the library expands its coverage while actively maintaining the reliability of stored knowledge.

knowledge, but from incorrect decisions about when existing knowledge should be trusted. Such unreliability severely limits long-term autonomous self-improvement.

Reliable skill evolution therefore requires more than adding new knowledge (Liu et al., 2026c; Gou et al., 2024). An evolving library must also test whether its existing knowledge remains consistent with new experience and revise it when evidence exposes an incorrect boundary. This motivates our central question:

> *How can an agent expand its skill library while continuously testing and correcting the knowledge already stored?*

To answer this, we propose treating each skill as a versioned, falsifiable contract. In addition to semantic intent and a reusable procedure, a skill must specify its conditions of applicability and the observable behavior expected when it is followed. Interaction outcomes can then serve not only as feedback for learning how to act, but also as empirical evidence for testing what the library currently claims. For instance, a rubric that accepts a failed trajectory is likely too permissive, whereas one that rejects a successful trajectory may be too restrictive.

Based on this idea, we introduce Skill-V, a verifiable self-evolving skill library for interactive agents (Figure 1(b)). Skill-V augments each skill with a contract that links its semantic intent to observable behavioral criteria. During training, each interaction provides two complementary signals for evolution. First, task failures expose missing strategies and drive the addition of new skills. Second, rubric–outcome disagreements identify existing contracts whose boundaries require revision. Candidate revisions are committed only when they preserve the skill's semantic identity and do not regress on previously observed evidence. Skill-V thereby expands its coverage while conservatively improving the reliability of its stored knowledge.

Skill-V also separates semantic relevance from task applicability when using the evolved library. It first retrieves semantically related candidates and then filters skills that are confidently inapplicable to the current task. The selected skills guide the policy throughout the interaction, and the resulting trajectory and outcome provide new evidence for subsequent library evolution. This closes the loop between using stored knowledge and testing whether that knowledge should be retained, expanded, or revised. The library evolves not only by expanding its coverage, but also by making its existing knowledge more selective, correctable, and reliable.

We evaluate Skill-V across embodied household tasks in ALFWorld (Shridhar et al., 2020) and online shopping in WebShop (Yao et al., 2022). We compare against static and growth-only skill libraries to assess overall task performance. We also analyze the effectiveness of applicability filtering and the ability of counterexample-driven revisions to correct existing skill boundaries. Empirical results confirm that verifiable evolution yields state-of-the-art capabilities. On WebShop, Skill-V achieves a 92.5 task score and an 85.9% success rate, outperforming existing memory-augmented agents. On ALFWorld, it achieves a 95.3% success rate and solves several subtasks perfectly while requiring only a fraction of the library size demanded by accumulation-centric methods. Beyond overall success metrics, our analyses demonstrate that Skill-V effectively corrects boundary errors and mitigates the catastrophic forgetting that typically degrades lifelong learning agents.

Our main contributions are:

- We formulate self-evolving skill learning as the joint problem of expanding knowledge coverage and maintaining the reliability of existing skills.
- We introduce Skill-V, which combines versioned skill contracts, outcome-grounded skill addition and revision, evidence-preserving validation, and applicability-aware skill use.
- We evaluate Skill-V on ALFWorld and WebShop, demonstrating that proactive skill maintenance achieves comparable performance, including success rates of 95.3% on ALFWorld and 85.9% on WebShop. Crucially, we show that Skill-V effectively corrects boundary errors and mitigates catastrophic forgetting while avoiding the severe library bloat that plagues accumulation-only baselines.

2 Related Work

Memory and skill reuse. LLM-based agents can learn from experience by retaining trajectories, reflections, or distilled knowledge (Yao et al., 2023; Zhu et al., 2023; Packer et al., 2023). Reflexion (Shinn et al., 2023) and ExpeL (Zhao et al., 2024) convert interaction feedback into reusable textual memory, while Voyager(Wang et al., 2024a) organizes experience as executable skills (Schick et al., 2023). These works establish external memory as a complement to parametric learning. Skill-V focuses on a later-stage problem of how persistent procedural knowledge should be tested and corrected after entering memory.

Self-evolving skill learning. Recent methods allow skill libraries to improve through interaction and feedback (Gou et al., 2024; Yang et al., 2024). SkillRL (Xia et al., 2026) recursively expands a hierarchical SkillBank alongside policy learning, while Skill1 (Shi et al., 2026) jointly optimizes skill selection, utilization, and distillation. Skill-R1 (Vishe et al., 2026) trains a lightweight generator to iteratively revise instance-level skills for a frozen task model, whereas SkillOS (Ouyang et al., 2026) learns a curator that updates an external SkillRepo from delayed downstream feedback. These approaches primarily study skill acquisition, credit assignment, or curation. Skill-V instead asks whether a stored skill remains applicable and behaviorally well-specified as new evidence arrives.

Reliable skill maintenance. SkillOpt (Yang et al., 2026) treats a single skill document as a trainable parameter updated via validation-gated edits. Concurrently, SkillOps (Pu et al., 2026) represents skills as typed contracts and maintains library-level utility, compatibility, risk, and validation. Skill-V differs in both its setting and learning signal. It studies skill maintenance within interactive reinforcement learning, where the policy and library evolve together. General agent reliability is often approached through post-hoc reflection or self-correction (Shinn et al., 2023; Gou et al., 2024). Skill-V instead uses disagreements between skill rubrics and environment outcomes to systematically identify boundary errors. Candidate revisions are admitted only when they preserve semantic intent and do not regress on observed counterexamples. This strict validation is designed to mitigate the catastrophic forgetting that typically degrades lifelong learning agents (French, 1999; Wang et al., 2024b). Thus, Skill-V extends skill evolution from acquiring more knowledge to maintaining when existing knowledge should be trusted.

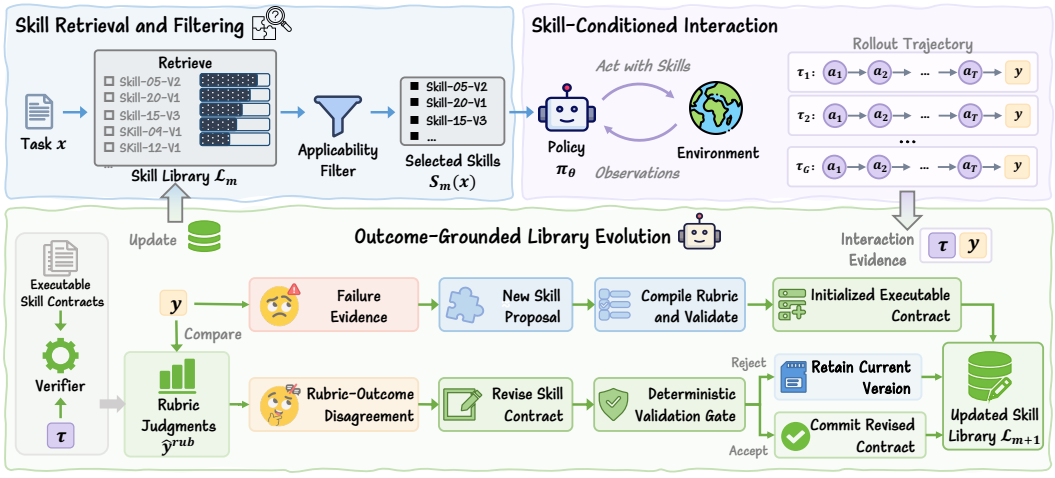

Figure 2: **Overview of Skill-V.** Skill-V first retrieves and filters task-relevant skills to guide policy interaction, then uses the resulting trajectories and environment outcomes to evolve the skill library. Failures expose missing strategies and lead to the addition of new semantic skills, whereas rubric–outcome disagreements provide counterexamples for revising existing executable skill contracts. Contract revisions are committed only after deterministic validation, and the updated library is reused in future interactions.

3 METHOD

Skill-V turns an external skill library into an explicit object of learning. As illustrated in Figure 2, it forms a closed loop between skill use and library evolution. We formalize this interactive setting in Section 3.1 and represent skills as versioned, falsifiable contracts in Section 3.2. During training, environment outcomes drive library evolution through two complementary mechanisms detailed in Section 3.3. Task failures motivate the addition of new skills, while disagreements between contract rubrics and actual outcomes trigger targeted boundary revisions. Finally, Section 3.4 describes how the agent retrieves and filters these skills to ensure task applicability during inference.

3.1 PROBLEM FORMULATION

We consider an interactive agent that receives a task instruction x, produces a trajectory $\tau = (o_1, a_1, \ldots, o_T, a_T)$, and obtains an environment outcome $y(\tau) \in \{0, 1\}$. The agent maintains an external skill library $\mathcal{L}_m = \{s_i\}_{i=1}^{N_m}$, where m indexes the current library version. Each skill is a reusable natural-language strategy that describes when a behavior is useful and how it should be carried out. Rather than specifying a complete policy or a fixed action sequence, a skill provides procedural guidance that the policy adapts to the current interaction.

For a task x, the agent retrieves a subset of skills $\mathcal{S}_m(x) = \mathrm{Retrieve}(x, \mathcal{L}_m)$ (detailed in Section 3.4) and conditions the policy on the selected skills:

$$a_t \sim \pi_\theta \left(a_t \mid x, h_t, \mathcal{S}_m(x)\right), \tag{1}$$

where $h_t = (o_1, a_1, \ldots, o_t)$ denotes the interaction history.

During training, the agent collects experience $\mathcal{D}_m = \{(x_n, \tau_n, y_n)\}_{n=1}^{M}$ and updates the library as $\mathcal{L}_{m+1} = \mathrm{Update}(\mathcal{L}_m, \mathcal{D}_m)$. The update may add a missing skill, revise an existing skill, or leave the library unchanged. The central problem is therefore not simply how to grow the library, but how to determine which change is supported by experience.

3.2 SKILLS AS VERIFIABLE CONTRACTS

Skill-V represents each skill $s_i \in \mathcal{L}_m$ as

$$s_i = (\phi_i,\ p_i,\ \rho_i,\ \nu_i), \qquad \rho_i \in \{\emptyset,\ (A_i, \mathcal{C}_i)\}, \tag{2}$$

where ϕ_i specifies the semantic intent and intended applicability of the skill, p_i describes its reusable procedure, and ν_i records its skill and rubric versions. The rubric ρ_i remains $\emptyset$ while its executable specification is pending. Otherwise, it comprises applicability conditions A_i and a set of weighted behavioral criteria $\mathcal{C}_i = \{c_{ij}\}_j$.

The semantic specification determines what knowledge the skill represents and how it should guide the policy. However, free-form language alone does not provide a systematic way to test whether the skill remains consistent with experience. The executable rubric therefore captures observable implications of the skill, including required steps, forbidden behaviors, preconditions, postconditions, and termination conditions. Each criterion is grounded in evidence from the task instruction, executed actions, observations, or event sequences. Some criteria are designated as hard constraints and must be satisfied regardless of the aggregate rubric score. Ultimately, these two components serve complementary roles. The semantic specification defines what the skill means, while the executable rubric makes its current operational boundary falsifiable. Skill-V can therefore revise when and how a skill should apply without silently replacing the knowledge that the skill is intended to represent.

Contract initialization. Existing and newly added skills may initially contain only semantic descriptions and reusable procedures, resulting in $\rho_i = \emptyset$. Such skills can already guide the policy but cannot yet participate in executable contract evaluation. Given applicable experience, a contract updater proposes observable criteria based on the interaction history. Provided this proposal preserves the skill's semantic intent, admits non-empty executable support, and satisfies the contract specification, it becomes the first executable version of the skill. Subsequent modifications are recorded as new versions. This strict versioning makes the evolution of each skill explicit and traceable.

3.3 Outcome-Grounded Skill Evolution

Skill-V uses trajectories and environment outcomes as interaction evidence $e = (x, \tau, y)$ to derive two complementary signals for skill library evolution. A task failure suggests that the library may lack a reusable strategy, motivating the addition of a new skill. Furthermore, to identify existing contracts that may be mis-specified, Skill-V evaluates the executable contracts that apply to task x. Let $\mathcal{A}(x) = \{s_i \in \mathcal{L}_m : A_i(x) = 1\}$ denote this set of applicable skills. For each criterion c_{ij} within these contracts, $m_{ij}(\tau) \in \{0, 1\}$ indicates whether the trajectory satisfies the criterion, and w_{ij} denotes its assigned weight. Skill-V aggregates these applicable criteria to produce an overall rubric judgment:

$$q(\tau) = \frac{\sum_{s_i \in \mathcal{A}(x)} \sum_j w_{ij} m_{ij}(\tau)}{\sum_{s_i \in \mathcal{A}(x)} \sum_j w_{ij}}. \tag{3}$$

The final contract-level judgment is then produced by applying a strict threshold and checking for hard constraint violations:

$$\hat{y}^{\,\mathrm{rub}}(\tau) = \mathbb{I}\left[q(\tau) \geq \eta \ \wedge \ H(\tau) = 0\right], \tag{4}$$

where $H(\tau)$ indicates whether any applicable hard criterion is violated. When no executable criterion applies, Skill-V does not produce a contract-level judgment for that trajectory.

Alongside the aggregate judgment, Skill-V retains individual rubric scores and failed criteria to identify applicable contracts that contradict the trajectory. Comparing rubric judgments with environment outcomes provides diagnostic evidence for skill maintenance. A rubric pass followed by task failure may indicate an overly permissive or incomplete executable specification, whereas a rubric failure on a successful trajectory may indicate an overly restrictive criterion. Agreement cases provide supporting or negative evidence, and joint failures may also motivate procedural repair. Because environment outcomes are defined for complete trajectories, these signals do not establish that a particular skill caused success or failure. Skill-V uses them to prioritize candidate updates, which are subsequently screened by the validation gate.

Skill addition. A failed trajectory may reveal that the current library lacks an appropriate strategy. To address this capability gap, a language-model updater analyzes the failed interaction to propose new procedural knowledge. The system adds this resulting skill in semantic form with its executable rubric initially pending. The skill nevertheless becomes immediately available for subsequent interactions. This proactive addition continuously expands the library's coverage.

Skill revision and validation. A disagreement between the rubric judgment and environment outcome provides a counterexample to the current operational boundary. Using the per-skill diagnostics, the updater proposes targeted revisions to the relevant contracts. A proposal may refine the skill's applicability conditions, procedure, or executable criteria, but must preserve the protected semantic constraints that encode the identity of the skill. However, such counterexamples do not justify arbitrary rewrites. For each skill s_i, let $\mathcal{E}_i$ denote its observed evidence set, which is maintained via a bounded replay buffer of applicable trajectories. Skill-V evaluates the current rubric ρ_i and candidate rubric ρ_i' by treating rubric pass/fail as a prediction of the environment outcome. We measure their consistency using the overall disagreement rate Disc, the false-negative rate on successful trajectories FNR, and balanced accuracy BAcc.

We require revisions to be non-regressive on observed evidence. Specifically, a candidate ρ_i' is considered evidence-preserving relative to ρ_i only if it yields a lower or equal Disc and FNR, along with an equal or higher BAcc on the evidence set $\mathcal{E}_i$. To be eligible for acceptance, this evidence-preserving revision must also maintain non-empty executable support, satisfy the contract specification, and preserve the semantic constraints of its source skill. Furthermore, if the revision alters a hard executable criterion, Skill-V requires a strict improvement in at least one of these three statistical metrics. An accepted revision is committed as the next version of the skill contract, whereas failing candidates are rejected to retain the current version. This rigorous constraint allows Skill-V to refine a falsified boundary without sacrificing behavior that remains supported by experience.

3.4 Applicability-Aware Skill Use

The evolved skill library must determine which skills are useful for a particular task. Semantic similarity provides an efficient first-stage retrieval signal, but a semantically related skill may not be applicable under the current task constraints. Skill-V therefore separates candidate retrieval from applicability judgment. Given an embedding function f, Skill-V ranks stored skills by the cosine similarity between the task embedding $f(x)$ and their semantic representations. It retrieves an expanded candidate set from both general and task-specific portions of the library, allowing the applicability stage to inspect more candidates than will eventually be provided to the policy.

To evaluate task applicability, Skill-V combines deterministic contract checks with an LLM-based applicability judge. The judge inspects each candidate with respect to the current task and returns one of $\{\textsc{Applicable}, \textsc{NotApplicable}, \textsc{Uncertain}\}$ together with a confidence score. Skill-V removes a candidate only when it is judged not applicable with confidence above a threshold δ. This conservative rule retains uncertain but potentially useful skills while filtering confident mismatches. Among the remaining candidates, Skill-V retains those whose semantic similarity lies within a margin ϵ of the highest-scoring candidate and selects the top K skills. If applicability judgments are unavailable, Skill-V falls back to semantic Top-K retrieval. The selected skills are provided to the policy at the beginning of the episode and used throughout the interaction. This execution closes the learning loop. The skill library guides the policy, the environment evaluates the resulting behavior, and the observed outcome provides new interaction evidence for adding missing skills or revising mis-specified contracts.

4 Experiments

4.1 Experimental Setup

Environments and metrics. We evaluate Skill-V on ALFWorld (Shridhar et al., 2020) and WebShop (Yao et al., 2022) using the same evaluation protocol as SkillRL and Skill1 (Xia et al., 2026; Shi et al., 2026). ALFWorld is a text-based embodied environment requiring agents to complete multi-step tasks. We report the overall success rate alongside success rates across its six task types (Pick, Look, Clean, Heat, Cool, and Pick2). WebShop evaluates interactive search and purchase under natural-language constraints, where we report both the average task score and success rate.

Baselines. We compare Skill-V against a comprehensive suite of baselines, including closed-source LLM agents, prompting- and memory-based methods, RL-based agents, and memory-augmented RL methods. This latter category serves as our primary point of comparison and includes MemRL (Zhang et al., 2026a), EvolveR (Wu et al., 2025), Mem0+GRPO (Chhikara et al.,

Table 1: Main results on ALFWorld and WebShop. ALFWorld reports success rates (%) for each task type and overall. WebShop reports the average task score and success rate (%). The best result is shown in **bold** and the second best is underlined. †Skill1 expands its ALFWorld skill library to its capacity of 5,000 skills; stored units may not be directly comparable across methods.

	ALFWorld							WebShop	
Method	Pick	Look	Clean	Heat	Cool	Pick2	**All**	**Score**	**Succ.**
Closed-source LLMs									
GPT-4o	75.3	60.8	31.2	56.7	21.6	49.8	48.0	31.8	23.7
Gemini-2.5-Pro	92.8	63.3	62.1	69.0	26.6	58.7	60.3	42.5	35.9
Training-free agents with Qwen2.5-7B-Instruct									
Qwen2.5	33.4	21.6	19.3	6.9	2.8	3.2	14.8	26.4	7.8
ReAct	48.5	35.4	34.3	13.2	18.2	17.6	31.2	46.2	19.5
Reflexion	62.0	41.6	44.9	30.9	36.3	23.8	42.7	58.1	28.8
Mem0	54.0	55.0	26.9	36.4	20.8	7.7	33.6	23.9	2.0
ExpeL	21.0	67.0	55.0	52.0	71.0	6.0	46.3	30.9	11.2
MemP	54.3	38.5	48.1	56.2	32.0	16.7	41.4	25.3	6.4
SimpleMem	64.5	33.3	20.0	12.5	33.3	3.84	29.7	33.2	8.59
RL-trained agents with Qwen2.5-7B-Instruct									
RLOO	87.6	78.2	87.3	81.3	71.9	48.9	75.5	80.3	65.7
GRPO	90.8	66.1	89.3	74.7	72.5	64.7	77.6	79.3	66.1
Memory-augmented RL agents with Qwen2.5-7B-Instruct									
MemRL	62.8	38.5	22.2	12.5	8.0	0.0	21.4	29.5	9.2
EvolveR	64.9	33.3	46.4	13.3	33.3	33.3	43.8	42.5	17.6
Mem0+GRPO	78.1	54.8	56.1	31.0	65.0	26.9	54.7	58.1	37.5
SimpleMem+GRPO	89.5	36.3	60.0	50.0	64.9	26.3	62.5	67.8	46.9
SkillRL	97.9	71.4	90.0	90.0	95.5	87.5	89.9	85.2	72.7
RetroAgent	97.9	90.9	99.2	92.9	85.3	91.0	94.9	88.9	82.3
Skill1†	**100.0**	98.6	97.3	99.2	**96.1**	**96.5**	**97.5**	89.7	82.9
Skill-V (Ours)	93.3	**100.0**	**100.0**	**100.0**	90.9	92.9	95.3	**92.5**	**85.9**

2025), SimpleMem+GRPO (Liu et al., 2026a), SkillRL (Xia et al., 2026), RetroAgent (Zhang et al., 2026b), and Skill1 (Shi et al., 2026).

Implementation details. Skill retrieval uses Qwen3-Embedding-0.6B, followed by an applicability judge implemented with DeepSeek-V4-Flash (DeepSeek-AI, 2025). We retrieve up to six general skills and five task-specific skills after filtering. The initial candidate pool is twice the final retrieval budget, and candidates judged inapplicable with confidence above 0.8 are discarded. Executable rubrics employ a pass threshold of $\eta = 0.7$, and the skill library is periodically evolved using recent training trajectories. Complete details and hyperparameters are provided in Appendix B and C.

4.2 Main Results

Skill-V substantially improves over task-reward RL. Table 1 demonstrates the strong performance of Skill-V across both interactive domains. Compared with GRPO, Skill-V improves the overall ALFWorld success rate from 77.6% to 95.3%, a gain of 17.7 percentage points. On WebShop, it improves the average score from 79.3 to 92.5 and the success rate from 66.1% to 85.9%. Since both methods optimize the policy using environment task rewards, these gains suggest that evolving reusable external knowledge provides benefits beyond pure policy optimization.

Skill-V is competitive with the strongest skill-augmented agents. As the most direct baseline, SkillRL also combines reinforcement learning with an evolving skill library. Compared to SkillRL, Skill-V elevates the overall ALFWorld success rate from 89.9% to 95.3%. The improvement spans several task types, including gains of 28.6, 10.0, 10.0, and 5.4 percentage points on Look, Clean, Heat, and Pick2, respectively. Skill-V consequently reaches 100% success on Look, Clean, and Heat. The gains are even larger on WebShop, where Skill-V improves the average score from 85.2 to 92.5 and the success rate from 72.7% to 85.9%. These results indicate that verifying skill applicabil-

Table 2: Knowledge-source analysis on ALFWorld. We fix the policy at the end of training and vary only the external knowledge available during evaluation.

Knowledge Condition	Success	Δ vs. No Skill	Δ vs. Initial
No External Skills	78.59	–	–
Initial Skill Library	88.99	+10.40	–
Skill-V Library	**95.30**	**+16.71**	**+6.31**

Table 3: Ablation of skill evolution mechanisms. Failure-based revision corrects permissive contracts, whereas success-based revision corrects restrictive ones.

Method	Acquire	Fail Rev.	Succ Rev.	Success (%) ↑
Static Library	×	×	×	73.40
Acquisition Only	✓	×	×	68.80
Failure-only	✓	✓	×	67.20
Full Skill-V	✓	✓	✓	**95.30**

ity and revising existing skill boundaries provide substantial benefits beyond acquisition-based skill evolution. Skill-V demonstrates similar superiority over more recent memory-augmented agents. It outperforms RetroAgent by 0.4 percentage points on ALFWorld, and achieves gains of 3.6 points in both average score and success rate on WebShop. While Skill-V marginally trails Skill1 by 2.2 percentage points on the overall ALFWorld metric, it achieves notable gains of 2.8 points in average score and 3.0 percentage points in success rate on WebShop.

Competitive performance does not require aggressive library growth. The comparison with Skill1 reveals a substantial difference in library scale. Skill1 expands its ALFWorld library to its capacity of $5{,}000$ skills (Shi et al., 2026), whereas Skill-V contains approximately 200 stored entries at the end of training. This represents less than 4% of the library size demanded by Skill1. Nevertheless, Skill-V achieves 100% success on the Look, Clean, and Heat task types, outperforming Skill1 by 1.4, 2.7, and 0.8 percentage points, respectively. The combination of a highly compact library and superior subtask performance proves that effective skill evolution depends not only on how much knowledge is accumulated, but also on whether stored skills are strictly applicable and correctly specified. By prioritizing verifiable maintenance over unrestrained growth, Skill-V successfully circumvents the severe knowledge bloat that typically degrades long-term agent autonomy.

4.3 Ablation Studies

Validated skill evolution provides gains beyond policy learning. Because the policy and skill library evolve together during training, final task performance alone cannot reveal whether the improvement is encoded in the policy parameters or maintained in the external library. We disentangle these sources by fixing the policy checkpoint at the end of training and varying only the external knowledge provided during evaluation (Table 2). Without external skills, the frozen policy achieves a 78.59% success rate. Providing the initial skill library raises performance to 88.99%, demonstrating that external procedural knowledge remains highly useful even after extensive policy training. Using the fully evolved Skill-V library further improves success to 95.30%. Unlike a library expanded through acquisition alone, this library contains newly acquired skills alongside existing contracts revised through rubric-outcome diagnosis and non-regressive validation. This additional 6.31-point gain over the initial library confirms that the evolved skill library provides useful knowledge beyond the initial library, even when the policy parameters are fixed.

Reliable evolution requires more than skill accumulation. Adding new skills alone fails to recover the performance of the full method (Table 3). Specifically, the Acquisition Only variant reaches 68.8% success, compared with 95.3% for Skill-V. Restricting revision exclusively to failed trajectories is also insufficient, reaching only 67.2%. In contrast, full Skill-V combines skill acquisition with both directions of contract correction. Failed trajectories expose missing strategies and overly permissive boundaries, while successful counterexamples identify overly restrictive boundaries. Together, these mechanisms outperform all incomplete variants by at least 21.9 points. These results strongly support our central claim that reliable skill evolution benefits from combining library expansion with bidirectional boundary correction.

Evidence Gate filters regressive skill revisions. Structural validity alone does not guarantee that a revision preserves contract reliability. Without evidence-based validation, 9 of the 46 committed revisions regress on at least one replay metric, and only 27 satisfy the complete acceptance criteria

Table 4: Effect of the Evidence Gate on ALFWorld task performance and revision reliability. Non-regressive revisions do not worsen Disc, FNR, or BAcc on replay evidence. Passing the full gate additionally requires satisfying all remaining acceptance conditions. Regression counts are not mutually exclusive.

	Task perf.	Revisions	Retrospective audit		Metric regressions ↓		
Variant	**Success (%) ↑**	**Committed**	**Non-regressive**	**Pass full gate**	**Disc**	**FNR**	**BAcc**
w/o Evidence Gate	75.00	46	37 (80.40%)	27 (58.70%)	9	0	3
Full Skill-V	**95.30**	10	**10 (100.00%)**	**10 (100.00%)**	**0**	**0**	**0**

Table 5: Effect of the LLM applicability judge on ALFWorld. Isolates the contribution of LLM-based applicability judgment.

Variant	Embed. Retr.	Contract Check	LLM Judge	Success (%) ↑
Semantic Retr. Only	✓	×	×	71.90
w/o LLM Judge	✓	✓	×	89.10
Full Skill-V	✓	✓	✓	**95.30**

Table 6: Effect of rubric-based process reward. The default policy uses only the environment task reward.

Method	Success (%) ↑	Δ vs. Default
Skill-V (default)	**95.30**	–
+ Process reward	93.80	−1.5

under retrospective evaluation. In contrast, all revisions committed by full Skill-V pass the complete gate. Consequently, enforcing this validation improves the overall task success rate from 75.0% to 95.3%. This substantial gain provides consistent downstream evidence that preventing unsupported revisions critically impacts library evolution.

Applicability-aware routing improves skill use. Relying solely on semantic retrieval severely limits the success rate to 71.9% (Table 5). Retaining deterministic contract checks recovers performance to 89.1%, omitting the LLM judge still drops success from the full 95.3%. Throughout training, this judge labels 29.6% of retrieved candidates as confidently inapplicable, confirming that semantic similarity frequently surfaces irrelevant skills that misguide the policy. These stepped ablations separate the critical contributions of both deterministic and contextual filtering.

Skill-V performs strongly without additional reward shaping. Although Skill-V uses rubrics to diagnose and revise skill contracts, these rubrics could theoretically be converted into process rewards to directly supervise the policy. This raises the alternative hypothesis that the benefits of Skill-V stem from reward shaping rather than improved external knowledge. We examine this possibility by augmenting the default Skill-V objective with a rubric-based process reward (Table 6). Incorporating this dense reward yields a slightly lower success rate of 93.8%, compared to 95.3% for the default configuration. Consequently, we retain the simpler task-reward-only objective for our main experiments. Although process rewards might prove beneficial in other contexts, this comparison confirms that Skill-V achieves its strong performance through verifiable knowledge evolution rather than dense reward shaping.

5 Conclusion

We introduced Skill-V, a verifiable self-evolving skill library that shifts agent learning from pure knowledge accumulation to rigorous knowledge maintenance. By representing skills as falsifiable contracts, our framework uses task failures to drive skill addition and leverages contract-environment disagreements to trigger targeted boundary revisions. An evidence gate ensures these self-generated updates remain entirely non-regressive. During inference, an applicability-aware filter effectively excludes confidently inapplicable candidates, preventing irrelevant skills from misguiding the policy. Empirical evaluations across ALFWorld and WebShop demonstrate that Skill-V achieves state-of-the-art performance while maintaining a highly compact library, successfully circumventing the catastrophic forgetting that plagues growth-oriented methods. As interactive agents advance toward recursive self-improvement, the ability to safely verify and correct procedural memory becomes critical. By prioritizing continuous maintenance over unrestrained expansion, Skill-V provides a principled foundation for reliable and autonomously improving agents.

AI USE STATEMENT

We used AI tools for language polishing, including improving grammar, and clarity. All AI-assisted edits were reviewed by the authors to preserve the intended scientific meaning. In addition, our method, Skill-V, uses an LLM-based agent policy and employs DeepSeek-V4-Flash for applicability judgment, failure-driven skill proposals, and rubric compilation and revision. The model configurations are detailed in the experimental setup and Appendix C.4.

REFERENCES

Anthropic. The claude 3 model family: Opus, sonnet, haiku. URL https://api.semanticscholar.org/CorpusID:270640496.

Yanping Chen, Weijie Shi, Wen Yang, and Jiajie Xu. Task decomposition-guided reranking for adaptive agent skill retrieval. *CoRR*, abs/2607.06283, 2026. doi: 10.48550/ARXIV.2607.06283. URL https://doi.org/10.48550/arXiv.2607.06283.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. In Inês Lynce, Nello Murano, Mauro Vallati, Serena Villata, Federico Chesani, Michela Milano, Andrea Omicini, and Mehdi Dastani (eds.), *ECAI 2025 - 28th European Conference on Artificial Intelligence, 25-30 October 2025, Bologna, Italy - Including 14th Conference on Prestigious Applications of Intelligent Systems (PAIS 2025)*, volume 413 of *Frontiers in Artificial Intelligence and Applications*, pp. 2993–3000. IOS Press, 2025. doi: 10.3233/FAIA251160. URL https://doi.org/10.3233/FAIA251160.

DeepSeek-AI. Deepseek-v3.2: Pushing the frontier of open large language models. *CoRR*, abs/2512.02556, 2025. doi: 10.48550/ARXIV.2512.02556. URL https://doi.org/10.48550/arXiv.2512.02556.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=Sx038qxjek.

Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. Simplemem: Efficient lifelong memory for LLM agents. *CoRR*, abs/2601.02553, 2026a. doi: 10.48550/ARXIV.2601.02553. URL https://doi.org/10.48550/arXiv.2601.02553.

Shuaiqi Liu, Zhengkai Lin, Yuxiang Zhang, Yuanyi Ren, Yue Wu, Yongbin Li, Zheng Wang, Zhihang Fu, and Jieping Ye. The path to recursive self-improving agents: Foundation, framework, and future directions. 2026b.

Yuxuan Liu, Zhaochen Su, Lingyun Xie, Yuhao Zhang, Qing Zong, Jiahe Guo, Zhongwei Xie, Yiyan Ji, Yauwai Yim, Hongyu Luo, Xiyu Ren, Ruan Chenyu, Haoran Li, and Yangqiu Song. Skillrevise: Improving llm-authored agent skills via trace-conditioned skill revision. *CoRR*, abs/2606.01139, 2026c. doi: 10.48550/ARXIV.2606.01139. URL https://doi.org/10.48550/arXiv.2606.01139.

OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/ARXIV.2303.08774. URL https://doi.org/10.48550/arXiv.2303.08774.

Siru Ouyang, Jun Yan, Yanfei Chen, Rujun Han, Zifeng Wang, Bhavana Dalvi Mishra, Rui Meng, Chun-Liang Li, Yizhu Jiao, Kaiwen Zha, Maohao Shen, Vishy Tirumalashetty, George Lee, Jiawei Han, Tomas Pfister, and Chen-Yu Lee. Skillos: Learning skill curation for self-evolving agents. *CoRR*, abs/2605.06614, 2026. doi: 10.48550/ARXIV.2605.06614. URL https://doi.org/10.48550/arXiv.2605.06614.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph Gonzalez. Memgpt: Towards llms as operating systems. *ArXiv*, abs/2310.08560, 2023. URL https://api.semanticscholar.org/CorpusID:263909014.

Hongji Pu, Xinyuan Song, and Liang Zhao. Skillops: Managing LLM agent skill libraries as self-maintaining software ecosystems. *CoRR*, abs/2605.13716, 2026. doi: 10.48550/ARXIV.2605.13716. URL https://doi.org/10.48550/arXiv.2605.13716.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551. Curran Associates, Inc., 2023. doi: 10.52202/075280-2997. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/d842425e4bf79ba039352da0f658a906-Paper-Conference.pdf.

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. *CoRR*, abs/2605.06130, 2026. doi: 10.48550/ARXIV.2605.06130. URL https://doi.org/10.48550/arXiv.2605.06130.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew J. Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *CoRR*, abs/2010.03768, 2020. URL https://arxiv.org/abs/2010.03768.

Gemini Team. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805, 2023. doi: 10.48550/ARXIV.2312.11805. URL https://doi.org/10.48550/arXiv.2312.11805.

Kimi Team. Kimi K2: open agentic intelligence. *CoRR*, abs/2507.20534, 2025. doi: 10.48550/ARXIV.2507.20534. URL https://doi.org/10.48550/arXiv.2507.20534.

Yash Vishe, Rohan Surana, Xunyi Jiang, Zihan Huang, Xintong Li, Nikki Lijing Kuang, Tong Yu, Ryan A. Rossi, Jingbo Shang, Julian J. McAuley, and Junda Wu. Skill-r1: Agent skill evolution via reinforcement learning. *CoRR*, abs/2605.09359, 2026. doi: 10.48550/ARXIV.2605.09359. URL https://doi.org/10.48550/arXiv.2605.09359.

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. *Trans. Mach. Learn. Res.*, 2024, 2024a. URL https://openreview.net/forum?id=ehfRiF0R3a.

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), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 1529–1550. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.69. URL https://doi.org/10.18653/v1/2026.acl-long.69.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5362–5383, 2024b. doi: 10.1109/TPAMI.2024.3367329.

Rong Wu, Xiaoman Wang, Jianbiao Mei, Pinlong Cai, Daocheng Fu, Cheng Yang, Licheng Wen, Xuemeng Yang, Yufan Shen, Yuxin Wang, and Botian Shi. Evolver: Self-evolving LLM agents through an experience-driven lifecycle. *CoRR*, abs/2510.16079, 2025. doi: 10.48550/ARXIV.2510.16079. URL https://doi.org/10.48550/arXiv.2510.16079.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, Zeyu Zheng, Cihang Xie, and Huaxiu Yao. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. *CoRR*, abs/2602.08234, 2026. doi: 10.48550/ARXIV.2602.08234. URL https://doi.org/10.48550/arXiv.2602.08234.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. 2024.

Yifan Yang, Ziyang Gong, Weiquan Huang, Qihao Yang, Ziwei Zhou, Zisu Huang, Yan Li, Xuemei Gao, Qi Dai, Bei Liu, Kai Qiu, Yuqing Yang, Dongdong Chen, Xue Yang, and Chong Luo. Skillopt: Executive strategy for self-evolving agent skills, 2026. URL https://arxiv.org/abs/2605.23904.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/82ad13ec01f9fe44c01cb91814fd7b8c-Abstract-Conference.html.

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 *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Xunjian Yin, Xinyi Wang, Liangming Pan, Li Lin, Xiaojun Wan, and William Yang Wang. Gödel agent: A self-referential agent framework for recursively self-improvement. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27890–27913, 2025.

Shengtao Zhang, Jiaqian Wang, Ruiwen Zhou, Junwei Liao, Yuchen Feng, Weinan Zhang, Ying Wen, Zhiyu Li, Feiyu Xiong, Yutao Qi, Bo Tang, and Muning Wen. Memrl: Self-evolving agents via runtime reinforcement learning on episodic memory. *CoRR*, abs/2601.03192, 2026a. doi: 10.48550/ARXIV.2601.03192. URL https://doi.org/10.48550/arXiv.2601.03192.

Xiaoying Zhang, Zichen Liu, Yipeng Zhang, Xia Hu, and Wenqi Shao. Retroagent: From solving to evolving via retrospective dual intrinsic feedback. *CoRR*, abs/2603.08561, 2026b. doi: 10.48550/ARXIV.2603.08561. URL https://doi.org/10.48550/arXiv.2603.08561.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: LLM agents are experiential learners. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pp. 19632–19642. AAAI Press, 2024. doi: 10.1609/AAAI.V38I17.29936. URL https://doi.org/10.1609/aaai.v38i17.29936.

YanZhao Zheng, ZhenTao Zhang, Chao Ma, YuanQiang Yu, JiHuai Zhu, Yong Wu, Tianze Xu, Baohua Dong, Hangcheng Zhu, Ruohui Huang, and Gang Yu. Skillrouter: Skill routing for llm agents at scale, 2026. URL https://arxiv.org/abs/2603.22455.

Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, Y. Qiao, Zhaoxiang Zhang, and Jifeng Dai. Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory. *ArXiv*, abs/2305.17144, 2023. URL https://api.semanticscholar.org/CorpusID:258959262.

A Limitations and Broader Impacts

Environment types. Skill-V evaluates contracts using discrete text-based criteria. Applying this framework to visual or continuous control environments will require new multimodal evaluation mechanisms. Future research could integrate vision-language models to visually verify physical state changes. Alternatively, continuous environments might benefit from programmatic reward functions acting as executable rubrics.

Skill deletion. Our framework only updates operational boundaries to prevent unintended behavior changes. If an agent initially acquires a fundamentally flawed strategy, the system will tightly restrict its use rather than discarding it entirely. Removing these inefficient strategies will therefore require additional mechanisms for explicit skill deletion. Future systems could implement automated skill deprecation to permanently retire skills that consistently fail applicability checks.

Historical dependency. The strict validation step relies heavily on past trajectories to prevent performance regression. If the underlying rules of an environment change over time, these historical trajectories may no longer represent correct behavior. Adapting this historical replay approach to rapidly changing environments remains a necessary direction for future research. Future evolution mechanisms could adopt time-weighted buffers to decay obsolete evidence.

Broader impacts. Skill-V supports more transparent agent adaptation by making skill revisions explicit and checking them against observed evidence. Its contract framework could be extended to incorporate safety requirements beyond task success. Future work could develop domain-specific constraints and shared evaluation protocols to assess whether these mechanisms help detect and mitigate harmful updates, with the broader goal of enabling trustworthy recursive self-improvement.

B Implementation Details

This appendix describes the implementation of Skill-V in sufficient detail to separate the conceptual method from engineering choices. Skill-V maintains one training skill library and a synchronized evaluation copy. The training copy is updated exclusively using training trajectories. Before each validation round, the current training library is copied to the evaluation workers and remains strictly read-only throughout the validation. Rubric evaluations serve solely to diagnose and evolve the library. Unless explicitly stated otherwise, they are strictly excluded from the policy reward.

B.1 End-to-End Training Procedure

The Algorithm 1 summarizes the complete training loop. The same experience supports two complementary updates. Environment rewards optimize the policy, while terminal outcomes test the external skill library. Library evolution is performed every F training steps, with $F = 5$ in our experiments.

Here, DiagnoseContracts executes every applicable contract possessing an available rubric and compares its judgment with the environment outcome. SelectEvidence prioritizes rubric-outcome disagreements while retaining supporting success and failure cases. ProposeContractUpdates serves two roles. It compiles rubrics for pending semantic skills and proposes targeted revisions to existing contracts. A newly added semantic skill becomes retrievable immediately. In contrast, a compiled or revised contract becomes active only after passing deterministic validation. ValidateContractUpdate combines schema and traceability checks, semantic preservation, non-empty executable support, and evidence non-regression.

B.2 Stored Skills as Versioned Contracts

Free-form instructions alone do not distinguish what a skill means from how its execution can be checked. Skill-V therefore stores each skill as a versioned contract $s_i = (\phi_i, p_i, \rho_i, \nu_i)$. Here, ϕ_i specifies the semantic identity and intended scope, p_i provides the reusable procedure, ρ_i connects the skill to observable trajectory evidence, and ν_i records its evolution. This explicit separation allows the executable boundary to be corrected without silently altering the core identity of the skill.

Algorithm 1 Skill-V training and library evolution

Require: Policy π_θ, skill library $\mathcal{L}$, update interval F, retrieval budgets K_g, K_t

for training step $t = 1, \ldots, T$ **do**
Skill-conditioned interaction
  Sample task batch $\mathcal{X}_t$ and initialize $\mathcal{E}_t \leftarrow \emptyset$
  for all $x \in \mathcal{X}_t$ **do**
    $C(x) \leftarrow \textsc{SemanticRetrieve}(x, \mathcal{L}, K_g, K_t)$
    $S(x) \leftarrow \textsc{ApplicabilityFilter}(x, C(x))$
    $(\tau, y, r) \leftarrow \textsc{Rollout}(\pi_\theta, x, S(x))$
    $\mathcal{E}_t \leftarrow \mathcal{E}_t \cup \{(x, \tau, y, r)\}$
  end for
  Compute GRPO advantages from environment rewards r
  if $t \bmod F = 0$ **then**
Outcome-grounded library evolution
    $\mathcal{E} \leftarrow$ recent training experiences
    $\mathcal{F} \leftarrow \{e \in \mathcal{E} : y(e) = 0\}$
    $\mathcal{P}_{\text{new}} \leftarrow \textsc{ProposeNewSkills}(\mathcal{F}, \mathcal{L})$
    $\mathcal{L} \leftarrow \textsc{AddPendingSkills}(\mathcal{L}, \mathcal{P}_{\text{new}})$
    $D \leftarrow \textsc{DiagnoseContracts}(\mathcal{L}, \mathcal{E})$
    $\widetilde{\mathcal{E}} \leftarrow \textsc{SelectEvidence}(D)$ ▷ disagreements first
    $\mathcal{P} \leftarrow \textsc{ProposeContractUpdates}(\mathcal{L}, \widetilde{\mathcal{E}})$
    for all $p \in \mathcal{P}$ **do**
      if $\textsc{ValidateContractUpdate}(p, \mathcal{L}, \mathcal{E})$ **then**
        $\mathcal{L} \leftarrow \textsc{CommitVersion}(\mathcal{L}, p)$
      else
        Retain the current contract version
      end if
    end for
  end if
  $\theta \leftarrow \text{GRPO}\textsc{Update}(\theta, \mathcal{E}_t)$
end for
return π_θ and $\mathcal{L}$

Table 7: Fields of a stored Skill-V contract and their roles in skill use and evolution.

Component	Stored content	Role
Semantic specification ϕ_i	`skill_id`, `title`, `principle`, `when_to_apply`, protected `skill_constraints`	Defines what capability the skill represents and establishes the semantic boundary that a revision is not allowed to overwrite.
Reusable procedure p_i	Ordered natural-language steps	Conditions the policy during interaction and may be refined when experience reveals an incomplete strategy.
Executable rubric ρ_i	Applicability, required steps, forbidden behaviors, preconditions, postconditions, and termination conditions	Translates the intended behavior into conservative criteria that can be checked deterministically against an executed trajectory.
Evidence interface	Criterion identifier, severity, weight, match mode, source, operator, and pattern	Grounds each executable criterion directly in the task, action, observation, or trajectory rather than in the policy's hidden reasoning.
Evolution record ν_i	Skill version, rubric version, traceability, update reason, and evidence case identifiers	Links each committed version to its source skill and to the specific experience that motivated the change.

Table 8 traces two revisions selected from the step-150 checkpoint of our main ALFWorld experiment. Each case follows the exact same evidence-to-update sequence. An environment outcome

Table 8: How outcome evidence drives contract evolution in the main ALFWorld experiment at step 150. Skill-V tightens an under-specified boundary that admits failed behavior and relaxes an over-restrictive boundary that rejects successful behavior. Each candidate is evaluated against its previous version on identical replay support. The second revision is accepted as a non-regressive improvement, although it has not yet reached target thresholds.

Case 1: Tightening an under-specified boundary				`gen_004`: *Track Counts & Progress*
Skill intent, preserved: Track how many goal objects remain and terminate only after the required count is reached.				
Counterexample *Contract: pass; outcome: failure.* Only one target object was collected in a quantity-two task.	$\Longrightarrow$	**Rubric change** **Before:** no executable distinct-object count. **After:** require ≥ 2 `take` events on distinct objects with a hard `event_count` criterion.	$\Longrightarrow$	**Replay validation** **Old** **Revised** Disc $\downarrow$ 0.400 **0.100** FNR $\downarrow$ 1.000 **0.000** BAcc $\uparrow$ 0.500 **0.917** Support: $10 \rightarrow 10$ **Accepted** Skill v1$\rightarrow$v2; Rubric v1$\rightarrow$v2.
Case 2: Relaxing an over-restrictive boundary				`gen_008`: *Avoid Redundant Rechecks*
Skill intent, preserved: Record inspected locations and avoid revisiting them unless new evidence suggests a state change.				
Counterexample *Contract: fail; outcome: success.* A successful trajectory contained a brief pair of identical consecutive actions.	$\Longrightarrow$	**Rubric change** **Before:** flag two identical consecutive actions as a loop. **After:** preserve loop detection but trigger it after three identical consecutive actions.	$\Longrightarrow$	**Replay validation** **Old** **Revised** Disc $\downarrow$ 0.500 **0.333** FNR $\downarrow$ 1.000 **0.667** BAcc $\uparrow$ 0.500 **0.667** Support: $12 \rightarrow 12$ **Accepted** Skill v1$\rightarrow$v2; Rubric v2$\rightarrow$v3.

contradicts the current rubric, the executable boundary is modified while the skill intent remains fixed, and the candidate is replayed on the identical observed support before being committed.

The first revision compiles a semantic multiplicity requirement into an executable distinct-object counter. The second revision utilizes successful replay as a counterexample against an over-restrictive loop detector. These cases demonstrate that rubric evolution is not equivalent to simply accumulating constraints. Instead, it dynamically calibrates the executable boundary in either direction and commits a new version only when replay evidence proves strictly non-regressive.

The checked-in initial libraries utilize a compact sentence-rubric schema for readability. At load time, this representation is normalized into the internal criterion groups outlined in Table 7. Each executable criterion contains a stable identifier, a description, a severity in $\{\text{HARD}, \text{SOFT}\}$, a weight in $[0, 1]$, a match mode in $\{\text{ANY}, \text{ALL}\}$, and at least one observable evidence matcher. Hard criteria employ unit weight. Protected semantic constraints reside within the same contract but are never treated as executable evidence unless a conservative proxy has been explicitly compiled for them.

To ensure strict security and stability, Skill-V never executes model-generated code. Evidence matchers remain restricted to a predefined declarative operator set. All domains support case-insensitive literal matching and regular expressions over the task, executed actions, environment observations, or their flattened trajectory. ALFWorld additionally supports bounded event-sequence and event-count monitors. The flattened trajectory contains only actions and observations. Policy thoughts are strictly excluded, ensuring that a planned but unexecuted action cannot fraudulently satisfy a rubric criterion.

B.3 EXECUTABLE RUBRIC EVALUATION

For a given task x, the verifier first evaluates the declarative applicability conditions of each contract. Empty applicability conditions are treated as universally applicable. Invalid generated contracts fail closed and are automatically excluded from the policy prompt. If a valid contract is deemed applicable, the verifier systematically evaluates its execution criteria against trajectory τ.

Let $m_{ij}(\tau) \in \{0, 1\}$ indicate whether criterion c_{ij} is satisfied after accounting for its criterion type. In particular, a forbidden-behavior criterion passes when its specified evidence pattern remains absent. Let w_{ij} denote its weight. Across all applicable contracts possessing executable criteria, Skill-V computes

$$q(\tau) = \frac{\sum_{s_i \in \mathcal{A}(x)} \sum_j w_{ij} m_{ij}(\tau)}{\sum_{s_i \in \mathcal{A}(x)} \sum_j w_{ij}}, \qquad \hat{y}^{\mathrm{rub}}(\tau) = \mathbb{I}[q(\tau) \geq \eta \ \wedge \ H(\tau) = 0], \tag{5}$$

where $H(\tau)$ indicates whether any applicable hard criterion is violated. We utilize a threshold of $\eta = 0.7$. When no applicable contract possesses an executable criterion, the aggregate score and judgment are undefined, and the trajectory generates no rubric-outcome evidence.

In addition to the aggregate judgment, the implementation logs per-skill applicability, pass/fail state, hard-criterion state, weighted score, and failed criterion identifiers. These diagnostics localize candidate revisions. They are deliberately not interpreted as causal credit assignment, because the environment outcome is evaluated for the complete trajectory rather than attributed to an individual skill.

B.4 FAILURE-DRIVEN SKILL ACQUISITION

At each library-update step, the acquisition updater receives failed training trajectories alongside the titles of existing skills. It considers at most five failure examples and retains the final five interaction steps of each example to provide context. The external language model then proposes up to a configured maximum number of reusable skills. Each proposal contains a title, principle, and intended scope. Existing titles serve as a deduplication hint, and the implementation assigns collision-free identifiers locally.

A newly acquired skill is immediately converted into a semantic contract. Its principle serves as its initial procedure, and an immutable semantic constraint is instantiated with the status PENDINGCOMPILATION. At this point, the skill can participate in embedding retrieval, applicability judgment, and policy prompting. However, it cannot produce a rubric judgment because it lacks executable criteria. This strict separation prevents an uncalibrated model-generated rule from acting as a verifier.

Pending skills receive prioritization by the contract updater. A successful compilation adds conservative executable proxies and transitions the skill into its first executable contract. A failed or rejected compilation retains the pending status, allowing the semantic skill to remain available to the policy for subsequent interactions.

B.5 OUTCOME-GROUNDED CONTRACT DIAGNOSIS

For each recent training experience (x, τ, y), Skill-V evaluates all valid contracts whose declarative applicability conditions match x. Comparing the aggregate rubric judgment with the environment outcome produces the four diagnostic cases detailed in Table 9.

The updater selects at most 12 recent cases, prioritizing the two disagreement quadrants while retaining agreement cases from both outcome classes. It then retrieves up to 24 relevant contracts to form the proposal context. Relevance favors contracts implicated by failed criteria, contracts whose semantic fields overlap the current task, and pending contracts awaiting compilation.

The external updater may refine applicability conditions, procedural steps, or executable criteria. For an existing skill, however, every protected semantic constraint must be copied perfectly. Consequently, a revision can successfully repair the operational boundary of a skill, but it remains structurally prohibited from silently replacing its semantic identity.

Table 9: Interpretation of rubric-outcome evidence.

Rubric	Outcome	Agreement	Interpretation
Pass	Success	Yes	Evidence supporting the current operational boundary.
Fail	Failure	Yes	Consistent negative evidence.
Pass	Failure	No	The contract may be too permissive or underspecified.
Fail	Success	No	The contract may be too restrictive.

B.6 Deterministic Revision Validation

Language-model proposals are never committed directly. Skill-V first verifies the contract schema, unique criterion identifiers, allowed matcher types, criterion weights and severities, criterion-to-procedure traceability, source skill identity, and monotonically increasing versions. Following structural validation, it performs shadow replay on the observed evidence pool associated with the update.

For a skill s_i, rubric pass/fail is evaluated as a prediction of the environment outcome. With the standard definitions of true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN), we compute

$$\mathrm{Disc} = \frac{\mathrm{FP} + \mathrm{FN}}{N}, \qquad \mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{TP} + \mathrm{FN}}, \qquad \mathrm{BAcc} = \frac{1}{2}\left(\frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} + \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}\right). \quad (6)$$

Let ρ_i and ρ_i' denote the current and proposed rubrics. A proposal is considered evidence-preserving only if

$$\mathrm{Disc}(\rho_i') \le \mathrm{Disc}(\rho_i), \quad \mathrm{FNR}(\rho_i') \le \mathrm{FNR}(\rho_i), \quad \mathrm{BAcc}(\rho_i') \ge \mathrm{BAcc}(\rho_i). \quad (7)$$

It must additionally possess non-empty executable support. If a hard executable criterion changes, at least one quantity in Eq. 7 must improve strictly. The target values used to prompt the updater ($\mathrm{FNR} \le 0.05$ and $\mathrm{Disc} \le 0.15$) merely indicate whether an accepted proposal reaches a desired operating region. They do not constitute additional hard acceptance requirements.

The implementation robustly handles degenerate evidence pools using standard conventions. When no valid applicable and executable trajectory exists, support is zero, disagreement and FNR default to one, and balanced accuracy defaults to zero. The candidate is subsequently rejected by the non-empty-support condition. If the evidence contains no positive outcomes, FNR is zero and the missing true-positive rate is treated as one. If it contains no negative outcomes, the missing true-negative rate is treated as one. Current and candidate contracts are replayed over the exact same observed trajectory pool, although their effective supports may diverge when a revision alters applicability.

An accepted proposal increments both the skill and rubric versions and replaces the current contract in place. A rejected proposal leaves the current version completely unchanged. Because revision occurs in place, accepted revisions fundamentally do not increase library size.

B.7 Semantic Retrieval and Applicability Filtering

Skill-V embeds the task instruction and each skill's semantic representation using Qwen3-Embedding-0.6B. General and task-specific skills are ranked separately. For final budgets $K_g = 6$ and $K_t = 5$, the implementation first forms candidate pools of size $\lceil \alpha K_g \rceil$ and $\lceil \alpha K_t \rceil$, where $\alpha = 2$.

An external language model labels each candidate as Applicable, NotApplicable, or Uncertain, returning this label alongside a confidence score. Only NotApplicable predictions with confidence at least $\delta = 0.8$ are removed. Applicable and uncertain candidates remain fully eligible. Among these remaining candidates, Skill-V removes those whose cosine similarity falls more than $\epsilon = 0.15$ below the highest remaining score, ultimately selecting up to the original category-specific budget.

The judge must return one valid decision for every supplied candidate. If an API request fails or returns an incomplete decision set after two attempts, the system gracefully falls back to the original embedding Top-K result. Decisions are cached using the normalized task, model version, prompt

version, skill identifier, skill and rubric versions, and a fingerprint of the semantic fields. Updating a skill therefore invalidates stale applicability decisions naturally through the cache key.

C Experimental Details

C.1 Environments and Metrics

We evaluate Skill-V in two interactive domains. ALFWorld contains embodied household tasks and reports success rates for Pick, Look, Clean, Heat, Cool, Pick2, and the complete task set. Its binary outcome is derived from a positive terminal task reward. WebShop requires an agent to search for and purchase a product satisfying a natural-language request. We report its average task score on a 0–100 scale alongside its overall success rate. For contract diagnosis, a WebShop trajectory is labeled successful only when its normalized environment score reaches at least 0.99.

All headline results in Table 1 correspond to the selected step-150 checkpoints. We deliberately avoid choosing a separate peak checkpoint for each reported metric. The main table consequently reports a 95.3% overall success rate for ALFWorld, and an average score of 92.5 with an 85.9% success rate for WebShop.

C.2 Training and Evaluation Configuration

Both domains employ GRPO with eight rollouts per instruction, a learning rate of 10^{-6}, a KL loss coefficient of 0.01, and an invalid-action penalty of 0.1. The policy is initialized from the corresponding 7B SFT checkpoint and trained across eight GPUs. Table 10 lists domain-specific settings. These settings reflect the effective launch configuration of the reported runs, incorporating any subsequent command-line overrides.

Table 10: Domain-specific configuration for the reported step-150 results. Update limits refer to per-library-update event budgets rather than global limits.

Configuration	ALFWorld	WebShop
Training batch size	16	32
Validation batch size	64	128
Rollouts per instruction	8	8
Maximum environment steps	50	15
Maximum prompt length	8192	6000
Maximum response length	512	768
Validation sampling	sampling	greedy
Validation temperature	0.4	0.4 (sampling disabled)
Library update frequency	5 steps	5 steps
Maximum new semantic skills	10	3
Maximum contract revisions	10	3
Rubric pass threshold η	0.7	0.7
Outcome success threshold	positive task reward	normalized score ≥ 0.99
General/task-specific Top-K	6/5	6/5
Candidate multiplier α	2.0	2.0
Applicability confidence δ	0.8	0.8
Embedding margin ϵ	0.15	0.15
Tensor parallel size	4	4
Checkpoint/validation interval	10/5 steps	10/5 steps
Rubric process reward	disabled	disabled

C.3 Initial Skill Libraries

The initial ALFWorld library contains 12 general skills, 32 task-specific skills, and 11 common mistakes. The 44 general and task-specific entries function as retrievable contracts. Common mistakes are stored separately, with up to five randomly inserted into the prompt as negative guidance. The initial WebShop library contains 15 general skills, 39 task-specific skills, and 12 common mistakes. Thus, the complete initial stored-library sizes total 55 and 66 entries, respectively.

Table 11: Skill-library evolution through step 150. Total entries include common mistakes, which are stored separately from retrievable contracts.

Domain	Initial	Step 150	Net Growth	Accepted Revisions
ALFWorld	55	205	150	10
WebShop	66	153	87	14

The initial contracts were compiled offline from previously generated procedural skills. Unsupported semantic requirements remain protected constraints but are disabled as executable proxies. This prevents the initial library from erroneously presenting a deterministic matcher as complete verification of a richer semantic claim.

C.4 External Models

Qwen3-Embedding-0.6B is utilized exclusively for semantic retrieval. The reported runs employ `deepseek-v4-flash` as the external language model for applicability judgment, failure-driven skill proposals, and rubric compilation or revision. The policy itself acts as a separate 7B actor and remains the only model updated by gradient-based reinforcement learning. External-model responses are parsed as JSON and verified locally. The updater structurally cannot bypass the contract validator or the evidence-preservation gate.

The applicability judge operates under a limit of 2,048 completion tokens, two attempts, and a 30-second request timeout. A five-minute circuit-breaker cooldown avoids rapid repeated requests after service failures. Semantic acquisition and contract maintenance utilize limits of 4,096 and 8,192 completion tokens, respectively, with two parsing attempts.

D Library Evolution and Diagnostics

D.1 Library Evolution Statistics

Table 11 reports the library state at the selected checkpoints. Because revisions replace existing versions in place, they fundamentally do not contribute to size growth. For ALFWorld, the step-150 library contains 162 general skills, 32 task-specific skills, and 11 common mistakes. The 150-entry net growth arises entirely from newly acquired semantic skills. Ten contract revisions were successfully accepted by step 150 at update steps 5, 60, 65, 85, 120, 125, and 150. The corresponding numbers of accepted revisions were 2, 1, 2, 1, 1, 1, and 2. This exact count is confirmed independently by both the summed training metric and the version history stored within the checkpoint.

WebShop contains 153 stored entries at step 150 and records 14 accepted revisions. Its net growth is computed purely from the final and initial library sizes, as the revisions themselves execute as in-place updates. These robust statistics demonstrate that the reported method does far more than simply grow its library. It actively calibrates the operational boundaries of persistent skills as empirical evidence accumulates.

D.2 Checkpoint Diagnostics

The ALFWorld training log retains additional aggregate diagnostics for the selected checkpoint (Table 12). These metrics characterize the final training state but do not substitute for the standard task-success metrics utilized for comparison with prior work.

The extremely low aggregate disagreement value indicates that executable rubric judgments remain highly consistent with environment outcomes at this checkpoint. This value serves a descriptive purpose rather than acting as a causal measure of policy quality, as the outcome is trajectory-level and several contracts may apply simultaneously to the same task.

Table 12: Aggregate ALFWorld diagnostics at the selected step-150 checkpoint.

Metric	Value
Validation success (%)	95.3
Validation task score	8.130
Training-episode success (%)	86.7
Validation rubric score	0.894
Rubric-outcome disagreement	0.033
Hard-criterion violation rate	0.062
Stored library entries	205

E PROMPT TEMPLATES

This section records the effective instructions supplied to the external language models. Runtime prompts dynamically substitute the task, candidate skills, trajectory cases, diagnostics, and current contracts into the bracketed fields.

E.1 APPLICABILITY JUDGMENT

Decide whether each candidate skill is semantically applicable to the task objective. Applicability means the skill could be useful for solving this task at some point in the episode. Judge only the task objective and declared skill scope. Do not require that a precondition already holds in the current environment state. Handle paraphrases and implied state changes.

Use `applicable` when the skill is clearly compatible, `not_applicable` when it is clearly incompatible, and `uncertain` when compatibility cannot be established confidently. Return every supplied skill identifier exactly once as a JSON object containing the decision, confidence in $[0, 1]$, and a short analytical reason.

[Input Fields] Task objective; Candidate skills (ID, Title, Principle, Scope).

E.2 FAILURE-DRIVEN SEMANTIC ACQUISITION

Analyze the supplied failed trajectories and suggest new skills to add to the skill library. Avoid duplicating existing skill titles. Generate between one and the configured maximum number of actionable, reusable skills that could prevent these failures. Each skill must contain a title, a brief principle, and an explicit condition describing when it should be applied. Return only a JSON object containing a `skills` array.

[Input Fields] Failed trajectories (up to five failures, final five steps each); Existing skill titles.

E.3 CONTRACT COMPILATION AND REVISION

Maintain a skill library under explicit hybrid contracts. First derive semantic constraints from the skill principle and make every constraint traceable to the procedure. Then add conservative executable proxies using safe, domain-appropriate matchers. Never claim that a proxy completely verifies a richer semantic constraint.

Use rubric-fail/outcome-fail cases to repair a skill while preserving valid criteria. A rubric-pass/outcome-fail case indicates a missed failure and may require an added or conditionalized criterion. A rubric-fail/outcome-pass case indicates an over-restrictive rubric and may require narrower applicability or relaxed soft criteria. Preserve a supported contract unless repeated evidence justifies a change. For any revision, copy every protected semantic constraint exactly. Prefer revising a relevant contract over adding a duplicate. Return only JSON that perfectly follows the supplied contract schema and update limits.

[Input Fields] Trajectory cases; Per-skill rubric diagnostics; Relevant existing contracts.