Library Drift: Diagnosing and Fixing a Silent Failure Mode in Self-Evolving LLM Skill Libraries

Xing Zhang [1] **Yanwei Cui** [1] **Guanghui Wang** [1] **Ziyuan Li** [2] **Wei Qiu** [2] **Bing Zhu** [2] **Peiyang He** [1]

Abstract

Self-evolving skill libraries face a silent failure mode we term *library drift*: unbounded skill accumulation without outcome-driven lifecycle management causes retrieval degradation, false-positive injections, and performance stagnation. Recent evaluation confirms the symptom (LLM-authored skills deliver +0.0pp gain while human-curated ones deliver +16.2pp; SkillsBench (Li et al., 2026)), yet the underlying mechanism has not been isolated. We provide (1) a **reproducible trigger**: ablations that isolate drift: one disables skill injection (flat floor, +0.002), one imposes premature retirement (active harm, −0.019); (2) **trace-level diagnostics**: an append-only evidence log with per-skill contribution scores, attribution verdicts, and router engagement metrics that make the failure visible before it reaches end-task scores; and (3) a **verified fix**: a minimal governance recipe (outcome-driven retirement + bounded active-cap + meta-skill authoring prior) that lifts held-out pass@1 from a 0.258 baseline to a late-window mean of 0.584 (rolling gain +0.328) on MBPP+ hard-100 over 100 rounds. Eight ablations decompose which governance mechanisms are load-bearing and which are subsumed, providing a concrete playbook for diagnosing library drift in any self-evolving agent.

1. Introduction

Self-evolving skill libraries, pioneered by Voyager (Wang et al., 2023), let frozen LLM agents accumulate reusable procedural knowledge without weight updates. The promise is compounding: each solved task deposits a skill that accelerates future tasks. Yet SkillsBench (Li et al., 2026) reveals a striking gap: LLM-authored skills deliver +0.0pp over no-skill baselines while human-curated ones deliver +16.2pp. A recent survey (Zhang et al., 2026a) of 20+ such systems reports that lifecycle management (versioning, conflict detection, deprecation) is "largely neglected." The libraries grow, but the agents do not improve.

We identify a specific, diagnosable failure mode behind this gap: **library drift**. As skills accumulate without quality gates, the library degrades retrieval precision, injects stale or harmful guidance, and the agent's effective performance stagnates or drops below its no-skill baseline. The failure is *silent*: end-task metrics decline gradually with no explicit error signal, making trace-level diagnostics essential. While we demonstrate drift on single-call code tasks, the mechanism applies more acutely to multi-step agents, where a single misleading injection compounds through downstream tool calls and planning decisions.

This paper makes three contributions toward understanding and fixing failure modes in agentic systems:

1. **Failure definition and reproducible trigger** (Sec. 3): We define library drift operationally and show two ablations that bracket it: one establishes the no-skill floor (A1, +0.002), one triggers active harm via premature retirement (A4, −0.019), providing minimal reproductions for any system.

2. **Trace-level diagnostics beyond final scores** (Sec. 4): An append-only evidence log with per-skill contribution scores, attribution verdicts, and router engagement metrics exposes drift at per-skill granularity, before it reaches aggregate pass@1.

3. **Verified fix with documented trade-offs** (Sec. 5): A minimal governance recipe (Ratchet): outcome-driven retirement, bounded active-cap, meta-skill authoring prior, lifts held-out pass@1 by +0.328. Eight ablations decompose which mechanisms are load-bearing and which are safely removed.

[1]AWS Generative AI Innovation Center [2]HSBC Holdings Plc., HSBC Technology Center, China. Correspondence to: Peiyang He <peiyan@amazon.com>.

Accepted to the ICML 2026 Workshop on Failure Modes in Agentic AI (FAGEN@ICML 2026), Seoul, South Korea.

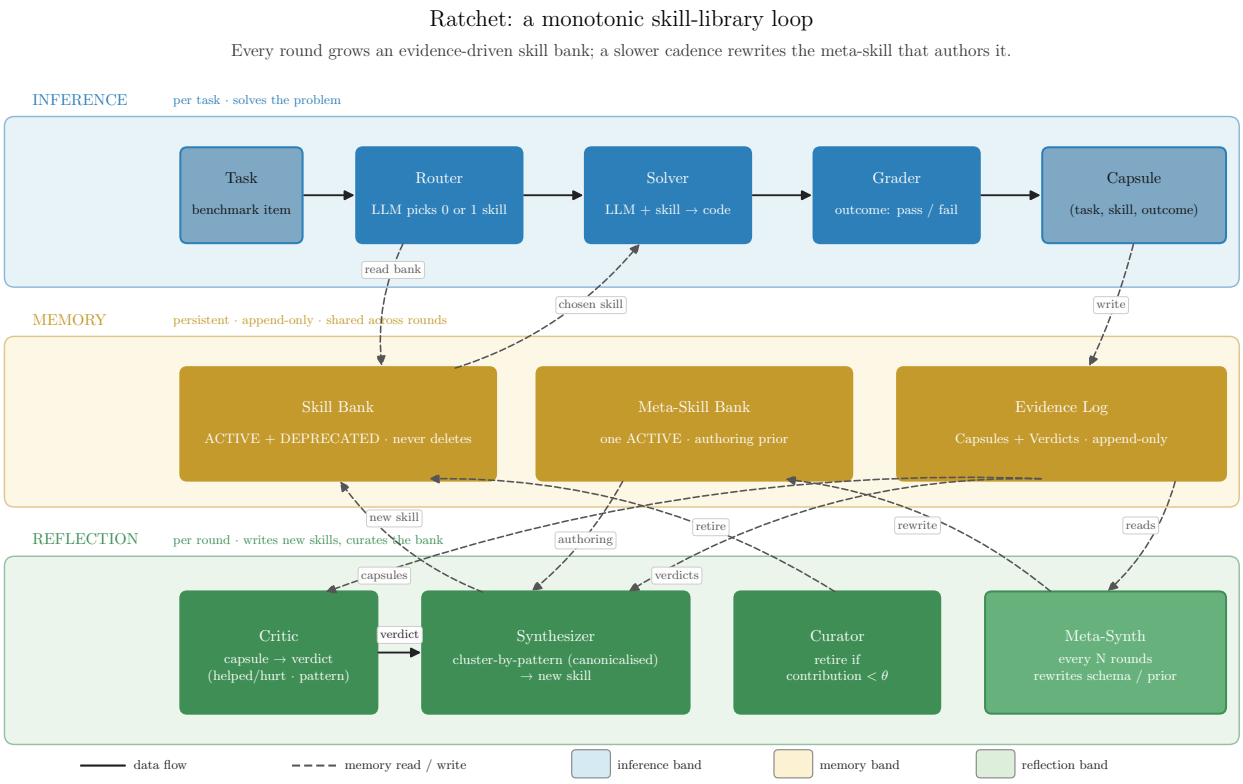

Figure 1. The Ratchet loop and where library drift is diagnosed and fixed. **Inference** (top): each task flows through Router → Solver → Grader → Capsule. **Memory** (middle): Skill Bank, Meta-Skill, and Evidence Log. **Reflection** (bottom): the Critic produces attribution verdicts (the diagnostic signal); the Curator retires under-performers and enforces the bounded cap (the fix). Without outcome-driven retirement and the bounded cap, the Skill Bank accumulates unchecked and library drift emerges.

2. Background and Related Work

Self-evolving skill libraries. Voyager (Wang et al., 2023) pioneered the ever-growing executable skill library in Minecraft; ExpeL (Zhao et al., 2024) extracts textual insights from trajectories but lacks outcome-driven retirement; AutoManual (Chen et al., 2024) compiles instruction manuals; CASCADE (Huang et al., 2025) pairs meta-skills with cumulative creation; AutoSkill (Yang et al., 2026) adds versioning but does not retire on measured per-task contribution. Concurrent work advances orthogonal axes: EvolveR (Wu et al., 2025) distills trajectories into strategic principles; Trace2Skill (Ni et al., 2026) induces skills from trace pools; Strategy Genes (Wang et al., 2026) attaches failure history as metadata. A recent survey (Zhang et al., 2026a) of 20+ such systems reports that lifecycle management is "largely neglected"; none pairs outcome-driven retirement with a bounded active-cap, the two mechanisms our ablation analysis identifies as load-bearing for preventing drift.

Agent failure modes. Prior work catalogs failure modes in tool use (Schick et al., 2023), planning (Yao et al., 2023), and self-correction (Shinn et al., 2023; Madaan et al., 2023). Library drift is a distinct failure mode specific to *persistent cross-task memory*: it emerges only when accumulated artifacts degrade future performance through a retrieval bottleneck. Catastrophic forgetting in weight-update systems occurs when new gradients overwrite previously learned representations (Kirkpatrick et al., 2017). Library drift is its frozen-weight counterpart: persistent skill artifacts replace neural weights as the degradation substrate, but the same fundamental issue applies: accumulating new information can degrade the system's prior performance unless lifecycle management actively protects useful knowledge. Where EWC adds a regulariser over parameter space, our fix adds retirement and a bounded cap over skill space.

Diagnostic signals. LLM-as-judge (Zheng et al., 2023) reports >80% agreement with human preferences; we extend this into per-skill *attribution* verdicts (helped/hurt/neutral) from a closed label set, requiring ≥3 failures sharing a canonical pattern before any skill is born. This detects drift before it surfaces in aggregate metrics, an early-warning system rather than a post-hoc evaluator.

3. Library Drift: Definition and Trigger

3.1. Operational definition

Library drift occurs when a self-evolving skill library's accumulated artifacts *reduce* the agent's expected performance below its no-skill baseline on a fixed task distribution. Let $\mathcal{S}_t$ be the active skill set at round t and p_0 the no-skill pass rate. Drift manifests when:

$$\mathbb{E}[\text{pass@1} \mid \mathcal{S}_t] < \mathbb{E}[p_0] \tag{1}$$

for some $t > 0$. Crucially, drift can occur even when every individual skill appears reasonable; the failure is systemic, arising from retrieval dilution and stale injection at the *library* level.

3.2. Mechanism

Library drift proceeds through three compounding stages:

1. **Accumulation without quality signal.** Skills are born from failure patterns but never validated against outcomes. Marginal or harmful skills dilute the retrieval pool with each round.
2. **Retrieval degradation.** The router selects skills by surface similarity. As the bank grows unbounded, near-duplicate or stale skills crowd out useful ones, degrading precision at constant recall.
3. **Silent injection harm.** A stale skill injected into the solver prompt actively misleads, but produces no explicit error; the solver simply fails more often, indistinguishable from inherent task difficulty.

In practice, these stages produce three distinguishable drift sub-modes depending on the governance regime: *stagnation*: skills accumulate but fail to reach the solver (no routing signal, no learning; A1 reproduces this floor); *bloat*: unbounded growth degrades retrieval until injections become harmful (the failure mode of systems without a cap (Wang et al., 2023; Zhao et al., 2024); prevented in our Default by the active-cap); and *erosion*: over-aggressive governance destroys useful skills faster than they accumulate, collapsing the bank (A4 retains only 2 active skills). Which sub-mode dominates depends on governance strength: too little permits bloat, too much causes erosion, and disconnecting injection produces stagnation.

3.3. Reproducible triggers

We demonstrate library drift with two ablations of our full system (Ratchet; Sec. 5), both run on MBPP+ hard-100 (Liu et al., 2023) with Claude Opus 4.7 over 100 rounds:

A1 (no skill injection): The Router is forced to NONE: the full pipeline (synthesizer, critic, curator) still runs, but no skill is ever injected into the solver prompt. The critic produces 0 calls because no skill-attributed failures exist to judge. Result: +0.002±0.005 gain (the no-skill floor). This isolates the routing effect: skill creation alone, without injection, produces no gain.

A4 (harsh retirement): Evidence floor $N_{\min}$ reduced from 100 to 20; threshold τ tightened to 0.0. Result: -0.019 ± 0.010, *below* the no-skill baseline. The library actively harms performance. With only 20 trials, the Hoeffding deviation is $\epsilon \approx 0.44$; skills with true contribution $c \in [-0.44, 0]$ are prematurely retired on unlucky draws, while genuinely harmful skills may survive early rounds before enough evidence accumulates. The bank collapses to 2 active skills, and the router's 18.9% engagement means it rarely injects, yet when it does, the surviving skills hurt. The effect is consistent across all three seeds (-0.005, -0.027, -0.025; Table 3), confirming that harsh retirement reliably triggers erosion rather than reflecting a single unlucky run.

Together, these triggers bracket the failure mode from both sides: A1 shows that skills must be *injected* to matter; A4 shows that naïve lifecycle management can be *worse* than none. A4 is a cautionary negative result: governance is not uniformly beneficial: acting on insufficient evidence ($N_{\min} = 20$ vs. the Default's 100) actively harms performance rather than merely failing to help.

4. Trace-Level Diagnostics

End-task metrics (pass@1, solve rate) are insufficient to diagnose library drift: they conflate task difficulty, model stochasticity, and library quality into a single number. We introduce three complementary diagnostic signals that decompose the failure into actionable components:

4.1. Per-skill contribution score

For each active skill s, we track:

$$\hat{c}(s) = \frac{\text{successes}(s) - \text{failures}(s)}{\text{trials}(s)} \tag{2}$$

computed from the evidence log (capsules where s was injected). A declining mean $\hat{c}$ across the bank signals drift: the library is accumulating skills that do not help. In the Default condition, mean $\hat{c}$ rises over rounds (skills that hurt are retired); in A4, it oscillates because premature retirement removes useful skills.

4.2. Attribution verdicts

For every failure capsule, a separate Critic LLM call produces a structured verdict: HELPED, HURT, NEUTRAL, or INAPPLICABLE, plus a pattern label and confidence. These verdicts serve dual purpose: (1) synthesis substrate (clusters of $\geq$3 failures sharing a canonical pattern trigger new skills),

and (2) early-warning diagnostic. A rising proportion of HURT verdicts is a leading indicator of drift: in A4, the bank collapses to 2 active skills by round 30 (visible in router engagement dropping to 18.9%), while aggregate pass@1 declines only gradually because the router adaptively selects NONE on most tasks.

4.3. Router engagement metrics

We track what fraction of tasks the router assigns a skill versus NONE. Table 1 shows that healthy conditions maintain 70–80% engagement, while A4 (drifting) drops to 19% as the bank empties. A2 (retrieval-only, no LLM gate) shows 98% engagement but lower quality; the LLM gate's ability to decline injection is itself a drift-prevention mechanism.

5. Verified Fix: The Ratchet Recipe

Having defined the failure mode and the diagnostics that expose it, we now describe the fix. Ratchet is a single-agent loop where a frozen LLM writes, retrieves, curates, and retires its own natural-language skills. Three governance mechanisms jointly prevent library drift:

5.1. Outcome-driven retirement

A skill is retired once $n(s) \geq N_{\min}$ trials have accumulated *and* the empirical contribution $\hat{c}(s) \leq -\tau$. The Default uses $N_{\min} = 100$, $\tau = 0.10$, conservative enough that useful skills survive stochastic noise (Hoeffding $\epsilon \approx 0.20$) but aggressive enough that harmful skills are eventually removed. This directly addresses the "accumulation without quality signal" stage of drift.

5.2. Bounded active-cap

A hard cap C (Default: 50) forces the curator to evict the lowest-contribution skill when synthesis would exceed the cap. This directly addresses the "retrieval degradation" stage: by bounding the bank, retrieval precision cannot decay with unbounded growth. Combined with retirement, it provides a formal non-divergence guarantee:

Proposition 1. Under bounded cap C and retirement threshold τ, the expected eval pass@1 cannot drift below the no-skill floor by more than $\tau + \epsilon + C\delta$, where ϵ is the Hoeffding estimation tolerance and δ is the per-skill failure probability.

Systems without bounded C and τ (Wang et al., 2023; Zhao et al., 2024; Chen et al., 2024) have no finite analogue of this bound: library drift is unconstrained.

5.3. Meta-skill authoring prior

A meta-skill document constrains the Synthesizer to produce stylistically consistent skills. This addresses the "silent injection harm" stage at its source: by enforcing structural homogeneity, fewer harmful or redundant skills are born in the first place, reducing the load on downstream retirement. Ablation A3 (no meta-skill) shows that removing it costs 43% of the Default's gain; it is the single most valuable component.

Surprisingly, explicit deduplication mechanisms (pattern canonicalisation, A5; cover-guard, A6) are *not* necessary given the meta-skill: A5 and A6 slightly *exceed* the Default. The meta-skill subsumes explicit dedup at this scale, a design insight relevant to any system with LLM-authored artifacts.

6. Experiments

6.1. Protocol

We evaluate on **MBPP+ hard-100**: from the 378-task MBPP+ test split (Liu et al., 2023), we discard tasks that Claude Opus 4.7 solves on all 5 baseline seeds (∼273 tasks) and randomly sample 100 from the remainder (60 train / 40 eval, fixed seed). This isolates tasks where a skill library can plausibly help. All LLM calls use Claude Opus 4.7; embeddings use Cohere embed-v4 (both via Amazon Bedrock). The Solver is a single direct LLM call with no execution feedback, no self-refinement, and no tool use, isolating the skill library's contribution. Each run is 100 rounds; we report mean ± std over 3 seeds.

6.2. Main results

Table 1 and Figure 2 summarize results. The Default (full governance) lifts pass@1 from 0.258 baseline to 0.584 late-window mean (peak 0.658), more than doubling performance on genuinely difficult tasks. A1 establishes the no-skill floor (+0.002); A4 confirms that drift can push the library *below* it (−0.019).

6.3. Decomposing the fix

Eight ablations (A1–A8) each modify one knob from the Default to test whether the corresponding mechanism is load-bearing (3 seeds per condition; full settings in Table 2). We group them into two questions.

A1–A3: which components are necessary? **A1** (no skill injection) forces the Router to always select NONE: the synthesis pipeline still runs but no skill reaches the solver. Result: +0.002 (the no-skill floor). **A2** (retrieval-only routing) bypasses the LLM gate and injects the top-ranked retrieval hit directly. Result: +0.077 (24% of the Default gain),

Table 1. MBPP+ hard-100 (mean±std, 3 seeds). *Baseline*: round-0 pass@1 (no skill active in any condition; cross-condition variation is sampling noise from $n=40$ eval tasks; gain is computed within each run, cancelling this). *Gain*: mean(last 10) − mean(first 10) of held-out pass@1. *Router*: % of eval tasks assigned a skill. *Active/Retired*: bank state at round 100 (A1 has 42 active because the pipeline still synthesises skills, they are just never injected; A4's aggressive retirement empties the bank to 2, directly causing harm). *Critic*: LLM verdict calls (0 = no quality signal).

Condition	Baseline	Peak	Gain	Router	Active	Retired	Critic
Default	0.258±0.047	0.658±0.042	+0.328±0.018	73	50	89	4299
A1 no injection	0.283±0.031	0.375±0.000	+0.002±0.005	0	42	15	0
A2 retrieval	0.242±0.012	0.492±0.042	+0.077±0.065	98	42	69	5740
A3 no meta	0.200±0.035	0.592±0.047	+0.187±0.036	80	50	84	4676
A4 harsh retire	0.300±0.035	0.433±0.042	−0.019±0.010	19	2	51	1090
A5 no canon	0.275±0.020	0.708±0.012	+0.374±0.023	80	50	76	4393
A6 no guard	0.217±0.024	0.700±0.035	+0.363±0.033	70	50	94	3871
A7 cap=100	0.292±0.042	0.650±0.089	+0.317±0.110	75	100	55	4609
A8 meta refresh	0.250±0.035	0.725±0.020	+0.372±0.017	74	50	131	4388

confirming the LLM gate contributes beyond tf-idf∪embed similarity. **A3** (no meta-skill) removes the authoring prior from the Synthesizer prompt. Result: +0.187 (57% of the Default gain), making the meta-skill the single most valuable component (−0.141 when removed).

A4: harsh retirement is harmful. A4 lowers the evidence floor $N_{\min}$ from 100 to 20 and tightens τ to 0.0. Result: −0.019, *below* the no-skill floor. At $N_{\min} = 20$, the Hoeffding deviation is $\epsilon \approx 0.44$; skills with true contribution $c \in [-0.44, 0]$ can be retired on unlucky draws, collapsing the bank to 2 active skills. This validates Prop. 1: the retirement threshold is insufficient without a sufficiently large evidence floor. A4 is a cautionary result: governance is not uniformly beneficial; acting on insufficient evidence actively harms performance rather than merely failing to help.

A5–A6: explicit dedup is not necessary. A5 (no canonicalisation) raises τ_{canon} to 1.0, disabling pattern dedup. A6 (no cover-guard) disables duplicate-cluster skipping. Both slightly *exceed* the Default: A5 +0.374, A6 +0.363 (vs. Default +0.328; within $\pm 2\sigma$ at $n=3$). The meta-skill's authoring guidance enforces enough stylistic consistency that the explicit filter's false positives discard more useful skills than duplicates it prevents, a transferable design insight for any system with LLM-authored artifacts.

A7–A8: relaxed cap and meta-skill refresh. A7 (bank cap=100) doubles the active-cap. Result: +0.317 comparable mean but substantially higher variance (±0.110 vs. ±0.018); the cap primarily controls variance rather than mean performance at this scale. A8 (meta-synth refresh) regenerates the meta-skill every 10 rounds. Result: +0.372 and the highest peak (0.725), but at 55% more wall time (10.1 h vs. 6.5 h) due to additional synthesis rounds. Conclusion: more frequent refresh does not meaningfully improve the learning curve but incurs substantial compute overhead, not justified at this scale.

6.4. Operational cost

The Default uses ∼14.5k LLM calls per 100 rounds (10k solver + 4.3k critic + 152 synthesis), 43% more than the no-skill baseline (A1, 10k calls). Wall time is 6.5 h vs. A1's 2.3 h (2.8×). The solver dominates cost in all conditions (63–99% of total calls).

7. Discussion

Library drift as a general failure mode. Although we demonstrate drift in a skill library, neither the definition (Eq. 1) nor the diagnostics are specific to skills. Rule-based systems (Zhang et al., 2026b), workflow memories (Wang et al., 2024), and episodic stores all persist LLM-authored artifacts across tasks and face the same accumulation-without-retirement failure. We argue that library drift deserves recognition as a first-class failure category alongside tool-use errors and planning failures.

Diagnostics precede fixes. The trace-level diagnostics detected drift before end-task metrics declined. In A4, the bank collapsed to 2 active skills by round 30, but pass@1 showed only gradual decline because the router adaptively chose NONE more often. Without per-skill diagnostics, drift would appear as "random task difficulty variation," invisible and unfixable. Critically, these diagnostics are not merely descriptive; they enable counterfactual intervention. In the Default condition, monitoring mean $\hat{c}$ and triggering retirement only after $N_{\min} = 100$ trials prevents the premature eviction that destroys A4. An operator watching router engagement drop below 50% could halt synthesis and investigate before drift reaches end-task metrics. The recipe (per-artifact contribution scores, attribution verdicts, and engagement metrics) generalizes to any system where an LLM agent accumulates persistent artifacts across episodes.

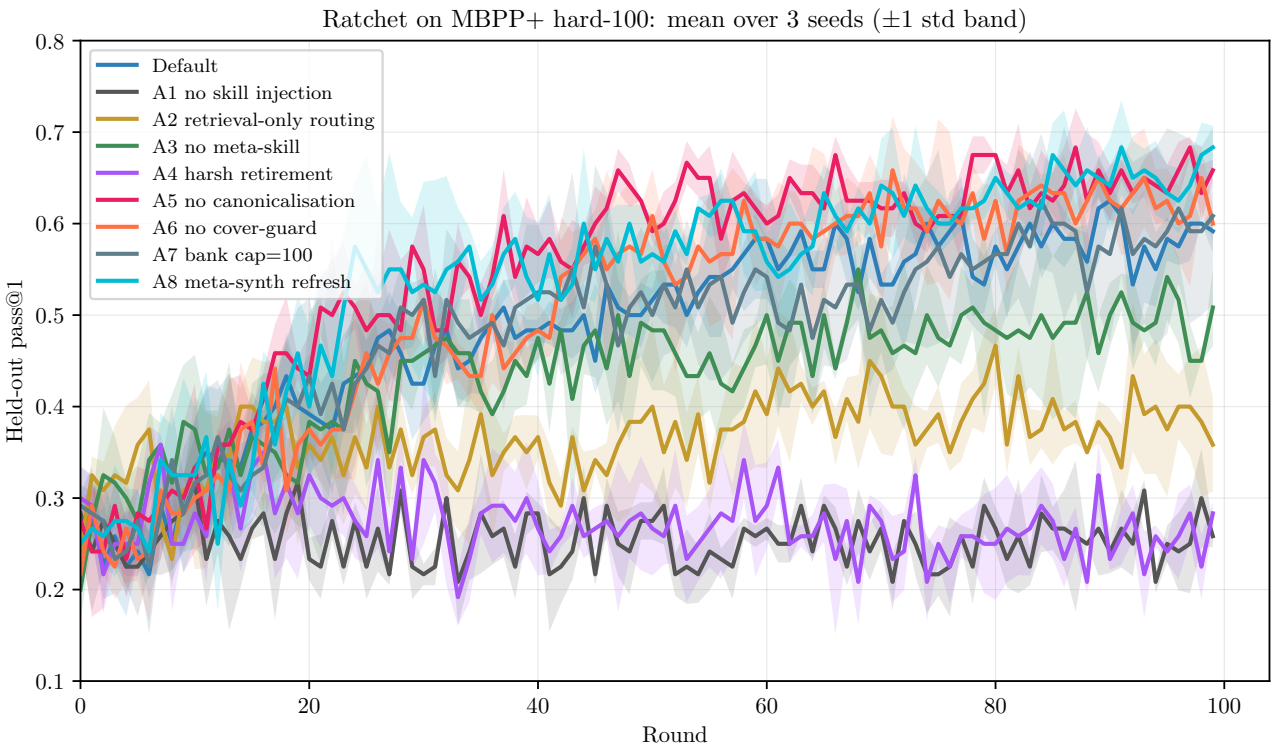

Figure 2. Held-out pass@1 by round (3-seed mean ±1 std). A1 (flat floor) and A4 (below floor) exhibit library drift. A5/A6 (relaxed dedup) slightly exceed the Default; the meta-skill subsumes explicit filtering. A7 (doubled cap) shows comparable mean but higher variance. A8 (meta-synth refresh) matches A5/A6 gains but at 55% more wall time (10.1 h vs. 6.5 h).

We suggest adopting these trace-level signals as a standard diagnostic vocabulary for failure-mode analysis in agentic systems.

Expected vs. unexpected results. Of eight ablations, three produced surprises. *A4* fell below the no-skill baseline; we expected harsh retirement to underperform, but not to actively harm, confirming that the evidence floor is load-bearing even in principle. *A5/A6* slightly exceeded the Default, refuting our design assumption that explicit dedup would help: the meta-skill already enforces enough homogeneity. *A8* yielded only marginal gain despite 55% more wall time, challenging the intuition that a stale meta-skill limits synthesis quality. For practitioners diagnosing drift in their own systems: governance mechanisms can be counterproductive; the diagnostic recipe (Sec. 4) is what makes failures visible and safe intervention possible.

Scale-dependence of governance knobs. The Default is broadly safe: every condition except A4 substantially outperforms the no-skill baseline, and the non-divergence guarantee (Prop. 1) holds regardless of task distribution. However, exact knob settings are scale-dependent. On our 100-task suite the meta-skill alone provides sufficient stylistic consistency that canonicalisation (A5) and cover-guard (A6) are unnecessary; on larger suites with hundreds of candidate patterns, we expect explicit deduplication to become load-bearing. Similarly, the bounded cap ($C = 50$) is safe to double (A7 achieves comparable mean gain) but at the cost of substantially higher variance: per-seed gains range from $+0.172$ to $+0.440$ (Table 3), confirming that a relaxed cap exposes the loop to luck-of-synthesis-order. The principled approach: treat the Default as a safe starting point, then relax governance knobs guided by the diagnostic signals once the operator observes the meta-skill suffices.

Limitations. (i) One benchmark (MBPP+ hard-100); generalization to broader domains (multi-step agents, SWE-Bench) is left to future work. (ii) Single model (Claude Opus 4.7); cross-model stability not established. (iii) The diagnostic thresholds (when to intervene) are empirically chosen; principled calibration is future work. (iv) We hypothesize that drift signals (contribution scores, HURT verdicts) become more sensitive in multi-step agents where per-step verdicts can isolate which step a stale skill harmed; empirical validation is future work.

8. Conclusion

The bottleneck in self-evolving skill libraries is not the author; it is the librarian. Library drift, unbounded accumulation without lifecycle management, is a silent, systemic failure mode: the same frozen LLM that stagnates at $+0.002$ without governance delivers $+0.328$ once retirement, a bounded cap, and an authoring prior are added. We provide (1) reproducible triggers that bracket the failure from both sides (A1: no injection floor; A4: governance-induced harm), (2) trace-level diagnostics that detect drift before end-task metrics decline, and (3) eight ablations documenting which mechanisms are load-bearing (retirement, meta-skill) and which are safely removed (explicit dedup). Three surprises (A4 harmful, A5/A6 exceeding the Default, A8 marginal despite cost) demonstrate that naïve governance can be worse than none, making the diagnostic recipe essential for safe intervention. The playbook (per-artifact contribution scores, attribution verdicts, and engagement metrics) transfers to any system that persists LLM-authored artifacts across episodes.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Chen, M., Li, Y., Yang, Y., Yu, S., Lin, B., and He, X. AutoManual: Generating instruction manuals by LLM agents via interactive environmental learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Huang, X., Chen, J., Fei, Y., Li, Z., Schwaller, P., and Ceder, G. CASCADE: Cumulative agentic skill creation through autonomous development and evolution. *arXiv preprint arXiv:2512.23880*, 2025.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Li, X., Chen, W., Liu, Y., Zheng, S., Chen, X., He, Y., Li, Y., You, B., Shen, H., Sun, J., et al. SkillsBench: Benchmarking how well agent skills work across diverse tasks. *arXiv preprint arXiv:2602.12670*, 2026.

Liu, J., Xia, C. S., Wang, Y., and Zhang, L. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in Neural Information Processing Systems*, 36, 2023.

Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36, 2023.

Ni, J., Liu, Y., Liu, X., Sun, Y., Zhou, M., Cheng, P., Wang, D., Jiang, X., and Jiang, G. Trace2Skill: Parallel inductive skill distillation for LLM agents. *arXiv preprint arXiv:2603.25158*, 2026.

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., and Scialom, T. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36, 2023.

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., and Yao, S. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2023.

Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., and Anandkumar, A. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Wang, J., Ren, Y., and Zhang, H. From procedural skills to strategy genes: Towards experience-driven test-time evolution. *arXiv preprint arXiv:2604.15097*, 2026.

Wang, Z. Z., Mao, J., Fried, D., and Neubig, G. Agent workflow memory. *arXiv preprint arXiv:2409.07429*, 2024.

Wu, R., Wang, X., Mei, J., Cai, P., Fu, D., Yang, C., Wen, L., Yang, X., Shen, Y., Wang, Y., et al. Self-evolving LLM agents through an experience-driven lifecycle. *arXiv preprint arXiv:2510.16079*, 2025.

Yang, Y., Li, J., Pan, Q., Zhan, B., Cai, Y., Du, L., Zhou, J., Chen, K., Chen, Q., Li, X., et al. AutoSkill: Experience-driven lifelong learning via skill self-evolution. *arXiv preprint arXiv:2603.01145*, 2026.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. R., and Cao, Y. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Zhang, X., Wang, G., Cui, Y., Qiu, W., Li, Z., Zhu, B., and He, P. Experience compression spectrum: Unifying memory, skills, and rules in LLM agents. *arXiv preprint arXiv:2604.15877*, 2026a.

Zhang, X., Wang, G., Cui, Y., Qiu, W., Li, Z., Zhu, B., and He, P. Do agent rules shape or distort? guardrails beat guidance in coding agents. *arXiv preprint arXiv:2604.11088*, 2026b.

Zhao, A., Huang, D., Xu, Q., Lin, M., Liu, Y.-J., and Huang, G. ExpeL: LLM agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., and Stoica, I. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

A. System Architecture

Ratchet operates a five-phase loop per round: (1) **Eval**: routes active skills on held-out tasks; (2) **Train**: same pipeline on train split, generating failure substrate; (3) **Critic**: per-failure attribution verdict (LLM call); (4) **Synthesizer**: clusters failures by canonical pattern, authors new skills from clusters with $\geq$3 members; (5) **Curator**: computes contribution scores, retires under-performers, enforces cap.

The Router selects one skill or NONE per task via two-stage retrieval (tf-idf + embedding, K=10 each) followed by an LLM gate.

B. Hyperparameters

Table 2. Default configuration hyperparameters.

Knob	Purpose	Value
Active-cap C	Max active skills	50
$N_{\min}$	Evidence floor	100
τ	Retirement threshold	0.10
Canon threshold	Pattern dedup	0.85
Cover threshold	Cluster skip	0.85
Bank-dedup threshold	Skill YAML dedup	0.85
Skill length budget	YAML char cap	1500
Synth lookback	Verdict window	6 rounds
Min cluster size	Synthesis trigger	3
Max skills/round	Synthesis cap	2
Router cutoff	Full-bank mode	20
Retrieval K	Shortlist size	10
Rollback τ_{rb}	Regression depth	0.10
Rollback persistence	Consecutive rounds	5
Rounds	Per run	100
Seeds	Per condition	42, 7, 13

C. Per-Seed Results

D. Non-Divergence Proof Sketch

Proposition 1. Let $|\mathcal{S}_t| \leq C$, and suppose the Curator retires skill s when $n(s) \geq N_{\min}$ and $\hat{c}(s) \leq -\tau$. Under Hoeffding bounds with tolerance ϵ and per-skill failure probability δ:

With probability $\geq 1 - C\delta$ (union bound over C skills), every surviving skill has $c(s) \geq -\tau - \epsilon$. Tasks routed to a surviving skill have expected pass probability $\geq p_0(x) - \tau - \epsilon$; tasks routed to NONE achieve $p_0(x)$. Taking expectations: $\mathbb{E}[\text{pass@1}] \geq \mathbb{E}[p_0] - (\tau + \epsilon) - C\delta$.

With Default values ($\tau = 0.10$, $N_{\min} = 100$, $C = 50$, $\delta = 10^{-3}$): $\epsilon \approx 0.20$, floor $= \mathbb{E}[p_0] - 0.35$. The bound is loose at our scale (the system gains +0.328 rather than losing 0.35) but rules out unbounded degradation.

Table 3. Per-seed rolling-mean gain and peak on MBPP+ hard-100.

Condition	Seed	Baseline	Peak	Gain
Default	42	0.225	0.675	+0.303
	7	0.225	0.600	+0.340
	13	0.325	0.700	+0.343
A1 no injection	42	0.325	0.375	+0.003
	7	0.275	0.375	−0.005
	13	0.250	0.375	+0.008
A2 retrieval	42	0.250	0.475	+0.148
	7	0.250	0.550	+0.095
	13	0.225	0.450	−0.010
A3 no meta	42	0.175	0.525	+0.148
	7	0.250	0.625	+0.177
	13	0.175	0.625	+0.235
A4 harsh retire	42	0.325	0.450	−0.005
	7	0.325	0.375	−0.027
	13	0.250	0.475	−0.025
A5 no canon	42	0.300	0.725	+0.342
	7	0.275	0.700	+0.385
	13	0.250	0.700	+0.395
A6 no guard	42	0.200	0.725	+0.365
	7	0.200	0.725	+0.402
	13	0.250	0.650	+0.322
A7 cap=100	42	0.350	0.525	+0.172
	7	0.250	0.700	+0.340
	13	0.275	0.725	+0.440
A8 meta refresh	42	0.200	0.750	+0.365
	7	0.275	0.700	+0.355
	13	0.275	0.725	+0.395